Journal of Computational Design and Engineering, Vol. 1, No. 2 (2014) 140~151
www.jcde.org

Intelligent 3D packing using a grouping algorithm for automotive container engineering

Youn-Kyoung Joung and Sang Do Noh*

[1] *Department of Systems Management Engineering, Sungkyunkwan University, 300 Cheoncheon-dong, Jangan-gu, Suwon, Gyeonggi-do, South Korea*

(Manuscript Received January 19, 2014; Revised February 27, 2014; Accepted March 19, 2014)

Abstract

Storing, and the loading and unloading of materials at production sites in the manufacturing sector for mass production is a critical problem that affects various aspects: the layout of the factory, line-side space, logistics, workers' work paths and ease of work, automatic procurement of components, and transfer and supply. Traditionally, the nesting problem has been an issue to improve the efficiency of raw materials; further, research into mainly 2D optimization has progressed. Also, recently, research into the expanded usage of 3D models to implement packing optimization has been actively carried out. Nevertheless, packing algorithms using 3D models are not widely used in practice, due to the large decrease in efficiency, owing to the complexity and excessive computational time. In this paper, the problem of efficiently loading and unloading freeform 3D objects into a given container has been solved, by considering the 3D form, ease of loading and unloading, and packing density. For this reason, a Group Packing Approach for workers has been developed, by using analyzed truck packing work patterns and Group Technology, which is to enhance the efficiency of storage in the manufacturing sector. Also, an algorithm for 3D packing has been developed, and implemented in a commercial 3D CAD modeling system. The 3D packing method consists of a grouping algorithm, a sequencing algorithm, an orientating algorithm, and a loading algorithm. These algorithms concern the respective aspects: the packing order, orientation decisions of parts, collision checking among parts and processing, position decisions of parts, efficiency verification, and loading and unloading simulation. Storage optimization and examination of the ease of loading and unloading are possible, and various kinds of engineering analysis, such as work performance analysis, are facilitated through the intelligent 3D packing method developed in this paper, by using the results of the 3D model.

Keywords: Packing, 3D CAD model, Grouping algorithm, Container engineering

1. Introduction

As the life-cycles of products are decreasing, and competition between global corporations is becoming more intensive, in order to provide new and diverse ranges of goods that customers want, product development, production-period reduction, production-cost reduction, quality improvement, small-quantity batch production, and mass-customization have become the top priorities of the manufacturing sector. To compete and survive in the marketplace, manufacturing companies have undertaken various kinds of R&D. Packing has been used widely throughout the automobile sector, to solve the problems of storing parts in containers.

In the case of an automobile factory in Korea, 3,300 engineering containers in the press factory, 3,900 engineering containers in the chassis factory, and 700 containers (bulk containers: 200, engineered containers: 500) in the assembly factory are used, respectively, for the mixed-model production of four models. Also, in the case of an automobile factory in the US, more than 3,000 main products are needed for container nesting plans, and between 2002 and 2009, 2.5 billion dollars were invested in containers.

The intelligent packing algorithm developed in this paper enables better design of containers and packing; also, the cost of containers can be estimated when they are designed at first. For these reasons, storing and loading/unloading materials at production sites in the manufacturing sector for mass production is a critical problem that affects various aspects: the layout of the factory, line-side space, logistics, workers' work paths and ease of work, automatic procurement of components, and transfer and supply. There are two types of containers, bulk and engineering.

1.1 Bulk container (Car truck packing)

The bulk container packing problem is similar to the car truck packing problem. Therefore, we applied the algorithm that is developed in this paper to the car truck packing problem, instead of the bulk container packing problem. The

[*] Corresponding author. Tel.: +82-31-290-7603, Fax.: +82-31-290-7610
E-mail address: sdnoh@skku.edu

doi: 10.7315/JCDE.2014.014

computation of baggage capacity has a significant influence on the design process of cars [1]. Automobile manufacturers report the trunk capacity using one of the two published standards for the cargo volume, which can be considered as specializations of the general 3D packing problem [20]. Two widely used standards are the SAEJ1100 (for the USA) standard, and the DIN 70020 (for the European Union) standard [16, 6]. Cooperation with a German car manufacturer has led to several efficient approximation schemes for the trunk packing problem, according to the German standard DIN 70020. This cooperation has continued by exploring the baggage volume capacity according to the U.S. standard SAE J1100. Extensive research has been performed into the development of packing algorithms both for the case of SAEJ100, and DIN70020 [20, 8, 7, 11, 14, 15].

DIN70020 is a type of a box (200mm×100mm×50mm), and the maximum number of this type of boxes seated in a trunk becomes the capacity of the trunk [8]. Figure 1 shows the real trunk used for calculating its capacity with DIN70020.

1.2 Engineering container

An engineering container stores only one type of a part; moreover, it has a unique feature, in that there has to be consideration for the jig & fixture, which allows for stability of the parts when they are being packed, and the fact that all the parts are directed one way.

Engineering containers mainly store panel-type parts, and require clearance between parts, in order to prevent parts from being damaged during loading/unloading works. Currently, the clearance value between parts is empirically decided, by engineering container designers. In this paper, the bulk container problem is examined using the SAE standard J1100, and the result is compared with the result from Santosh, by using their genetic algorithm [20, 16].

1.3 Engineering container

Cagan proposed some approaches for the packing problem, as follows, and many recent researches approach the packing problems on the basis of their proposals [3].

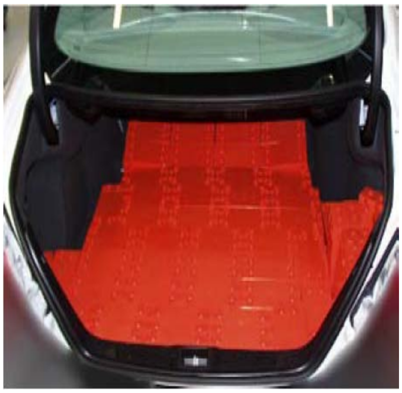


Figure 1. Trunk capacity calculation by DIN70020 standard [7].

1.3.1 Heuristic rule-based approaches

This approach is a methodology of enhancing the efficiency of packing, by analyzing the sense and experience of experts. Realistic limitation is applied, and it is stable. Also, it has the advantage of short packing calculation time on complicated forms. But it has the disadvantage of difficulty in applying in other industries, and generalizing the algorithms.

1.3.2 Traditional optimization approaches

These approaches include branch and bound methods, as in Scheithauer, linear programming methods, as in Beasley, and gradient-based methods, as in Landon [2, 13, 17]. The arithmetic operation speed of single run methods is fast; however, these find only to the nearest local sum. Therefore, other multiple run methods from the initial state need to overcome this drawback.

1.3.3 Genetic algorithms

In this approach, design variables are mapped into a string of symbols. A population of strings is maintained, and the design is changed, by mutating the strings. A fitness function is used to sift out the promising seeds for the next iteration of mutation. Genetic algorithms are stochastic algorithms, and can control the end of repetition. This algorithm has the strength that it simplifies the calculating method of packing, which needs complex calculation; however, the calculating time is still long, and the result value occurs randomly. Moreover, if there is an increase in the number of repetitions for better results, the calculating time will be prolonged, as well.

1.3.4 Simulated annealing algorithms

Imitation-based simulated annealing, which simulates the process of melting metal, is used to solve a complicated optimization problem. Simulated annealing algorithms have been applied to 2D circulation design. Sechen and Cagan extended this method from 2D to 3D, and applied it to mechanical and electrical layout design [5, 18, 19]. Kolli expanded this method, by solving the components of geometrical structure, and restriction of rotation [12]. Simulated annealing is not affected by early stages due to the stochastic searching strategy, and a 3D layout's space search is also possible. However, it costs a lot to get a good result, because lots of changes in motion are required.

1.3.5 Extended pattern search

A fundamental pattern search algorithm is a deterministic algorithm, direct search, which was introduced by Hooke, and developed by Torczon [10, 21]. Yin developed this by adding the stochastic characteristic, which helps to not get a local sum, and is applied to the layout problem [3]. Through

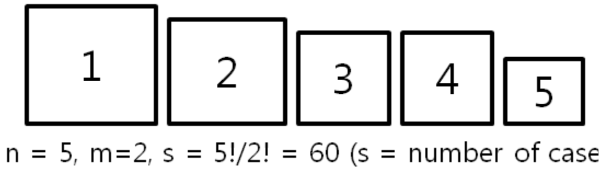


Figure 2. Example regarding packing sequences.

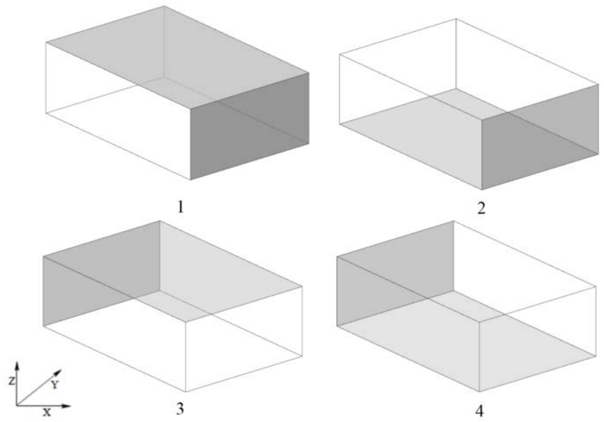


Figure 3. Orthogonal orientations for a prismatic object.

the tests, he revealed that this approach method can reduce the time to reach the same result as the simulated annealing algorithm.

1.3.6 Hybrid approaches

This approach is the method that combines two or more search techniques. The method remedies each approach's shortcomings, and creates new approaches. Detailed information will be provided from Cagan [3].

In this way, packing optimization is actively carried out. Nevertheless, a packing algorithm using 3D CAD model is not widely used in actual work, due to the rapid decrease of efficiency according to the complexity of the form, and long operation time. Therefore, in this research, it is applied to the bulk container problem from SAE, and the result is compared with the result from the genetic algorithm that Santosh developed, to investigate the excellence of the algorithm proposed in this research [20]. Further, an algorithm that considers the loading direction and the clearance value between parts is applied to the engineering container that is used in the real world.

2. Packing problem

2.1 The packing sequence

The packing sequence problem is an order decision problem concerning the loading of parts. If n parts are loadable, the loading sequences of parts number n!. Assuming there are m parts of the same type, the number of sequences is n!/m!. Figure 2 shows an example of the above.

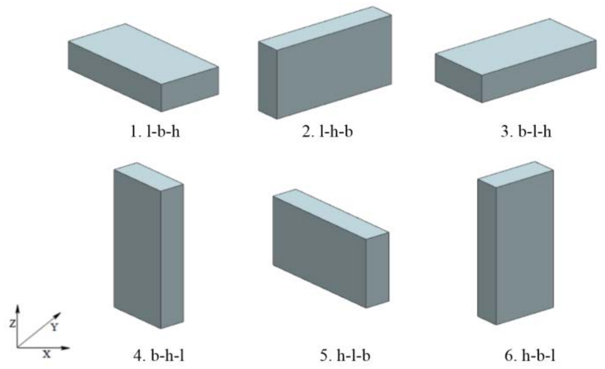


Figure 4. Four possible orientations for the same bounding box.

2.2 Part orientation

The part-orientation problem is the problem of deciding the orientation of parts. In the case of 2D, only the orientation between 0 degrees and 360 degrees has to be calculated; but in the case of 3D, the orientation regarding each axis has to be calculated, which makes the computation impossible. Santosh et al. defined the orientation of parts under the criterion of 90 degrees, and defined the problem as follows [20].

2.2.1 Prismatic objects with orthogonal orientations

If the part is a right cuboid of dimensions (l × b × h), six orientations can be derived as shown in Figure 3.

2.2.2 Free-form objects with orthogonal orientations

A freeform object has a different form in each face; thus, four more orientations can be derived, as shown in Figure 4.

The next formula considers the orientation and the sequence of parts. With n different freeform parts, (6 × 4)n! cases are obtained. For example, assuming that three parts are loaded, 144 cases have to be considered. The packing problem entails loading from several tens of parts, up to several hundred parts; thus, computing for all the cases is very time-intensive. Moreover, the loading locations of parts and container capacity and type have to be considered as well.

3. Approach

As mentioned in Section 2, the packing problem can be discretized in the following way: First, the rotation of the boxes can be limited to axis-oriented placements. Second, the possible placements are restricted to the cells of a grid. Clearly, these restrictions reduce the solution space, and possibly eliminate the optimal solution of the original problem. Reichel shows similar problems (Discrete-Box-Packing for equal boxes) to be NP-complete, and the continuous version to be NP-hard [14].

Thus, we cannot hope to find the optimal solution. In this study, work analysis is made of the use of workers in trunk packing, as well as for the group-technology methodology used in the manufacturing sector; thus, the algorithm has been developed. As aforementioned, efficiently automating the loading of parts with complicated shapes into containers is a difficult task. For the automation, Ding and Cagan summarized the work pattern in packing by truck experts, as follows [4]:

Packing by truck experts

1) The technicians usually put in luggage pieces with large volumes, to occupy as much space as possible.
2) Those pieces are rearranged in the trunk, in order to get a good initial loading.
3) Smaller components, including H-boxes, are added, to fill the remaining space

Grouping by similarity of form

1) Parts having more than a certain level of similarity in terms of form are grouped together.
2) The group with the largest average volume of parts is to be loaded first.
3) The parts in the same group have to be loaded in the same direction and posture, if possible.
4) Loading must commence from the bottom corner.

Here, packing is a problem of loading parts of different sizes and shapes into a container of erratic form; primarily, there is a loading-sequence problem, and a parts-orientation problem.

4. Intelligent 3D packing method through grouping algorithm

In this paper, we solve the packing problem, using an intelligent 3D packing by grouping algorithm. The grouping algorithm decides the loading sequence, orientations, and displacements of the parts that will be packed in the container. The algorithm developed in this paper consists of 4 sub-algorithms. The algorithm developed in this research includes 4 sub-algorithms, and each of them is a CAD function. First, the grouping algorithm groups parts in similar shapes, through comparison on the shape of each part, and the sequencing algorithm makes the sequence of parts by which they are loaded into the container. After executing two algorithms, the orientating algorithm decides the poses of parts to be loaded, and the loading algorithm finds the optimized location where the parts are loaded. In the case of failure, it loops back to the sequencing algorithm, and decides the sequence again. The algorithm is terminated after all parts are loaded, or at the point when there is no sufficient space in the container. Also, all of the sub-algorithms are executed by CAD software. The entire algorithm is explained below. Figure 5 represents the whole process of the algorithm.

4.1 The grouping algorithm

The grouping algorithm is an algorithm that groups objects that are similar in shape or size, and it is developed on the basis of a total shape-based comparative methodology to find similar shapes and sizes [9]. The grouping algorithm sets a bounding box around a part, and divides the box into 1000 smaller cells, and creates points in cells where the part's edge falls in. Then the algorithm decides to which group a part should be in, by comparing the distribution of points created in cells along the edge of the part. The distribution rate that decides the group is input by users. Figure 6 shows the entire flow of the grouping algorithm.

4.1.1 Calculation of the bounding-box size for the parts

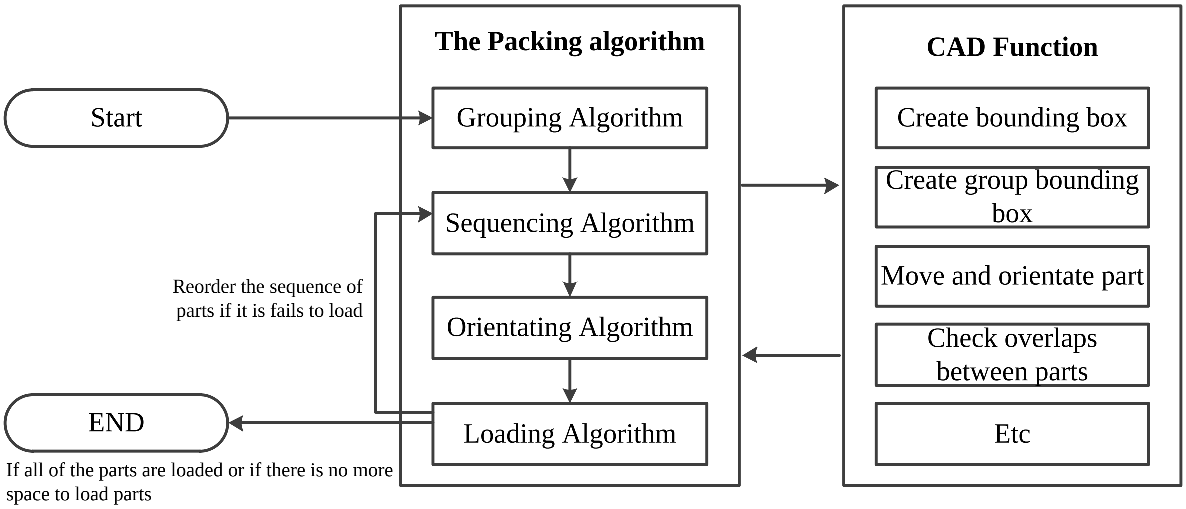

The Packing algorithm
Start
Grouping Algorithm
Sequencing Algorithm
Orientating Algorithm
Loading Algorithm
END
Reorder the sequence of parts if it is fails to load
If all of the parts are loaded or if there is no more space to load parts
CAD Function
Create bounding box
Create group bounding box
Move and orientate part
Check overlaps between parts
Etc

Figure 5. Procedure of the intelligent 3D packing method by grouping algorithm.

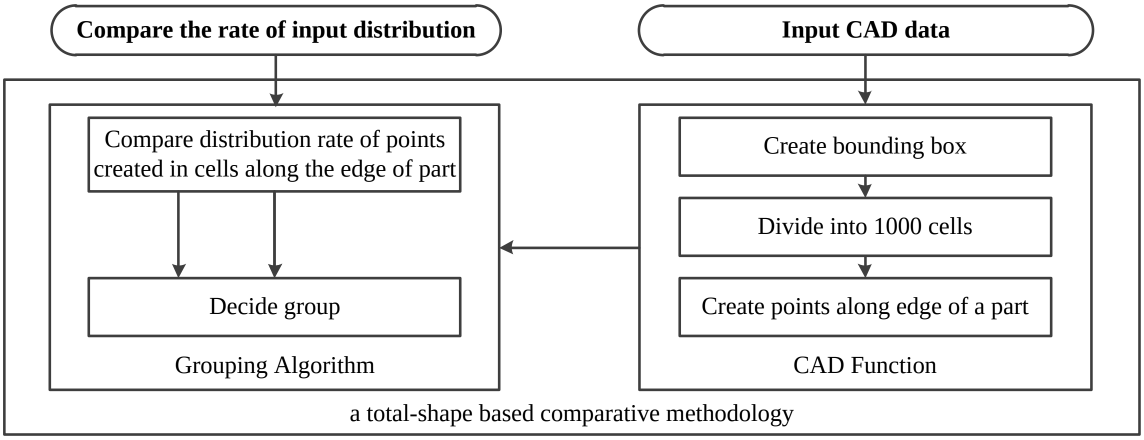

Compare the rate of input distribution
Input CAD data
Compare distribution rate of points created in cells along the edge of part
Decide group
Grouping Algorithm
Create bounding box
Divide into 1000 cells
Create points along edge of a part
CAD Function
a total-shape based comparative methodology

Figure 6. Procedure of the grouping algorithm.

The volume of each bounding box is the minimum in which a part can just be filled, and the bounding box technique is utilized in this research, to achieve faster calculation of a shape. The calculation of the bounding-box volume is as shown in Figure 7.

4.1.2 Classification of parts by the bounding-box volume

Using the volume of the bounding box for the reduction of time, the parts that will be grouped are firstly classified. For the classification, the following are determined, such that the comparing part is GP, the object part is CP, and the compare-rate is TR. If 100(CP/GP) > TR, the part is going to be grouped, and if 100(CP/GP) < TR, it will be not considered for grouping.

4.1.3 Point formation for a part

For point creation, the faces in the solid are examined, and the edges in each face are examined. After the edges are found, the attributes of the edges are scanned. The edge of the hole will be deleted from the list, and all the edges left in the edge list create a point. Figure 8 explains the above procedure.

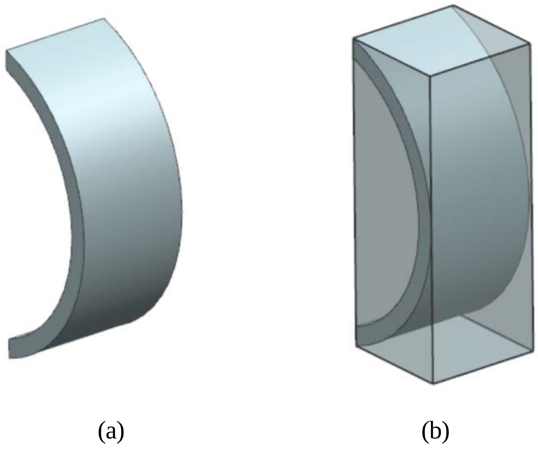


Figure 7. Calculation of the bounding-box volume: (a) part, (b) Apply bounding box.

4.1.4 Distribution-chart examination

Firstly, 1,000 cells are created in relation to (widths, lengths, and heights)/10 of the part's bounding box. Then, the points produced in the fourth process of “Point formation for a part” have to be ascertained, to know which cells they belong to. Figure 9 represents the erstwhile process. Actually, the cells are not created as shown in the figure; each cell is calculated to find the cell to which the point belongs, which reduces the comparison time.

4.1.5 Grouping

If the number of newly created points in a similar cell exceeds the compare-rate, for accuracy of the result, an exami-

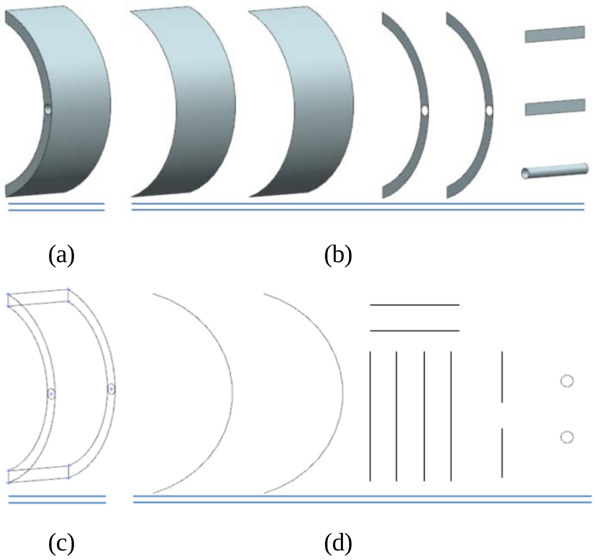
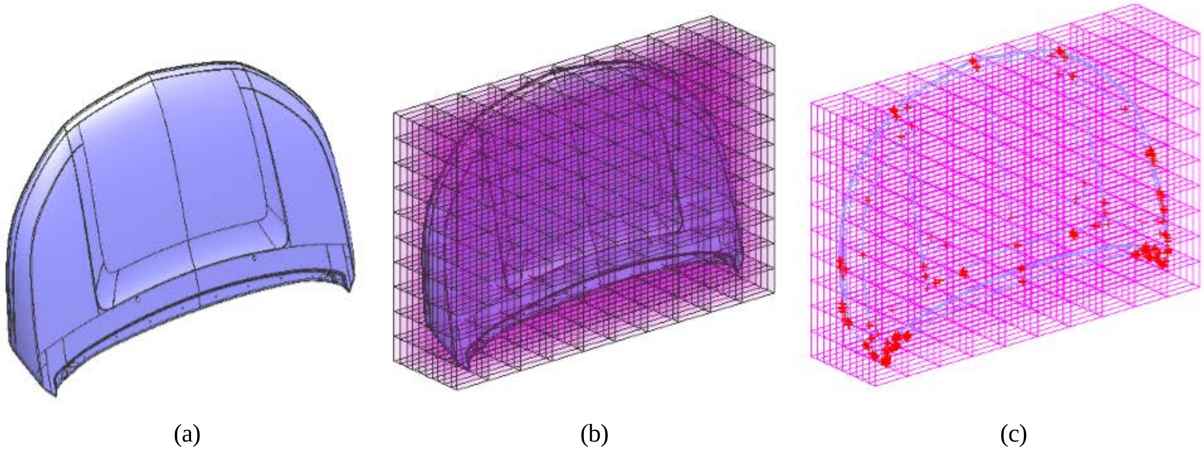


Figure 8. The point formation process: (a) solid, (b) faces, (c) edges, (d) points.



(a) (b) (c)

Figure 9. The distribution-chart examination: (a) part, (b) 1000 cells are created in the bounding box of the part, (c) points are created on edges of the part.

Table 1. Result of point creation.

Edge type	Linear edge	Circular edge	Ellipse edge	Spline edge	Total number
Number	60	232	0	1,818	2,110

nation of the distribution chart is carried out, using the attributes of the edges that are the basis for point creation. Table 1 displays the result of the creation of a point in Figure 9. Also, parts of the same form with different orientations must be compared in relation to the same orientation, and grouped.

4.2 The sequencing algorithm

The sequencing algorithm decides the sequence of parts to be loaded in the order of deciding the sequence of the group that results from the grouping algorithm, and the sequence of parts in each group. The sequencing algorithm decides the sequence of parts to be loaded with ‘Packing by truck experts approach’, according to the size of parts: from bigger parts to smaller parts. The size of a part is calculated by calculating the size of the bounding box that holds a part, since the packing problem is more affected by the shape of parts to be loaded, than the volume of parts. The flow of the sequencing algorithm is explained below.

1) Calculation of the average bounding-box volume of the parts in the group
2) Decision regarding the loading order for groups. The average volume of the group that is calculated from (1) is denoted as EGV. The values of EGV are arranged in descending order as G0, G1……… Gn-1; thus, the loading order of the groups is decided (n = number of the groups).
3) Decision regarding the loading order for the parts within a group

The sequence of part loading is decided by the bounding-box volume of the parts in the group in descending order. After the decision regarding the loading order for groups is made, the loading orientation decisions and loading of the parts are accomplished. The part to be loaded will affect the next one; if it fails, the loading sequence moves to the next group. Failure in loading is determined by the loading algorithm.

4.3 The orientating algorithm

The orientating algorithm decides the pose of parts to be loaded. The algorithm regarding decision on orientation involves the orientation decisions of the first and n+1th (n > 0) loaded parts. After deciding the pose of the first part to be loaded, poses of the following parts are decided, and loaded into container in the order that is decided by the sequencing algorithm. The flow of the orientating algorithm is as in Figure 10.

(1) Orientation decision for the first loaded part

The G00 and G01 parts are firstly compared (Gij: i is an index of the group, j indexes parts within a group). If group G0 has one part, the G00 and G10 parts are compared. As mentioned before, if part-numbers 1 and 2 are in the same group, then they are judged as having the same form; thus, each part has six orientations (GijO0, GijO1, …, GijO5). If part-numbers 1 and 2 belong to different groups, then each orientation results in three additional orientations, which makes each part with 24 orientations, such as GijO0, GijO1……GijO23. If part number 1 and 2 are in the same group, the order in which orientations are compared is the same as in the figure below, and the number of possible cases is 6×6. If part number 1 and 2 belong to different groups, the order in which orientations are compared is the same as below, and the number of possible cases is 24×24. Figure 11 shows a comparison between the pose of part number 1, and 6 poses of part number 2. Poses of two parts are compared in the order that is shown in Figure 12, and the pose of the part

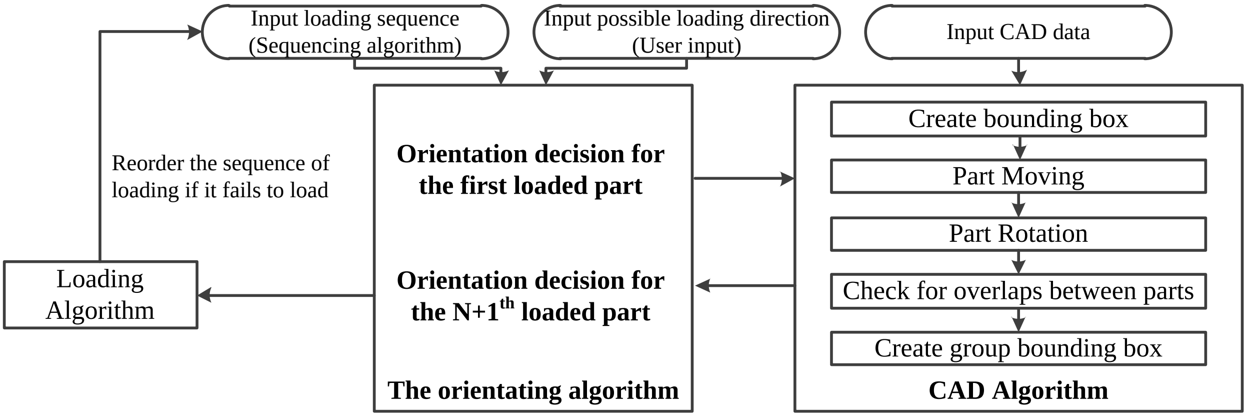

Input loading sequence
(Sequencing algorithm)
Input possible loading direction
(User input)
Input CAD data
Reorder the sequence of loading if it fails to load
Orientation decision for the first loaded part
Orientation decision for the N+1th loaded part
The orientating algorithm
Loading Algorithm
Create bounding box
Part Moving
Part Rotation
Check for overlaps between parts
Create group bounding box
CAD Algorithm

Figure 10. The distribution-chart examination.

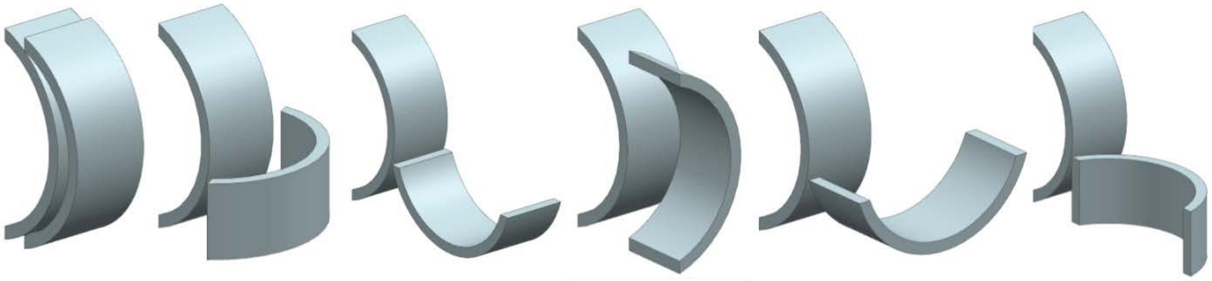


Figure 11. Comparison between 1st part's 1st posture, and 2nd part's 1st~6th postures.

$G_{00}P_0$->$G_{01}P_0$, $G_{01}P_1$......$G_{01}P_{6-1}$
$G_{00}P_1$->$G_{01}P_0$, $G_{01}P_1$......$G_{01}P_{6-1}$
⋮
$G_{00}P_{6-1}$->$G_{01}P_0$, $G_{01}P_1$......$G_{01}P_{6-1}$

(a)

$G_{00}P_0$->$G_{01}P_0$, $G_{01}P_1$......$G_{01}P_{24-1}$
$G_{00}P_1$->$G_{01}P_0$, $G_{01}P_1$......$G_{01}P_{24-1}$
⋮
$G_{00}P_{24-1}$->$G_{01}P_0$, $G_{01}P_1$......$G_{01}P_{24-1}$

(b)

Figure 12. Sequence of comparison orientation parts: (a) sequence of comparison postures parts with same group, (b) sequence of comparison postures parts with other group.

is decided when the volume of the bounding box is the minimum.

The group bounding box is the box with the minimum volume in which the two parts can be filled. Creating the group bounding box is to decide the optimized loading efficiency, after comparing poses of each of two parts rotated 90 degrees, and this approach eases the calculation, regardless of the complexity of shapes of parts. Figure 13 represents the computational procedure for the group's bounding-box volume. The first part of the bounding box is set as fPb, and the second part of the bounding box is set as sPb. The minimum and maximum values of both parts of the group's bounding box are Gbmin = fPbmin and Gbmax = sPbmax, respectively, and the formula for calculating Gbv (the group's bounding-box volume) is the same as the formula for the part's bounding-box volume.

After deciding the pose of the part, the part is loaded onto the container through the loading algorithm, and if a part fails to be loaded, the first part in the next group becomes the next part to be loaded. After the loading, every number should be reset, except the loaded part, and the group for which every part has been loaded. Through the above process, the first part is loaded; in the fifth process, the loading position is decided by the loading algorithm. The number of possible orientation comparisons for loading the first part is derived through the formal procedure outlined below.

P_n: The number of parts
G_n: The number of groups
G_{n1}: Group in which number of the parts in the group is one
S_{gpn}: The possible postures when two parts are in the same group (6 × 6)
D_{gpn}: The possible postures when two parts are not in the same group (24 × 24)
SG_{cn}: The number of cases in which posture comparison be-tween the same groups is possible ($G_n – G_{n1}$)
DG_{cn}: The number of cases that an postures compare be-tween other groups is possible ($G_n – 1$)
F_c: The number of cases in which a first object is loaded

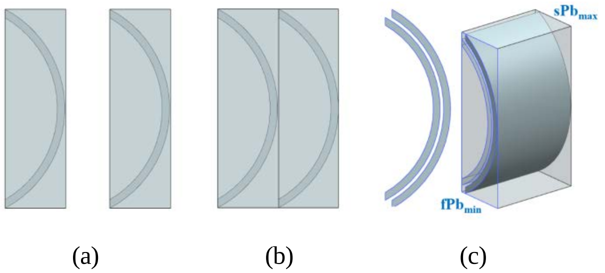


Figure 13. Computational process for the group's bounding-box volume: (a) bounding box is created for two parts, (b) compare poses of two parts, (c) group bounding box is created.

$$F_c = 6\times6(G_n\text{-}G_{n1}) + 24\times24(G_{n\text{-}1})$$
$$= 6\times6(SG_{cn}) + 24\times24(DG_{cn})$$
$$\text{Ex) } P_n\text{: } 9,\ G_n\text{: } 4,\ G_{n1}\text{: } 1,\ F_c = ?$$
$$F_c = 6\times6(4 - 1) + 24\times24(4 - 1) = 108 + 1728= 1836$$

(2) Orientation decision of the N+1th part to be loaded

Compared to deciding the orientation of the first part, the way in which it decides the orientation of the N+1th part has some different points. First, the part being loaded has to go through a collision check between the container and the parts already loaded. Second, it needs to find a comparison part (among those that are already loaded) in the container, in order to decide the orientation and loading direction of the part being loaded. The comparison part and loading direction are decided through the "Loading Algorithm." Once the comparison part is found, the orientation of the N+1th part can be decided by the same method of deciding the orientation of the first part. Nevertheless, since the orientation of one part is fixed by comparing two parts, the number of cases in which the loading orientation of the N+1th part is decided through a posture comparison is either 6 or 24. In case the part for comparison and the part that is to be loaded are in the same group, and the loading direction is the –X direction, when the part for comparison is loaded, the orientation comparison occurs in the same group. Thus, the optimized orientation can be found after 6 orientation comparisons. If the loading direction is not the –X direction although the parts are in the same group, 24 orientation comparisons should be carried out (as opposed to none earlier). Twenty-four orientation comparisons between parts belonging to adjacent groups must be performed regardless of the loading direction.

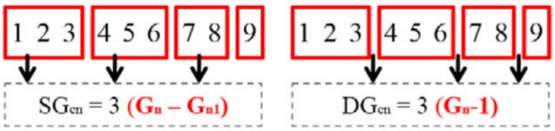


Figure 14. The number of possible orientation comparisons for loading the first part.

4.4 The loading algorithm

The loading algorithm decides the loading direction, and the best location of the part, once the loading orientation is decided. The part is loaded in the order of Bottom-left-back-fill (BLBF), and the loading direction is decided through the "Find face algorithm" that employs the BLBF method.

(1) The bottom-left-back-fill (BLBF) algorithm

BLBF is an extended version of bottom-left-fill (BLF) [20]. Loading is the process of finding the optimized location, by moving the part from the initial position, in order to avoid collision with the loaded parts and the container. The first loaded part has its initial position at the bottom-left-rear corner of the container; therefore, it is better that the part is moved towards the bottom-left-rear corner, to find the best position. If the part is not the first to be loaded, the initial position is decided by the earlier loaded part; thus, the direction in which the optimized location of the part is found, differs from the direction of loading the earlier part. The following situation is explained by the find-face algorithm. The loading encounters failure on some occasions, e.g. if there is

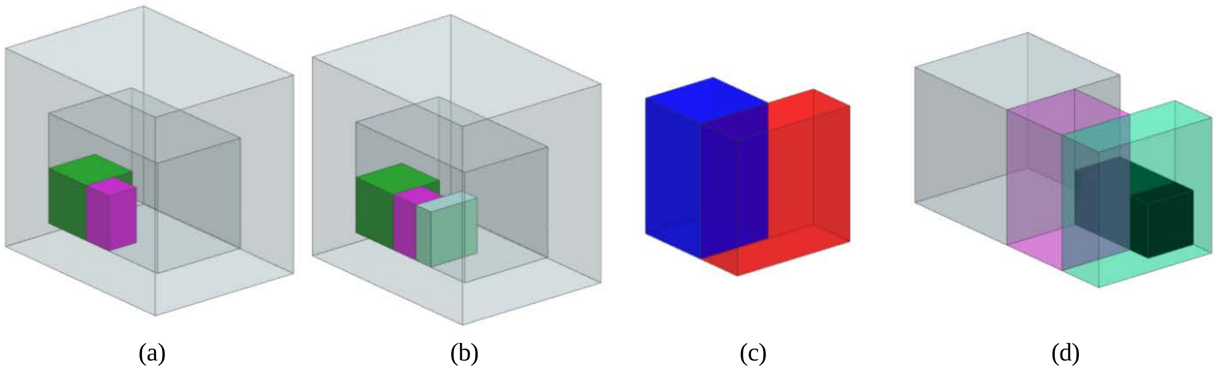


Figure 15. Exception handling procedure: (a) load part 1 and 2, (b) load the 3rd part, (c) example of handling the exception (Y-axis of it is longer than that of the loaded part), (d) the reason of the blue part in C is being handled as an exception (it influences the load of yellow part).

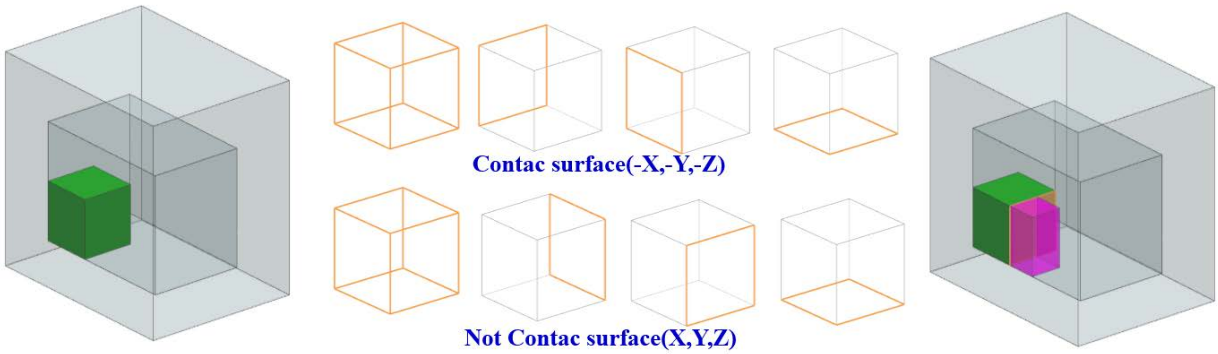


Figure 16. Actual load face and possible load face.

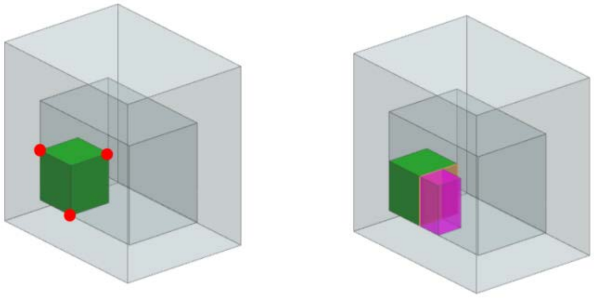
no loadable space for the part being loaded. Even if the loading succeeds, if the next loaded part is affected, the loading is considered as fail. In Figure 15, part 3 is loaded after part 2, and part 3 affects the yellow part that is to be loaded afterwards. This case occurs when part 3 has a greater value in the Y direction than part 2; this case is considered a failure, even though the loading succeeds.

(2) The find-face algorithm

The find-face algorithm starts as soon as the first part is loaded. The loading algorithm loads the part by the BLBF method, and the next part cannot be loaded at the location of faces of the bounding box of the loaded part with (-1,0,0), (0,-1,0), (0,0,-1). In contrast, the face with (1,0,0), (0,1,0), (0,0,1) can be employed for loading the next part, and Figure 16 explains how. The process of the find-face algorithm is as follows.

The priority is decided by the BLBF method, in the order of X(1,0,0), Y(0,1,0), Z(0,0,1) of the face vector. The priorities among the same vectors are determined by the face closest to the bottom left corner of the container's bounding box, as explained in Figure 17.

The movement of the part that is loaded is opposite to that of the first part, viz. from the top-right-front to the bottom-left-rear. Loading grouped parts in the same direction yields good loading efficiency; thus, the chosen face moves until it does not collide with the parts loaded in the –X(-1,0,0) direction. In this manner, the optimized position is found. In this paper, to distinguish the loading directions, if the direction of the face is decided, the X loading direction of the part is the X face, and the Y, Z directed face are the Y face, Z face, respectively.



(a) (b)

Figure 17. Example of moving the part to be loaded to the initial position, after scanning the faces: (a) the initial location of the part that is going to be loaded, (b) example of parts loaded at the initial location.

5. Implementation

The algorithm is developed using the UG OPEN API of UG NX4. The composition of the program and the functions of each module are presented in Figure 18. The GUIs for executing the program consist of an execution button, form-similarity input, total number of parts, and total volume of parts. Figure 19 displays the execution dialog. Upon initial execution, the total number of parts and the total volume of parts are calculated and presented; the user presses the Run button, after inputting the form similarity. After execution terminates, the result appears as a dialog, as shown in Figure 19.

The container form implemented in this program is the 3D free-form. The payload (parts) shape is 3D free-form, and collision checking is non-overlapped. The orientation comparison of parts is based on 90 degrees, and the efficiency is evaluated on the basis of the number of loaded parts, part

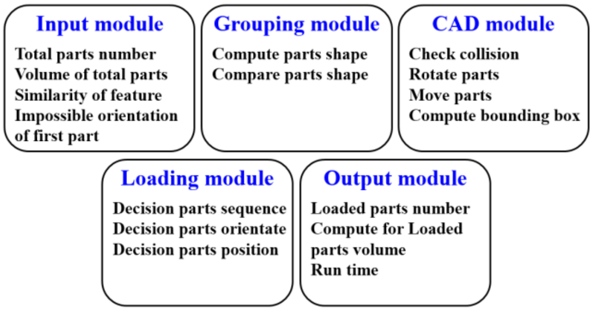


Figure 18. Composition of the program modules.

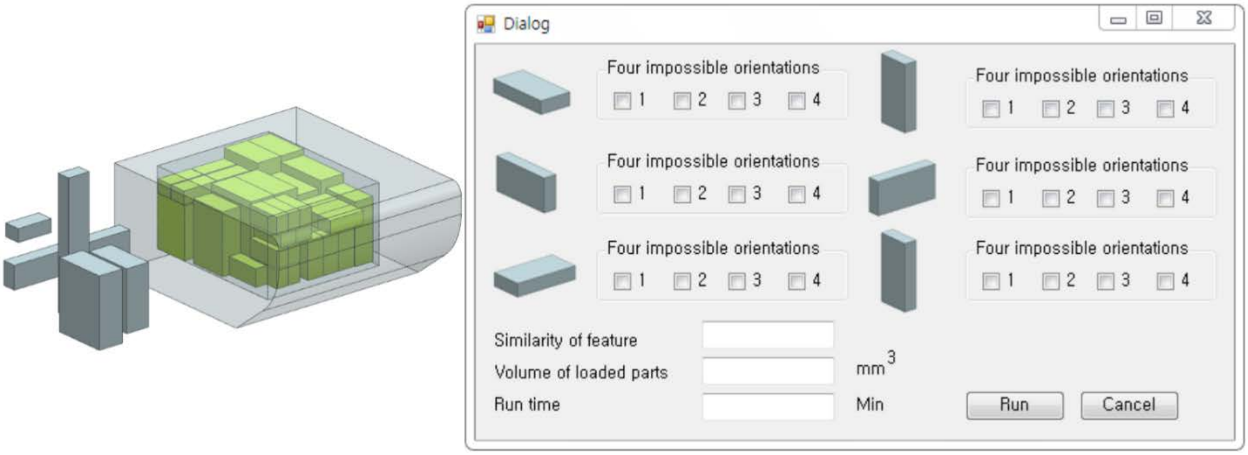

Dialog
Four impossible orientations
□ 1 □ 2 □ 3 □ 4
Four impossible orientations
□ 1 □ 2 □ 3 □ 4
Four impossible orientations
□ 1 □ 2 □ 3 □ 4
Four impossible orientations
□ 1 □ 2 □ 3 □ 4
Four impossible orientations
□ 1 □ 2 □ 3 □ 4
Four impossible orientations
□ 1 □ 2 □ 3 □ 4
Similarity of feature
Volume of loaded parts
mm3
Run time
Min
Run
Cancel

Figure 19. Dialog for 3D packing.

density, and operation time. To test the algorithms developed in this paper, problems involving 6, 10, and 16 boxes have been considered.

5.1 Bulk containers (SAE, 200. SAE standard J1100)

The bulk container application is in accordance with the SAE standard J1100, and the result is compared with the solution by Santosh and Georges [20]. The SAE standard problem is how to load the maximum amount of materials in a car trunk. The composition of the parts of SAE standard J1100 is as in Table 2, and the luggage capacity is defined by SAE J1100 [16]. The result of applying the method of this paper is shown in Figure 20.

5.2 Engineering containers

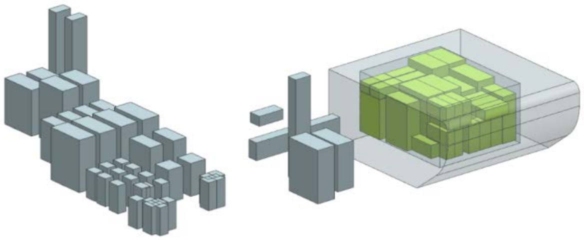
The engineering-container problem is examined under the assumption of fixtures and jigs that hold the parts in the container. The engineering-container problem deals with the maximum possible number of a single type of part that can be loaded in a given container. The jig and the fixture that holds the parts diminish the loading space, although the capacity of the container is huge. In this problem, the jig and the fixture are replaced by an imaginary boundary, and the distances between parts are randomly set as 3mm. Figure 21 and 22 show the results.

Table 2. Standard luggage set.

Luggage piece description	No. pieces/set	Size (in mm) L × W × H	Liters
Men’s two-suiter	4	610 × 229 × 483	67
Small overnight	4	457 × 165 × 330	25
Pullman	2	660 × 229 × 406	61
Wardrobe	2	533 × 216 × 457	53
Train case	2	381 × 203 × 229	18
Large overnight	2	533 × 178 × 356	34
Golf bag	2	1143 × 204 × 127	43
H-box	20	152 × 114 × 325	5.6
Total	38		

6. Conclusions

In this paper, a packing algorithm has been developed to solve the optimization problem of packing regarding a 3D model, wherein the efficiency rapidly decreases, due to the complexity of form, resulting in excessive operation times. Further, it has been implemented on a 3D CAD modeling system. The packing algorithm is composed based on group-technology methodology, and an analysis of the work performance of workers. The developed program is capable of similarity comparison of objects, interference awareness and execution between objects, position decisions regarding objects, orientation comparisons between objects, and loading simulation. The efficiency of the algorithm is examined, and has been verified with the solution, using a genetic algorithm. Through the 3D packing method using the group-technology method developed in this paper, storage optimization and loading and unloading ease can be examined; further, various engineering activities, such as work performance analysis, can be carried out, using the 3D model.



(a) (b)

Figure 20. The SAE standard J1100 problem: (a) SAE J1100 parts, (b) a picture shows the packing program is applied.

Table 3. Comparison of the proposed method with the genetic algorithm [20] for SAE standard J1100.

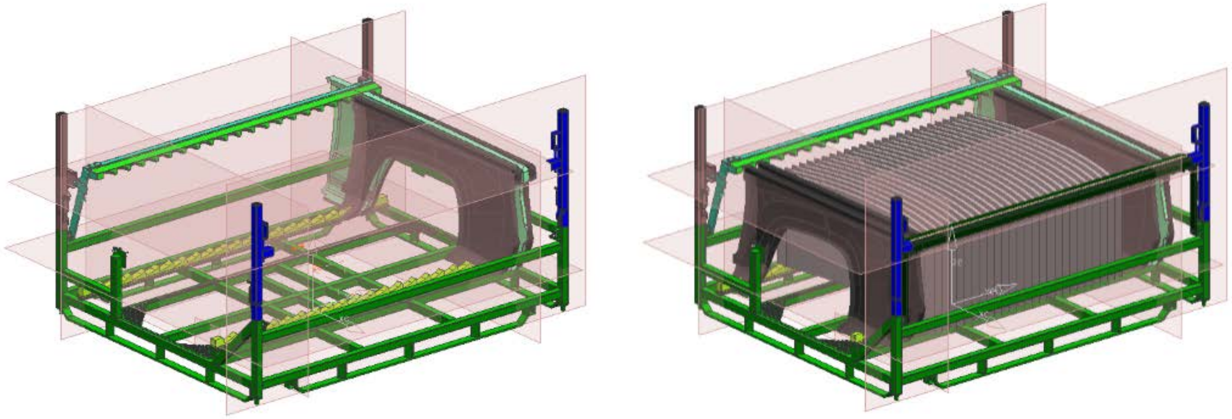
	Container volume	Total parts volume	Efficiency	Loaded pieces	Run time (min)
The proposed method	826,447,063.3	872,199,422	0.8138	35	27
Genetic algorithm	826,447,063.3	872,199,422	0.6974	21	68

Table 4. Result for engineering-container problem 1.

Container volume	Total parts volume	Loaded volume	Efficiency	Loaded pieces	Run time (min)
1,545,464,392	Infinity	646,779,432	0.42	26	6

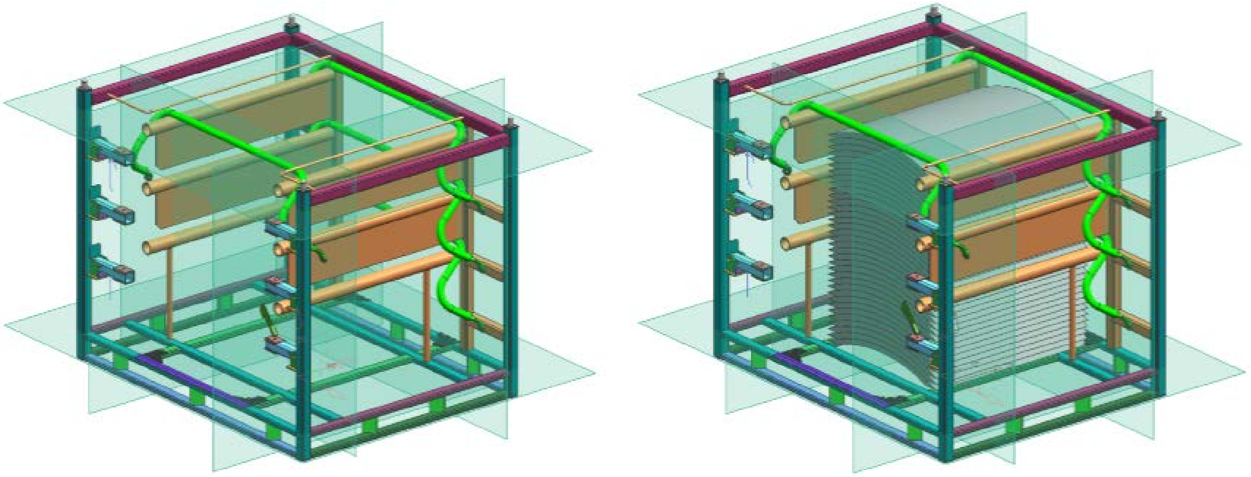
Table 5. Result for engineering-container problem 2.

Container volume	Total parts volume	Loaded volume	Efficiency	Loaded pieces	Run time (min)
779,339,050	Infinity	524,671,035	0.67	35	7



(a) (b)

Figure 21. Engineering-container problem 1: (a) engineering container, (b) a picture shows the packing program is applied.



(a) (b)

Figure 22. Engineering-container problem 2: (a) engineering container, (b) a picture shows the packing program is applied.

Acknowledgments

This research work is inspired by Mr. Erle Robertson and Mr. Steven Smith from Global Container Center, General Motors. The authors want to give very special thanks for their help and advice.

References

[1] Althaus E, Baumann T, Schömer E, Werth K. Trunk packing revisited. LNCS. 2007; 4525: 420-430.

[2] Beasley JE. An exact two-dimensional non-guillotine cutting tree search procedure. Operations Research. 1985; 33(1): 49-64.

[3] Cagan J, Shimada K, Yin S. A survey of computational approaches to three-dimensional layout problems. Computer-Aided Design. 2002; 34(8): 597-611.

[4] Cagan J. Ding Q. Automated trunk packing with extended pattern search. SAE Technical Paper. 2003; No. 2003-01-0671.

[5] Szykman S, Cagan J. Synthesis of optimal nonorthogonal routes. Journal of Mechanical Design. 1996; 118(3): 419-424.

[6] Deutsches Institut für Normung. Automotive engineering, maximum speed, acceleration and other terms, definitions and tests. DIN 70020. 1993.

[7] Eisenbrand F, Funke S, Karrenbauer A, Reichel J, Schömer E. Packing a trunk: now with a twist!. In: Proceedings of the 2005 ACM Symposium on Solid and Physical Modeling; 2005 Jun 13-15; p. 197-206.

[8] Eisenbrand F, Funke S, Reichel J, Schömer E. Packing a trunk. In: Algorithms-ESA 2003, 11th Annual European Symposium; 2003 Sep 16-19; Budapest, Hungary; p. 618-629.

[9] Hong LS, Lee KW, Kim SC. Similarity comparison of mechanical parts. CAD/CAM Conference; 2000 Aug 4-11; Korea; p. 316-326.

[10] Hooke R, Jeeves TA. "Direct search" solution of numerical and statistical problems. Journal of the ACM (JACM). 1961; 8(2): 212-229.

[11] Karrenbauer A. Packing boxes with arbitrary rotations [master's thesis]. Universit ät des Saarlandes, Saarbrücken. 2004.

[12] Kolli A, Cagan J, Rutenbar R. Packing of generic, three-dimensional components based on multi-resolution modeling. In: Proceedings of the 1996 ASME Design Engineering Technical Conferences and Computers in Engineering Conference: Design Automation Conference; 1996 Aug 18-22; 96-DETC/DAC-1479.

[13] Landon MD, Balling RJ. Optimal packaging of complex parametric solids according to mass property criteria. Journal of Mechanical Design. 1994; 116(2): 375-381.

[14] Reichel J. Combinatorial approaches for the Trunk packing problem [dissertation]. Universität des Saarlandes, Saarbrücken. 2007.

[15] Rieskamp J. Automation and optimization of Monte Carlo based trunk packing [master's thesis]. Universität des Saarlandes, Saarbrücken. 2005.

[16] Society of Automotive Engineers. Motor vehicle dimensions. SAE J1100. 2005.

[17] Scheithauer G, Terno J. A branch-and-bound algorithm for solving one-dimensional cutting stock problems exactly. Applicationes Mathematicae. 1995; 23(2): 151-167.

[18] Szykman S, Cagan J. A simulated annealing-based approach to three-dimensional component packing. Journal of Mechanical Design.1995; 117(2A): 308-314.

[19] Szykman S, Cagan J. Constrained three-dimensional component layout using simulated annealing. Journal of Mechanical Design. 1997; 119(1): 28-35.

[20] Santosh T, Fadel G, Fenyes P. A fast and efficient compact packing algorithm for free-form objects. In: ASME 2008 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference; 2008 Aug 3-6; New York, NY; p. 543-552.

[21] Torczon V, Trosset MW. From evolutionary operation to parallel direct search: Pattern search algorithms for numerical optimization. Computing Science and Statistics. 1998; 29: 396-401.