

Optimization of a robot-served cart capacity using the three-dimensional single bin packing problem

A-Ra Khil · Kang-Hee Lee

Published online: 12 January 2014
© Springer Science+Business Media New York 2014

Abstract Given a set of rectangular-shaped items such as dishes, cups, saucers, or forks and a rectangular tray of a cart, the three-dimensional single bin packing problem (3D-BPP) involves orthogonally packing a subset of the items within the tray. If the value of an item is given by its volume, the objective is to maximize the covered volume of the tray. Thus, this paper aims to optimize the transport capacity of a serving robot carrying a cart. This experiment, the first of its type, proves the feasibility of this endeavor efficiently.

Keywords Three-dimensional single bin packing problem · Three-dimensional knapsack problem · Serving robot · Optimization · Cart · Tray

1 Introduction

James Kuffner at Google introduced the term “Cloud Robotics” to describe a new approach to robotics that takes advantage of the Internet as a resource for massively parallel computation and sharing of vast data resources [4]. The robot specialized company, Willow Garage, is funded from Google, has received attention to introduce “Cloud Robotics”, is exploiting the cloud resource of Google, for the services such as the context-awareness and the object manipulation that are required to process enormous data. This paper, based on the concept of Cloud Robotics, would rather focus on optimizing the task plan-

A.-R. Khil
School of Computer Science and Engineering, College of Information Technology, Soongsil University,
511 Sangdo-Dong, Dongjak-Gu, Seoul, Republic of Korea
e-mail: ara@ssu.ac.kr

K.-H. Lee (✉)
Global School of Media, College of Information Technology, Soongsil University, 511 Sangdo-Dong,
Dongjak-Gu, Seoul, Republic of Korea
e-mail: kanghee.lee@ssu.ac.kr

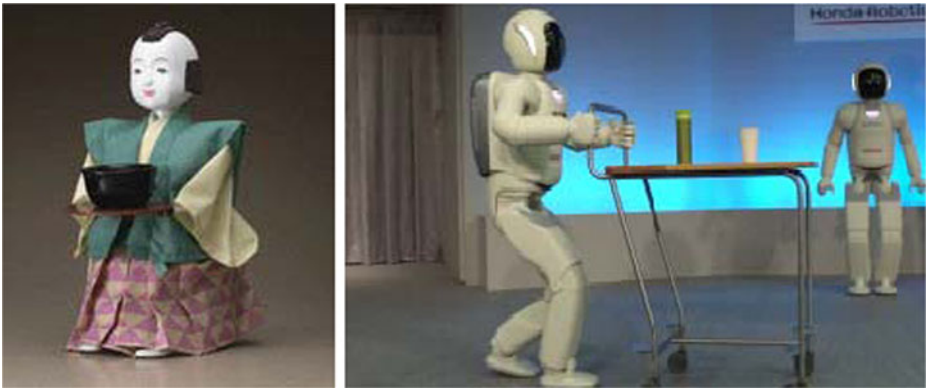


Fig. 1 World's first tea serving robot: Karakuri (*leftside*) and world's best universal serving robot: ASIMO (*rightside*)

ning in order to serve effectively dishes by using a cart or tray than do simple motion planning [7].

When we have dinner at a corner table in a crowded restaurant, we want to be served by a kind and experienced waiter/waitress. This person should also be strong, as he/she serves dinner on a heavy dish cart or a tray filled with the ordered food, which can reduce the waiting time as a result. This mechanism can also be applied to a serving robot. If a stronger serving robot carries twice the number of dishes compared to other normal serving robots, it will be able to save time when multiple dishes of food are served. The longer the time taken and the greater the distance covered, the more important storage efficiency becomes when a robot serves in this manner.

Thus, this paper formulates this problem as a three-dimensional finite bin packing problem (3D-BPP) and aims to optimize the amount of a dish cart capacity which the serving robot can carry at one time. The rest of the paper is composed as follows. Sections 1.1 and 1.2 introduce the existing works related to the contribution of this paper: serving robots and 3D-BPP. In Section 2, we formulate the optimization problem of a cart storage using the 3D-BPP. Section 3 describes the structure of the ONEBIN algorithm [2, 5, 6] for finding possible positions for placing items. The experiments in Section 4 are performed to demonstrate the feasibility of the proposed scheme in Sections 2 and 3. The empirical and theoretical analyses follow to investigate its characteristics. Concluding remarks follow in Section 5.



Fig. 2 PR2 can reach for a variety of bottles, grasp, and places them on a cart using its 'Pick and Place' operation based on the cloud resources



Fig. 3 PR2 is clearing up tables with a cart (*leftside*: stacked homogeneous plates and *rightside*: stacked heterogeneous plates, bowls, and cups)

1.1 Paradigm shift of serving robot

The world's first serving robot is the fascinating Karakuri, which serves tea as shown in Fig. 1. It was designed nearly four centuries ago and today remains a remarkable example of Japan's keen sense of robotics [9]. Currently, the all-new ASIMO, developed by Honda, can perform serving tasks, such as carrying a dish cart, picking up a glass bottle and twisting off the cap, or gently holding a soft paper cup while pouring a liquid into it, with great dexterity [1].

While the ASIMO is a kind of standalone robot, the PR2 [7, 8], developed by Willow Garage is a representative cloud robot which performs the dexterous tasks similar to the ASIMO's as shown in Figs. 2, 3 and 4. Thus while state-of-the-art serving robots focus only on mobility, manipulability, and types of foods to be served, we aim to increase the transport or storage capacity of the dish cart carried by a serving robot. Hence, it is formulated as a three-dimensional finite bin packing problem (3D-BPP) in order to optimize the amount of dish cart storage which the serving robot can carry at one time in the following section.

1.2 Three-dimensional single bin packing problem

Figure 3 shows the examples of how a serving robot can take advantage of the 3D-BPP concept. Here each item is considered as a rectangular-shaped item. For the detail, the next section describes the 3D-BPP and models a cart problem using it. The Three-Dimensional

Fig. 4 A serving robot carrying a dish cart. The depth of the tallest item(s) becomes D

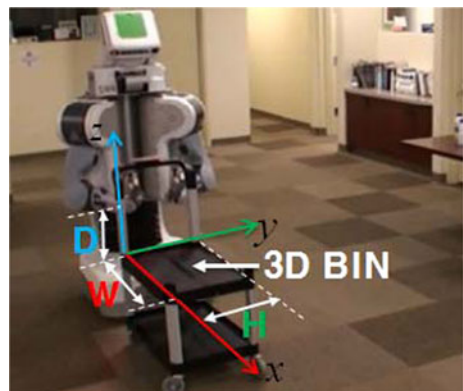


Table 1 Item list


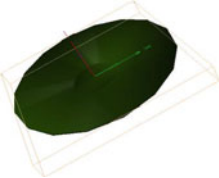
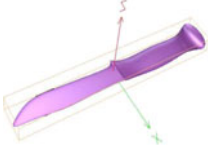






j	3D shape	w_j	h_j	d_j	Description
1		5	32	99	FORK
2		25	25	3	PLATE
3		4	22	99	KNIFE
4		16	16	99	SAUCERCUP
5		3	12	99	TEASPOON
6		10	7	99	BOWL
7		7	7	99	CAN
8		6	6	8	COCKTAILGLASS
9		8	8	99	COLACUP

Table 2 Parameters and lower bounds for 3D-BPP

Index	Par.	Description
1	I	Instance defined by item j and all the pieces currently assigned to bin i
2	i	Bin index
3	n	Number of items or rectangular pieces
4	J	$\{1, \dots, n\}$ A set of items
5	j	Item index or item name
6	W	Bin (tray) width
7	H	Bin (tray) height
8	D	Bin (tray) depth
9	w_j	Width of item j
10	h_j	Height of item j
11	d_j	Depth of item j
12	I'	Instance defined by the unassigned items and all the pieces currently assigned to active bins
13	c	Number of currently closed bins
14	L_0	$L_0 = \lceil \frac{\sum_{j=1}^n h_j w_j d_j}{HWD} \rceil$
15	L_1^{WH}	Refer to [6]
16	L_2^{WH}	Refer to [6]
17	L_1	$\max \{L_1^{WH}, L_1^{WD}, L_1^{HD}\}$
18	L_2	$\max \{L_2^{WH}, L_2^{WD}, L_2^{HD}\}$
19	z^*	Best incumbent solution value
20	$C(I)$	Resulting corner points
21	$\hat{C}(I)$	Set of corner points of the envelope
22	$S(I)$	$\{(x, y, z) : \forall i \in x \geq x_i + w_i \text{ or } y \geq y_i + h_i \text{ or } z \geq z_i + d_i\}$
23	$\hat{S}(I)$	Feasible $S(I)$

Bin Packing Problem (3D-BPP) is a feasible enumeration of Two-Dimensional Bin Packing Problem (2D-BPP). 3D-BPP is NP-hard in the strong sense [6].

2 Modeling a dish cart problem with a 3D-BPP

We model a dish cart capacity filing problem (DCCP) with a three-dimensional single bin packing problem (3D-BPP) [6]. This 3D-BPP consists of orthogonally packing into one bin as many items as possible. The aim is to find a feasible packing of a subset of the items on the cart (tray) that maximizes the total volume of the packed items. We assume that the origin of the coordinate system is in the left-bottom-back corner of the bin. We are given a set of rectangular-shaped items, each characterized by width $w_j \leq W$, height $h_j \leq H$, and depth $d_j \leq D$ ($j \in J = \{1, \dots, n\}$), and one three-dimensional bin (container) having width W , height H , and depth D . The items such as dishes, cups, forks, and knives are different from shapes, styles, and materials as listed in Table 1. While the depths of other

Algorithm 2D-CORNERS:

```

begin
  if  $I = 0$  then  $\hat{C}(I) := \{(0, 0)\}$  and return;
  comment: Phase 1 (identify the extreme items  $e_1, \dots, e_m$ );
   $\bar{x} := m := 0$ ;
  for  $j := 1$  to  $|I|$  do
    if  $x_j + w_j > \bar{x}$ 
      then  $m := m + 1$ ;  $e_m := j$ ;  $\bar{x} := x_j + w_j$ ;
  comment: Phase 2 (determine the corner points);
   $\hat{C}(I) = \{(0, y_{e_1} + h_{e_1})\}$ ;
  for  $j := 2$  to  $m$  do
     $\hat{C}(I) := \hat{C}(I) \cup \{(x_{e_{j-1}} + w_{e_{j-1}}, y_{e_j} + h_{e_j})\}$ ;
   $\hat{C}(I) := \hat{C}(I) \cup \{(x_{e_m} + w_{e_m}, 0)\}$ ;
  comment: Phase 3 (remove infeasible corner points);
  for each  $(x'_j) + \min_{i \in \hat{J} \setminus I} \{w_i\} > W$  or  $y'_j + \min_{i \in \hat{J} \setminus I} \{h_i\} > H$ 
    then  $\hat{C}(I) := \hat{C}(I) \setminus \{(x_j, y'_j)\}$ 
end.
  
```

Fig. 5 Algorithm 2D-CORNERS

items are equal to 99, the depths of PLATE and COCKTAILGLASS are equal to 3 ($d_2 = 3$) and 8 ($d_8 = 8$), respectively. It means that only both items can be stacked.

In this study, we use an enumerative algorithm for the exact and approximate solution of 3D-BPP that Martello, Pisinger, and Vigo [6] proposed. Therefore, the parameters including lower bounds, in Table 2 can be followed directly. The detailed theorems and proofs can be referred in [6].

Especially, the worst-case performance ratio of continuous lower bound L_0 is 1/8 if rotation of the pieces (by any angles) is allowed. It is intuitively easy to understand. L_0 and L_1 do not dominate each other, while L_2 dominates L_0 and L_1 . L_1 and L_2 are valid lower bounds of WH , WD , and HD planes of the bin volume by $B = WHD$ for 3D-Bin packing problem. They are described in Table 2 and can be referred in [6].

Fig. 6 Two-dimensional single bin filling (The envelop associated with the placed items is marked by a dashed line, and black points indicate corner points in $\hat{C}(I)$)

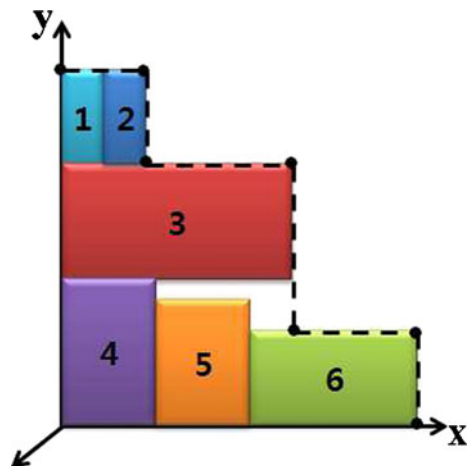
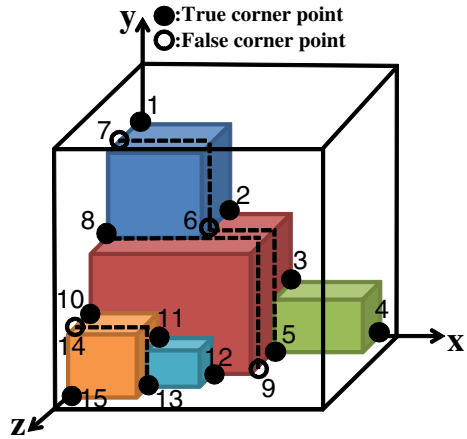


Fig. 7 Three-dimensional single bin filling (Corner points $C(I)$ are found by applying algorithm 2D-CORNERS six times on for each value of z'_k i.e., such that (1, 2, 3, 4)→(5, 6, 7)→(8, 9)→(10, 11, 12)→(13, 14)→ (15))



3 Exact algorithm for filling a single bin and approximation algorithms for optimal 3D-bin packing

This section describes the algorithm called ONEBIN [6] for finding the best filling of a single bin using items from a given set \bar{J} (refer to Sections 3.1 and 3.2). And two different heuristic algorithms called Approximation Algorithms are described for optimally packing three dimensional bin packing (refer to Section 3.3).

Initially, the bin volume is $B = WHD$ and no item is placed, so $C(0) = \{(0, 0, 0)\}$ as shown in Fig. 4. At each iteration, given the set $I \subset \bar{J}$, currently packed items, set $C(I)$ is

Algorithm 3D-CORNERS:
begin

```

if  $I = 0$  then  $\hat{C}(I) := \{(0, 0, 0)\}$  and return;
 $T := \{0\} \cup \{z_i + d_i : i \in I\}$  (comment: do not duplicate equal values in  $T$ );
sort  $T$  by increasing values, and let  $T = \{z'_1, \dots, z'_r\}$ ;
 $C(I) := \hat{C}(I_0) := 0; k = 1;$ 
while  $k \leq r$  and  $z'_k + \min_{i \in \bar{J} \setminus I} \{d_i\} \leq D$  do
  begin
     $I_k := \{i \in I : z_i + d_i > z'_k\}$ ;
    apply 2D-CORNERS to  $I_k$  yielding  $\hat{C}(I_k)$ ;
    comment: add true corner points to  $C(I)$ 
    for each  $(x'_j, y'_j) \in \hat{C}(I_k)$  do ;
      if not  $(x'_j, y'_j) \in \hat{C}(I_{k-1})$ 
        then  $C(I) := C(I) \cup \{x'_j, y'_j, z'_k\}$  ;
     $k^* := k; k = k + 1;$ 
  end
end.

```

Fig. 8 Algorithm 3D-CORNERS

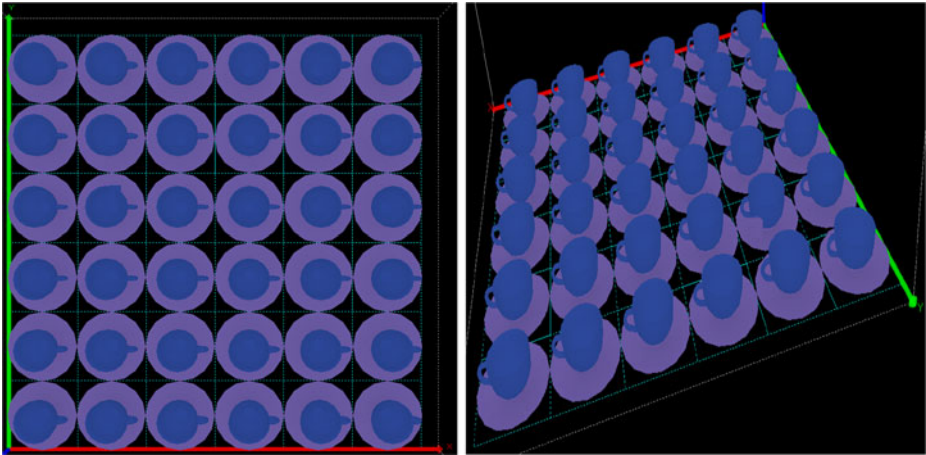


Fig. 9 Maximum packing for 36 SAUCERCUPS (*leftside: top view, right side: 3D view*)

determined through 3D-CORNERS together with the corresponding volume $V(I)$. If F is the total volume achieved by the current best filling, we may backtrack whenever

$$\sum_{i \in I} v_i + (B - V(I)) \leq F \tag{1}$$

If no more items fit into the bin (i.e., if $C(I) = 0$), we possibly update F and backtrack. Otherwise, for each position $(x_j, y_j, z_j) \in C(I)$ and for each item $(x_j, y_j, z_j) \in C(I)$ we assign the item $j \in \bar{J} \setminus I$ to this position and call the procedure recursively.

3.1 Algorithm 2D-CORNERS: determining the corner points in 2D single bin

Given an item set I , it is quite easy to find, in two dimensions, the set $\hat{C}(I)$ of corner points of the envelope associated with the feasible region $\hat{S}(I)$ defined by the $x - y$ faces of the items in I . There are two preconditions. First, any packing of a bin can be replaced by an equivalent packing where no item may be moved leftward, downward, or backward. Second, an ordering of the items in an optimal solution exists such that, if $i < j$, $x_i + w_i \leq x_j$ or $y_i + h_i \leq y_j$ or $z_i + d_i \leq z_j$. Following these, let us order the items according to their end points $(x_j + w_j, y_j + h_j)$ so that the values of $y_j + h_j$ are nonincreasing, breaking ties by the largest value of $x_j + w_j$. The following algorithm (refer to Fig. 5) for determining the corner points set consists of three phases.

Consider the example in Fig. 6. The extreme items are 2, 3, and 6, and the resulting corner points are indicated by black dots; Phase 3 could remove some of the first and/or last corner points. The time complexity of 2D-CORNERS is $O(|I|)$, plus $O(|I| \log |I|)$ for the initial item sorting, i.e., plus $O(n)$ plus $O(n \log n)$. Assume that the resulting corner points are $\hat{C}(I) = \{(x'_1, y'_1), \dots, (x'_t, y'_t)\} \neq \emptyset$. Then the area occupied by the envelope is

$$A(I) = x'_1 H + \sum_{i=2}^t (x'_i - x'_{i-1}) y'_{i-1} + (W - x'_t) y'_t \tag{2}$$

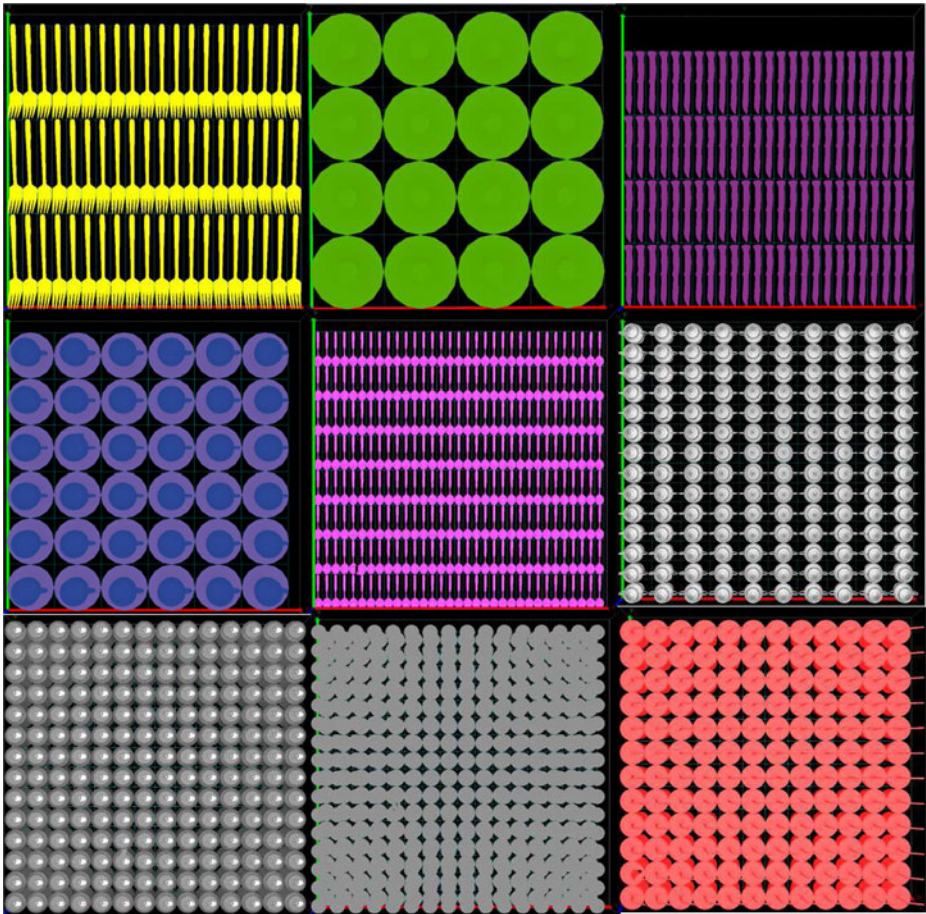


Fig. 10 Maximum packing for each item from FORK to COLACUP (Table 1) using 2D-BPP

where the first (resp. last) term is nonzero whenever the first (resp. last) corner point found in Phase 2 has been removed in Phase 3. In the special case where $\hat{c}(I) = 0$, we obviously set $A(I) = WE$.

3.2 Algorithm 3D-CORNERS: finding possible positions for placing an item

Algorithm 2D-CORNERS set $C(I)$ of corner points in three dimensions, where I is the set of three-dimensional items currently packed into the bin. One may apply the algorithm for $z = 0$ and for each distinct z coordinate where an item of I ends, by increasing z values. For each such coordinate z' , 2D-CORNERS can be applied to the subset of those items $i \in I$ that end after z' , i.e., such that $z_i + d_i > z'$ adding the resulting corner points to $C(I)$. However, as shown in Fig. 7, some false corner points like 6, 7, 9, and 14 can be obtained because they are corner points in the two-dimensional cases. To remove such points, the following 3D-CORNER algorithm (Fig. 8) is used such that no corner point will be generated inside

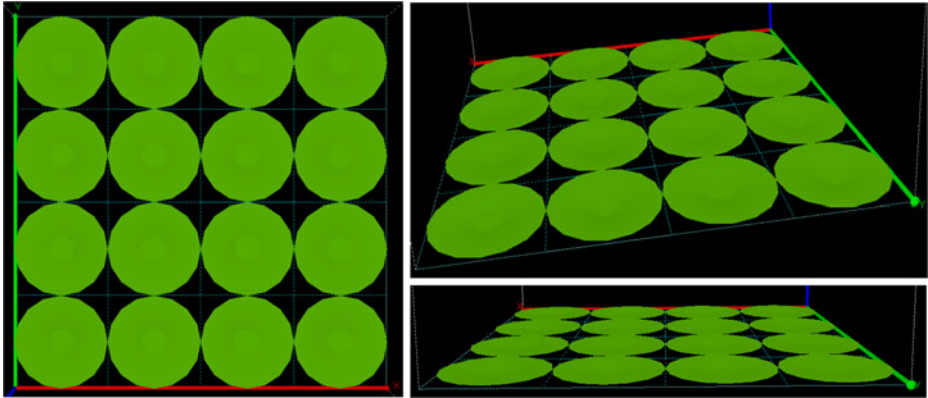


Fig. 11 Maximum packing for 16 PLATES by 2D-BPP

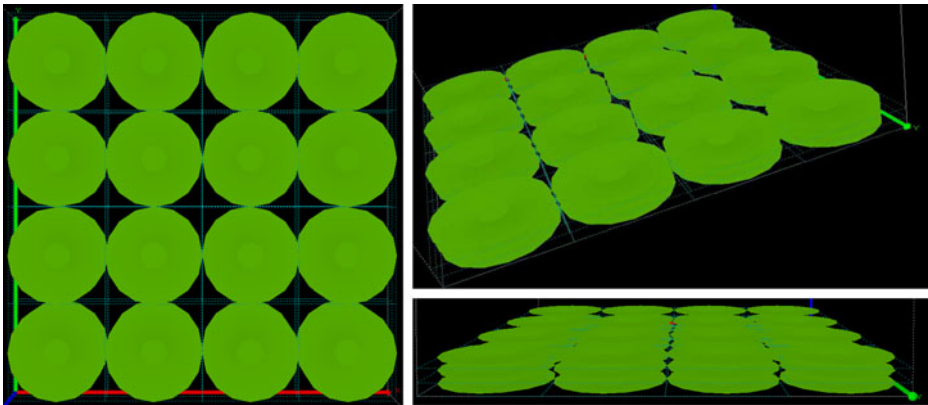


Fig. 12 Maximum packing for 48 PLATES by 3D-BPP

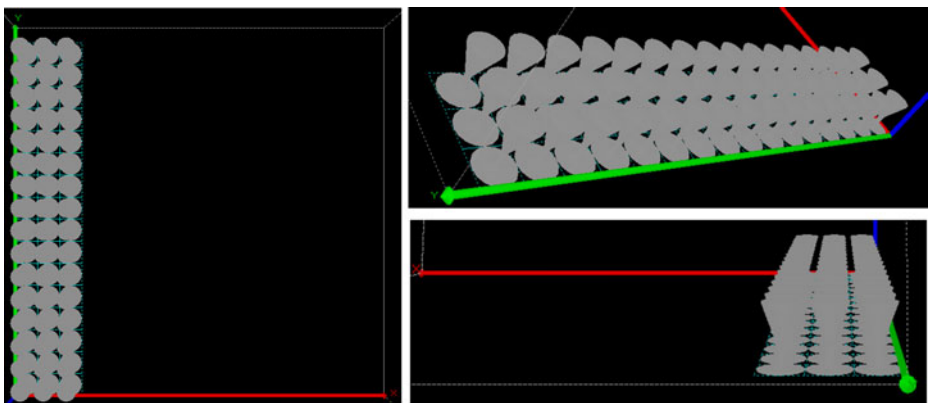


Fig. 13 Optimal packing for 48 COCKTAILGLASS by 2D-BPP

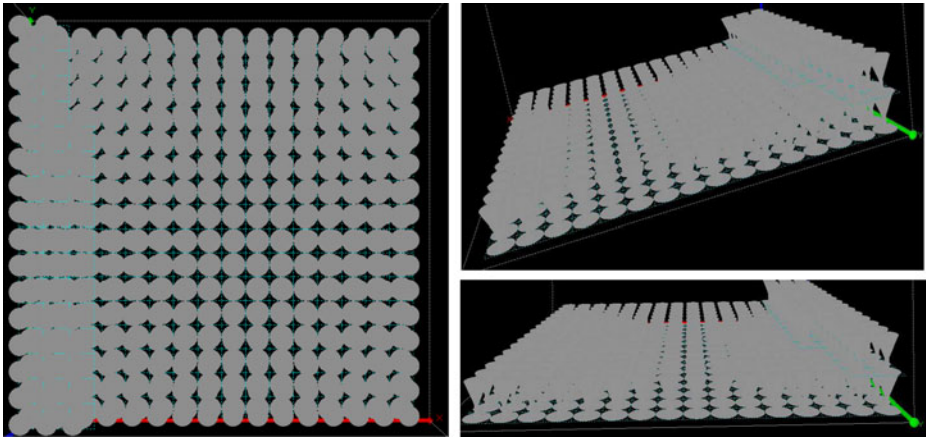


Fig. 14 Maximum packing for 300 COCKTAILGLASS by 3D-BPP

the three-dimensional envelope. The volume $V(I)$ occupied by the envelope associated with I is

$$V(I) = \sum_{k=2}^{k^*} (z'_k - z'_{k-1}) A(I_{k-1}) + (D - z'_{k^*}) A(I_{k^*}) \tag{3}$$

where the last term is nonzero whenever $k^* < r$.

3.3 Approximation algorithm: optimal 3D-BPP

To obtain a good upper bound at the root node of the branching tree and to limit the number of executions of ONEBIN, two complementary heuristics are used. The first heuristic, H1, is based on a layer building principle derived from shelf approaches used by several authors for 2D-BPP [2, 3]. The second heuristic, H2, repeatedly fills a single bin. Let \bar{J} be the set of items to be packed [6]. Heuristic H2 initializes a set T to \bar{J} and sorts it by nondecreasing volume. Then H2 iteratively applies ONEBIN to T and removes the packed items from it until T becomes empty.

4 Experiments

The experiments are performed to demonstrate the feasibility of the proposed scheme. In these test problems, the sizes of the pieces to be packed were varied and the bin sizes (tray) were always same as $W \times H \times D = 100\text{ cm} \times 100\text{ cm} \times 100\text{ cm} = 1000000\text{ cm}^3$. Accordingly, all experiments in this paper are measured in *cm*. The experiment was implemented in visual C++ 2008, OpenGL 2.0, and WinXP. It works well on an Intel PC with core i7 3.0 GHZ CPU and 8 GB RAM. We verify three classes of experiments in the following.

The first experiment is to pack a two-dimensional bin (tray) with each single item as shown in Figs. 9 and 10. The goal is to find a feasible arrangement of a subset of the items (totally 9 kinds of items) on the tray that maximizes the total value of the packed item using 2D-BPP. For example, the 2D tray is tightly packed with SAUCERCUPS ($j = 4$) in Fig. 9 and they are optimally maximizing the covered area of the tray. Figure 10 shows that there are 68 FORKS (0.13 s), 16 PLATES (0.001 s), 100 KNIFES (0.02 s), 36

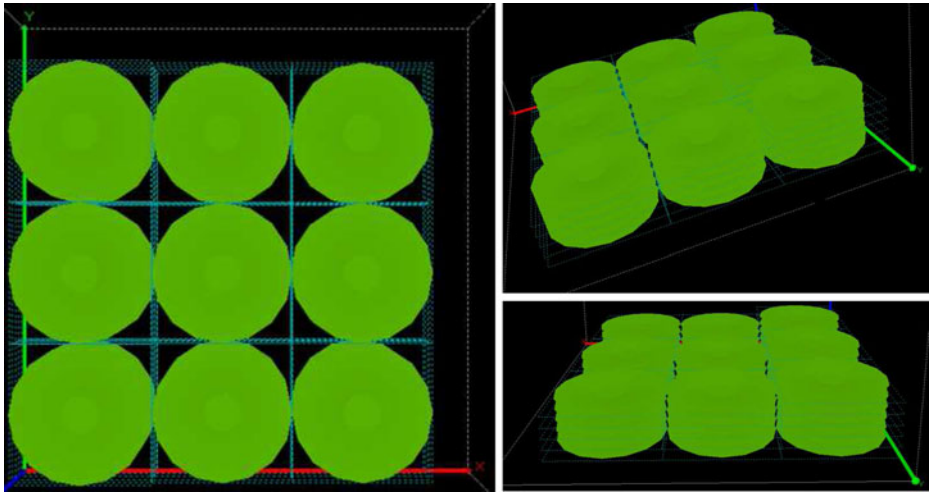


Fig. 15 Maximum packing for 48 PLATES by 3D-BPP. The size of PLATE is equal to $w_j = 30, h_j = 30, d_j = 3$

SAUCERCUPS (0.001 s), 264 TEASPOONS (0.02 s), 140 BOWLS (0.01 s), 196 CANS (0.01 s), 256 COCKTAILGLASSES (0.02 s), and 144 COLARCUPS (0.01 s) and they are optimized at each finite bin. As like taken 0.13 s, 0.001 s, \dots , 0.06 s for each item, it is feasible for a serving robot to plan the real-time task in various situation optimally.

The goal of the second experiment is to verify that the 3D-BPP is definitely superior to the 2D-BPP. Figures 11, 12, 13 and 14 show the 3D-BPP would allow a serving robot to effectively serve lots of stackable items such as PLATES and COCKTAILGLASSES at once.

Finally, the third experiment makes a change in the item size or the tray size. The 3D-BPP shows a stable and high performance in spite of the bigger PLATES (refer to Fig. 15) than the previous ones in Fig. 12. Likewise, it can cope with large numbers of stackable items and different size of trays as shown in Fig. 16.

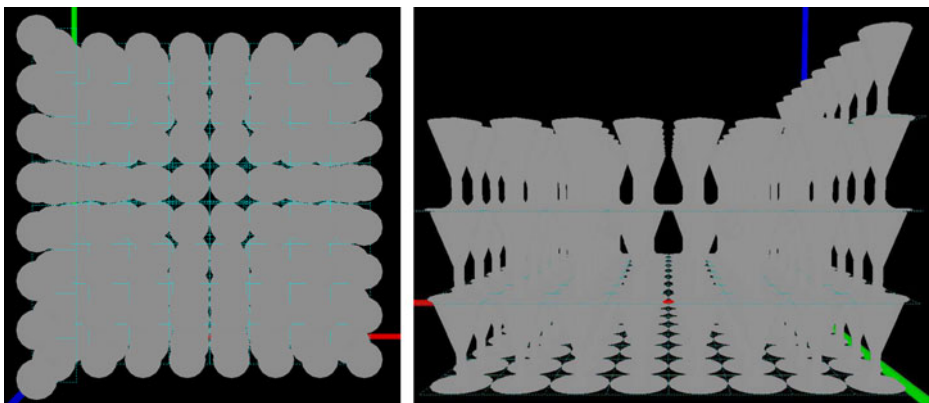


Fig. 16 Maximum packing for 200 COCKTAILGLASS by 3D-BPP. The bin size is equal to $W \times H \times D = 50 \times 50 \times 50 = 125000$

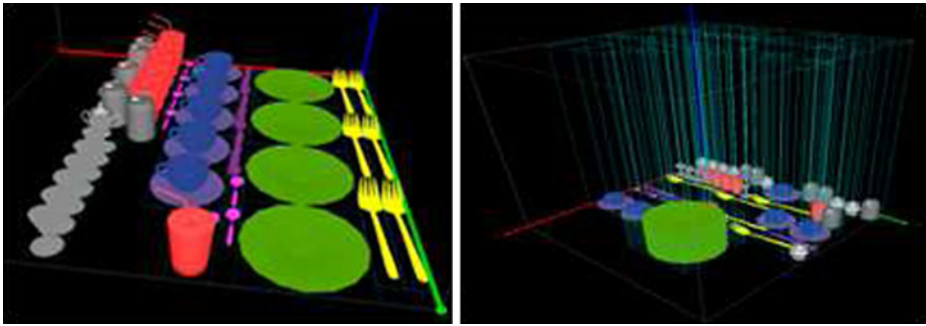


Fig. 17 Maximum packing for *leftside*: 40 items-2D-BPP and *rightside*: 48-items-3D-BPP

Figure 17 shows general cases of randomly generated problem instances having different combinations of items. Compared to the 2D-BPP result, the 3D-BPP result shows less covered area of the tray in spite of larger numbers of items because PLATEs were stacked.

By using the proposed mechanism, we show that limited cart capacity can be filled with various items. Hence, both serving and cleaning time by serving robot can be shortened.

5 Conclusions

We presented the world's first 3D-BPP applied to the situation of the maximization of the capacity of a cart for a serving robot. By using this concept, more effective service can be offered by various types of serving robots. For the future works, if evolutionary knapsack algorithms or more expert systems of bin packing problem are applied and quantitatively compared to the hybrid 3D-BPP, more feasible and realistic solution can be expected.

Acknowledgments This research was supported by the National Research Foundation of Korea Grant funded by the Korean Government (NRF-2013-S1A5A8020988).

References

1. ASIMO. <http://world.honda.com/ASIMO>
2. Berkey JO, Wang PY (1987) Two dimensional finite bin packing algorithms. *J Oper Res Soc* 38:423–429
3. Chung FKR, Garey MR, Johnson DS (1982) On packing two-dimensional bins. *SIAM J Algebr Discret Methods* 3:66–76
4. Kuffner JJ (2010) Cloud-enabled robots. In: *IEEE-RAS international conference on humanoid robots*. Nashville
5. Martello S, Vigo D (1998) Exact solution of the two-dimensional finite bin packing problem. *Manag Sci* 44:388
6. Martello S, Pisinger D, Vigo D (2000) The three-dimensional bin packing problem. *Oper Res* 48:256–267
7. PR2 Cleans Up with a Cart. <http://www.willowgarage.com/pages/pr2/overview>
8. The 2012 PR2 Workshop in Freiburg. <http://www.youtube.com/watch?v=kvahil80ayw>
9. 400 year old tea-serving robot kit. <http://www.boingboing.net/2005/10/29/400-year-old-teaserv.html>



A-Ra Khil received the B.S. degree in Computer Science from Ewha Women's University in 1987, the M.S. and PhD degrees in Computer Science from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 1990 and 1997, respectively. Since 1995, she has been a Senior Engineer in Serome Ltd., Korea. Since 2003, she has been a member of Board of Directors in Dialpad Ltd., USA. Since 1997, she is with School of Computer Science and Engineering, Soongsil University, Seoul, Korea. Her current research interests include the real-time Systems, emotional robotics, ubiquitous communications, wireless sensor networks, and embedded Operating system. Contact her at ara@ssu.ac.kr.



Kang-Hee Lee received the B.S., M.S. and PhD degrees in Electrical Engineering and Computer Science from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 1999, 2001, and 2006, respectively. Since 2006, he has been a Senior Engineer in Digital Media & Communication Research Center, Samsung Electronics Company, Ltd., Korea. He has been a dispatched researcher in the Robotics Institute, Carnegie Mellon University in 2008. Since moving to Soongsil University in 2009, he is with Global School of Media, Soongsil University, Seoul, Korea. His current research interests include the areas of ubiquitous robotics, evolutionary robotics, emotional robotics, media robotics, cognitive task planning system, and knowledge-based reasoning system. Contact him at kanghee.lee@ssu.ac.kr.