2nd International Conference on Engineering Optimization
September 6-9, 2010, Lisbon, Portugal

On the three-dimensional bin packing using rotations

Ana Maria de Almeida and Marisa Figueiredo

CISUC - Centre for Informatics and Systems of the University of Coimbra
and Department of Mathematics of the University of Coimbra
Largo D.Dinis, Apartado 3008, 3001–454 Coimbra, Coimbra, Portugal
Email: amca@mat.uc.pt; mbfig@dei.uc.pt

Abstract

One of the most interesting variations of the problem of the 3-Dimensional Bin Packing Problem (3BPP) is the determination of the minimum number of three-dimensional rectangular bins that are required for orthogonally allocating a given set of three-dimensional rectangular items without overlap and minimizing the occupied space: the 3BPP–min problem. This is, of course, yet another NP-Hard multi-criteria combinatorial optimization. One of the most obvious applications for this problem is the one that occurs daily in warehouses management systems, either by using manual or automatic packing.

We present a new approach for the 3BPP–min problem where 90 degrees rotations are allowed in order to allow for a more compact packing. Most of the known heuristic solutions for this type of packing are based on the well-known works of Martello, Pisinger and Vigo. Boschetti has recently introduced new lower bounds specifically tailored for packing using the possibility of rotations of items that we used for designing the new heuristic algorithm. This algorithm uses both this new lower bounds and the theory of non-dominated solutions for deciding on the best packing for a 3BPP–min instance. Computational results show the effectiveness of the new approximation algorithm that shows to be faster and achieve lower occupation of space, thus, better compaction.

Keywords: Three-dimensional bin packing, multi-criteria optimization problem, non-dominated solutions, space compaction.

1. Introduction

The *Three-dimensional Bin–Packing Problem*(3BPP) [3] is an extension of the *Bin–Packing Problem* for packing solid objects into three–dimensional bins. Packing problems have various real-world applications in areas like container loading, storing goods, and cutting objects out of a piece of material. These problems are usually NP-hard, thus the research for solvability is focused on the design of polynomial time approximation algorithms and schemes. The problem here addressed arises frequently when in need to store goods in warehouses and can be informally described as: given a set of different bins and a set of diverse boxes, find a packing of the boxes into the smallest number of bins minimizing space occupation. This is a variant for NP-hard three-dimensional bin packing problem (3BPP). We call this problem the 3BPP–min. Hereinafter we only consider offline problems.

During the last decade, several papers on the general subject of orthogonal 3D-BPP have been published. The most cited one is probably the reference [3] by Martello, Pisinger, and Vigo. The lower bounds there derived have been used since throughout the related literature. The authors also present a stripping two-phase heuristic, H_1, here called MPV, for approximating the solution of the tree-dimensional bin packing problem. Each container is divided into several shelves and, for each shelf, a two-dimensional bin packing problem is solved. When all the boxes are placed into the shelves, a one-dimensional problem is solved, searching for the best combination of shelves for each container. The time needed for MPV to obtain a solution can exceed one hour for instances having no more than 90 boxes. Another heuristic algorithm is also presented, H_2, using the concept of three dimensional *corner points* that allow for the definition of the envelop of the region with the already packed items. In fact, the most common approach to the general 3BPP problem is the so called *Slicing* (or *Strip*) strategy for the resolution of the Three-dimensional Packing Problem. It consists in repeatedly solve 2-D packing problems by defining horizontal slices on the height of a bin.

Next section describes some basic concepts related to the problem and presents a new heuristic using the strip approach for the 3BPP–min together with the concept of non-dominated solution. Section 3) describes some numerical experiments using several instantiations for the problem and the correspondent results analysis. We finish with a section where conclusions are drawn and directions for future work are laid.

2. SRC – A new algorithm for approximating the 3DPP–min

Consider the Euclidean space $\mathbb{R}^3$ with the XYZ coordinate system to represent a packing. The down-back-left vertex of a bin represents the origin. Each rectangular box R_i is given with an initial orientation related to the coordinate axes and has its dimensions denoted as (w_i, h_i, d_i) representing the weight, height and depth, respectively. The input of a general 3BPP problem consists of a list $L = \{R_1, R_2, \ldots, R_n\}$ of 3-dimensional rectangular boxes and a list of 3-dimensional rectangular bins $B = \{B_1, B_2, \ldots, B_m\}$, where box dimensions must be smaller than the maximum of the dimensions of the bins. We assume that there are enough bins to pack all the boxes.

All the boxes of the list L are to be packed into bins without overlap. The edges of both packed boxes and bins must always remain parallel to the coordinate axes. When rotations are allowed, the boxes to be placed may be rotated around any of the axes by 90^o. However, under this context, rotations around the Z axis provide the same general orientation and are therefore redundant for this study. Allowing rotations greatly increases the difficulty of finding a good solution since the search space expands significantly comparing to the 3D-BPP where the boxes have fixed orientations.

This section presents a new heuristic algorithm for the 3BPP–min with rotations that we call SRC – *Strip Rotation and Compaction*. The new approach is based on the ideas found in [3], namely the MPV heuristic algorithm, and the non-dominated rectangle compaction approach in [1]. Box rotations are only allowed for two of the dimensions: around the Y axis and around the X axis. The basic structure of MPV is used, i.e., the problem is solved by a first phase where 2D-Packing problems defining strips are solved. When all the boxes are placed and the several strips defined (which can be looked at as virtual shelves inside the bin) a one-dimensional packing problem is solved searching for the best combination of shelves for each container. The major innovation is the way the boxes are placed (packed) in each shelf. For each one, we solve the two-dimensional bin packing strip problem assuming that all the boxes must be placed and no ordering has yet been defined between the boxes themselves (e.g. by order of volume).

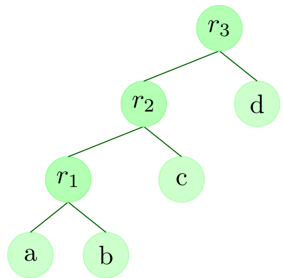

Figure 1: Binary tree for packing 4 boxes into a shelf (strip) (r_i represents the ith sub-root).

We will represent the sequence of boxes already chosen for placement (packing) by a binary tree structure (Fig. 1). This means that it will always be a degenerate tree: the right child of any internal node is always a leaf, representing a single box, while the root (and each sub-root) represents a packing (sequence of one or more boxes). Associated with each leaf is a list of dimensions representing the two possible ways to pack a pair of rectangle: using a horizontal or a vertical orientation. For single boxes (leaf nodes), assume that a horizontal box orientation fixes the smallest of the box's dimensions for the Y axis coordinate and assume a vertical orientation otherwise.

It is now important to realise that we order the list of the rectangles' dimensions satisfying (strictly) decreasing order of heights, $h_i > h_{i+1}$, and (strictly) increasing order of widths, $w_i < w_{i+1}$, where i represents the i^{th} element in this list as described in (Fig. 2). We will next see that this ordering is crucial for the heuristic method next described.

A sequence of already relatively placed boxes can be considered, in itself, a virtual box whose lateral (list of) dimensions are the ones of the placement's bounding rectangle or envelop. Then any two boxes (virtual or not) can be placed next to each other under two possible orientations: one on the right of the other (meaning that this placement can be divided by a vertical slice) or on top of each other (we have a horizontal slice dividing the placement). Thus we only consider slicing placements for the packing

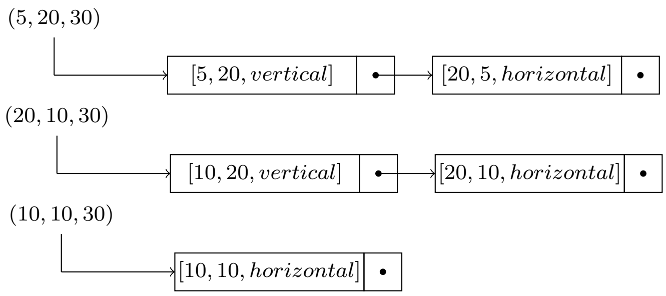

Figure 2: The associated lists of boxes with dimensions: (a) (5, 20, 30), (b) (20, 10, 30) and (c) (10, 10, 30). The latter is squared having only one possible orientation.

of any two boxes (recurrently, for placing a box into an already decided arrangement of boxes). The list associated with a node represents all the possible placements (arrangements) for all the possible orientations between both boxes.

The packing is obtained using a procedure where the goal in each iteration is to place a box into a previous packing, which was obtained through a slicing sequence of box's placements. At first, a list with all the possible combinations deriving from the use of a vertical cut is created. Next the solutions obtained assuming the existence of a horizontal cut are inserted into the list. Each new possible solution (the dimensions for the enveloping rectangle and the associated cutting direction) is joined into the already existing solutions list obeying the order

$$h_i > h_{i+1} \wedge w_i < w_{i+1} \, , 1 \leq i \leq s \, , \tag{1}$$

where s is the number of elements in the list. Furthermore, any new solution is only added to the list if and only if it is a non-dominated solution, that is, there is no solution that is strictly better than it on both dimensions*. A solution (a, b) is considered dominated by (c, d) if they are equal in each coordinate or if $(c \leq a \wedge d < b) \vee (c < a \wedge d \leq b)$. As an example, if we were to place the boxes (a) and (c) from Fig. 2 next to each other, the node representing all the possible and non-dominated placement combinations would present, as associated list of feasible solutions, the four possibilities shown in Figure 3.

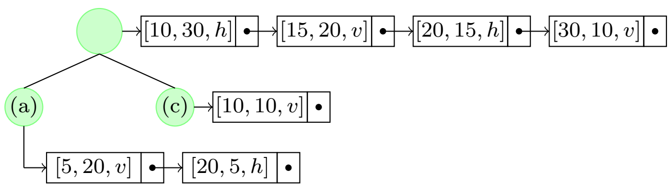

Figure 3: New tree node aggregating two leaf-nodes (representing boxes (a) and (c)) of Fig. 2.

Given two nodes and their associated lists of dimensions, an algorithm for constructing the list of possible placement solutions when joining the box in the right node with the box (or enveloping boxes rectangle) represented by the left node can be described as:

SRC Packing Procedure

- Starting from the head of the list and while both lists associated with the children nodes have elements:
  1. Assuming a vertical slice, join the new box to the right of this placement solution adding both widths and using the maximum hight as new rectangular dimensions;
  2. Insert the new solution into the list of this root node solutions obeying the order previously defined;
  3. If the width of the element from the list of the left child was the greatest then advance for the next element in this list;

*In the sense of Pareto optimality.

4. Else if the width of the element from the list of the right child was the greatest then advance for the next element in this list
5. Otherwise advance in both lists.

- Repeat the steps above starting from the end of the list and replacing vertical by horizontal and width by hight.

The insertion of a recently build rectangular solution first tests this solution with respect to the all possibilities already in the list of solutions, to find out if this one is a non-dominated solution. In the latter case the new solution it is inserted in the list in order. Otherwise, this solution being dominated is not inserted. The test is made using the defined ordering which simplifies the number of tests to be performed (for more details please refer to [1]).

Other important step in the algorithm is that, whenever a solution presents one of the dimensions equal to the respective bin's width or height but there is still space on the opposite direction to accommodate one more box, the associated rectangle suffers a 90^o rotation so that the remaining box can still be placed.

The process above described is repeated until there are no boxes to be placed or the bin shelf comports no more of the remaining boxes. The best packing will be the solution from the list at the root of the final tree that presents the greater compaction rate. At the end of this procedure, a sub-set of the boxes are packed at a shelf. The remaining ones, that were rejected during the search for a feasible solution or could no more be accommodated into the present bin's shelf, return to the list of boxes and the placement and search for another feasible solution using a new shelf are repeated. If the bin is full, a new bin is chosen. These procedures are repeated until all the boxes are packed into feasible shelfs, this is, until we find a feasible solution for the overall packing. In the limit, there may be shelves/bins having only one placed box.

3. Numerical Results

To perform the computational experiments to assert the validity of this new approximation algorithm tests were run using an Intel(R) Core(TM)2 Quad CPU at 2.83GHz with 4GB RAM.

Different types of items, i.e, items with different dimensions were considered (Table 1). Classes 1 to 5 and 7 to 9 are the ones already used in the related literature, namely in [3]. Class 6 consists of boxes having randomly generated dimensions in $[1, 50]$. While the bins used for classes 1 to 6 were always cubic bins of side 50, for the types found in 7, 8 and 9 the bins have dimensions $W = H = D = 10$, $W = H = D = 40$ e $W = H = D = 100$, respectively.

Table 1: The several types of boxes in each instance class.

Dimensions	1	2	3	4	5	6	7	8	9
w_j	$[1, \frac{1}{2}W]$	$[\frac{2}{3}W, W]$	$[\frac{2}{3}W, W]$	$[\frac{1}{2}W, W]$	$[1, \frac{1}{2}W]$	$[1, W]$	$[1, 10]$	$[1, 35]$	$[1, 100]$
h_j	$[\frac{2}{3}H, H]$	$[1, \frac{1}{2}H]$	$[\frac{2}{3}H, H]$	$[\frac{1}{2}H, H]$	$[1, \frac{1}{2}H]$	$[1, H]$	$[1, 10]$	$[1, 35]$	$[1, 100]$
d_j	$[\frac{2}{3}D, D]$	$[\frac{2}{3}D, D]$	$[\frac{1}{2}D, D]$	$[\frac{1}{2}D, D]$	$[1, \frac{1}{2}D]$	$[1, D]$	$[1, 10]$	$[1, 35]$	$[1, 100]$

The new heuristic algorithm, SRC, was compared with heuristic H_1 from [3] already referred to in the introductory section and here denominated MPV. In [2] Boschetti presents new lower bounds for the 3BPP showing that these dominate the ones in [3]. The author also provides an extension for packings allowing for 90^o box rotations. Thus we used Boschetti results for the evaluation of the expected minimum number of bins for each test instance.

This section reports the running times needed to terminate the heuristics and the number of bins occupied in the solutions found. These values are averages of 30 instances runs in each class. Both heuristics were tested using different quantities of each type of boxes to pack. The number of items to be placed ranged from 10 to 90 and from 100 to 500. Notice that the boxes instances' of the first three classes (1,2, and 3) are very similar, always having two of their dimensions very large when compared to the remaining one.

Table 2 presents the average number of bins allocated to the packing solutions. Observe that for $N = 10$ both algorithms achieved the lower bound or only used one more bin (type 6). Nevertheless, in general, SRC presents better compaction ratios then those obtained by MPV packings since it occupies less bins for the same instances. Note also that, for types 1 and 2, the results are very similar: MPV occupies the same bins while SRC uses less bins although not exactly the same number. This is due to the fact that these boxes are shaped very similarly, having $w_j \in \left[1, \frac{1}{2}W\right]$, $\left[\frac{2}{3}W, W\right]$ and $h_j \in \left[1, \frac{1}{2}H\right]$, $\left[\frac{2}{3}H, H\right]$, respectively. Taking in consideration the fact that the new procedure performs 90^o rotations this effect was expected. For class 3 it is noticeable that the values are the same for both heuristics. Moreover, both always achieved the LB values. Since the new heuristic only performs rotations over 2 of the dimensions - width and hight and this type of boxes have dimensions that are bigger than half of the bin's respective dimension, the possibility of increasing the overall compaction rate by rotation is here redundant. For class 4, notice that the new heuristic obtains only slightly better results that MPV. The boxes are instantiated having all the three dimensions bigger than half on the bin's dimension. Again, whichever gain is obtained by the rotations, it is not very noticeable due to the huge dimensions of the boxes involved when compared with the bins' dimensions. Thus the algorithm SRC consistently presents better results than the MPV heuristic in the sense that it obtains more compact solutions when the boxes are relatively smaller than the bins where they are to be packed on.

Table 2: Table with the average number of allocated bins for packing each of the 30 instances (LB – Lower bound).

		Number of allocated bins								
N		Type 1	Type 2	Type 3	Type 4	Type 5	Type 6	Type 7	Type 8	Type 9
	LB	3	3	3	10	1	2	2	2	3
10	MPV	3	3	3	10	1	3	3	2	3
	SRC	3	3	3	10	1	3	3	2	3
	LB	11	11	13	47	1	8	8	5	8
50	MPV	13	13	13	48	3	11	10	8	11
	SRC	13	12	13	48	2	10	9	7	10
	LB	21	21	25	94	2	15	14	10	15
100	MPV	25	25	25	95	4	20	18	15	20
	SRC	24	24	25	94	3	18	17	13	19
	LB	31	31	38	142	3	22	21	14	22
150	MPV	38	38	38	142	6	28	26	20	28
	SRC	36	36	38	142	4	27	24	18	27
	LB	42	42	50	188	4	29	27	19	29
200	MPV	50	50	50	189	7	36	33	26	36
	SRC	48	48	50	189	5	35	32	24	35
	LB	52	52	63	235	4	36	33	23	35
250	MPV	63	63	63	236	8	44	41	31	44
	SRC	59	59	63	235	7	43	38	29	43
	LB	62	62	75	282	5	43	40	27	42
300	MPV	75	75	75	283	9	52	48	36	52
	SRC	70	70	75	283	8	51	45	34	51
	LB	73	73	88	330	6	49	46	31	49
350	MPV	88	88	88	330	10	60	55	42	60
	SRC	81	82	88	330	9	58	52	39	59
	LB	83	83	100	377	7	55	52	36	55
400	MPV	100	100	100	378	11	67	62	47	68
	SRC	92	93	100	377	10	65	59	44	66
	LB	94	94	113	424	8	62	59	40	61
450	MPV	113	113	113	425	12	75	69	52	75
	SRC	104	104	113	424	11	73	66	49	74
	LB	104	104	125	471	8	68	65	44	68
500	MPV	125	125	125	472	13	82	76	57	83
	SRC	116	115	125	472	12	81	73	54	82

Figure 4 shows the average running times for both heuristics where the red line represents SRC and the blue line MPV values. The X axis represents the number of items to be packed for each of the 30 instances group. The instances generated for classes 1 and 2 have very similar results for both algorithms so type 1 results were chosen to illustrate the algorithmic performance. The same happens for the results in classes 7, 8 and 9 and therefore only type 8 results are here included. It is obvious that, in virtually all situations, the new algorithm is faster than the MPV, except for type 3 boxes. In this case, the relation between boxes and the bins dimensions allows to place only a box per shelf. Since SRC has an extra procedure to solve the one dimensional packing of the shelves, it ends up doing extra work when compared to MPV. For class 4, the items to be placed are big and only for the bigger numbers of boxes (from 200 up) do we see SRC faster then MPV. Overall, it is obvious that SRC is much faster than MPV and the plots in Fig. 4 seem to indicate that asymptotically, SRC is much more efficient than MPV, with the only exception of class 3 instances.

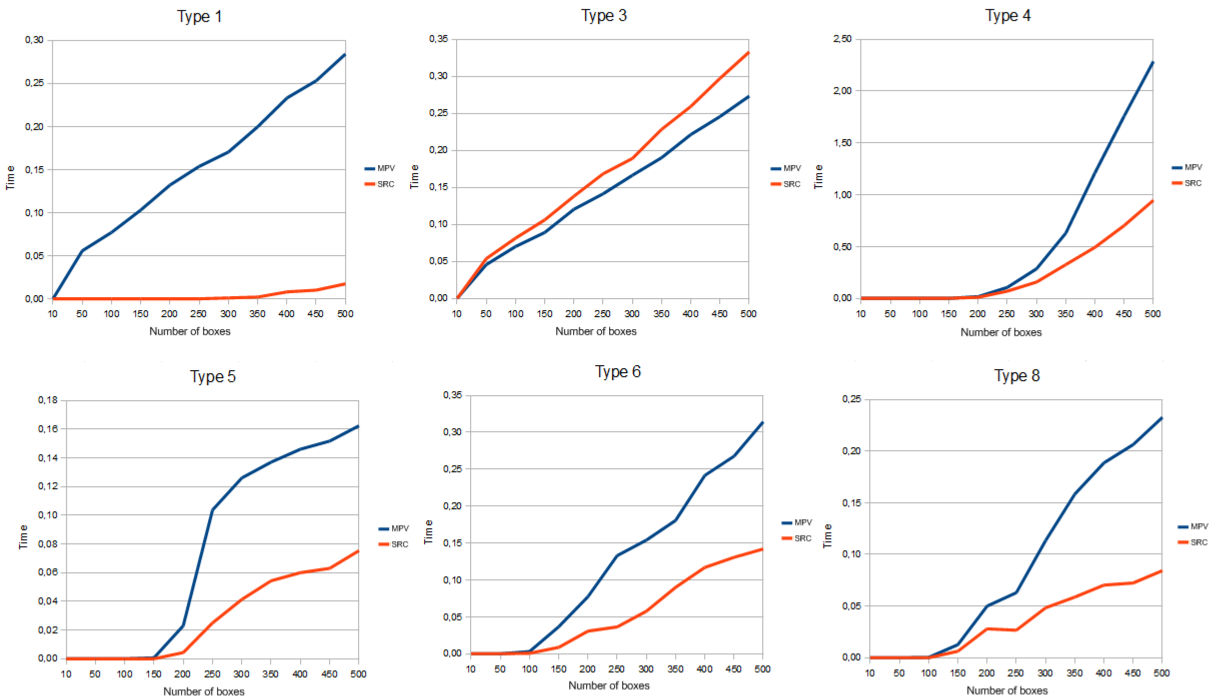

Figure 4: Average running times for 30 instances (seconds).

4. Conclusions

This work presents a new heuristic approach to solve a three dimensional bin packing problem allowing for 90^o rotations, 3BPP–min. The method is based on a stripping approache for the three dimensional bin packing problem and from the concepts presented by Almeida *etal.* for the compaction of two dimensional packings allowing for item rotations.

A new approximation algorithm, SRC, is presented and the comparison of its results with the ones obtained with the benchmarking heuristic MPV from Martello *etal.* are positively favorable towards the new heuristic. This is true either in terms of achieved compaction rations as for performance times.

The experiment and conclusions encouraged us into further explore and extend this method. The more so if, as we expect, if the study of the theoretical complexity bounds for the SRC approximation algorithm shows that it is polynomially bounded.

References

[1] A.M.C. Almeida, E. Q. V. Martins and R. D. Rodrigues, Optimal cutting directions and rectangle orientation algorithm, *European Journal of Operational Research*, 109 (3), 660-671, 1998.

[2] M. A. Boschetti, New lower bounds for the three-dimensional finite bin packing problem, *Discrete Appl. Math.*, 140 (1-3), 241-258, 2004.

[3] S. Martello, D. Pisinger and D. Vigo, The three-dimensional bin packing problem, *Operations Research*, 48(2), 256-267, 2000.