Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Discrete Mathematics 36 (2010) 993–1000

Electronic Notes in DISCRETE MATHEMATICS

www.elsevier.com/locate/endm

A Linear Programming Approach for the Three-Dimensional Bin-Packing Problem

Mhand Hifi*, Imed Kacem**, Stéphane Nègre*, Lei Wu*

* *MIS, Axe GOC-ODR, University of Picardie Jules Verne, Amiens, France*[1]

** *LITA, Paul Verlaine University of Metz, 57000 Metz - France*[2]

Abstract

In this paper we consider the three-dimensional bin packing problem (3D-BPP), when the bins are identical with the aim of minimizing the number of the used bins. We introduce a mixed-integer linear programming formulation (MILP1). Some special valid inequalities will be presented in order to improve the relaxed lower bound of MILP1. A large set of experimental tests has been carried out. The obtained results show that our method provides a satisfactory performance.

Keywords: bin-packing, integer linear programming, scheduling.

1 Introduction

In this paper we consider the three-dimensional bin packing problem (3D-BPP) when the bins are identical. The first review related to this topic was published by Martello et al [7] which introduced a lower bound and an exact approach. A few years later, Fekete and Schepers [4] proved that their lower bounds dominate the one proposed by Martello et al, and in [5] they introduced

[1] Email: `[hifi, stephane.negre, lei.wu]@u-picardie.fr`

[2] Email: `kacem@univ-metz.fr`

1571-0653/$ – see front matter © 2010 Elsevier B.V. All rights reserved.

doi:10.1016/j.endm.2010.05.126

an exact algorithm following the outline of Martello et al. Then, Boschetti [2] introduced some improved lower bounds. Unfortunately, it is not easy to extend these bounds to some practical industrial situations. For this reason, we aim in this paper to construct a more adaptive approach based on an integer-linear programming. Contrary to constructive heuristic algorithms, such as (*next, first, best*)-fit and *bottom-left-back* methods, we could integrate the practical issues in the linear model we propose for 3D-BPP by formulating them as inequalities. This paper is organized as follows. In Section 2, we first present a *mixed-integer linear programming* (MILP) model to the 3D-BPP. Section 3 presents a set of lower bounds based on the introduction of valid inequalities in the MILP. The experimental results are discussed in Section 4. Finally, Section 5 concludes the paper.

2 Mixed integer linear programming model

Our problem consists in packing a set of n rectangular items into a set of identical bins. Every item i $(i = 1, 2, ..., n)$ is characterized by a width w_i, a height h_i and a depth d_i $(i = 1, 2, ..., n)$ while the bins have the same width W, the same height H, and the same depth D. We aim to find a feasible packing by minimizing the number of the necessary bins. We start with a sufficient number of bins (an upper bound) $\overline{\gamma}$ $(n \geq \overline{\gamma})$. We define the vector $(x_i, y_i, z_i, \gamma_i)$ as the geometrical location of item i, where $x_i \geq 0$, $y_i \geq 0$, $z_i \geq 0$ and $\gamma_i \geq 1$ $(i = 1, \cdots, n)$. We use (x_i, y_i, z_i) to denote the *left-bottom-back* coordinate of item i while we assume that the coordinate of *left-bottom-back* corner of the bin is $(0, 0, 0)$. γ_i is defined as the label of the bin to which item i is assigned $(i = 1, \cdots, n)$. The aim is to minimize the greatest label of the used bin $\gamma = \max_{1 \leq i \leq n}\{\gamma_i\}$.
The proposed *Mixed Integer Linear Programming* for the 3D-BPP, noted by MILP1, can be formulated as follows:

$$
\begin{array}{llll}
 & (MILP1) \quad \min \quad \gamma & & \\
s.c. & l_{ij} + l_{ji} + u_{ij} + u_{ji} + b_{ij} + b_{ji} + c_{ij} + c_{ji} = 1, & i < j = 1, \cdots, n, & (1) \\
 & x_i - x_j + W(l_{ij} - c_{ij} - c_{ji}) \leq W - w_i, & i \neq j = 1, \cdots, n, & (2) \\
 & y_i - y_j + H(u_{ij} - c_{ij} - c_{ji}) \leq H - h_i, & i \neq j = 1, \cdots, n, & (3) \\
 & z_i - z_j + D(b_{ij} - c_{ij} - c_{ji}) \leq D - d_i, & i \neq j = 1, \cdots, n, & (4) \\
 & (\overline{\gamma} - 1)(l_{ij} + l_{ji} + u_{ij} + u_{ji} + b_{ij} + b_{ji}) & & \\
 & \quad + \gamma_i - \gamma_j + \overline{\gamma} c_{ij} \leq \overline{\gamma} - 1, & i \neq j = 1, \cdots, n, & (5) \\
 & l_{ij}, u_{ij}, b_{ij}, c_{ij} \in \{0, 1\}, & i \neq j = 1, \cdots, n, &
\end{array}
$$

$$
\begin{aligned}
&0 \le x_i \le W - w_i, && i = 1, \cdots, n,\\
&0 \le y_i \le H - h_i, && i = 1, \cdots, n,\\
&0 \le z_i \le D - d_i, && i = 1, \cdots, n,\\
&0 \le \gamma_i \le \gamma \le \overline{\gamma}, && i = 1, \cdots, n.
\end{aligned}
$$

where $l_{ij} = 1$ if item i is in the left of item j, $u_{ij} = 1$ if item i is under item j, $b_{ij} = 1$ if item i is in the back item j and $c_{ij} = 1$ if $\gamma_i < \gamma_j$.
The condition that no overlap between two packed items is guaranteed by constraints (1, 2, 3, 4). Constraints (5) implies that when $c_{ij} = 1$ or $c_{ji} = 1$ the items i, j are located in different bins and when one of $l_{ij}, l_{ji}, u_{ij}, u_{ji}, b_{ij}, b_{ji}$ is equal to 1, items i and j are necessarily located in the same bin. The parameter $\overline{\gamma}$ is a valid upper bound on γ. $\overline{\gamma}$ can be obtained by using a fast heuristic, for instance, *Wall-Building* heuristic which was first proposed by George *et al* [6]. It is well-known that entirely solving MILP1 is not practical. For this reason, we return our attention to solve MILP1 stage by stage: we first try to find a feasible solution, and then, with starting from this node, we set out the tree search to find an optimal solution. The feasible solution is obtained by solving an adapted similar formulation of a single-bin packing problem MILP2.

3 Lower bounds based on valid inequalities

The lower bounds in this paper are based on the resolution of LP-relaxation of MILP1. A trivial lower bound LB_0 is the value of the optimal solution of LP-relaxation of (MILP1). Unfortunately, LB_0 is always weak and a huge unnecessary tree nodes would be created when we apply branch and bound (B&B) algorithm to solve MILP1. In terms of performance, when the amount of items is greater than 60 Cplex.11 is not able to find a feasible solution in a reasonable time. This is the main motivation to present valid inequalities for MILP1 inspired from Bekrar and Kacem [1].

3.1 Valid inequalities related to the parallel-machine scheduling problem

First, let us recall the lower bound of Eastman proposed for the weighted flow-time minimization on identical parallel-machine [3]. Given m machines and n jobs ($j = 1, ..., n$) where to every j a processing time a_j and a weight g_j are associated ($a_1/g_1 \le a_2/g_2 \le .. \le a_n/g_n$). Let C_j be the completion time of job j in a feasible schedule. Any feasible solution minimizing the weighted completion times of jobs must verify the following inequality:

$$\sum_{j=1}^{n} g_j C_j \geq LB_{eastman} = \frac{1}{m}\sum_{j=1}^{n} g_j(\sum_{k=1}^{j} a_k) + \frac{m-1}{2m}\sum_{j=1}^{n} g_j a_j \tag{6}$$

Now, we generalize the valid inequalities introduced by Bekrar and Kacem [1] to improve a linear model for the *Two-dimensional Guillotine Strip Packing Problem* (cf. [1]). With a similar reasoning, an item can be considered as a set of jobs and a single bin can be considered as a set of machines. For example, an item i of the size (w_i, h_i, d_i) can be recognized as $h_i d_i$ (resp. $w_i d_i$, $w_i h_i$) jobs having the same processing time w_i (resp. h_i, d_i), and a bin can be seen as HD (resp. WD, WH) machines. In other words, the fact that item i is located at $(x_i, y_i, z_i, \gamma_i)$ can be imagined as $h_i d_i$ (resp. $w_i d_i$, $w_i h_i$) jobs, of a processing time w_i, that are performed at time $(\gamma_i - 1)W + x_i$ (resp. $(\gamma_i - 1)H + y_i$, $(\gamma_i - 1)D + z_i$) $(i = 1, \cdots, n)$. Hence, by associating to each item a fictitious weight, the sum of weighted completion times related to any configuration of items (packing) must be greater than $LB_{eastman}$ applied to the associated instance of parallel-machine scheduling problem. Consequently, we can add the following valid inequalities to (MILP1). Moreover, by exchanging the roles played by the positions x, y and z, the so-generated constraints are also valid to (MILP1). In the rest of this section, we present only the valid inequalities formulated by considering x-positions. Therefore, the associated parallel-machine scheduling problem corresponds to sequence $\sum_{i=1}^{n} h_i d_i$ jobs on HD machines.

$$\begin{aligned}\sum_{i=1}^{n} \tilde{p}_i h_i d_i (x_i + w_i + (\gamma_i - 1)W) \geq \frac{1}{HD}\sum_{i=1}^{n} \tilde{p}_{[i]} \left(h_{[i]} d_{[i]} \sum_{j=1}^{i-1} vol_{[j]} + \sum_{k=1}^{h_{[i]}d_{[i]}} k w_{[i]} \right) \\ + \frac{HD-1}{2HD}\sum_{i=1}^{n} \tilde{p}_i vol_i \end{aligned} \tag{7}$$

where $[i]$ is the i^{th} element according to the non-increasing order of $\frac{\tilde{p}_i}{w_i}$, vol_i is the volume of i and $\tilde{p}_i$ is a fictitious weight associated to item i for every $i = 1, \cdots, n$.

Reciprocally, if we take the right face of the bin as the starting time, the completion time of job i can be seen equal to $W - x_i$. Furthermore, in the case of several bins, the completion time of job i is defined as $\gamma W - (\gamma_i - 1)W - x_i$.

Hence, the following inequality is also valid for MILP1:

$$\sum_{i=1}^{n} \tilde{p}_i h_i d_i(\gamma W - (\gamma_i - 1)W - x_i) \geq \frac{1}{HD} \sum_{i=1}^{n} \tilde{p}_{[i]} \left(h_{[i]} d_{[i]} \sum_{j=1}^{i-1} vol_{[j]} + \sum_{k=1}^{h_{[i]}d_{[i]}} kw_{[i]} \right) + \frac{HD-1}{2HD} \sum_{i=1}^{n} \tilde{p}_i vol_i \qquad (8)$$

Differently from the mechanism used in (7) and (8) in which the bins have been sequenced as a train, the bins could be organized in a shape. Indeed, we can neglect the labels or assemble all the bins together and therefore we have finally $\overline{\gamma}HD$ machines. The third type of valid inequalities is written as follow:

$$\sum_{i=1}^{n} \tilde{p}_i h_i d_i(x_i + w_i) \geq \frac{1}{\overline{\gamma}HD} \sum_{i=1}^{n} \tilde{p}_{[i]} \left(h_{[i]} d_{[i]} \sum_{j=1}^{i-1} vol_{[j]} + \sum_{k=1}^{h_{[i]}d_{[i]}} kw_{[i]} \right) + \frac{\overline{\gamma}HD-1}{2\overline{\gamma}HD} \sum_{i=1}^{n} \tilde{p}_i vol_i \qquad (9)$$

3.2 Valid inequalities based on precedence considerations

The following set of valid constraints (10 - 13) hold due to the precedence considerations. For example, inequality (10) means that the total volume of the items which are located in the left of item i cannot be greater than the volume of the total space being left to item i. The same reasoning is also valid for the others.

$$W - w_i - \frac{\sum_{j\neq i}(w_j h_j d_j l_{ij})}{HD} \geq x_i \geq \frac{\sum_{j\neq i}(w_j h_j d_j l_{ji})}{HD} \qquad (10)$$

$$H - h_i - \frac{\sum_{j\neq i}(w_j h_j d_j u_{ij})}{WD} \geq y_i \geq \frac{\sum_{j\neq i}(w_j h_j d_j u_{ji})}{WD} \qquad (11)$$

$$D - d_i - \frac{\sum_{j\neq i}(w_j h_j d_j b_{ij})}{WH} \geq z_i \geq \frac{\sum_{j\neq i}(w_j h_j d_j b_{ji})}{WH} \qquad (12)$$

$$\gamma - \frac{\sum_{j\neq i}(w_j h_j d_j c_{ij})}{WHD} \geq \gamma_i \geq \frac{\sum_{j\neq i}(w_j h_j d_j c_{ji})}{WHD} + 1 \qquad (13)$$

3.3 New bounds for the 3D-BPP

We note LB_1 as a lower bound of 3D-BPP which is calculated by adding all constraints introduced before. However, LB_1 may be far away from the optimum due to the presence of special shape items like the one which leads us to the worst-case. With respect to the principles developed in Martello *et al* [7] and Fekete Schepers [4], we write the valid constraints according to the size of items. Thus, a better lower bound LB_2 can be obtained by classifying items with respect to their sizes.

4 Computational experiments

In order to test MILP1, we adopt the 9 groups randomly instances described by Martello *et al* in [7]. The first five groups structured by Martello and Vigo can be generated as follows:

- mvp-1: bin size $W = H = D = 100$; items with $w_i \in [1, \frac{1}{2}W]$, $h_i \in [\frac{2}{3}H, H]$ and $d_i \in [\frac{2}{3}D, D]$.
- mvp-2: bin size $W = H = D = 100$; items with $w_i \in [\frac{2}{3}W, W]$, $h_i \in [1, \frac{1}{2}H]$ and $d_i \in [\frac{2}{3}D, D]$.
- mvp-3: bin size $W = H = D = 100$; items with $w_i \in [\frac{2}{3}W, W]$, $h_i \in [\frac{2}{3}H, H]$ and $d_i \in [1, \frac{2}{3}D]$.
- mvp-4: bin size $W = H = D = 100$; items with $w_i \in [\frac{1}{2}W, W]$, $h_i \in [\frac{1}{2}H, H]$ and $d_i \in [\frac{1}{2}D, D]$.
- mvp-5: bin size $W = H = D = 100$; items with $w_i \in [1, \frac{1}{2}W]$, $h_i \in [1, \frac{1}{2}H]$ and $d_i \in [1, \frac{1}{2}D]$.

For group k $(k = 1, \cdots, 5)$, every item is of type k with a probability of 60% and of the other four types with a probability of 10%. It is important to note that mvp-1, mvp-2 and mvp-3 generate a greater proportion of special shape items whose two of their dimensions are greater than 50% of the corresponding dimension of bins. In other words, this kind of items can be fitted along only one dimension.

The next group is a generalization of the instances described by Berkey and Wang (1987). The three classes have been defined as follows:

- mvp-6: bin size $W = H = D = 10$; items with $w_i, h_i, d_i \in [1, 10]$.
- mvp-7: bin size $W = H = D = 40$; items with $w_i, h_i, d_i \in [1, 35]$.
- mvp-8: bin size $W = H = D = 100$; items with $w_i, h_i, d_i \in [1, 100]$.

The last class mvp-9 has been generated in a *all-fill* way. The items are generated by decomposing a large item which is identical to the bin. Thus, these classes have a known optimal solution corresponding to the number of the used large items. Furthermore, for this class of instances the continuous bound is equal to the optimal solution.

The algorithm was implemented in C++ while MIP-Solver Cplex 11.20 was called to solve MILP2. The experiments were run on an UltraSparc10 (250 Mhz and with 1 Gb of RAM). The outcome of our experiments is shown in Table (2). We limited the computation time spent to solve a single filling bin problem to 300 seconds. In Table (1) we summarize the average deviation of the upper bound U_{mpv} found by algorithms H1 and H2 (see [7]) with respect to the best lower bound value L_2 (cf. [7]), computed over all the instances. From these results, we can observe that the quality of the lower bound (reps. the upper bound) strongly depends on the presence of the large items. For instance, in the class 4 which contains a sufficient number of large items, one always successfully both find nice lower bounds and upper bounds.

	Class								
n	1	2	3	4	5	6	7	8	9
10	8.3	15.0	29.2	2.5	30.0	28.3	48.3	25.0	0.0
15	16.5	18.3	22.0	1.1	47.5	30.7	19.2	23.3	0.0
20	19.2	14.5	27.7	0.8	26.7	20.5	25.8	18.3	6.7
25	13.2	14.0	18.7	2.4	23.7	20.2	30.8	21.8	30.0
30	16.8	15.7	16.6	3.9	22.3	20.2	37.5	25.9	40.0
35	15.7	11.6	15.5	2.5	18.1	25.0	33.5	28.1	40.0
40	16.1	14.7	17.3	2.2	34.1	19.3	39.4	28.1	53.3
45	15.6	12.5	12.1	3.2	25.2	23.0	45.6	30.4	56.7
50	17.4	14.4	17.1	3.5	31.6	25.6	33.7	35.2	66.7
60	15.2	15.6	14.0	2.8	33.3	23.5	41.0	23.3	60.0
70	14.9	14.8	15.8	3.6	31.0	24.1	47.2	32.2	83.3
80	17.0	14.9	12.8	2.7	32.6	22.5	37.9	35.2	93.3
90	13.4	14.0	13.8	3.4	30.1	21.3	45.0	32.8	103.3

Table 1

Average deviation of U_{mvp} with respect to L_2

Table (2) presents the average deviation of our lower and upper bounds. The values mentioned in bold in Table (2) means that we have the better results than Martello *et al*'s. We can easily notice that our algorithm yields better results for several classes, which proves that it is an effective improvement.

5 Conclusion

This paper considers the 3D Bin-Packing problem and investigates the elaboration of a new lower bound based on an LP-relaxation improved by the design of valid inequalities. These inequalities are based on some analogies with the parallel-machine scheduling problems. The theoretical models have been tested on several instances from the literature. The obtained results

	Class								
n	1	2	3	4	5	6	7	8	9
10	**3.3**	**5.8**	**15.0**	**0.0**	**10.0**	**25.8**	**5 .0**	**16.7**	**0.0**
15	19.0	20.3	**20.3**	**0.0**	**10.8**	**12.8**	21.6	**20**	**0.0**
20	24.8	19.6	**23.5**	**0.0**	**11.2**	**13.0**	**11.6**	19.8	**0.0**
25	21.4	27.5	21.5	**1.3**	**15.8**	26.6	**27.0**	**14.6**	**3.3**
30	19.1	28.0	24.8	**1.1**	**19.5**	26.7	**21.5**	**19.4**	**36.6**
35	19.0	23.6	23.1	2.9	32.6	**19.1**	**27.3**	**27.9**	43.3
40	28.7	18.6	19.2	**0.9**	**19.0**	22.5	**24.5**	**17.6**	56.7
45	26.7	19.7	22.3	**1.2**	**17.0**	23.5	**36.8**	**24.4**	**50**
50	19.6	18.3	17.5	**0.7**	**18.1**	**18.7**	**23.3**	**21.3**	**66.7**
60	20.4	18.4	17.9	**0.3**	**30.1**	**20.0**	**28.2**	24.2	73.3
70	16.4	20.3	18.4	**0.5**	**26.4**	**21.8**	**25.2**	**17.1**	**73.3**
80	20.6	18.6	14.1	**0.8**	37.9	**20.5**	39.7	**21.3**	**83.3**
90	17.5	15.9	18.2	**0.9**	**20.4**	**19.2**	**31.9**	**24.1**	**93.3**

Table 2

Average deviation of our upper and lower bounds

are satisfactory. As a perspective for this work, we aim to improve the used upper-bound in order to enhance the exact method performance.

References

[1] Bekrar A., and I. Kacem. An exact method for the 2D guillotine strip packing problem. *Advances in Operations Research*, volume 2009 (2009), article ID 732010.

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

[3] Eastman W.L., Even S., and I. M. Isaacs. Bounds for the optimal scheduling of n jobs on m processors. *Management Science* 11:2, (1964)

[4] Fekete S.P., and J. Schepers. New classes of lower bounds for bin packing problems. *Mathematical Programming* 91, 11-31 (2001)

[5] Fekete S.P., Schepers J., and J.C. van der Veen. An Exact Algorithm for Higher-Dimensional Orthogonal Packing. *Operations Research* 55:3, 569-587 (2007)

[6] George J.A., and D.F. Robinson. A heuristic for packing boxes into a container. *Computers and Operations Research* 7:3, 147-156 (1980)

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