

Models and Bounds for Two-Dimensional Level Packing Problems

Andrea Lodi, Silvano Martello, Daniele Vigo

*Dipartimento di Elettronica, Informatica e Sistemistica, University of Bologna
Viale Risorgimento, 2 - 40136 - Bologna (Italy)*

E-mails: `alodi@deis.unibo.it`, `smartello@deis.unibo.it`, `dvigo@deis.unibo.it`

Abstract

We consider two-dimensional bin packing and strip packing problems where the items have to be packed by levels. We introduce new mathematical models involving a polynomial number of variables and constraints, and show that their LP relaxations dominate the standard area relaxations. We then propose new (combinatorial) bounds that can be computed in $O(n \log n)$ time. We show that they dominate the other bounds, and establish their absolute worst-case behavior. The quality of models and bounds is evaluated through extensive computational experiments.

1 Introduction

In two-dimensional packing problems one is given a set of n rectangular *items*, each having *width* w_j and *height* h_j ($j = 1, \dots, n$), and the object is to orthogonally allocate them, without overlapping, to rectangular containers by minimizing the unused space. It is assumed that the items have fixed orientation, i.e., they cannot be rotated. Two main specific problems are considered in the literature:

- the *two-dimensional bin packing problem* (2BP), where an infinite number of identical finite containers (*bins*), having width W and height H , is available, and the object is to minimize the number of bins used;
- the *two-dimensional strip packing problem* (2SP), where a single container (*strip*), having width W and infinite height, is available, and the object is to minimize the height to which the strip is used.

The reader is referred to Lodi, Martello and Vigo [16] for a recent survey on two-dimensional packing, and to Dyckhoff, Scheithauer and Terno [10] for an annotated bibliography on cutting and packing.

Most of the approximation algorithms for 2BP and 2SP (see, e.g., Coffman, Garey, Johnson and Tarjan [6], Chung, Garey and Johnson [5], Baker and Schwarz [1], Berkey

and Wang [3], Frenk and Galambos [11], Lodi, Martello and Vigo [15]) find a bin/strip solution by packing the items, from left to right, in rows forming *levels*. The first level is the bottom of the bin/strip, and subsequent levels are created on the horizontal line coinciding with the top of the tallest item packed on the level below. This kind of packing has also practical relevance: in most cutting applications it is required that the patterns are such that the items can be obtained through a sequence of edge-to-edge cuts parallel to the edges of the bin (*guillotine* cuts), and it is easily seen that level packings fulfil this constraint.

In this paper we consider 2BP and 2SP with the additional constraint that the items are packed by levels. We denote the resulting problems as *two-dimensional level bin/strip packing problem* (2LBP/2LSP).

The origin of these problems can be found in the seminal paper by Gilmore and Gomory [12]. They first introduce the “exact two-stage guillotine cutting stock problem”, which is a 2LBP with the additional constraint that all items packed in a level have the same height. They then consider and analyze the more practical (and more difficult) case arising if a third trimming stage is permitted to separate an item from a waste area. Gilmore and Gomory call this problem, which coincides with 2LBP, the “non-exact two-stage guillotine cutting stock problem”. Nowadays, this kind of pattern is denoted as *two-stage guillotine cutting with trimming* in the knapsack and cutting stock literature, and as *level packing* in the bin and strip packing literature. The case where the third stage can separate items superposed within the same level is referred to as *three-stage guillotine cutting*, with or without trimming.

Besides their algorithmic interest in the solution of subproblems (see above), staged guillotine cutting problems have practical relevance in all the cases where the raw material to be cut has a low cost with respect to the industrial costs involved in the cutting process, such as, e.g., in many wood, paper or glass cutting applications. Real world applications involving two-stage cutting problems can be found, e.g., in de Carvalho and Rodrigues [8, 9] (steel industry), while an industrial three-stage cutting problem was recently considered by Vanderbeck [21].

Both 2LBP and 2LSP can be viewed as generalizations of the *one-dimensional bin packing problem* (1BP), where n elements, having size w_j ($j = 1, \dots, n$), have to be partitioned into the minimum number of subsets so that the sum of the sizes in each subset does not exceed a given capacity W . It is known that 1BP is strongly NP-hard. Given any instance of 1BP, we can construct both an equivalent instance of 2LBP, by defining $h_j = H$ for $j = 1, \dots, n$, and an equivalent instance of 2LSP, by defining $h_j = 1$ for $j = 1, \dots, n$. It follows that 2BP and 2SP are strongly NP-hard.

Several mathematical models presented in the literature for two-dimensional cutting and packing problems involve a non-polynomial number of variables. The classical model presented by Gilmore and Gomory [12] for 2LBP is based on a column generation approach. Models based on a discrete representation of the geometrical space and the explicit use of coordinates at which items may be allocated (see, e.g., Beasley [2], Hadjiconstantinou and Christofides [13], Hifi [14]), require a huge, non-polynomial, number of binary variables, such as, for example, $x_{ipq} = 1$ iff item i is placed with its bottom left-hand corner at

coordinate (p, q) ($i = 1, \dots, n, p = 1, \dots, W, q = 1, \dots, H$).

A mathematical mixed-integer model for a three-dimensional bin packing problem, involving a polynomial number of variables and constraints, was presented by Chen, Lee and Shen [4]. Their model, which can be seen as an extension to the three-dimensional case of the modeling technique proposed by Onodera, Taniguchi and Tamaru [20] for a two-dimensional block placement problem, is based on the enumeration of all possible relative placements of each pair of items. The computational experiments presented in [4] show however that the proposed model is quite ineffective in the practical solution of packing instances. Moreover, there is no easy way to adapt this modeling technique to level packing. The same technique has been used by Daniels, Milenkovic and Li [7] to model the packing of general (two-dimensional) polygons. In this case too, direct use of the model proves to be ineffective in practice.

In this paper we show that the level restriction can be exploited so as to obtain mathematical models (both for 2LSP and 2LBP) that involve a polynomial number of variables and constraints and are effective from a computational point of view. The models are presented in the next section. In Section 3 we show that the LP relaxations of these models dominate the standard area relaxations. In Sections 4.1 and 4.2 we introduce new (combinatorial) bounds that can be computed in $O(n \log n)$ time. We show that they dominate the other bounds, and establish their absolute worst-case behavior. The quality of models and bounds is evaluated in Section 5 through extensive computational experiments.

We assume in the following, without loss of generality, that all input data are positive integers. We will consider relaxations in which items (and levels) may be split through vertical or horizontal cuts: we will call *slices* (resp. *sectors*) the portions produced by vertical (resp. horizontal) cuts.

2 Mathematical Models

We start with a couple of simple observations, which will allow to obtain a compact formulation. For any optimal level solution there exists an equivalent solution in which

- (i) the first (leftmost) item packed in each level is the tallest item in the level;
- (ii) the first (bottom) level packed in each bin/strip is the tallest level in the bin/strip.

Hence we will consider only solutions satisfying these conditions. Moreover, if an item is the first in a level, we will say that it *initializes* the level; similarly, if a level is the first in a bin/strip, we will say that it *initializes* the bin/strip. We will also assume that

- (iii) the items are sorted so that $h_1 \geq h_2 \geq \dots \geq h_n$.

Let us first consider the two-dimensional level bin packing problem 2LBP. Our model uses four sets of variables: the first two sets refer to the packing of items into levels, the remaining two to the packing of levels into bins. The model assumes that n potential levels

are available, each associated with a different item i which initializes it, hence having the corresponding height h_i .

The first decision variable is thus

$$y_i = \begin{cases} 1 & \text{if item } i \text{ initializes level } i \\ 0 & \text{otherwise} \end{cases} \quad (i = 1, \dots, n) \quad (1)$$

and observe that, by (i) and (iii) above, only items j satisfying $j > i$ may be packed in level i (if this level is actually initialized by item i). Therefore the item packing is modeled by

$$x_{ij} = \begin{cases} 1 & \text{if item } j \text{ is packed into level } i \\ 0 & \text{otherwise} \end{cases} \quad (i = 1, \dots, n-1; j > i) \quad (2)$$

Similarly, we assume that n potential bins are available, each associated with a potential level k which initializes it. The third decision variable is thus

$$q_k = \begin{cases} 1 & \text{if level } k \text{ initializes bin } k \\ 0 & \text{otherwise} \end{cases} \quad (k = 1, \dots, n) \quad (3)$$

and observe that, by (ii) and (iii) above, only levels i satisfying $i > k$ may be allocated to bin k (if this bin is actually initialized by level k). Therefore the level packing is modeled by

$$z_{ki} = \begin{cases} 1 & \text{if level } i \text{ is allocated to bin } k \\ 0 & \text{otherwise} \end{cases} \quad (k = 1, \dots, n-1; i > k) \quad (4)$$

The ILP model follows:

$$(2\text{LBP}) \quad \min \quad \sum_{k=1}^n q_k \quad (5)$$

$$\text{subject to} \quad \sum_{i=1}^{j-1} x_{ij} + y_j = 1 \quad (j = 1, \dots, n) \quad (6)$$

$$\sum_{j=i+1}^n w_j x_{ij} \leq (W - w_i) y_i \quad (i = 1, \dots, n-1) \quad (7)$$

$$\sum_{k=1}^{i-1} z_{ki} + q_i = y_i \quad (i = 1, \dots, n) \quad (8)$$

$$\sum_{i=k+1}^n h_i z_{ki} \leq (H - h_k) q_k \quad (k = 1, \dots, n-1) \quad (9)$$

$$y_i \in \{0, 1\} \quad (i = 1, \dots, n) \quad (10)$$

$$x_{ij} \in \{0, 1\} \quad (i = 1, \dots, n-1; j > i) \quad (11)$$

$$q_k \in \{0, 1\} \quad (k = 1, \dots, n) \quad (12)$$

$$z_{ki} \in \{0, 1\} \quad (k = 1, \dots, n-1; i > k) \quad (13)$$

The objective function (5) minimizes the number of bins used. Equations (6) impose that each item is packed exactly once, either by initializing a level or in a level initialized by a preceding (taller) item. Equations (7) impose the width constraint to each used level. Equations (8) impose that each used level is allocated exactly once, either by initializing a bin or in a bin initialized by a preceding (taller) level. Finally, equations (9) impose the height constraint to each used bin.

A couple of observations can be done on this model. First, from (6) and (8) we immediately have $y_1 = q_1 = 1$. Second, an equivalent model could replace variables y_i ($i = 1, \dots, n$) with x_{ii} ($i = 1, \dots, n$), and variables q_k ($k = 1, \dots, n$) with z_{kk} ($k = 1, \dots, n$). In both cases we feel that the given model has to be preferred for the sake of clarity.

It is easy to derive from (5)–(13) a mathematical model for the two-dimensional level strip packing problem 2LSP. Indeed, it is enough to introduce the appropriate objective function, and to drop all variables and constraints related to the packing of the levels into the bins. We get the ILP model

$$\begin{aligned} (2LSP) \quad & \min \quad \sum_{i=1}^n h_i y_i \\ & \text{subject to} \quad (6), (7), (10), (11) \end{aligned} \quad (14)$$

2.1 Variants

The computational experiments we present in Section 5 show that models (2LBP) and (2LSP) may also be profitably used in practice. Indeed, their direct use with a commercial ILP solver produces very good solutions (and, in many cases, the optimal solution) to realistic sized instances involving tens of items within short CPU times. In addition, several variations of 2LBP and 2LSP can be easily modeled by modifying some of the constraints, or by adding linear constraints to the models.

For example, a relevant variant of both problems allows rotation of the items by 90° . Given any such instance, let us define a new instance by generating, for each item j ($j = 1, \dots, n$), a companion item $n + j$ with $w_{n+j} = h_j$ and $h_{n+j} = w_j$. Suppose now that the $2n$ resulting items are sorted and re-numbered by non-increasing height, and let α_j be the index of the rotated counterpart of j ($j = 1, \dots, 2n$). In order to obtain models for the variants where rotation is allowed, we just need to impose that, for each pair of companion items, exactly one is packed. This is easily obtained by replacing equation (6) with

$$\sum_{i=1}^{j-1} x_{ij} + y_j + \sum_{i=1}^{\alpha_j-1} x_{i\alpha_j} + y_{\alpha_j} = 1 \quad (j = 1, \dots, 2n; j < \alpha_j) \quad (15)$$

and by replacing n with $2n$ in all the remaining equations of (2LBP) and (2LSP).

As a second example, consider the case where an upper bound U on the number of items packed in each level is imposed. This can be handled by adding to (2LBP) or (2LSP) cover-type constraints of the form

$$y_i + \sum_{j=i+1}^n x_{ij} \leq U \quad (i = 1, \dots, n - U) \quad (16)$$

Other variants involving incompatibilities or couplings among items (e.g., pairs of items that cannot be packed in the same level, or that have to be packed in the same level) may be handled in similar ways.

For all such cases, the models represent a practical tool for a simple solution approach. In Section 5 we present some results on the variant where rotation of the items by 90° is allowed.

3 Lower Bounds from LP relaxations

Obvious lower bounds for 2LBP and 2LSP are derived by splitting each item j into $w_j h_j$ unit squares, thus obtaining, respectively:

$$L_a^b = \left\lceil \frac{\sum_{j=1}^n w_j h_j}{WH} \right\rceil \quad L_a^s = \left\lceil \frac{\sum_{j=1}^n w_j h_j}{W} \right\rceil \quad (17)$$

These are often called ‘continuous’ bounds. Since however we are going to use this term in a different context too, we will refer to them as *area bounds*.

The mathematical models of the previous section produce the *continuous bounds* by relaxing the integrality requirements of the variables, i.e., by replacing, in (10)–(13), $\xi \in \{0, 1\}$ with $0 \leq \xi \leq 1$ ($\xi \in \{y_i, x_{ij}, q_k, z_{ki}\}$). Let Z_c^b and Z_c^s denote the solution values of the two relaxations: the continuous bounds are then $L_c^b = \lceil Z_c^b \rceil$ and $L_c^s = \lceil Z_c^s \rceil$, respectively for 2LBP and 2LSP. We show that these bounds dominate the *area bounds*.

Theorem 1 *For any instance of 2LSP, $L_c^s \geq L_a^s$.*

Proof Observe that the solution implied by L_a^s is a strip of height $\sum_{j=1}^n w_j h_j / W$, whose area is totally occupied by “liquified” items. An optimal solution to the continuous relaxation of model (2LSP) is a strip of height Z_c^s . In this solution we may have:

- (i) fractional x_{ij} values, implying that a $(w_j x_{ij}) \times h_j$ slice of item j is packed in level i ;
- (ii) fractional y_i values, implying that item i initializes a level of packable width $W y_i$ (see (7)) and gives a contribution $h_i y_i$ to the objective function value (see (14)).

The proof that $Z_c^s \geq \sum_{j=1}^n w_j h_j / W$ easily follows from the following observation. Consider a level i with $y_i > 0$ (fractional or not) in the continuous relaxation, and any horizontal

sector of height $h_i y_i$ in the area relaxation: they give the same contribution $h_i y_i$ to the objective function, and pack a total item area which is: (a) not greater than $W h_i y_i$ in the continuous relaxation, and (b) exactly equal to $W h_i y_i$ in the area relaxation. Hence the thesis, since we could match each level of the continuous solution, in any order, to a different horizontal sector of the area solution having the same cost and packing a larger area, until no more feasible horizontal sector is available. \square

It is not difficult to extend to 2LBP the proof of Theorem 1 by observing that in an optimal solution to the continuous relaxation of model (2LBP), in addition to the fractional values described in (i) and (ii) above we may have fractional z_{ki} values, and fractional q_k values. Hence

Theorem 2 *For any instance of 2LBP, $L_c^b \geq L_a^b$.*

4 Better Lower Bounds

In this section we introduce combinatorial lower bounds for 2LSP and 2LBP, and discuss their relevant properties.

4.1 A Lower Bound for 2LSP

Consider the relaxation of 2LSP, say R(2LSP), obtained by allowing any item to be split into slices of integer width through vertical cuts. We next show that this relaxed instance can be solved to optimality in linear time (plus $O(n \log n)$ time for the initial sorting). Consider indeed the following simple algorithm.

Algorithm CUT-S

Assume that the items are sorted by non-increasing h_j values. Initialize the first level at height h_1 , consecutively pack into it items $1, 2, \dots$, until the first item i is found which does not fit. Split item i into two slices: one having width $\delta = W - \sum_{j=1}^{i-1} w_j$, the other having width $w_i - \delta$. Pack the first slice (possibly null, if $\delta = 0$) into the first level, and initialize the next level at height h_i by packing the second slice into it. Proceed in the same way until all items are packed. Let H_1, H_2, \dots, H_t (with $H_i \geq H_{i+1} \forall i$) be the heights of the t resulting levels.

Theorem 3 *Value*

$$L_{cut}^s = \sum_{j=1}^t H_j \tag{18}$$

is a valid lower bound for 2LSP.

Proof It is enough to prove that the solution S given by algorithm CUT-S is optimal for R(2LSP). Observe that a property of S is that the total width of items and slices packed in any level (but possibly the last one) is exactly equal to W . Consider now an optimal

solution to R(2LSP), sort the levels by non-increasing heights \widetilde{H}_i and note that, if it does not satisfy the above property, we can obtain an equivalent optimal solution \widetilde{S} satisfying it by simply shifting left and down the items/slices, possibly by introducing further cuts. Consider the first level of S and \widetilde{S} , note that we must have $H_1 = \widetilde{H}_1 = h_1$. Split the items and slices packed in these two levels into unit width slices, sort them (in each level) by non-increasing height, and let x be the first unit slice of S for which the slice y in the same position in \widetilde{S} is different (if any). By construction, $h_x > h_y$. Let \tilde{i} be the level where x is packed in \widetilde{S} : since $\widetilde{H}_{\tilde{i}} \geq h_x$, we can swap the two slices in \widetilde{S} without increasing the corresponding solution value. By iterating, we obtain an optimal solution with the first level identical to the one in S . We can now consider the residual instance obtained by removing the items and slices packed in the first level, and repeat until $S = \widetilde{S}$. \square

We will use later the following

Corollary 1 *Any feasible solution to R(2LSP) has at least as many levels as those produced by algorithm CUT-S, and, if these are sorted by non-increasing height, each level is high at least as the corresponding level of CUT-S.*

Proof Immediate from the considerations introduced in the proof of Theorem 3. \square

Corollary 2 *Although relaxation R(2LSP) allows any item to be cut into any number of integral slices, it admits an optimal solution where at most one cut per item is introduced.*

Proof Immediate from the description of algorithm CUT-S, which produces an optimal solution to R(2LSP). \square

We now show that lower bound L_{cut}^s dominates the bounds of the previous section. Indeed,

Theorem 4 *For any instance of 2LSP, $L_{cut}^s \geq L_c^s$.*

Proof We prove that L_{cut}^s is the optimal solution value to a less relaxed problem with respect to the LP relaxation of the ILP model (2LSP) of Section 2. To this end, consider the MIP relaxation of (2LSP) in which only the integrality requirement on variables x_{ij} is drop, with the additional requirement that, for all i and j , $w_j x_{ij}$ be integer. In the resulting problem, the items initializing the levels cannot be split, while vertical cuts are allowed for the remaining items. Problem R(2LSP) is a further relaxation in which cuts are allowed for all items. However, given any solution produced by CUT-S, we can obtain an equivalent solution as follows. Let j be the first item split by CUT-S, with its first slice, s_1 , packed in level i and its second slice, s_2 , initializing level $i + 1$. Since $H_i \geq H_{i+1}$, the following operation is feasible: move s_2 to level $i + 1$, cut a portion of the remaining packing of level $i + 1$ having the same width as s_1 and move it to level i . By iterating the process we obtain an equivalent solution which is feasible for the MIP relaxation. \square

We finally discuss the worst-case behavior of the new bound. Given a minimization problem P and a lower bounding procedure L , let $L(I)$ and $OPT(I)$ denote the values

of the lower bound and the optimal solution, respectively, for an instance I of P . The *absolute worst-case performance ratio* of L is then defined as

$$r(L) = \inf_{I \in P} \left\{ \frac{L(I)}{OPT(I)} \right\} \quad (19)$$

Theorem 5 $r(L_{cut}^s) = \frac{1}{2}$.

Proof We first show that, for any instance I of 2LSP, $L_{cut}^s(I) \geq \frac{1}{2}OPT(I)$. Consider the solution S to relaxation R(2LSP) produced by algorithm CUT-S. We may derive from S a feasible solution to 2LSP having height at most twice that of S . Consider indeed the lowest level which packs a split item and observe that, by construction, the only slice on the level is in the rightmost position. Replace the level by two levels, say s_a and s_b , having the same height as the original one: pack the unsplit items in s_a , join the slice to the remaining part of the item (currently packed in the leftmost position of the next level) and pack the reconstructed item in s_b . By iterating, we obtain an unsplit solution whose height is at most $2L_{cut}^s(I)$. To prove that the worst-case bound is tight, consider the series of instances I with $n = W$, $w_j = W/2 + 1$ and $h_j = 1$ for $j = 1, \dots, n$: we have $OPT(I) = n$ and $L_{cut}^s(I) = \lceil (n(W/2 + 1))/W \rceil = n/2 + 1$, so the ratio $L_{cut}^s(I)/OPT(I)$ is arbitrarily close to $1/2$ for n sufficiently large. \square

4.2 A Lower Bound for 2LBP

Consider the relaxation of 2LBP, say R(2LBP), obtained by allowing: (i) any item to be split into slices of integer width through vertical cuts (as for 2LSP) and (ii) any level to be split into sectors of integer height through horizontal cuts. This relaxed instance can be optimally solved in linear time (once the items are sorted) through the following two-step algorithm.

Algorithm CUT-B

Step 1: execute algorithm CUT-S of the previous section.

Step 2: consecutively pack levels $1, 2, \dots$ in the first bin, until the first level k is found which does not fit. Horizontally split level k (i.e., the items and slices packed into it) into two sectors: one having height $\delta = H - \sum_{i=1}^{k-1} H_i$, the other having height $H_k - \delta$. Pack the first sector (possibly null, if $\delta = 0$) into the first bin, and initialize the next bin with the second sector. Proceed in the same way until all levels are packed.

Theorem 6 *Value*

$$L_{cut}^b = \lceil L_{cut}^s/H \rceil \quad (20)$$

is a valid lower bound for 2LBP.

Proof It is easy to prove that the solution produced by algorithm CUT-B is optimal for R(2LBP). We have already shown, in the proof of Theorem 3, that algorithm CUT-S gives

the set of levels with minimum total height L_{cut}^s . It is then enough to observe that Step 2 of CUT-B produces bins (but possibly the last one) with no empty sector. \square

In practice, Step 2 of CUT-B determines a solution to R(2LBP) by solving a continuous relaxation of the one-dimensional bin packing problem induced by the level heights H_1, H_2, \dots, H_t produced by CUT-S, with bin capacity H . It follows that a better bound can be obtained by computing a more accurate lower bound for this instance, e.g. by using lower bound L_2 of Martello and Toth [18, 17]. Indeed, as shown by Corollary 1, the above 1BP instance is minimal both with respect to the number of elements (levels) and to their sizes (heights). The effectiveness of this approach is computationally tested in Section 5, where the improved bound is referred to as L_{cut}^b .

Lower bound L_{cut}^b dominates the bounds of Section 3:

Theorem 7 *For any instance of 2LBP, $L_{cut}^b \geq L_c^b$.*

Proof It is not difficult to extend the proof of Theorem 4. In this case we show that L_{cut}^b is the optimal solution value to the MIP relaxation of (2LBP) in which the integrality requirement on variables x_{ij} and z_{ki} is drop, with the additional requirements that $w_j x_{ij}$ be integer for all i and j and $h_i z_{ki}$ be integer for all i and k , i.e., to a less relaxed problem with respect to the LP relaxation of model (2LBP) of Section 2. In this MIP, the levels initializing the bins cannot be split, while horizontal cuts are allowed for the remaining levels. In R(2LBP) all the levels can be horizontally split. However, for any solution produced by CUT-B, we can obtain an equivalent MIP solution where the levels initializing the bins are not split, by an easy extension of the technique given in the proof of Theorem 4 for re-joining the split items initializing levels. \square

We can also extend the proof of Theorem 5 to prove the following

Theorem 8 $r(L_{cut}^b) = \frac{1}{4}$.

Proof We show that, for any instance I of 2LBP a feasible solution exists which uses no more than $4L_{cut}^b(I)$ bins. We have shown, in the proof of Theorem 5, that, for any instance I of 2LBP, the solution produced by Step 1 of algorithm CUT-B can be transformed into a feasible solution for 2LSP consisting of a strip of height not exceeding $2L_{cut}^s(I)$. Suppose now that we execute Step 2 of CUT-B on this solution. We can then produce a feasible solution for 2LBP as follows. Consider the first bin which packs a split level: remove the two level sectors (from this bin and from the next one), re-join them and pack the reconstructed level into a new bin. By iterating, we obtain a feasible 2LBP solution using at most $2 \lceil 2L_{cut}^s(I)/H \rceil \leq 4 \lceil L_{cut}^s(I)/H \rceil$ bins. To prove that the worst-case bound is tight, consider the series of instances I with $W = H = 2n$, $w_j = h_j = n + 1$ for $j = 1, \dots, n$: we have $OPT(I) = n$ and, by assuming that n is such that the ‘ $\lceil \cdot \rceil$ ’ operations have no effect, $L_{cut}^b(I) = n/4 + 1/2 + 1/(4n)$. Hence the ratio $L_{cut}^b(I)/OPT(I)$ is arbitrarily close to $1/4$ for n sufficiently large. \square

5 Computational Experiments

In this section we evaluate the average quality of the proposed lower bounds through computational experiments on randomly generated instances. In addition, we test the effectiveness of models (2LBP) and (2LSP), by using the Cplex 6.5.3 package (with no specific tailoring) to compute both their continuous relaxation and their optimal integer solution. The computational testings presented in Tables 1 and 2 have been performed on a Digital Alpha 533 MHz, on ten classes of instances proposed in the literature for 2BP. The first four classes were proposed by Martello and Vigo [19], and are based on the generation of items of four different types:

type 1 : w_j uniformly random in $[\frac{2}{3}W, W]$, h_j uniformly random in $[1, \frac{1}{2}H]$;

type 2 : w_j uniformly random in $[1, \frac{1}{2}W]$, h_j uniformly random in $[\frac{2}{3}H, H]$;

type 3 : w_j uniformly random in $[\frac{1}{2}W, W]$, h_j uniformly random in $[\frac{1}{2}H, H]$;

type 4 : w_j uniformly random in $[1, \frac{1}{2}W]$, h_j uniformly random in $[1, \frac{1}{2}H]$.

Class k ($k \in \{1, 2, 3, 4\}$) is then obtained by generating an item of type k with probability 70%, and of the remaining types with probability 10% each. The strip width is always $W = 100$, and the bin size $W = H = 100$. The next six classes have been proposed by Berkey and Wang [3]:

Class 5 : $W = 10$, w_j and h_j uniformly random in $[1, 10]$;

Class 6 : $W = 30$, w_j and h_j uniformly random in $[1, 10]$;

Class 7 : $W = 40$, w_j and h_j uniformly random in $[1, 35]$;

Class 8 : $W = 100$, w_j and h_j uniformly random in $[1, 35]$;

Class 9 : $W = 100$, w_j and h_j uniformly random in $[1, 100]$;

Class 10 : $W = 300$, w_j and h_j uniformly random in $[1, 100]$.

For 2LBP, we have $H = W$.

For each class and value of n ($n \in \{20, 40, 60, 80, 100\}$), ten instances have been generated. The 500 instances, as well as the generator code, are available on the internet at <http://www.or.deis.unibo.it/ORinstances/2BP/>.

Table 1 gives the results for 2LSP. The entries give the average ratios of the area, continuous and combinatorial lower bound (L_a^s , L_c^s , and L_{cut}^s , respectively) computed with respect to the best incumbent solution value z determined by the branch-and-bound of Cplex 6.5.3 within a time limit of 300 CPU seconds. (Notice that instances which were not solved to optimality get smaller L/z ratios than if sufficient time was given to the algorithm.) In addition, the table reports the number of instances solved to proved

Table 1: Results for 2LSP. Random problem instances proposed by Martello and Vigo (Classes 1-4), and by Berkey and Wang (Classes 5-10).

Class	n	Lower Bounds			Cplex		
		L_a^s/z	L_c^s/z	L_{cut}^s/z	<i>Solved</i>	<i>TimeB&B</i>	<i>Nodes</i>
1	20	0.82	0.86	0.90	10	0.01	2.0
	40	0.83	0.85	0.87	10	0.05	7.1
	60	0.85	0.86	0.88	10	0.42	38.7
	80	0.85	0.86	0.87	10	0.63	45.0
	100	0.86	0.87	0.88	10	0.81	39.6
2	20	0.87	0.90	0.97	10	17.40	9188.6
	40	0.92	0.93	0.97	5	166.08	14064.2
	60	0.93	0.94	0.97	0	300.06	11535.0
	80	0.94	0.94	0.96	0	300.10	6363.5
	100	0.94	0.95	0.96	0	300.17	3827.1
3	20	0.80	0.82	0.85	10	0.01	1.6
	40	0.80	0.81	0.82	10	0.03	2.1
	60	0.79	0.80	0.81	10	0.06	2.0
	80	0.80	0.80	0.81	10	0.12	1.5
	100	0.82	0.82	0.83	10	0.33	7.9
4	20	0.85	0.88	0.98	10	0.87	218.3
	40	0.89	0.92	0.96	10	33.68	2765.6
	60	0.91	0.92	0.96	5	220.51	9865.6
	80	0.92	0.93	0.95	3	239.66	5531.2
	100	0.93	0.94	0.96	0	300.13	3514.4
5	20	0.88	0.91	0.95	10	0.09	34.0
	40	0.91	0.92	0.95	10	1.09	265.4
	60	0.92	0.93	0.95	9	31.17	5461.6
	80	0.93	0.93	0.94	9	31.43	7522.7
	100	0.96	0.96	0.97	8	91.05	5601.7
6	20	0.81	0.89	1.00	10	0.74	193.8
	40	0.89	0.91	1.00	4	220.06	35329.0
	60	0.91	0.92	0.98	0	300.04	20581.5
	80	0.92	0.93	0.98	0	300.07	15579.2
	100	0.93	0.93	0.97	0	300.11	11971.1
7	20	0.84	0.88	0.94	10	0.14	31.2
	40	0.88	0.89	0.93	10	2.41	245.5
	60	0.90	0.91	0.93	9	33.98	1265.9
	80	0.90	0.91	0.92	10	10.75	378.1
	100	0.92	0.92	0.94	6	144.97	2891.6
8	20	0.79	0.89	0.99	10	0.87	112.8
	40	0.87	0.91	0.99	6	167.25	17464.2
	60	0.89	0.91	0.97	0	300.05	14235.7
	80	0.90	0.91	0.95	0	300.10	7711.6
	100	0.90	0.91	0.95	0	300.13	5783.4
9	20	0.84	0.87	0.92	10	0.11	16.8
	40	0.87	0.88	0.91	10	0.97	75.6
	60	0.89	0.90	0.92	10	12.42	761.3
	80	0.89	0.90	0.91	10	23.57	1231.2
	100	0.92	0.93	0.94	6	136.12	2658.1
10	20	0.77	0.88	1.00	10	0.88	120.2
	40	0.86	0.90	0.99	4	197.56	21029.3
	60	0.88	0.91	0.97	0	300.04	11562.9
	80	0.89	0.91	0.96	0	300.08	8007.6
	100	0.89	0.90	0.94	0	300.13	5630.5

Table 2: Results for 2LBP. Random problem instances proposed by Martello and Vigo (Classes 1-4), and by Berkey and Wang (Classes 5-10).

Class	n	Lower Bounds				Cplex		
		L_a^b/z	L_c^b/z	L_{cut}^b/z	$L2_{cut}^b/z$	<i>Solved</i>	<i>TimeB&B</i>	<i>Nodes</i>
1	20	0.83	0.87	0.90	0.92	10	0.13	8.9
	40	0.84	0.86	0.88	0.88	10	1.79	44.8
	60	0.87	0.87	0.90	0.90	10	30.35	606.7
	80	0.84	0.85	0.87	0.87	7	120.24	1682.8
	100	0.85	0.85	0.87	0.87	5	186.60	1101.1
2	20	0.79	0.82	0.85	0.98	10	0.13	7.7
	40	0.84	0.86	0.90	0.98	10	5.05	158.4
	60	0.86	0.87	0.89	0.99	9	51.62	997.4
	80	0.84	0.85	0.86	0.97	4	200.41	2010.8
	100	0.84	0.84	0.85	0.96	3	222.41	1073.0
3	20	0.66	0.66	0.71	0.83	10	0.01	1.0
	40	0.65	0.66	0.68	0.81	10	0.07	1.6
	60	0.63	0.64	0.65	0.79	10	0.12	1.0
	80	0.64	0.65	0.65	0.79	10	0.23	1.0
	100	0.65	0.65	0.66	0.81	10	0.38	1.0
4	20	0.84	0.93	0.93	0.96	10	0.41	5.7
	40	0.90	0.93	0.95	0.95	10	13.54	206.6
	60	0.87	0.92	0.92	0.93	6	130.37	1073.2
	80	0.89	0.90	0.93	0.93	5	193.98	865.5
	100	0.88	0.89	0.91	0.91	1	278.74	366.7
5	20	0.88	0.88	0.93	0.95	10	0.11	2.8
	40	0.88	0.89	0.91	0.94	10	1.33	30.5
	60	0.91	0.91	0.93	0.95	10	2.98	20.7
	80	0.91	0.91	0.93	0.95	10	11.01	132.0
	100	0.94	0.94	0.95	0.97	10	59.15	496.3
6	20	1.00	1.00	1.00	1.00	10	0.23	13.3
	40	0.95	1.00	1.00	1.00	10	3.52	76.4
	60	0.90	0.90	1.00	1.00	9	63.15	1566.5
	80	0.90	0.90	0.95	0.95	7	155.70	2076.4
	100	0.88	0.88	0.90	0.90	5	194.94	1458.3
7	20	0.83	0.83	0.87	0.95	10	0.21	1.0
	40	0.84	0.86	0.88	0.95	10	2.87	21.6
	60	0.90	0.91	0.93	0.96	10	36.78	279.8
	80	0.88	0.88	0.89	0.92	9	52.06	437.9
	100	0.88	0.89	0.90	0.92	4	207.85	484.3
8	20	1.00	1.00	1.00	1.00	10	0.25	13.1
	40	0.95	0.95	1.00	1.00	10	3.90	60.0
	60	0.83	0.90	0.93	0.93	7	126.24	2322.8
	80	0.80	0.83	0.88	0.88	5	188.68	2245.2
	100	0.87	0.87	0.92	0.92	5	174.74	1236.9
9	20	0.81	0.83	0.84	0.88	10	0.17	1.0
	40	0.82	0.85	0.87	0.90	10	2.61	43.0
	60	0.86	0.87	0.88	0.92	10	16.99	152.5
	80	0.86	0.86	0.88	0.90	10	33.70	318.2
	100	0.88	0.89	0.89	0.91	5	203.12	747.6
10	20	1.00	1.00	1.00	1.00	10	0.18	10.6
	40	0.80	0.95	1.00	1.00	10	3.66	43.9
	60	0.87	0.87	0.90	0.90	7	102.44	1726.2
	80	0.93	0.93	0.93	0.93	7	115.10	1222.1
	100	0.81	0.81	0.86	0.86	4	199.40	1327.7

optimality by Cplex (within 300 CPU seconds), the average computing time required by the branch-and-bound algorithm, and the average number of explored nodes.

Table 2 gives the same information for 2LBP. In this case we consider the area, continuous and combinatorial lower bound (L_a^b , L_c^b , and L_{cut}^b , respectively), as well as the improved bound $L2_{cut}^b$.

The results show that the LP relaxations of (2LSP) and (2LBP) improve on the area relaxations, respectively by 1.9 and 1.5 percentage points on average. Using these LP relaxations, a standard branch-and-bound package was able to solve to optimality 324 2LSP instances out of 500, and 414 2LBP instances out of 500, within five minutes. For the cases where the optimum was not attained, the package always produced an integer solution within the imposed time limit.

Despite the fact that the worst-case behavior is better for the 2SBP bounds than for the 2LBP bounds, there is no significant difference in their average performance. However, the level strip packing problem turns out to be substantially harder than the level bin packing problem, probably because the ILP solver has more difficulty in closing a higher absolute gap between lower bound and optimal solution value. Indeed, the range of possible solution values is much higher for the strip packing instances.

The combinatorial bounds considerably improve on the LP relaxations of (2LSP) and (2LBP), respectively by 3.9 and 2.2 percentage points on average. For 2LBP, the use of the 1BP bound in the final computation leads to a further improvement by 3.6 percentage points on average. These results suggest that specifically tailored branch-and-bound or branch-and-cut algorithms could be successfully implemented both for 2LSP and 2LBP.

It is interesting to note that problems of Classes 2, 4, 6, 8, and 10 are in general more difficult to solve, although the lower bounds are quite close to the best known solution value. A possible explanation of this behavior is the fact that, for such classes, the ratio $W/(\text{average item width})$ is considerably larger than for the other classes. Hence the instances have a more complex combinatorial structure (each level packs, on average, more items), while the continuous relaxations adopted in the various bounds have a somehow lesser effect.

We finally consider the use of the models for the relevant variant of both problems in which we allow rotation of the items by 90° . We performed experiments on the models introduced in Section 2.1, by using the same instances generated for 2LSP and 2LBP (Tables 1 and 2). Since these models have twice as many variables as (2LSP) and (2LBP), we found it reasonable to set to 600 seconds the time limit for the Cplex 6.5.3 ILP solver. The results are summarized in Table 3, where each entry refers to the 50 instances generated for a class. The LP relaxations of the models hardly improve on the area relaxations. Nevertheless, the ILP solver was able to solve to optimality 209 2LSP instances out of 500, and 316 2LBP instances out of 500. For the cases where the optimum was not attained, the package produced an integer solution within the imposed time limit in all the 2LSP instances and in 470 out of 500 2LBP instances. These results confirm that the proposed models are a practical tool for a simple solution approach to level packing problems.

Table 3: Results for 2LSP and 2LBP with rotation allowed.

Class	2LSP with rotation					2LBP with rotation				
	Lower Bounds		Cplex			Lower Bounds		Cplex		
	L_a^s/z	L_c^s/z	Solved	TimeB&B	Nodes	L_a^b/z	L_c^b/z	Solved	TimeB&B	Nodes
1	0.96	0.96	17	417.86	6358.9	0.85	0.85	34	272.75	629.6
2	0.90	0.91	10	489.80	5108.0	0.60	0.60	25	349.60	408.1
3	0.96	0.96	15	444.26	11370.2	0.88	0.88	28	307.82	669.8
4	0.96	0.96	14	438.26	9849.0	0.88	0.88	27	302.43	568.0
5	0.90	0.90	50	4.08	266.8	0.64	0.64	50	0.83	3.0
6	0.95	0.96	15	456.29	5969.5	0.85	0.85	27	324.70	352.2
7	0.99	0.99	47	86.66	2252.5	0.91	0.91	43	183.50	305.8
8	0.94	0.94	15	457.59	7404.7	0.67	0.67	29	293.69	366.1
9	0.96	0.96	16	435.54	6783.7	0.85	0.85	27	321.60	427.5
10	0.92	0.92	10	494.29	5999.0	0.65	0.65	26	335.96	609.2

Acknowledgments

We thank the Ministero dell'Università e della Ricerca Scientifica e Tecnologica (M.U.R.S.T.) and the Consiglio Nazionale delle Ricerche (C.N.R.), Italy, for the support given to this project. The computational experiments have been executed at the Laboratory of Operations Research of the University of Bologna (Lab.O.R.). Thanks are also due to two anonymous referees for helpful comments.

References

- [1] B.S. Baker and J.S. Schwarz. Shelf algorithms for two-dimensional packing problems. *SIAM Journal on Computing*, 12:508–525, 1983.
- [2] J.E. Beasley. An exact two-dimensional non-guillotine cutting tree search procedure. *Operations Research*, 33:49–64, 1985.
- [3] J.O. Berkey and P.Y. Wang. Two dimensional finite bin packing algorithms. *Journal of the Operational Research Society*, 38:423–429, 1987.
- [4] C.S. Chen, S.M. Lee, and Q.S. Shen. An analytical model for the container loading problem. *European Journal of Operational Research*, 80:68–76, 1995.
- [5] F.K.R. Chung, M.R. Garey, and D.S. Johnson. On packing two-dimensional bins. *SIAM Journal of Algebraic and Discrete Methods*, 3:66–76, 1982.
- [6] E.G. Coffman, Jr., M.R. Garey, D.S. Johnson, and R.E. Tarjan. Performance bounds for level-oriented two-dimensional packing algorithms. *SIAM Journal on Computing*, 9:801–826, 1980.
- [7] K. Daniels, V.J. Milenkovic, and Z. Li. Multiple containment methods. Technical Report 12-94, Center for Reserch in Computing Technology, Division of Applied Sciences, Harvard University, 1994.

- [8] J.M.V. de Carvalho and A.J.G. Rodrigues. A computer based interactive approach to a two-stage cutting stock problem. *INFOR*, 32:243–252, 1994.
- [9] J.M.V. de Carvalho and A.J.G. Rodrigues. A LP-based approach to a two-stage cutting stock problem. *European Journal of Operational Research*, 84:580–589, 1995.
- [10] H. Dyckhoff, G. Scheithauer, and J. Terno. Cutting and Packing (C&P). In M. Dell’Amico, F. Maffioli, and S. Martello, editors, *Annotated Bibliographies in Combinatorial Optimization*. John Wiley & Sons, Chichester, 1997.
- [11] J.B. Frenk and G.G. Galambos. Hybrid next-fit algorithm for the two-dimensional rectangle bin-packing problem. *Computing*, 39:201–217, 1987.
- [12] P. C. Gilmore and R. E. Gomory. Multistage cutting problems of two and more dimensions. *Operations Research*, 13:94–119, 1965.
- [13] E. Hadjiconstantinou and N. Christofides. An exact algorithm for the orthogonal, 2-D cutting problems using guillotine cuts. *European Journal of Operational Research*, 83:21–38, 1995.
- [14] M. Hifi. Contribution à la résolution de quelques problèmes difficiles de l’optimization combinatoire. Habilitation thesis, PRiSM, Université de Versailles St-Quentin en Yvelines, 1999.
- [15] A. Lodi, S. Martello, and D. Vigo. Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems. *INFORMS Journal on Computing*, 11:345–357, 1999.
- [16] A. Lodi, S. Martello, and D. Vigo. Recent advances on two-dimensional bin packing problems. *Discrete Applied Mathematics*, 2001 (to appear).
- [17] S. Martello and P. Toth. *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, Chichester, 1990.
- [18] S. Martello and P. Toth. Lower bounds and reduction procedures for the bin packing problem. *Discrete Applied Mathematics*, 28:59–70, 1990.
- [19] S. Martello and D. Vigo. Exact solution of the two-dimensional finite bin packing problem. *Management Science*, 44:388–399, 1998.
- [20] H. Onodera, Y. Taniguchi, and K. Tamaru. Branch-and-bound placement for building block layout. In *Proceedings of the 28th ACM/IEEE Design Automation Conference*, pages 433–439, 1991.
- [21] F. Vanderbeck. A nested decomposition approach to a 3-stage 2-dimensional cutting stock problem. *Management Science*, 47:864–879, 2001.