A new lower bound for the non-oriented two-dimensional bin-packing problem

François Clautiaux[a*], Antoine Jouglet [a] and Joseph El Hayek [a]

[a]HeuDiaSyC, UMR CNRS 6599, Université de Technologie de Compiègne

We propose a new scheme for computing lower bounds for the non-oriented bin-packing problem when the bin is a square. It leads to bounds that theoretically dominate previous results. Computational experiments show that the bounds are tight. We also discuss the case where the bin is not a square.

Keywords

non-oriented bin-packing problem, lower bounds

1. Introduction

The two-dimensional discrete bin-packing problem ($2BP$) consists in minimizing the number of identical rectangles used to pack a set of smaller rectangles. This problem is *NP-complete*. It occurs in industry if pieces of steel, wood, or paper have to be cut from larger rectangles. It belongs to the family of *cutting and packing* (C & P) problems, more precisely *Two-Dimensional Single Bin Size Bin-Packing Problems* (2SBSBPP), following the typology of Wäscher *et al.* [10].

A $2BP$ instance D is a pair (I, B). It is composed of the set $I = \{1, \ldots, n\}$ of items i to pack, and a bin $B = (W, H)$ of width W and height H ($W, H \in \mathbb{N}$). For the oriented case ($2BP|O$), an item i is of width w_i and height h_i ($w_i, h_i \in \mathbb{N}$). We consider the non-oriented version of the problem ($2BP|R$), *i.e.* where the items can be rotated. So we consider the two possible orientations: $w_i^1 = h_i^2$ and $w_i^2 = h_i^1$. Before rotation, an item is of size (w_i^1, h_i^1), otherwise it is of size (w_i^2, h_i^2). An orientation of the set of items is an application r from I to $\{1, 2\}$ that associates an orientation $r(i)$ to each item i in I.

Many methods [2, 4, 5, 7, 9] have been proposed for computing lower bounds for $2BP|O$. Most of them can be computed by means of so-called *dual-feasible functions*. These functions cannot be directly applied if the orientation of the items is not known. Fewer results exist for $2BP|R$ [3, 6]. Two schemes were proposed for computing such bounds: cutting the items into squares [3,6], or explicitly using dual-feasible functions [3]. It would appear that the difference between upper and lower bounds is larger than for the oriented case.

In this paper we propose a new general scheme for computing lower bounds for $2BP|R$ when the bin is a square. It is based on the construction of a $2BP|O$ instance composed of $2n$ items: each item appears once for each orientation. We show that if the original instance needs z bins, the $2n$-items instance needs at most $2z$ bins. When this instance is computed, bounds designed for the oriented case can be applied. We show that when the bin is a square, the bounds of Carlier *et al.* [5] used in this general framework dominate those of Dell' Amico *et al.* [6], as well as those of Boschetti and Mingozzi [3]. When the bin is not a square, the above method can be used, if a suitable preprocessing is performed.

We tested our method against well-known benchmarks proposed by [1, 9]. Computational results show that the dominance is not only theoretical, as the bounds are improved for several instances.

In Section 2 we describe several results from the literature for the oriented case, which are used in

*Corresponding author: F. Clautiaux, Université de Technologie de Compiègne, BP 20529, 60205 Compiègne, France. Email: francoisclautiaux@gmail.com

our new bounds. We also recall previous results for the non-oriented case. Section 3 deals with the new lower bounds. We present several dominance results in Section 4. Section 5 is devoted to computational experiments.

2. Literature Review

In this section, we recall the lower bounds that are useful in this paper. First we describe the bounds of [5] for the oriented case. These bounds are used in our new lower bounds described in Section 3. Next we describe the bounds of [2] for the oriented case. Finally we describe the frameworks used to compute bounds for the non-oriented case [3,6]. In Section 4, these frameworks are compared to our new method.

2.1. Lower bounds for $2BP|O$

2.1.1. The bound L_{CCM} of [5]

A classical lower bound for $2BP$ is the continuous bound L_0. $L_0 = \lceil \frac{\sum_{i \in I} w_i h_i}{WH} \rceil$. One way of improving this method is to apply *Dual-Feasible Functions* (DFF) [8] on both dimensions of the instance [7]. The bounds of [5] use the framework of [7], using a discretization of DFF.

Definition 2.1. *A* Discrete DFF f *is a discrete application from* $[0, X]$ *to* $[0, X']$ *(*X *and* X' *integers), such that* $x_1 + x_2 + \ldots + x_k \leq X \Rightarrow f(x_1) + f(x_2) + \ldots + f(x_k) \leq f(X) = X'$.

We only consider DFF that are increasing and superadditive, *i.e.* for a given DFF f, $x + y \leq z$ implies $f(x) + f(y) \leq f(z)$.

In [5] the authors also introduced the concept of *Data-Dependent DFF (DDFF)*. These functions behave like DFF for the specific instance considered. The authors used a classical family of DFF (f_0^k), an improvement on a DFF used by [3] (f_2^k), and a new family of DDFF (f_1^k). We now give the definitions of f_0^k, f_1^k, and f_2^k, which are required below. In what follows, C is an integer value, and $k = 1, \ldots, C/2$.

- $f_0^k(x) = 0$ if $x < k$, $f_0^k(x) = x$ if $k \leq x \leq C - k$, and $f_0^k(x) = C$ if $C - k < x \leq C$.
- The DDFF f_1^k is defined for C and a list of integer values $c_1, c_2, \ldots, c_n$. We introduce the set $J = \{i \in I : \frac{C}{2} \geq c_i \geq k\}$ and for a given integer Y, $M_C(Y, J)$ is the maximum number of items c_i such that $i \in J$, which can be packed together in a bin of size Y. $f_1^k(x) = 0$ if $x < k$, $f_1^k(x) = 1$ if $k \leq x \leq \frac{C}{2}$, and $f_1^k(x) = M_C(C - x, J)$ if $\frac{C}{2} < x$.
- $f_2^k(x) = 2\lfloor \frac{x}{k} \rfloor$ if $x < \frac{C}{2}$, $f_2^k(x) = \lfloor \frac{C}{k} \rfloor$ if $x = \frac{C}{2}$, and $f_2^k(x) = 2\lfloor \frac{C}{k} \rfloor - 2\lfloor \frac{C-x}{k} \rfloor$ if $x > \frac{C}{2}$.

$$L_{CCM}^{DDFF} = \max_{\substack{1 \leq k \leq \frac{W}{2}, 1 \leq l \leq \frac{H}{2}, \\ u \in \{0,1,2\}, v \in \{0,1,2\}}} \left\{ \left\lceil \sum_{i \in I} \frac{f_u^k(w_i) f_v^l(h_i)}{f_u^k(W) f_v^l(H)} \right\rceil \right\}$$

It is strengthened by using the bounds of [5] with the framework of $L_{BM}^{F,2}$ [2] described below. The obtained bound is termed L_{CCM}^F [5].

2.1.2. The bound L_{BM}^F of [2]

The bound L_{BM}^F of [2] is composed of four bounds $L_{BM}^{F,2a}$, $L_{BM}^{F,2b}$, $L_{BM}^{F,3}$, and $L_{BM}^{F,4}$. The first, $L_{BM}^{F,2a}$, consists in creating a $1BP$ instance by multiplying the width and the height of each item. The second, $L_{BM}^{F,2b}$, consists in considering *large*, *tall* and *wide* items separately (see [2] or [5] for more details). The bound $L_{BM}^{F,3}$ can be computed by means of two functions f_1^k (see [5]). Similarly, the bound $L_{BM}^{F,4}$ is the continuous bound of the instance obtained by applying a weaker version of f_2^k to each dimension of the instance. The bound L_{CCM}^F dominates L_{BM}^F [5].

2.1.3. The bounds of [4]

Caprara *et al.* [4] propose an exact method to find the best pair of DFF to be applied to a given instance of $2BP|O$. This problem is a relaxation of the two-dimensional bin-packing problem. The results obtained are good, which means that for many instances in the literature, the relaxation yields the same result as the original problem, *i.e.* there is a pair of DFF giving a bound equal to the optimal result. No clue is given for a generalization of this method for $2BP|R$.

2.2. Lower bounds for $2BP|R$

We now describe lower bounds from the literature for $2BP|R$. Two main schemes are used.

The first consists in cutting the items into squares [3, 6]. The second explicitly takes into account both dimensions of the items and the possibility of rotating them by 90 degrees [3].

2.2.1. The bound LB^R_{DA} of [6]

The first step of the bound LB^R_{DA} consists in replacing each item $i \in I$ by several squares. These square items are obtained by applying a pseudo-polynomial algorithm called CUTSQ [6]. Initially each rectangle is considered in the orientation r where $w_i^{r(i)} \geq h_i^{r(i)}$. At each iteration, CUTSQ cuts from each current rectangle (w_j, h_j) the maximum number of (h_j, h_j) squares, and rotates the residual rectangles by 90 degrees to conserve horizontal orientation. A lower bound for packing squares is then applied to the set of resulting squares denoted I_S. Let l_j be the resulting edge sizes of square $j \in I_S$. I_S is then divided into four subsets as follows. Given an integer $0 \leq k \leq \frac{1}{2}H$, five sets are defined: $I_1 = \{j \in I_S : l_j > W-k\}, I_2 = \{j \in I_S : W-k \geq l_j > \frac{1}{2}W\}, I_3 = \{j \in I_S : \frac{1}{2}W \geq l_j > \frac{1}{2}H\}, I_4 = \{j \in I_S : \frac{1}{2}H \geq l_j \geq k\}$, and $I_{23} = \{j \in I_2 \cup I_3 : l_j > H-k\}$

$$LB^R_{DA} = \max_{0 \leq k \leq \frac{1}{2}H} \Bigg\{ |I_1| + \tilde{L} + max\Bigg\{0, \left\lceil \frac{\sum_{j \in I_2 \cup I_3 \cup I_4} l_j^2 - (WH\tilde{L} - \sum_{j \in I_{23}} l_j(H-l_j))}{WH} \right\rceil \Bigg\}\Bigg\}$$

LB^R_{DA} is a valid lower bound for $2BP|R$, where $\tilde{L}$ is the following valid lower bound on the number of bins needed for packing items of $I_2 \cup I_3$:

$$\tilde{L} = |I_2| + max\left\{ \left\lceil \frac{\sum_{j \in I_3 \setminus \overline{I_3}} l_j}{W} \right\rceil, \left\lceil \frac{|I_3 \setminus \overline{I_3}|}{\left\lfloor \frac{W}{\lfloor H/2+1 \rfloor} \right\rfloor} \right\rceil \right\}$$

In this formulation, $\overline{I_3}$ is the set of the largest items of I_3 that can be packed into the bins that pack the items of I_2.

For a given value of k, when the bin is a square (W, W), the bound LB^R_{DA} can be written as follows:

$$L'^R_{DA}(k) = |I_1 \cup I_2| + max\{0, \lceil \frac{\sum_{j \in I_2 \cup I_4}(l_j)^2}{W^2} - |I_2| \rceil\}$$

2.2.2. Lower bound L^R_{BM} of [3]

The lower bound L^R_{BM} [3] is computed as the maximum of two lower bounds $L^{R,1}_{BM}$ and $L^{R,2}_{BM}$.

Lower bound $L^{R,1}_{BM}$ consists in transforming the items into items with fixed orientation. Thus, given the initial instance $D = (I, B)$, the items of size (w_j, h_j) are resized as follows: $(\overline{w_j}, \overline{h_j})$, where $\overline{w_j} = min(w_j, h_j)$ if $(w_j \leq H$ and $h_j \leq W)$, and $\overline{w_j} = w_j$ otherwise. $\overline{h_j} = min(w_j, h_j)$ if $(h_j \leq H$ and $w_j \leq W)$, and $\overline{h_j} = h_j$ otherwise.

This means that if an item cannot be rotated, it is left unchanged. Otherwise, only the largest square included in this item is considered. $L^{R,1}_{BM}$ is obtained by applying the lower bound L^F_{BM} [2] for $2BP|O$.

Lower bound $L^{R,2}_{BM}$ takes into account both dimensions of the items and the possibility of rotating them by 90 degrees. Two different subsets of items are considered: $I' = \{i \in I : w_i \neq \overline{w_i}\}$ and $I'' = I \setminus I'$. Given two integers k and l such that $1 \leq k \leq \frac{1}{2}H$ and $1 \leq l \leq \frac{1}{2}W$, and using our notation, a valid lower bound $L^{R,2}_{BM}$ can be written as follows

$$L^{R,2}_{BM} = \max_{k,l} \left\{ \left\lceil \frac{\sum_{i=1}^n \mu^{k,l}(i)}{\lfloor \frac{H}{k} \rfloor \lfloor \frac{W}{l} \rfloor} \right\rceil \right\}$$

where

$$\mu^{k,l}(i) = \begin{cases} min\{\eta^{l,W}(w_i) \times \eta^{k,H}(h_i), \\ \quad \eta^{l,W}(h_i) \times \eta^{k,H}(w_i)\} & \text{if } i \in I' \\ \eta^{l,W}(w_i) \times \eta^{k,H}(h_i) & \text{if } i \in I'' \end{cases}$$

and

$$\eta^{k,X}(x) = \begin{cases} \lfloor \frac{X}{k} \rfloor \lfloor \frac{X-x}{k} \rfloor & \text{if } x > \frac{X}{2} \\ \lfloor \frac{x}{k} \rfloor & \text{if } x \leq \frac{X}{2} \end{cases}$$

Note that $\eta^{k,X}$ is a version of f_2^k, where an item of size $\frac{X}{2}$ may have a smaller image.

The final bound proposed by [3] is $max(L^{R,1}_{BM}, L^{R,2}_{BM})$. We denote this bound L^R_{BM}.

3. A new lower bound for $2BP|R$

In this section we propose a new scheme for computing lower bounds for the non-oriented case when the bin is a square. It is based on the creation of a new $2BP|O$ instance. We also show how this instance can be additionally constrained for obtaining better results. At the end of the section, we discuss how our method can be generalized to rectangular bins.

Let $D^R = (I, B)$ be an instance of $2BP|R$. $I = \{1, \ldots, n\}$ is the set of items i and $B = (W, W)$ a bin. Now consider the following $2BP|O$ problem. $\widehat{D}^F = (\widehat{I}^F, B)$, with $\widehat{I}^F = \{1, \ldots, 2n\}$, such that

1. $w_i = w_i^1$ and $h_i = h_i^1$ for $i \in \{1, \ldots, n\}$
2. $w_i = w_{i-n}^2$ and $h_i = h_{i-n}^2$ for $i \in \{n + 1, \ldots, 2n\}$

Theorem 3.1 states that $\widehat{D}^F$ needs at most twice the number of bins needed for D^R.

Theorem 3.1. *If z bins are needed to pack items of D^R, there is a solution for $\widehat{D}^F$ with $2z$ bins.*

Proof. Consider the solution for D^R that needs z bins. In this solution, each i is used once in one of its two possible orientations. A solution with $2z$ bins can be found for $\widehat{D}^F$ by using the same configuration for the items that are in the solution of D^R, and packing the remaining items by rotating the z first bins by 90 degrees (see Figure 1). □

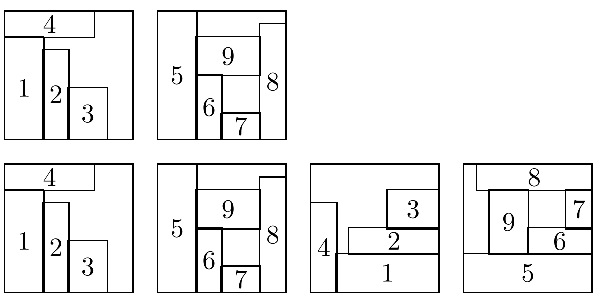

Figure 1. Obtaining a solution for $\widehat{D}^F$

In Figure 1, we show how a solution for $\widehat{D}^F$ with four bins (the four lower bins) can be obtained from a solution for D^R using two bins (the two upper bins). The first two bins use the same configuration as the initial problem. The configuration of bin three (resp. four) is obtained by rotating the configuration of bin one (resp. two) by 90 degrees.

Corollary 3.1. *If LB^F is a valid lower bound for $2BP|O$, then $\lceil LB^F(\widehat{D}^F)/2\rceil$ is a valid lower bound for the optimal solution of D^R.*

From Corollary 3.1, any lower bound for $2BP$ with fixed orientation can be used for finding a bound for a non-oriented problem. We use the bound L^F_{CCM} of [5], with a slight modification. In the bound we propose, we allow the two items related to the same original item i to be packed in the same bin. If this is not allowed, the bound remains valid. So the data-dependent set considered in f_1^k will only consider one item per couple (the one for which the considered dimension is smaller). We denote the obtained lower bound L^R_{CJE}. Its complexity is the same as the internal lower bound used ($O(n^3)$).

When the bin is not a square, the method can be used anyway. We suppose without loss of generality that $W > H$. In this case, we transform the bin (W, H) into a bin (W, W). To avoid over-relaxing the problem, dummy items of size $(W, W - H)$ are added (one for each bin) to fit the created area. The method is embedded in the following lifting procedure: assume the optimal solution needs z bins. In this case, z dummy items are added when the new instance is computed. If the lower bound is greater than z, the method is rerun with the value $z + 1$. The initial value z may be 1, or any valid lower bound for the problem. When f_1^k is used, we may ensure that only one dummy item at a time can pertain to the solution of a knapsack problem.

4. Dominance results

In this section, we show that when the bin is a square, the lower bound L^R_{CJE} proposed above dominates the methods of [6], and [3]. For this purpose, we introduce a new bound $L^{R,t}_{CJE}$, which dominates several previous bounds, and is dominated by our new method.

Let $\mathcal{F}$ be the set of valid functions f_0^k and f_2^k for an instance D. The bound $L_{CJE}^{R,t}$ generalizes the bound $L_{BM}^{R,2}$ of [3].

$$L_{CJE}^{R,t} = \max_{f,g\in\mathcal{F}} \left\lceil \sum_{i\in I} \frac{\min\left\{f(w_i^1)g(h_i^1), f(w_i^2)g(h_i^2)\right\}}{f(W)g(W)} \right\rceil$$

Proposition 4.1. *$L_{CJE}^{R,t}$ is a valid lower bound for $2BP|R$.*

Proof. If one computes all possible orientations for the set of items, and calculates a lower bound for $2BP|O$, the minimum value obtained is a lower bound. Let β be the set of all possible orientations r of the set of items.

$$OPT(I) \geq \min_{r\in\beta} \left\{ \max_{f,g\in\mathcal{F}} \left\lceil \sum_{i\in I} \frac{f(w_i^{r(i)})g(h_i^{r(i)})}{f(W)g(W)} \right\rceil \right\}$$

The expression above is greater than or equal to:

$$\min_{r\in\beta} \left\{ \max_{f,g\in\mathcal{F}} \left\lceil \sum_{i\in I} \frac{\min\{f(w_i^{r(i)})g(h_i^{r(i)}), f(h_i^{r(i)})g(w_i^{r(i)})\}}{f(W)g(W)} \right\rceil \right\}$$

The orientation chosen for the items does not change the expression above. The following is therefore a lower bound for $OPT(I)$.

$$\max_{f,g\in\mathcal{F}} \left\lceil \sum_{i\in I} \frac{\min\{f(w_i^1)g(h_i^1), f(w_i^2)g(h_i^2)\}}{f(W)g(W)} \right\rceil$$

□

We now show that $L_{CJE}^{R,t}$ is dominated by L_{CJE}^{R}. This result will help us to show that L_{CJE}^{R} dominates previous lower bounds.

Proposition 4.2. *If the bin is a square, the bound $L_{CJE}^{R,t}$ is dominated by L_{CJE}^{R}.*

Proof. Consider a weaker version of L_{CJE}^{R}, where only f_2 and f_0 are used. The obtained bound is the following:

$$LB = \left\lceil \frac{1}{2} \max_{f,g\in\mathcal{F}} \Big\{ \sum_{i=1}^{n} \frac{f(w_i^1)g(h_i^1)}{f(W)g(W)} + \sum_{i=n+1}^{2n} \frac{f(w_i^1)g(h_i^1)}{f(W)g(W)} \Big\} \right\rceil$$

By definition of $\widehat{I}^F$, we have for $i = 1, \ldots, n$, $w_i^1 = h_{i+n}^2$ and $h_i^1 = w_{i+n}^2$. Thus, the equation above can be rewritten:

$$LB = \left\lceil \max_{f,g\in\mathcal{F}} \left\{ \sum_{i=1}^{n} \frac{1}{2} \frac{f(w_i^1)g(h_i^1) + f(w_i^2)f(h_i^2)}{f(W)g(W)} \right\} \right\rceil$$

As $(f(w_i^1)g(h_i^1)+f(w_i^2)g(h_i^2))/2$ is greater than or equal to $\min\{f(w_i^1)g(h_i^1), f(w_i^2)g(h_i^2)\}$, the weaker bound proposed dominates $L_{CJE}^{R,t}$. So does L_{CJE}^{R}. □

We now show that $L_{CJE}^{R,t}$ dominates the bound of [6]. For the sake of comprehension, we first show a result concerning the DFF. Lemma 4.1 means that cutting items cannot lead to improved bounds when method $L_{CJE}^{R,t}$ is used. The lemma holds for both horizontal and vertical cuts.

Lemma 4.1. *For two DFF f and g, and four values w_a, w_b, w and h such that $w_a + w_b = w$, we have*

$$\begin{aligned} &\min\{f(w_a)g(h), g(w_a)f(h)\} \\ &+\min\{f(w_b)g(h), g(w_b)f(h)\} \\ &\qquad \leq \min\{f(w)g(h), g(w)f(h)\} \end{aligned}$$

Proof. Recall that we only consider superadditive DFF. As f is superadditive, we have $f(w_a) + f(w_b) \leq f(w)$. Thus $f(w_a)g(h) + f(w_b)g(h) \leq f(w)g(h)$. Then $\min\{f(w_a)g(h), g(w_a)f(h)\} + \min\{f(w_b)g(h), g(w_b)f(h)\} \leq f(w)g(h)$. Using the same property for g, we have $g(w_a)f(h) + g(w_b)f(h) \leq g(w)f(h)$. Thus $\min\{f(w_a)g(h), g(w_a)f(h)\} + \min\{f(w_b)g(h), g(w_b)f(h)\} \leq g(w)f(h)$.

□

Note that the result applies also when $w_a = h$, like in the method L^R_{DA} of [6].

Proposition 4.3. *When the bin is a square, L^R_{CJE} dominates the bound ${L'}^R_{DA}$ of [6].*

Proof. The bound ${L'}^R_{DA}$ [6] recursively cuts the set of rectangles of I into smaller rectangles. In both possible cases, ${L'}^R_{DA}(k)$ can be obtained by applying f_0^k or $f_0^{C/2}$ on both dimensions.

Now we prove that any method that cuts items and computes a lower bound using DFF on the residual instance cannot improve the results that consist in applying DFF to the original instance. Let i be an item in I, and J_i the set of items obtained by recursively cutting i into two rectangles. Using Lemma 4.1, we know that the area of i using $L^{R,t}_{CJE}$ is greater than the sum of the areas of the first two cut items. Using this result recursively, we obtain that the area of i modified by two DFF is greater than the total area of the items of J_i modified by two DFF. So by summing the obtained inequalities, we obtain that $L^{R,t}_{CJE}$ is greater than L^R_{DA}. As L^R_{CJE} dominates $L^{R,t}_{CJE}$ when the bin is a square, it also dominates L^R_{DA}. $\square$

We now show that our bound dominates the bound L^R_{BM} of [3].

Proposition 4.4. *When the bin is a square, L^R_{CJE} dominates the bound L^R_{BM} of [3].*

Proof. We have to show that L^R_{CJE} dominates the two bounds contained in L^R_{BM}.

When the bin is a square, the bound $L^{R,1}_{BM}$ (see Section 2) consists in considering the instance obtained by only keeping the largest square included in each item, and applying the three bounds $L^{F,2}_{BM}$, $L^{F,3}_{BM}$ and $L^{F,4}_{BM}$ described in [2] (see Section 2).

For $2BP|R$, when the bin is a square, the bound $L^{F,2a}_{BM}$ is equivalent to creating a $1BP$ instance where each item i has a size $(min\{w_i, h_i\})^2$ and the bin is of size W^2. Then DFF are applied (see Section 2). In the bound L^R_{CJE}, the instance considered is composed of items larger than those of $L^{F,2}_{BM}$. As the inner lower bound L^F_{CCM} uses the same technique as $L^{F,2a}_{BM}$, along with DFF that dominate those of [2], it dominates this bound (recall that the DFF we consider are superadditive).

When the smaller of the two dimensions is kept, only items with both large width and height are considered in the bound $L^{F,2b}_{BM}$. So the obtained bound cannot be larger than the bound obtained by applying $f_0^{\frac{W}{2}}$ to each dimension.

Carlier *et al.* [5] have shown that $L^{F,3}_{BM}$ can be computed by means of two functions f_1^k. Obviously, the total area of the items considered is larger when L^R_{CJE} is applied. Although the DFF we consider are superadditive, DDFF are not, so we need to recall that the set of data-dependent values J considered in L^R_{CJE} is the same as the set considered in $L^{R,1}_{BM}$. Indeed using the improvement on the bound L^F_{CCM} proposed in Section 3, only the value $min\{w_i^1, h_i^1\}$ is considered, which is the same as the value considered in $L^{R,1}_{BM}$. So the area obtained for each item is greater than or equal to the total area obtained using the method of [2, 3]. By summing the inequalities obtained, the dominance result holds.

Carlier *et al.* [5] have also shown that $L^{F,4}_{BM}$ can be computed by means of two DFF, which are dominated by f_2^k. Using Lemma 4.1 and Proposition 4.2 we directly deduce that L^R_{CJE} dominates the bound $L^{F,4}_{BM}$.

Finally the bound $L^{R,2}_{BM}$ [3] is obtained from the framework of bound $L^{R,t}_{CJE}$ using a weaker version of f_2^k, and thus it is directly dominated by $L^{R,t}_{CJE}$. Using Proposition 4.2, the bound $L^{R,2}_{BM}$ is dominated by L^R_{CJE}. $\square$

When the bin is not a square, the dominance relations do not hold any more. The following example illustrates this fact. Let us consider the following instance: $A = \{(4,1), (4,1), (3,3)\}$ and $B = (6,3)$. The method of [3] first determines a set of items that cannot be rotated: $(4,1)$ and $(4,1)$. So when f_0^3 is applied to the width and f_0^1 to the height, the lower bound obtained is equal to the following expression: $\lceil (6 \times 1 + 6 \times 1 + 3 \times 3)/(6 \times 3) \rceil = \lceil 21/18 \rceil = 2$. If our method is used, the following instance is created: $A = \{(6,3), (3,6), (4,1), (4,1), (3,3), (3,3), (1,4), (1,4)\}$

and $B = (6, 6)$. The value computed by our lower bound is one, whereas the lower bound computed by the method of [3] is two.

Lemma 4.1 does not apply when DDFF are used, since DDFF may not be superadditive. This means that the bounds obtained by using f_1^k in the framework of [6] can lead to results that are not dominated by L_{CJE}^R. However computational experiments we performed did not lead to any improvement compared to our method.

5. Computational experiments

We tested our method against well-known benchmarks derived from the literature [1, 9]. There are 10 classes of randomly generated test problems. Each class contains five groups of ten instances each.

In Table 1 we compare our results to those of [3], which are the best results known so far. For this purpose we report the number of times each bound is equal to the upper bound of [3] (columns Opt). To allow further comparisons, we also report the average value of each lower bound (column Avg). Columns BM refer to the results of [3], while columns CJE are related to the results of our new bound when the modified version of L_{CCM}^F [5] is used (see Section 3). In column CPU we report the computing time needed for one test case using our method.

The theoretical results are confirmed by experimentation. Moreover, 19 values of lower bounds are strictly improved compared to the results of [3]. They allow us to solve 15 additional instances when L_{CCM}^F is used.

6. Concluding remarks

Our bounds are tight when the bin is a square. However the drawback of our method is that it does not take into account the fact that some items may not be rotated. This may lead to weaker results when the bin is not a square, or when it is used within an enumerative method.

The framework we propose can be adapted to the three-dimensional bin-packing problem. In this case, the instance to construct would have $6n$ items.

REFERENCES

1. J. O. Berkey and P. Y. Wang. Two-dimensional finite bin-packing algorithms. *Journal of the Operational Research Society*, 38:423–429, 1987.
2. M. Boschetti and A. Mingozzi. The two-dimensional finite bin packing problem. part I: New lower bounds for the oriented case. *4OR*, 1:27–42, 2003.
3. M. Boschetti and A. Mingozzi. The two-dimensional finite bin packing problem. part II: New lower and upper bounds. *4OR*, 1:135–147, 2003.
4. A. Caprara, M. Locatelli, and M. Monaci. Bilinear packing by bilinear programming. In Michael Jünger and Volker Kaibel, editors, *Integer Programming and Combinatorial Optimization, 11th International IPCO Conference, Berlin, Germany, June 8-10, 2005*, volume 3509 of *Lecture Notes in Computer Science*, pages 377–391. Springer, June 8-10 2005.
5. J. Carlier, F. Clautiaux, and A. Moukrim. New reduction procedures and lower bounds for the two-dimensional bin-packing problem with fixed orientation. *Computers and Operations Research*, to appear, 2005.
6. M. Dell'Amico, M. Martello, and D. Vigo. A lower bound for the non-oriented two-dimensional bin packing problems. *Discrete Applied Mathematics*, 118:13–24, 2002.
7. S. Fekete and J. Schepers. A general framework for bounds for higher-dimensional orthogonal packing problems. *Mathematical Methods of Operations Research*, 60:311–329, 2004.
8. D. S. Johnson. Near optimal bin packing algorithms, 1973. Dissertation, Massachussetts Institute of Technology, Cambridge, Massachussetts.
9. S. Martello and D. Vigo. Exact solution of the two-dimensional finite bin packing problem. *Management Science*, 44:388–399, 1998.
10. G. Wäscher, H. Haussner, and H. Schumann. An improved typology of cutting and packing problems. *European Journal of Operational Research*, to appear, 2006.

	Problem		Opt		Avg		CPU
Class	$H \times W$	n	BM	CJE	BM	CJE	CJE
I	10×10	20	10	10	6.6	6.6	0.00
		40	8	9	12.7	12.8	0.00
		60	10	10	19.5	19.5	0.00
		80	9	10	26.9	27.0	0.00
		100	7	10	31.0	31.3	0.00
		Avg.	8.8	9.8	19.34	19.44	0.000
II	30×30	20	10	10	1.0	1.0	0.00
		40	10	10	1.9	1.9	0.00
		60	10	10	2.5	2.5	0.00
		80	10	10	3.1	3.1	0.00
		100	10	10	3.9	3.9	0.00
		Avg.	10.0	10.0	2.48	2.48	0.000
III	40×40	20	10	10	4.7	4.7	0.00
		40	6	7	9.0	9.1	0.00
		60	7	7	13.2	13.2	0.00
		80	7	8	18.1	18.2	0.01
		100	4	4	21.5	21.5	0.02
		Avg.	6.8	7.2	13.30	13.34	0.006
IV	100×100	20	10	10	1.0	1.0	0.00
		40	10	10	1.9	1.9	0.00
		60	8	8	2.3	2.3	0.00
		80	8	8	3.0	3.0	0.00
		100	9	9	3.7	3.7	0.01
		Avg.	9.0	9.0	2.38	2.38	0.002
V	100×100	20	10	10	5.9	5.9	0.00
		40	8	8	11.3	11.3	0.00
		60	4	6	16.9	17.1	0.02
		80	6	6	23.5	23.6	0.04
		100	3	3	27.2	27.2	0.09
		Avg.	6.2	6.6	16.96	17.02	0.030
VI	300×300	20	10	10	1.0	1.0	0.00
		40	8	8	1.5	1.5	0.00
		60	10	10	2.1	2.1	0.00
		80	10	10	3.0	3.0	0.03
		100	8	8	3.2	3.2	0.04
		Avg.	9.2	9.2	2.16	2.16	0.014
VII	100×100	20	5	5	4.7	4.7	0.00
		40	2	4	9.7	9.9	0.01
		60	1	1	14.0	14.0	0.03
		80	0	0	19.8	20.0	0.06
		100	0	0	23.9	23.9	0.11
		Avg.	1.6	2.0	14.42	14.50	0.042
VIII	100×100	20	6	7	4.9	5.0	0.00
		40	1	2	9.6	9.7	0.01
		60	0	0	14.1	14.2	0.03
		80	0	0	19.7	19.7	0.06
		100	0	0	24.2	24.2	0.11
		Avg.	1.4	1.8	14.50	14.56	0.042
IX	100×100	20	10	10	14.3	14.3	0.00
		40	10	10	27.5	27.5	0.00
		60	10	10	43.5	43.5	0.01
		80	10	10	57.3	57.3	0.02
		100	10	10	69.3	69.3	0.03
		Avg.	10.0	10.0	42.38	42.38	0.012
X	100×100	20	8	8	3.9	3.9	0.00
		40	6	7	6.9	7.0	0.00
		60	4	5	9.4	9.5	0.02
		80	4	4	12.2	12.2	0.04
		100	3	3	15.3	15.3	0.07
		Avg.	5.0	5.4	9.54	9.58	0.026

Table 1
Comparison with the bound of [3]