Packing problems with orthogonal rotations*

F.K. Miyazawa[1]** and Y. Wakabayashi[2]

[1] Instituto de Computação — Universidade Estadual de Campinas
Caixa Postal 6176 — 13084-971 — Campinas–SP — Brazil
fkm@ic.unicamp.br

[2] Instituto de Matemática e Estatística — Universidade de São Paulo
Rua do Matão, 1010 — 05508–090 — São Paulo–SP — Brazil
yw@ime.usp.br

Abstract. In this extended abstract, we present approximation algorithms for the following packing problems: the *strip packing problem*, the *two-dimensional bin packing problem*, the *three-dimensional strip packing problem*, and the *three-dimensional bin packing problem*. For all these problems, we consider orthogonal packings where $90°$ rotations are allowed. The algorithms we show for these problems have asymptotic performance bounds 1.613, 2.64, 2.76 and 4.89, respectively. We also present an algorithm for the *z-oriented three-dimensional packing problem* with asymptotic performance bound 2.64. To our knowledge the bounds presented here are the best known for each problem.

1 Introduction

We present approximation algorithms for packing problems allowing orthogonal rotations. We consider the following problems: the *strip packing problem*, the *two-dimensional bin packing problem*, the *three-dimensional strip packing problem*, and the *three-dimensional bin packing problem*. The packings must be orthogonal and $90°$ rotations are allowed. The algorithms we show for these problems have asymptotic performance bounds 1.613, 2.64, 2.76 and 4.89, respectively. We also present an algorithm for the *z-oriented three-dimensional packing problem* with asymptotic performance bound 2.64. This result improves our previous results in [10, 11].

Approximation algorithms for the oriented version of these packing problems have been extensively considered, but there are very few results when rotations are considered. For a survey on approximation algorithms for packing problems, see [3, 4].

In all problems considered in this paper, all items must be packed into recipients and they may not overlap. We consider orthogonal packings where orthogonal rotations are allowed, i.e. $90°$ rotations, around any axis.

There are many applications in which orthogonal rotations are allowed: cutting of hardboard, glass, cloth (when there is no oriented pattern), foam, etc. [3]. Interesting applications also occur in job scheduling problems [9, 11].

---

* This work has been partially supported by MCT/CNPq – Project ProNEx (Proc. 664107/97-4), and CNPq (Proc. 300301/98-7, 304527/89–0, 464114/00–4, 470608/01–3, 478818/03–3).

** Corresponding author.

The *strip packing problem*, SPP, is the following: given a list of rectangles $L = (r_1, \ldots, r_n)$, where $r_i = (x_i, y_i)$, and a rectangle $R = (a, \infty)$, find a packing of the rectangles of L into R minimizing the size of the packing in the unlimited direction of R.

In the *two-dimensional bin packing problem*, 2BP, we are given a list of rectangles $L = (r_1, \ldots, r_n)$, where $r_i = (x_i, y_i)$, and two-dimensional bins $R = (a, b)$, and we wish to pack the rectangles of L into bins R using the smallest number of bins.

In the *three-dimensional strip packing problem*, TPP, we are given a list of boxes $L = (b_1, \ldots, b_n)$, where $b_i = (x_i, y_i, z_i)$, and a box $B = (a, b, \infty)$, pack the boxes of L into B, minimizing the size of the packing in the unlimited direction of the recipient.

The *three-dimensional bin packing problem*, 3BP, is defined as follows: given a list of boxes $L = (b_1, \ldots, b_n)$, where $b_i = (x_i, y_i, z_i)$, and three-dimensional bins $B = (a, b, c)$, find a packing of the boxes of L into the smallest number of bins B.

We also consider a special version of the three-dimensional strip packing problem, called *z-oriented three-dimensional strip packing problem*, TPP^z, which is the same as in the definition of TPP, except that boxes are oriented in the z-axis. That is, a box can be rotated around the z-axis (height direction), but cannot be laid down.

In section 2, we present some notation and discuss some results when orthogonal rotations are allowed. In section 3, we present the approximation algorithm for the strip packing problem and in the section 4, the approximation algorithm for the two-dimensional bin packing problem. In section 5, we present the results we have obtained for the other packing problems. In section 6, we present some concluding remarks.

2 Preliminaries

To define the packings, we consider the Euclidean space $\mathbb{R}^3$, with the xyz coordinate system. An item e to be packed has its dimensions defined as $x(e)$, $y(e)$ and $z(e)$, also denoted as the length, width and height of item e, where each of these dimensions is the measure in the corresponding axis of the xyz system. For the one- and the two-dimensional cases, some of these values are not defined.

We denote by $\mathrm{SPP}(a)$, $\mathrm{2BP}(a, b)$, $\mathrm{TPP}(a, b)$, $\mathrm{TPP}^z(a, b)$ and $\mathrm{3BP}(a, b, c)$ the corresponding problems versions with the recipient sizes defined by values a, b and c. We denote by $S(r_i)$ the area of the rectangle $r_i = (x_i, y_i)$ and $V(b_i)$ the volume of the box $b_i = (x_i, y_i, z_i)$. Given a function $f : C \to \mathbb{R}$ and a subset $C' \subseteq C$, we denote by $f(C')$ the sum $\sum_{e \in C'} f(e)$. For all algorithms, we consider that the items are given in an initial configuration that can be packed. That is, given a box e we have $x(e) \leq a$, $y(e) \leq b$ and $z(e) \leq c$. Given an item $e = (a, b)$, we denote by $\rho(e) := (b, a)$. We also consider that each item dimension is not greater than a constant Z.

The following is a convenient notation to define and restrict the input list of items: $\mathcal{X}[p, q] := \{e : \ p \cdot a < x(e) \leq q \cdot a\}$, $\mathcal{Y}[p, q] := \{e : \ p \cdot b < y(e) \leq q \cdot b\}$, $\mathcal{C}\,[p_1, q_1\,;\, p_2, q_2] := \mathcal{X}[p_1, q_1] \cap \mathcal{Y}[p_2, q_2]$, $\mathcal{C}_m := \mathcal{C}\left[0, \frac{1}{m}\,;\, 0, \frac{1}{m}\right]$, $\wp_1 := \mathcal{C}\left[0, \frac{1}{2}\,;\, 0, \frac{1}{2}\right]$, $\wp_2 := \mathcal{C}\left[0, \frac{1}{2}\,;\, \frac{1}{2}, 1\right]$, $\wp_3 := \mathcal{C}\left[\frac{1}{2}, 1\,;\, 0, \frac{1}{2}\right]$, $\wp_4 := \mathcal{C}\left[\frac{1}{2}, 1\,;\, \frac{1}{2}, 1\right]$.

Given a list L of items to be packed, and an algorithm $\mathcal{A}$, we denote by $\mathcal{A}(L)$ the size of the packing generated by algorithm $\mathcal{A}$ when applied to list L. Such size can be the height of the packing or the number of bins used in the packing, depending on which version of packing problem we are considering. For a packing $\mathcal{P}$, we denote by $H(\mathcal{P})$ its height, and by $\#(\mathcal{P})$ the number of bins that is used by $\mathcal{P}$. We denote by $\mathrm{OPT}(L)$ the size of an optimum packing of L. We say that an algorithm $\mathcal{A}$ has *asymptotic performance bound* α if there exists a constant β such that $\mathcal{A}(L) \leq \alpha \cdot \mathrm{OPT}(L) + \beta$, for all input list L.

Since the one-dimensional bin-packing problem is a particular case of all problems considered in this paper, it follows that each problem considered here is $\mathcal{NP}$-hard.

One idea to solve problems allowing orthogonal rotations is to simply apply the algorithms developed for the oriented case, ignoring any possible rotation. It can be shown that there is no algorithm, developed in the way we described above, for the strip packing problem with asymptotic performance bound less than 2, and for the two-dimensional bin packing problem and the three-dimensional bin packing problem there is no known algorithm with asymptotic performance bound less than 3 (see [11]).

Most of the results concerning approximation results do not consider rotations and some questions where posed in the early 80's. The following quote where extracted from a paper of F.R. Chung, M.R. Garey and D.S. Johnson [2] on the two-dimensional bin packing problem:

> "*A second line of attack would be to design and analyze algorithms which could make use of the fact that, in some applications,* $90°$ *rotations of rectangles might be allowable.*
> *Algorithms which consider the possibility of rotations might well yield improvements. Can one prove worst case bounds that reflect these improvements ?*"

There are other papers in the literature that raise questions about orthogonal rotations, as for example, [3, 5]. Although these papers are from the early 80's, very few has been done about orthogonal rotations. In fact, when the scale does not affect the problem, we can show that for any of the general packing problems considered, the version allowing orthogonal rotations is as hard to approximate as the oriented version.

Theorem 1. *Let* PROB^r *be one of the problems defined previously, considering orthogonal rotations around some of the axes* x *or* y *or* z *(maybe in several axes),* α *and* β *constants and* $\mathcal{A}^r$ *an algorithm such that,* $\mathcal{A}^r(L) \leq \alpha \cdot \mathrm{OPT}(L) + \beta$ *for any instance* L *of* PROB^r*. Then, we can adapt this algorithm to another algorithm* $\mathcal{A}$ *for a variant of* PROB^r*, called* PROB*, where we fix the orientation of the items (with respect to some axis), in such a way that the following relation holds:* $\mathcal{A}(L) \leq \alpha \cdot \mathrm{OPT}(L) + \beta$ *for any instance* L *of* PROB.

3 Strip Packing Problem

In this section we consider the strip packing problem. In [5], Coffman, Garey, Johnson and Tarjan present the algorithms NFDH (Next Fit Decreasing Height) and FFDH (First Fit Decreasing Height) for the oriented case and prove that their asymptotic performance bounds are 2 and 1.7, respectively. Another algorithm with asymptotic performance bound 2 is the BLDW, Bottom Leftmost Decreasing Width, algorithm presented by Baker, Coffman and Rivest [1]. Kenyon and Remila [8], presented an asymptotic approximation scheme for the oriented strip packing problem.

When orthogonal rotations are allowed, the bound 2 of the algorithms BLDW and NFDH are also valid, since the proofs of the bounds are based only in area arguments.

The algorithm we present in this section is called ASP. It uses the critical set combination strategy presented in [10, 11]. The idea is to combine items that do not lead to packings with good space filling, if considered independently. We call these items *critical items*. We take two sets with critical items, called here *critical sets*, and generate a combined packing of the items in these two sets. Each algorithm that combines items of two critical sets has the property that at the end of this combination, one of these sets is totally packed in the combined packing. Moreover, the combined packing has a better configuration than the one we can obtain for each critical set.

Before presenting the algorithm, we describe an algorithm used as subroutine called TC (Two Column) which builds a packing with two columns, each one is a stack of rectangles packed one on top of the other. Each column is associated with only one critical set. The algorithm TC is called with parameters (L_1, L_2, x_1, x_2), where L_1 and L_2 are two critical sublists and x_1 and x_2 are positions where the columns are built aligned to the left, from the bottom of the strip. We call *height of the column* the sum of the rectangles heights in the corresponding column. To pack a rectangle, the algorithm chooses the first column with smallest height. Let h be the height of this column and L_i the list associated with this column. If not all rectangles of L_i have been packed, then the next rectangle of L_i, say r, is packed in the position (x_i, h). Then, the list L_i is updated (by removing the rectangle r). This process is repeated until one of the lists, L_1 or L_2 is totally packed. We assume the positions x_1 and x_2 and the lists L_1 and L_2 are such that they do not generate infeasible packings. Any algorithm that combines critical sets returns a pair $(\mathcal{P}', L')$, where $\mathcal{P}'$ is the packing generated and L' is the set of rectangles packed in $\mathcal{P}'$. The following lemma can be proved for the algorithm TC.

Lemma 1. *Let $\mathcal{P}$ be a packing of $L' \subseteq L_1 \cup L_2$ generated by the algorithm TC when applied to lists L_1 and L_2 for $\mathrm{SPP}(a)$. If $x(r) > l_i \cdot a$ for all $r \in L_i$, $i = 1, 2$, then $H(\mathcal{P}) \leq \frac{1}{l_1+l_2} \frac{S(L')}{a} + Z$.*

We denote the value s in inequalities of the form $H(\mathcal{P}) \leq \frac{1}{s} \frac{S(L)}{a} + Z$ as an *area guarantee* of the packing $\mathcal{P}$. The idea used in the algorithm ASP is to generate the final packing consisting of two parts: one associated with a partial optimum packing generated with rectangles with width greater than $\frac{a}{2}$, and the other associated with a packing with better area guarantee.

Note that in an oriented packing, if L_1 has only rectangles with width greater than $\frac{a}{2}$ then we can obtain an optimum packing only by placing one rectangle on top of the other. When rotations are allowed, we can also have a similar result. In this case, we first reorient and remove all rectangles of L_1 with width greater than $\frac{a}{2}$ but height at most $\frac{a}{2}$. In this way, the remaining rectangles are the ones that cannot be reoriented, or if they can, they continue with width greater than $\frac{a}{2}$. Clearly, the only way to pack these remaining rectangles is to pack one on top of the other. After this rotation step, we introduce another reorientation step for the new rectangles in L_1 in such a way that each rectangle stays with the lowest possible height, maintaining a feasible orientation. Thus, we can continue having a partial optimum packing only by packing one rectangle on top of the other.

The algorithm TC is used to pack the critical rectangles of L_1 and critical rectangles of the remaining part. The packing of the remaining non-packed rectangles is done using NFDH strategy, which consists in first sorting the rectangles in non-decreasing order of their height and then packing the rectangles in this order, placing rectangles side by side generating levels. When a rectangle cannot be packed in a level it is packed in a new consecutive parallel level (for more details of the algorithm NFDH, see [5]). The following result is valid for NFDH.

Lemma 2. *Let $N_1, \dots, N_v$ be the levels generated by NFDH for a list L, in the order they are generated. If $w(N_i)$ is the total sum of the width of the rectangles in N_i, and there exists a constant s such that $w(N_i) \geq s \cdot a$, for $1 \leq i \leq v-1$, then we have* $\mathrm{NFDH}(L) \leq \frac{1}{s}\frac{S(L)}{a} + Z$.

Algorithm ASP(L)

Input: List of rectangles L for $\mathrm{SPP}(a)$

1 Rotate all rectangles $r \in L$ with $x(r) > \frac{a}{2}$ and $y(r) \leq \frac{a}{2}$.
2 Rotate all rectangles $r \in L$ with $x(r) > \frac{a}{2}$ and $x(r) < y(r) \leq a$.
3 Let $p \leftarrow 1/\sqrt{6}$ and $L'_i \leftarrow \{s \in L : \frac{a}{i+1} < x(s) \leq \frac{a}{i}\}, \quad i = 1, \dots, 4,$
$L'_5 \leftarrow \{s \in L : (1-2p) \cdot a < x(s) \leq \frac{a}{5}\},\ L'_6 \leftarrow \{s \in L : x(s) \leq (1-2p) \cdot a\},$
$L_A \leftarrow \{r \in L'_1 : x(r) \leq (1-p) \cdot a\}, \qquad L_B \leftarrow \{r \in L'_2 \cup \dots \cup L'_5 : x(r) \leq p \cdot a\}.$
4 $(\mathcal{P}_{AB}, L_{AB}) \leftarrow \mathrm{TC}(L_A, L_B, 0, 1-p);\ L_i \leftarrow L'_i \setminus L_{AB},$ for $i = 1, \dots, 6$;
5 $\mathcal{P}_i \leftarrow \mathrm{NFDH}(L_i), \quad i = 1, \dots, 6$;
6 $\mathcal{P}_{opt} \leftarrow \mathcal{P}_1 \| \mathcal{P}_{AB}; \quad \mathcal{P}_{aux} \leftarrow \mathcal{P}_2 \| \dots \| \mathcal{P}_6$.
7 Return $\mathcal{P}_{opt} \| \mathcal{P}_{aux}$.

Theorem 2. *For any input list L for the* $\mathrm{SPP}(a)$*, where the rectangles of L have dimensions at most Z, we have* $\mathrm{ASP}(L) \leq 1.613 \cdot \mathrm{OPT}(L) + 6Z$.

Proof. Since each rectangle of L_A has width at least $\frac{a}{2}$ and each rectangle of L_B has width at least $(1-2p)a$, from Lemma 1 we conclude that the following inequality holds.

$$H(\mathcal{P}_{AB}) \leq \frac{1}{1/2 + (1-2p)}\frac{S(L_{AB})}{a} + Z \leq \frac{1}{1-p}\frac{S(L_{AB})}{a} + Z. \tag{1}$$

Denote by L_{opt} the set of rectangles packed in $\mathcal{P}_{opt}$. It is easy to see that $\mathcal{P}_{opt}$ is an asymptotic optimum packing of L_{opt} since the *large* rectangles (with $x(r) > \frac{a}{2}$) of $L_1 \cup \dots \cup L_6 \cup L_{AB}$ either cannot be rotated, or if they can, they remain in the set defined for L_1. Moreover, the large rectangles of L_{opt} are packed with the smallest possible height. Therefore, we have

$$H(\mathcal{P}_{opt}) \leq \mathrm{OPT}(L) + Z. \tag{2}$$

Now, we analyze two cases. In the first (second), we consider that all rectangles of L_A (all rectangles of L_B) are totally packed in $\mathcal{P}_{AB}$.

Case 1. $L_A \subseteq L_{AB}$. In this case, we have all rectangles of $L_1 := L'_1 \setminus L_{AB}$ with width greater than $(1-p) \cdot a$. Using (1), we obtain $S(L_{opt}) \geq (H(\mathcal{P}_{opt}) - Z) \cdot (1-p) \cdot a$. That is,

$$H(\mathcal{P}_{opt}) \leq \frac{1}{1-p}\frac{S(L_{opt})}{a} + Z. \tag{3}$$

For each list $L_i \quad (i = 2, \ldots, 6)$ the area guarantee in each level of $\mathcal{P}_i$, except perhaps in the last, is at least $\frac{2}{3}$. This area guarantee is obtained by considering the width occupation in each level of the packing generated by the algorithm NFDH. Since the packing $\mathcal{P}_{aux}$ is the concatenation of packings $\mathcal{P}_2, \ldots, \mathcal{P}_6$, from Lemma 2 we have

$$H(\mathcal{P}_{aux}) \leq \frac{1}{2/3} \frac{S(L_{aux})}{a} + 5Z. \tag{4}$$

Defining $h_1 := H(\mathcal{P}_{opt}) - Z$ and $h_2 := H(\mathcal{P}_{aux}) - 5Z$, we have

$$\mathrm{OPT}(L) \geq \frac{S(L)}{a} = \frac{S(L_{opt})}{a} + \frac{S(L_{aux})}{a} \geq (1-p) \cdot h_1 + \frac{2}{3} \cdot h_2 \, . \tag{5}$$

From (2) and (5) we have $\mathrm{OPT}(L) \geq \max\{h_1, (1-p) \cdot h_1 + \frac{2}{3} \cdot h_2\}$, and therefore, we obtain $H(\mathcal{P}) \leq \alpha_1 \cdot \mathrm{OPT}(L) + 6Z$, where $\alpha_1 = \frac{h_1 + h_2}{\max\{h_1, (1-p) \cdot h_1 + \frac{2}{3} \cdot h_2\}} \leq \frac{2+3p}{2}$. This last inequality can be proved by analyzing the two possible values attained by the maximum.

Case 2. $L_B \subseteq L_{AB}$. The analysis of this case uses the same arguments used in case 1. Therefore, we present only the inequality one could obtain

$$H(\mathcal{P}) \leq \alpha_2 \cdot \mathrm{OPT}(L) + 6Z,$$

where $\alpha_2 = \frac{h_1 + h_2}{\max\{h_1, \frac{1}{2} h_1 + 2p h_2\}} \leq \frac{4p+1}{4p}$.

Substituting the value of p in the bounds obtained in the two cases above, we obtain $H(\mathcal{P}) \leq 1.613 \cdot \mathrm{OPT}(L) + 6Z$. □

4 Two-Dimensional Bin Packing Problem

In 1982, Chung et al. [2] presented an algorithm with asymptotic performance bound 2.125, which is the best bound known for the oriented two-dimensional bin packing problem.

In this section, we present an algorithm using orthogonal rotations, called $\mathrm{BI}_{k,\epsilon}$, with asymptotic performance bound that can be made not larger than 2.64. This algorithm follows the same technique used in the algorithm ASP. It also defines critical sets and combination of them, although in more steps and phases.

Before presenting the algorithm, we describe some algorithms used as subroutines by the main algorithm: the first algorithm is the algorithm NFDH^x, used for the packing of simple sublists. The algorithm uses two subroutines to combine critical sets: the algorithms COMBINE-AB$_k$ and COMBINE-CD. The algorithms FL_ϵ and the algorithm FFD, both for one-dimensional bin packing problem are used to generate packings of special sets that can be interpreted as one-dimensional packing problems.

The algorithm FL_ϵ was developed by Fernandez de la Vega and Lucker [6]. The algorithm presented by these authors is a linear time algorithm with additive constant $O(1/\epsilon)$. For our purposes, we consider a polynomial time version of it with additive constant 1, which can be found in [12]:

Theorem 3. *[6, 12] For any $\epsilon > 0$, there exists a polynomial time algorithm FL_ϵ for the bin-packing problem such that $\mathrm{FL}_\epsilon(L) \leq (1+\epsilon) \cdot \mathrm{OPT}(L) + 1$.*

The algorithm FFD first sorts the items of L in non-increasing order of their lengths, and then packs items in the order given by L. To pack an item, the algorithm FFD tries to pack the new item into one of the previous bins, considering the order they were generated. If it is not possible to pack in the previous bins, it packs in a new bin. Johnson [7] proved that the asymptotic performance bound of the algorithm FFD is $11/9$.

The algorithm NFDH^x *(Next Fit Decreasing Height)* first sorts the input list L in non-decreasing order of height, then packs the rectangles side by side (first horizontally and then vertically) generating levels. When a rectangle cannot be packed in the current level, a new level is generated parallel to the last one, which becomes the current level. When a level cannot be generated in the current bin, it is generated at the bottom of a new bin, which becomes the current bin.

The variant of this algorithm that generates levels in the y-direction is denoted by NFDH^y. Subdividing a list $L \subseteq \mathcal{C}_m$ into more sublists, and applying an appropriate variant of the algorithm NFDH^x, we can obtain an algorithm, denoted by BI_m, for which the following lemma holds.

Lemma 3. *For any list of rectangles $L = (r_1, \ldots, r_n)$, where $x(r_i) \leq \frac{1}{m}$ and $y(r_i) \leq \frac{1}{m}$, we have $\mathrm{BI}_m(L) \leq \left(\frac{m+1}{m}\right)^2 S(L) + 6$.*

The following lemma gives an upper bound for the number of bins used in a packing that has a minimum area occupation in each bin.

Lemma 4. *If $\mathcal{P}$ is a packing of a list of rectangles L such that all bins (a, b), except perhaps k of them, have an area occupation of at least f, then $\#(\mathcal{P}) \leq \frac{1}{f}\frac{S(L)}{ab} + k$.*

The critical sets used by the algorithm $\mathrm{COMBINE\text{-}AB}_k$ are defined by the numbers $r_i^{(k)}$ and $s_i^{(k)}$, presented in [10, 11], defined as follows.

Definition 1. *Let $r_1^{(k)}, r_2^{(k)}, \ldots, r_{k+15}^{(k)}$ and $s_1^{(k)}, s_2^{(k)}, \ldots, s_{k+14}^{(k)}$ be real numbers such that $r_1^{(k)} < \frac{4}{9}$; $r_1^{(k)}\frac{1}{2} = r_2^{(k)}(1 - r_1^{(k)}) = r_3^{(k)}(1 - r_2^{(k)}) = \ldots = r_k^{(k)}(1 - r_{k-1}^{(k)}) = \frac{1}{3}(1 - r_k^{(k)})$; $r_{k+1}^{(k)} = \frac{1}{3}$, $r_{k+2}^{(k)} = \frac{1}{4}$, $\ldots$, $r_{k+15}^{(k)} = \frac{1}{17}$; $s_i^{(k)} = 1 - r_i^{(k)}$ for $i = 1, \ldots, k$; and $s_{k+i}^{(k)} = 1 - \left(\frac{2i+4-\lfloor\frac{i+2}{3}\rfloor}{4i+10}\right)$ for $i = 1, \ldots, 14$.*

For simplicity we omit the superscripts $^{(k)}$ of the notation $r_i^{(k)}, s_i^{(k)}$ when k is clear from the context. Using a continuity argument, we can prove that the numbers $r_1, r_2, \ldots, r_k$ are such that $r_1 > r_2 > \cdots > r_k > \frac{1}{3}$ and $r_1 \to \frac{4}{9}$ as $k \to \infty$. Now, we can define the following critical sets.

$$\mathcal{A}_i = \mathcal{C}\left[r_{i+1}, r_i\,;\, \tfrac{1}{2}, s_i\right],\;\; \mathcal{B}_i = \mathcal{C}\left[\tfrac{1}{2}, s_i\,;\, r_{i+1}, r_i\right],\;\; \mathcal{A} = \textstyle\bigcup_{i=1}^{k+14} \mathcal{A}_i,\;\; \mathcal{B} = \textstyle\bigcup_{i=1}^{k+14} \mathcal{B}_i.$$

Lemma 5. *Given a list L of rectangles for the $2\mathrm{BP}(a, b)$, the algorithm COMBINE-AB packs all rectangles of type $\mathcal{A}$ or all rectangles of type $\mathcal{B}$. Moreover, if $\mathcal{P}_{AB}$ is the packing generated and L_{AB} the rectangles in $\mathcal{P}_{AB}$, then $\#(\mathcal{P}_{AB}) \leq \frac{36}{17}\frac{S(L_{AB})}{ab} + 2k + 41$.*

After applying the algorithm COMBINE-AB, suppose that all rectangles of type $\mathcal{B}$ have been packed in $\mathcal{P}_{AB}$. Consider the lists $L_1, \ldots, L_{23}$, defined in step 4.3 of the algorithm $\mathrm{BI}_{k,\epsilon}$. The packing of lists L_1 and L_{18}, generated by the algorithm NFDH, has an area guarantee close to $\frac{4}{9}$, but for sublists $L_2, \ldots, L_{17}, L_{19}, \ldots, L_{23}$ the NFDH strategy generates packings with volume guarantee better than $\frac{4}{9}$ (namely at least $17/36$). Therefore, we define the critical set $L_D := L'_D \cup L''_D$ (see step 4.3) as the set of rectangles of $L_1 \cup L_{18}$ which leads to packings with area guarantee close to 4/9. The area guarantee we can obtain for rectangles in $L \cap (\wp_2 \cup \wp_4) \setminus L_{AB}$ is $\frac{1}{4}$. Therefore, we define the critical set L_C as the critical rectangles in $L \cap \wp_4$ (see step 4.3). Let us denote the algorithm that combines the sets L_C and L_D by COMBINE-CD. The packing that is generated has bins with one rectangle of L_C and one rectangle of L'_D, or one rectangle of L_C and two rectangles of L''_D. If $\mathcal{P}_{CD}$ is a packing generated by COMBINE-CD, and L_{CD} is the set of rectangles packed in $\mathcal{P}_{CD}$, then or $L_C \subseteq L_{CD}$ or $L_D \subseteq L_{CD}$. Moreover, the packing $\mathcal{P}_{CD}$ has an area guarantee close to $17/36$. More precisely,

Lemma 6. *If $\mathcal{P}_{CD}$ is a packing generated by the algorithm COMBINE-CD and L_{CD} is the set of rectangles in $\mathcal{P}_{CD}$, then $\#(\mathcal{P}_{CD}) \leq \frac{1}{1/4 + r_1/2} \frac{S(L_{CD})}{ab} + 2$.*

The packings $\mathcal{P}_{AB}$ and $\mathcal{P}_{CD}$ have area guarantee close to $\frac{17}{36}$. Depending on which set is totally packed in $\mathcal{P}_{CD}$, we can improve either the area guarantee of $\frac{1}{4}$, of $L \cap (\wp_2 \cup \wp_4) \setminus (L_{AB} \cup L_{CD})$ packing, or the area guarantee of $\frac{4}{9}$, of the $L \cap (\wp_1 \cup \wp_3) \setminus (L_{AB} \cup L_{CD})$ packing. Now, it is possible to obtain an algorithm for 2BP with asymptotic performance bound close to 2.64. For simplicity, we denote by Update(L) the procedure that removes from L all rectangles previously packed.

Algorithm $\mathrm{BI}_{k,\epsilon}(L)$

Input: List of rectangles L for the $2\mathrm{BP}(a,b)$

1 Rotate all rectangles $r \in L \cap \wp_4$, where $\rho(r) \in \wp_1 \cup \wp_2 \cup \wp_3$.
2 $t \leftarrow 0.4574271$.
3 $(\mathcal{P}_{AB}, L_{AB}) \leftarrow$ COMBINE-AB$_k(L)$. Update(L).
4 If all rectangles of type $\mathcal{B}$ were packed then
4.1 Rotate each rectangle of $L \cap \wp_2$ that fits in $\wp_1 \cup \wp_3$.
4.2 Rotate each rectangle of $L \cap (\wp_2 \cup \wp_4)$, so that if $b \in L \cap (\wp_2 \cup \wp_4)$ then $x(b) \leq y(b)$ or $\rho(b) \notin \mathcal{C}_1$.
4.3 Subdivide the list L into sublists $L_1, \ldots, L_{23}$ as follows.

$$L_i \leftarrow L \bigcap \mathcal{C}\left[\tfrac{1}{2}, 1\,;\, \tfrac{1}{i+2}, \tfrac{1}{i+1}\right], \text{for } i = 1, \ldots, 16, \qquad L_{17} \leftarrow L \bigcap \mathcal{C}\left[\tfrac{1}{2}, 1\,;\, 0, \tfrac{1}{18}\right],$$
$$L_{18} \leftarrow L \bigcap \mathcal{C}\left[\tfrac{1}{3}, \tfrac{1}{2}\,;\, \tfrac{1}{3}, \tfrac{1}{2}\right], \quad L_{19} \leftarrow L \bigcap \mathcal{C}\left[\tfrac{1}{3}, \tfrac{1}{2}\,;\, \tfrac{1}{4}, \tfrac{1}{3}\right], L_{20} \leftarrow L \bigcap \mathcal{C}\left[\tfrac{1}{3}, \tfrac{1}{2}\,;\, 0, \tfrac{1}{4}\right],$$
$$L_{21} \leftarrow L \bigcap \mathcal{C}\left[\tfrac{1}{4}, \tfrac{1}{3}\,;\, \tfrac{1}{3}, \tfrac{1}{2}\right], \quad L_{22} \leftarrow L \bigcap \mathcal{C}\left[0, \tfrac{1}{4}\,;\, \tfrac{1}{3}, \tfrac{1}{2}\right], L_{23} \leftarrow L \bigcap \mathcal{C}\left[0, \tfrac{1}{3}\,;\, 0, \tfrac{1}{3}\right],$$
$$L_C \leftarrow L \bigcap \mathcal{C}\left[\tfrac{1}{2}, 1\,;\, \tfrac{1}{2}, 1-t\right], L'_D \leftarrow L_1 \bigcap \mathcal{C}\,[0, t\,;\, 0, 1], \quad L''_D \leftarrow L_{18} \bigcap \mathcal{C}\,[0, t\,;\, 0, 1],$$
$$L_D \leftarrow L'_D \bigcup L''_D.$$

4.4 Generate packing $\mathcal{P}_{CD}$ as follows.
$(\mathcal{P}_{CD}, L_{CD}) \leftarrow$ COMBINE-CD(L); $\quad L_1 \leftarrow L_1 \setminus L_{CD}$; $L_{18} \leftarrow L_{18} \setminus L_{CD}$.
4.5 Generate packings $\mathcal{P}_1, \ldots, \mathcal{P}_{23}$ as follows.
$\mathcal{P}_i \leftarrow \mathrm{NFDH}^y(L_i)$ for $i = 1, \ldots, 21$; $\quad \mathcal{P}_i \leftarrow \mathrm{NFDH}^x(L_i)$ for $i = 22$;
$\mathcal{P}_{23} \leftarrow \mathrm{BI}_3(L_{23})$. $\quad$ Update(L). Now, we have $L \subseteq \wp_2 \cup \wp_4$.
4.6 Consider each rectangle of L as a one-dimensional item of length $x(r)$ and each two-dimensional bin as a one-dimensional bin of length a.
Let $\mathcal{P}_{\mathrm{FL}_\epsilon}$ be the packing obtained by $\mathrm{FL}_\epsilon(L)$ and let $\mathcal{P}_{\mathrm{FFD}}$ be the packing

$\mathrm{FFD}(L \cap \mathcal{X}[0, \frac{1}{3}]) \| \mathrm{FFD}(L \cap \mathcal{X}[\frac{1}{3}, \frac{1}{2}]) \| \mathrm{FFD}(L \cap \mathcal{X}[\frac{1}{2}, 1])$.
Let $\mathcal{P}_{\mathrm{UNI}}$ be the smallest packing in $\{\mathcal{P}_{\mathrm{FL}_\epsilon}, \mathcal{P}_{\mathrm{FFD}}\}$.

4.7 $\mathcal{P}_{aux} \leftarrow \mathcal{P}_{AB} \| \mathcal{P}_{CD} \| \mathcal{P}_1 \| \ldots \| \mathcal{P}_{23}$;

4.8 $\mathcal{P} \leftarrow \mathcal{P}_{\mathrm{UNI}} \| \mathcal{P}_{aux}$.

5 If all rectangles of type $\mathcal{A}$ were packed, then generate a packing $\mathcal{P}$ of L as in step 4 (in a symmetric way).

6 Return $\mathcal{P}$.

Theorem 4. *For any instance L of* 2BP*, we have* $\mathrm{BI}_{k,\epsilon}(L) \le \alpha_{k,\epsilon} \cdot \mathrm{OPT}(L) + \mathcal{O}(k)$*, where* $\alpha_{k,\epsilon} \to 2.63\ldots$ *as* $k \to \infty$ *and* $\epsilon \to 0$.

Proof. We present the proof for the case in which all rectangles of type $\mathcal{B}$ are packed in step 3. The proof for the other case (all rectangles of type $\mathcal{A}$ are packed) is analogous. This proof is divided in two cases, according to step 4.4 ($L_C \subseteq L_{CD}$ or $L_D \subseteq L_{CD}$).

Analyzing the configuration of each bin in the packing $\mathcal{P}_i, i \in \{1, \ldots, 23\} \setminus \{1, 18\}$, we can show that $\mathcal{P}_i$ have an area occupation of at least $\frac{17}{36} \cdot a \cdot b$ in each bin, except perhaps in a constant number of them. Therefore, applying lemmas 4, 3, 5 and 6 we have

$$\#(\mathcal{P}_i) \le \frac{36}{17} \frac{S(L_i)}{ab} + 1, \;\text{ for }\; i \in \{1, \ldots, 23\} \setminus \{1, 18\}. \tag{6}$$

$$\#(\mathcal{P}_{AB}) \le \frac{36}{17} \frac{S(L_{AB})}{ab} + (2k + 41), \quad\; \#(\mathcal{P}_{CD}) \le \frac{1}{\left(\frac{1}{4} + \frac{r_1}{2}\right)} \frac{S(L_{CD})}{ab} + 2. \tag{7}$$

Case 1. $L_C \subseteq L_{CD}$. For packings $\mathcal{P}_1$ and $\mathcal{P}_{18}$, we have

$$\#(\mathcal{P}_1) \le \frac{1}{r_1} \frac{S(L_1)}{ab} + 1, \quad\; \#(\mathcal{P}_{18}) \le \frac{1}{\frac{4}{9}} \frac{S(L_{18})}{ab} + 1. \tag{8}$$

By Theorem 3, and analyzing the area guarantee of $\mathcal{P}_{\mathrm{FFD}}$, we have

$$\#(\mathcal{P}_{\mathrm{UNI}}) \le \#(\mathcal{P}_{\mathrm{FL}_\epsilon}) \le (1 + \epsilon) \cdot \mathrm{OPT}(L_{\mathrm{UNI}}) + 1. \tag{9}$$

$$\#(\mathcal{P}_{\mathrm{UNI}}) \le \#(\mathcal{P}_{\mathrm{FFD}}) \le \frac{1}{(1 - t) \cdot \frac{1}{2}} \cdot \frac{S(L_{\mathrm{UNI}})}{ab} + 3. \tag{10}$$

Now, for the packing $\mathcal{P}_{aux} = \mathcal{P}_{AB} \| \mathcal{P}_1 \| \ldots \| \mathcal{P}_{23}$, using the inequalities (6)–(8) and the fact that $r_1 = \min\left\{\frac{17}{36}, r_1, \frac{1}{4} + \frac{r_1}{2}, \frac{4}{9}\right\}$, we obtain

$$\#(\mathcal{P}_{aux}) \le \tfrac{1}{r_1} \tfrac{S(L_{aux})}{ab} + (2k + 68),$$

where L_{aux} denotes the set of rectangles in $\mathcal{P}_{aux}$. Let $n_1 := \#(\mathcal{P}_{\mathrm{UNI}}) - 3$ and $n_2 := \#(\mathcal{P}_{aux}) - (2k + 68)$. From inequality (9) and the fact that $\mathrm{OPT} \ge \frac{S(L)}{ab}$, we have $\mathrm{OPT}(L) \ge \max\left\{\frac{1}{1+\epsilon} n_1, \frac{1-t}{2} n_1 + r_1 \cdot n_2\right\}$. Since $\#(\mathcal{P}) = \#(\mathcal{P}_{aux}) + \#(\mathcal{P}_{\mathrm{UNI}})$, now we have $\#(\mathcal{P}) = (n_2 + (2k + 68) + n_1 + 3) = n_1 + n_2 + (2k + 71)$. Therefore,

$$\mathrm{BI}_{k,\epsilon}(L) \le \alpha'_{k,\epsilon} \cdot \mathrm{OPT}(L) + (2k + 71),$$

where $\alpha'_{k,\epsilon} = (n_1 + n_2) / \max\left\{\frac{1}{1+\epsilon} n_1, \frac{(1-t)}{2} n_1 + r_1 \cdot n_2\right\} \le \frac{1}{r_1} - \frac{(1-t)}{2r_1} + (1 + \epsilon)$.

Case 2. $L_D \subseteq L_{CD}$. In this case, the proof is analogous and we can obtain that $\mathrm{BI}_{k,\epsilon}(L) \le \alpha''_{k,\epsilon} \cdot \mathrm{OPT}(L) + (2k + 71)$, where $\alpha''_{k,\epsilon} = \frac{(n_1 + n_2)}{\max\left\{\frac{1}{1+\epsilon} n_1, \frac{1}{4} \cdot n_1 + t \cdot n_2\right\}} \le \frac{1}{t} - \frac{(1+\epsilon)}{4t} + (1 + \epsilon)$.

From cases 1 and 2, we conclude that for $k \to \infty$ and $\epsilon \to 0$ the theorem follows. □

5 Three-Dimensional Packing Problems

Due to space limitations, we only present the results for the algorithms ATP, ATP^z and A3D we obtained by applying the same techniques to the problems TPP, TPP^z and 3BP, respectively.

Theorem 5. *For any instance L of* TPP*, we have* $\text{ATP}_{k,\epsilon}(L) \leq \alpha_{k,\epsilon} \cdot \text{OPT}(L) + \mathcal{O}\left(k + \frac{1}{\epsilon}\right) \cdot Z$*, where* $\alpha_{k,\epsilon} \to 2.76\ldots$ *as* $k \to \infty$ *and* $\epsilon \to 0$.

Theorem 6. *For any instance L of* TPP^z*, we have* $\text{ATP}^z_{k,\epsilon}(L) \leq \alpha_{k,\epsilon} \cdot \text{OPT}(L) + \mathcal{O}\left(k + \frac{1}{\epsilon}\right) \cdot Z$*, where* $\alpha_{k,\epsilon} \to 2.63\ldots$ *as* $k \to \infty$ *and* $\epsilon \to 0$.

This last result improves our previous result in [10, 11].

Theorem 7. *For any list of boxes L for* 3BP*, we have* $\text{A3D}(L) \leq \alpha_{k,\epsilon} \cdot \text{OPT}(L) + \beta_{k,\epsilon}$*, where* $\lim_{k\to\infty,\epsilon\to 0} \alpha_{k,\epsilon} \leq 4.88\ldots$*, and* $\beta_{k,\epsilon}$ *is constant for constant values of k and ϵ.*

6 Concluding Remarks

We presented several approximation algorithms for packing problems where orthogonal rotations are allowed. These problems have been less investigated in the literature. To our knowledge, the bounds presented are the best ones known for each problem. We would like to thank David S. Johnson, for his comments about the status of these problems.

References

1. B.S. Baker, E.G. Coffman Jr., and R.L. Rivest. Orthogonal packings in two-dimensions. *SIAM Journal on Computing*, 9:846–855, 1980.
2. F.R.K. Chung, M.R. Garey, and D.S. Johnson. On packing two-dimensional bins. *SIAM Journal on Algebraic and Discrete Methods*, 3:66–76, 1982.
3. E.G. Coffman, Jr., M.R. Garey, and D.S. Johnson. Approximation algorithms for bin packing - an updated survey. In G. Ausiello, M. Lucertini, and P. Serafini, editors, *Algorithms design for computer system design*, pages 49–106. Spring-Verlag, New York, 1984.
4. E.G. Coffman, Jr., M.R. Garey, and D.S. Johnson. *Approximation algorithms (ed. D. Hochbaum)*, chapter Approximation algorithms for bin packing - a survey. PWS, 1997.
5. E.G. Coffman, Jr., M.R. Garey, D.S. Johnson, and R.E. Tarjan. Performance bounds for level oriented two-dimensional packing algorithms. *SIAM J. on Computing*, 9:808–826, 1980.
6. W. Fernandez de la Vega and G.S. Lueker. Bin packing can be solved within $1 + \epsilon$ in linear time. *Combinatorica*, 1(4):349–355, 1981.
7. D.S. Johnson. *Near-optimal bin packing algorithms*. PhD thesis, Massachusetts Institute of Technology, Cambridge, Mass., 1973.
8. C. Kenyon and E. Remila. Approximate strip packing. In *37th Annual Symposium on Foundations of Computer Science*, pages 31–36, 1996.
9. K. Li and K-H. Cheng. Static job scheduling in partitionable mesh connected systems. *Journal of Parallel and Distributed Computing*, 10:152–159, 1990.
10. F.K. Miyazawa and Y. Wakabayashi. An algorithm for the three-dimensional packing problem with asymptotic performance analysis. *Algorithmica*, 18(1):122–144, 1997.
11. F.K. Miyazawa and Y. Wakabayashi. Approximation algorithms for the orthogonal z-oriented 3-D packing problem. *SIAM Journal on Computing*, 29(3):1008–1029, 2000.
12. V.V. Vazirani. *Approximation Algorithms*. Springer-Verlag, 2001.