INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

Two- and three-dimensional packings with orthogonal rotations

F. K. Miyazawa   *Y. Wakabayashi*

Technical Report - IC-04-007 - Relatório Técnico

July - 2004 - Julho

Two- and three-dimensional packings with orthogonal rotations*

F. K. Miyazawa[†] and Y. Wakabayashi[‡]

Abstract

We present approximation algorithms for the following packing problems: the *two-dimensional 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 ninety-degree 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 strip packing problem* with asymptotic performance bound 2.64. To our knowledge, these are the best bounds known for each problem.

1 Introduction

We focus on orthogonal packing problems where ninety-degree rotations are allowed. These problems have many real-world applications [5, 16]: job scheduling, container loading, cutting of hardboard, glass, foam, etc.

We present approximation algorithms for the 2-dimensional and the 3-dimensional versions of the strip packing and the bin packing problems. In the d-dimensional version of both problems, $d \geq 1$, the input consists of a list of d-dimensional items (not necessarily of equal sizes) and a d-dimensional bin B. In the *strip packing problem* (dSP), defined only for $d \geq 2$, one of the dimensions of the bin B, say height, is unlimited, and the goal is to pack the list of items into B so as to minimize the height of the packing. In the *bin packing problem* (dBP), the dimensions of the bin B are limited, and the goal is to pack the list of items into a minimum number of bins.

These problems and others of this nature have been more investigated in the version in which the packing is required to be *oriented*. In this version, the items and the bins are given with some orientation with respect to a coordinate system, and the items must be packed into the bins in this given orientation. In this paper, we consider packings that allow orthogonal rotations (that is, the items to be packed may be rotated by ninety degrees

*This research was partially supported by MCT/CNPq through PRONEX program (Proc. 664107/97-4) and CNPq (Proc. 470608/01–3, 468516/00–0, 464114/00–4, 478818/03–3, 300301/98–7 and 304527/89–0).

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

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

around any of the axes); to distinguish them we may refer to them as *r-packings* or *packings with rotations* (instead of saying non-oriented packing). In the 3-dimensional case, we also consider a restricted variant of r-packing, called here *z-oriented packing*. In this variant, ninety-degree rotations are allowed only around the z-axis.

To differentiate from the classical oriented versions, the problems to be considered here will be denoted $2SP^r$ (*2-dimensional strip packing with rotation*), $2BP^r$ (*2-dimensional bin packing with rotation*), $3SP^r$ (*3-dimensional strip packing with rotation*), and $3BP^r$ (*3-dimensional bin packing with rotation*). We also consider the *z-oriented 3-dimensional strip packing problem* ($3SP^z$), a variant of 3SP in which the packing has to be z-oriented, where z is the height direction.

We present approximation algorithms with asymptotic performance bounds 1.613, 2.64, 2.76, 4.89 and 2.64 for the problems $2SP^r$, $2BP^r$, $3SP^r$, $3BP^r$ and $3SP^z$, respectively. To our knowledge, these are the best bounds known for these problems.

Approximation algorithms for the oriented versions of these packing problems have been extensively considered. The most studied case is the 1-dimensional bin packing problem (1BP), for which the work of Johnson [12] in the early 1970s pioneered the approach of designing efficient approximation algorithms with worst-case performance guarantee for packing problems. Since 1BP is NP-hard and it is a particular case of all problems considered in this paper, it follows that each problem considered here is NP-hard. Moreover, 1BP cannot be approximated —in the absolute sense— within 3/2; thus, this negative result also holds for the problems considered here.

In what follows we only mention some previous results closely related to the problems we focus in this paper. For the problem 2SP, Kenyon and Rémila [13] obtained an asymptotic polynomial time approximation scheme (APTAS). For the problem 2BP, Chung, Garey and Johnson [4] proved that the algorithm HFF (Hybrid First Fit) has asymptotic performance bound 2.125. In 2001, Caprara [3] proved that this algorithm has asymptotic performance bound 2.077; and also presented an algorithm with asymptotic performance bound 1.691, the best bound known for the problem 2BP. Recently, Bansal and Sviridenko [2] proved that there is no APTAS for 2BP, unless P = NP. In the same paper, they showed an APTAS for a special version of 2BP in which the items and the bins are squares; this result was also obtained by Correa and Kenyon [8]. For the problem 3BP, Li and Cheng [14] and Csirik and van Vliet [9] designed algorithms with asymptotic performance bound 4.84. Their algorithms generalize to the problem dBP, giving an algorithm with asymptotic performance bound close to 1.691^d. For the problem 3SP, the best asymptotic performance bound previously known is 2.67, achieved by an algorithm we presented in [17].

When rotations are allowed, the bounds of some algorithms for the oriented versions may also hold; this happens when the proofs are based only on area arguments. Except for this type of results, very few approximation algorithms for r-packing problems have appeared in the literature.

In 2000, we presented in [19] an approximation algorithm for the problem $3SP^z$ with asymptotic performance bound 2.67. We also showed an algorithm with bound 2.53 for the special case of $3SP^z$ in which the bin has square bottom, and also for a more specialized version for packing boxes with square bottom. To our knowledge, [19] is the first paper to present approximation algorithms for r-packing problems where rotations are exploited in a non-trivial way.

As the problem $2\mathrm{BP}^{\mathrm{r}}$ can be seen as a particular case of $3\mathrm{SP}^{z}$, the algorithms presented in [19] also lead to algorithms with asymptotic performance bound 2.67 for $2\mathrm{BP}^{\mathrm{r}}$, and 2.53 for the special case of $2\mathrm{BP}^{\mathrm{r}}$ in which the bins are squares. Recently, Epstein [10] improved the bound for this special case of $2\mathrm{BP}^{\mathrm{r}}$, presenting an online algorithm with asymptotic performance bound 2.45.

Using the fact that there is no APTAS for 2BP (a result of Bansal and Sviridenko that we mentioned before), we may easily conclude that there is no APTAS for the problem 3SP, unless P = NP. From the latter statement, we may also conclude that there is no APTAS for the problem $3\mathrm{SP}^{z}$, unless P = NP, since the existence of such an APTAS for $3\mathrm{SP}^{z}$ would lead to an APTAS for 3SP (see comments on this in Section 2).

For a survey on approximation algorithms for packing problems, we refer the reader to [5, 6].

This paper is organized as follows. In Section 2, we define the problems, give some basic definitions and state some results. Each of the Sections 3 to 7 are devoted to each of these problems. In Section 8 we present some concluding remarks.

An extended abstract containing early versions of the results of this paper appeared in [20], without the description of all the algorithms and their analyses. The algorithm SPR we present in Section 3 is slightly different from its early version.

2 Preliminaries

In this section, we first define the problems we focus in this paper, then give some basic definitions, establish the notation, and mention some known results that we use in this paper. We also discuss some relations (reductions) between algorithms for the oriented version and the version with rotations.

The *two-dimensional strip packing problem with rotation*, $2\mathrm{SP}^{\mathrm{r}}$, is the following: given a list of rectangles $L = (r_1, \ldots, r_n)$, where $r_i = (x_i, y_i)$, and a bin $B = (a, \infty)$, find an r-packing of the rectangles of L into B that minimizes the size of the packing in the unlimited direction of B.

In the *two-dimensional bin packing problem with rotation*, $2\mathrm{BP}^{\mathrm{r}}$, we are given a list of rectangles $L = (r_1, \ldots, r_n)$, where $r_i = (x_i, y_i)$, and two-dimensional bins $B = (a, b)$, and we are asked to find an r-packing of the rectangles of L into a minimum number of bins B.

The *three-dimensional strip packing problem with rotation*, $3\mathrm{SP}^{\mathrm{r}}$, is defined as follows: given a list of boxes $L = (e_1, \ldots, e_n)$, where $e_i = (x_i, y_i, z_i)$, and a bin $B = (a, b, \infty)$, find an r-packing of the boxes of L into B, that minimizes the size of the packing in the unlimited direction of B.

In the *three-dimensional bin packing problem with rotation*, $3\mathrm{BP}^{\mathrm{r}}$, we are given a list of boxes $L = (e_1, \ldots, e_n)$, where $e_i = (x_i, y_i, z_i)$, and three-dimensional bins $B = (a, b, c)$, and we are asked to find an r-packing of the boxes of L into a minimum number of bins B.

We also consider a special version of $3\mathrm{SP}^{\mathrm{r}}$, called *$z$-oriented three-dimensional strip packing problem*, $3\mathrm{SP}^{z}$, defined analogously, except that, instead of an r-packing we require a z-oriented packing, where the z-axis is the unlimited direction of bin B.

We denote by $2\mathrm{SP}^{\mathrm{r}}(a)$, $2\mathrm{BP}^{\mathrm{r}}(a,b)$, $3\mathrm{SP}^{\mathrm{r}}(a,b)$, $3\mathrm{SP}^{z}(a,b)$ and $3\mathrm{BP}^{\mathrm{r}}(a,b,c)$ the corresponding problems versions with the bin sizes defined by values a, b and c.

2.1 Definitions and Notation

In all problems considered in this paper, the given list L of items (rectangles or boxes) must be packed orthogonally into bins B (strips, rectangles, boxes) in such a way that no two items overlap.

For all algorithms we assume that every item e in the input list L is given in a *feasible orientation*, that is, in an orientation that allows it to be packed into B without the need of any rotation (there is no loss of generality in assuming this, as the items can be rotated previously if needed). Moreover, we consider that the items have each of its dimensions not greater than a constant Z.

To refer to the packings we consider the Euclidean space $\mathbb{R}^3$, with the xyz coordinate system. An item e in L has its dimensions defined as $x(e)$, $y(e)$ and $z(e)$, also called its *length*, *width* and *height*, respectively. 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.

If e is a rectangle then we denote by $S(e)$ its *area*. If e is a box then the *bottom area* of e is the area of the rectangle $(x(e), y(e))$, and $V(e)$ denotes the *volume* of e. Given a function $f : C \to \mathbb{R}$ and a subset $C' \subseteq C$, we denote by $f(C')$ the sum $\sum_{e \in C'} f(e)$.

Although a list of items is given as an ordered n-tuple, when the order of the items is irrelevant we consider the corresponding list as a set. Therefore, if L is a list of items, we refer to the total area, respectively volume, of the items in L as $S(L)$, respectively $V(L)$.

If $L_1, L_2, \ldots, L_k$ are lists, where $L_i = (e_i^1, e_i^2, \ldots, e_i^{n_i})$, the *concatenation* of these lists, denoted by $L_1 \| L_2 \| \ldots \| L_k$, is the list $(e_1^1, \ldots, e_1^{n_1}, e_2^1, \ldots, e_2^{n_2}, \ldots, e_k^1, \ldots, e_k^{n_k})$.

The following is a convenient notation to define and restrict the input list of items.

$$
\begin{aligned}
\mathcal{X}[p,q] &:= \{e : \quad p \cdot a < x(e) \leq q \cdot a\}, \\
\mathcal{Y}[p,q] &:= \{e : \quad p \cdot b < y(e) \leq q \cdot b\}, \\
\mathcal{Z}[p,q] &:= \{e : \quad p \cdot c < z(e) \leq q \cdot c\},
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{C}^{xy}[p_1,q_1 \ ; \ p_2,q_2] &:= \mathcal{X}[p_1,q_1] \cap \mathcal{Y}[p_2,q_2], \\
\mathcal{C}^{yz}[p_1,q_1 \ ; \ p_2,q_2] &:= \mathcal{Y}[p_1,q_1] \cap \mathcal{Z}[p_2,q_2], \\
\mathcal{C}^{zx}[p_1,q_1 \ ; \ p_2,q_2] &:= \mathcal{Z}[p_1,q_1] \cap \mathcal{X}[p_2,q_2], \\
\mathcal{C}^{xyz}[p_1,q_1 \ ; \ p_2,q_2 \ ; \ p_3,q_3] &:= \mathcal{X}[p_1,q_1] \cap \mathcal{Y}[p_2,q_2] \cap \mathcal{Z}[p_3,q_3],
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{C}_m &:= \mathcal{C}^{xyz}\left[0,\frac{1}{m} \ ; \ 0,\frac{1}{m} \ ; \ 0,\frac{1}{m}\right], \\
\mathcal{C}_m^{xy} &:= \mathcal{C}^{xy}\left[0,\frac{1}{m} \ ; \ 0,\frac{1}{m}\right],
\end{aligned}
$$

$$
\wp_1 := \mathcal{C}^{xy}\left[0,\frac{1}{2} \ ; \ 0,\frac{1}{2}\right], \quad \wp_2 := \mathcal{C}^{xy}\left[0,\frac{1}{2} \ ; \ \frac{1}{2},1\right],
$$

$$\wp_3 := \mathcal{C}^{xy}\left[\frac{1}{2},1\ ;\ 0,\frac{1}{2}\right],\quad \wp_4 := \mathcal{C}^{xy}\left[\frac{1}{2},1\ ;\ \frac{1}{2},1\right],$$

$$\mathcal{C}^x_y[a,b] := \{e : \quad x(e) \geq y(e)\cdot\frac{a}{b}\},\quad \mathcal{C}^y_x[a,b] := \{e : \quad x(e) < y(e)\cdot\frac{a}{b}\}.$$

Given an item $e = (s_1, \ldots, s_t)$, we denote by $\rho_{xy}(e)$ the t-uple obtained from e by exchanging only the x and y coordinates. That is, if $e = (s_1, s_2, s_3)$ then $\rho_{xy}(e) = (s_2, s_1, s_3)$; and if $e = (s_1, s_2)$ then $\rho_{xy}(e) = (s_2, s_1)$. Analogously, the notation ρ_{xz} and ρ_{yz} are used to refer to the exchange of the coordinates x and z and the coordinates y and z, respectively. When the two axes are clear, or when we refer to any two axes, we use simply the notation ρ.

If $\mathcal{T}$ is a set of items, then we say that an item e in L *is of type* $\mathcal{T}$ if $e' \in \mathcal{T}$ for some permutation e' of e. For pairs $(L', \mathcal{T})$, where L' is a list and $\mathcal{T}$ is a set of items, we define the following functions:

$$\begin{aligned} \text{rtype}\,(L',\mathcal{T}) &:= \{e \in L' :\ \text{some permutation of } e \text{ belongs to } \mathcal{T}\}; \\ xy\text{-type}\,(L',\mathcal{T}) &:= \{e \in L' :\ e \in \mathcal{T} \text{ or } \rho_{xy}(e) \in \mathcal{T}\}. \end{aligned}$$

If $\mathcal{A}$ is an algorithm (for one of the packing problems), and L is a list of items to be packed, then $\mathcal{A}(L)$ denotes the size of the packing generated by algorithm $\mathcal{A}$ when applied to list L, and $\mathrm{OPT}(L)$ denotes the size of an optimal packing of L. The size can be the height of the packing or the number of bins used in the packing, depending on the problem we are considering. Although OPT will be used for distinct problems, its meaning will be clear from the context. We say that an algorithm $\mathcal{A}$ has *asymptotic performance bound* α if there exists a constant β such that

$$\mathcal{A}(L) \leq \alpha\,\mathrm{OPT}(L) + \beta, \qquad \text{for all input list } L.$$

If $\beta = 0$ than we say that algorithm $\mathcal{A}$ has *absolute performance bound* α.

2.2 Relations between algorithms for oriented packings and r-packings

One way to solve r-packing problems is to adapt algorithms developed for the oriented case. In [19] we mention that, for the problem $3\mathrm{SP}^z$, a simple algorithm that (rotates first all items so as to have them in a feasible orientation and) applies an algorithm for 3SP must have an asymptotic bound at least 3. It can be shown, using the same strategy, that no algorithm for $2\mathrm{SP}^\mathrm{r}$, designed as we described above, can have asymptotic performance bound smaller than 2. Similar results also hold for the problems $2\mathrm{BP}^\mathrm{r}$ and $3\mathrm{BP}^\mathrm{r}$: no algorithm with asymptotic performance bound smaller than 3 can be obtained as described above (for more details see [19]).

Most of the results concerning approximation results do not consider rotations. In the early 1980s, Coffman, Garey and Johnson [5] discussing the case where ninety-degree rotations are allowed, mentioned that *"no algorithm has been found (for the problem* 2BP*)* *that attains improved guarantees by actually using such rotations itself."* Chung, Garey and Johnson [4] also discussed this matter and raised the question about the possibility of obtaining algorithms with better worst-case bounds. For other papers that raise questions about rotations the reader may refer to [7, 13].

We can show that when scaling does not affect the problem, for any of the general packing problems considered, the version allowing rotations is as hard to approximate as the oriented version. More precisely, we can show the following result.

Theorem 2.1 *Let* $\mathrm{PROB}^{\mathrm{r}}$ *be one of the problems defined previously, for which orthogonal rotations around some of the axes* x *or* y *or* z *(possibly several axes) are allowed; and let* PROB *be a variant of* $\mathrm{PROB}^{\mathrm{r}}$*, obtained by fixing the orientation of the packing with respect to some axis. Let* α *and* β *be constants and* $\mathcal{A}^r$ *an algorithm for* $\mathrm{PROB}^{\mathrm{r}}$ *such that* $\mathcal{A}^r(L) \leq \alpha\, \mathrm{OPT}(L) + \beta$ *for any input list* L *of* $\mathrm{PROB}^{\mathrm{r}}$*. Then, there is an algorithm* $\mathcal{A}$ *for* PROB *such that the following holds:*

$$\mathcal{A}(L) \leq \alpha\, \mathrm{OPT}'(L) + \beta \quad \textit{for any input list } L \textit{ of } \mathrm{PROB}.$$

Moreover, the reduction is polynomial, if we consider a convenient representation for the instance.

We omit the proof of this theorem, and refer the reader to [19] for a proof for the problem $3\mathrm{SP}^z$.

2.3 One-Dimensional Bin Packing Problem

Many algorithms we shall describe use one-dimensional bin packing problem algorithms as subroutines. This section is devoted to these algorithms and related results (see Coffman, Garey and Johnson [6]).

The one-dimensional bin packing problem, 1BP, consists in packing a list L of one-dimensional items into a minimum number of one-dimensional bins B, of length a. Many algorithms have been designed for this problem. In what follows we describe the following: NF (Next Fit), FF (First Fit) and FFD (First Fit Decreasing).

The algorithm NF can be described as follows. Given a list of items L, it packs the items in the order given by L. The first item is packed into a bin which becomes the current bin; then as long as there are items to be packed, the next item is tested. If possible, it is packed into the current bin; if it does not fit in the current bin, then it is packed into a new bin, which becomes the current bin.

The algorithm FF also packs the items in the order given by L. It tries to pack each new item into one of the previous bins, considering the order they were generated. If it is not possible to pack an item in any of the previous bins, the algorithm packs it into a new bin.

The algorithm FFD first sorts the items of L in decreasing order of their length, and then applies the algorithm FF.

We also use the asymptotic polynomial time scheme designed by Fernandez de la Vega and Lucker [11], which we denote by FL_ϵ.

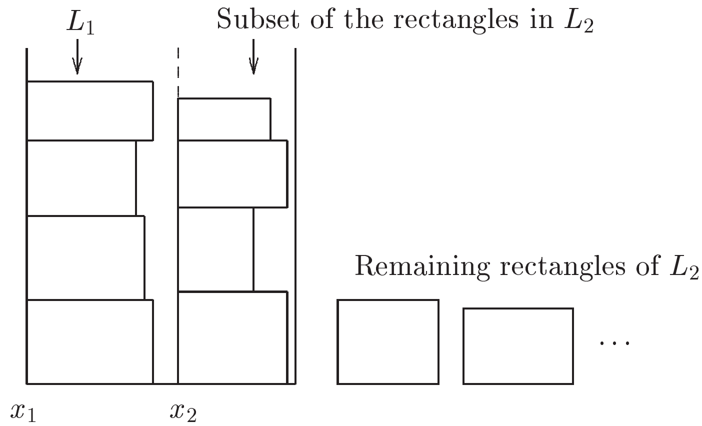

Figure 1: A packing generated by COLUMN$^{(s)}$

Theorem 2.2 *[11, 6] For any rational $\epsilon > 0$, there exists a linear-time algorithm* FL_ϵ *for the one-dimensional bin packing problem such that, for any input list L, $\mathrm{FL}_\epsilon(L) \leq (1+\epsilon)\,\mathrm{OPT}(L) + \beta_\epsilon$, where β_ϵ is a constant that depends on ϵ.*

3 Two-dimensional Strip Packing Problem

In this section we focus on the two-dimensional strip packing problem with rotation (2SP$^{\mathrm{r}}$). For the problem 2SP, Coffman, Garey, Johnson and Tarjan [7] presented the algorithms NFDH$^{(s)}$ (Next Fit Decreasing Height) and FFDH$^{(s)}$ (First Fit Decreasing Height) and proved that their asymptotic performance bounds are 2 and 1.7, respectively. Baker, Coffman and Rivest [1] presented another algorithm for 2SP, also with asymptotic performance bound 2, called BLDW (Bottom Leftmost Decreasing Width). More recently, Kenyon and Rémila [13] obtained an asymptotic approximation scheme for 2SP which we denote by KR_ϵ. The following holds for this algorithm.

Theorem 3.1 *[13]. For any rational $\epsilon > 0$, there exists a polynomial-time algorithm* KR_ϵ *for* 2SP *such that $\mathrm{KR}_\epsilon(L) \leq (1+\epsilon)\,\mathrm{OPT}(L) + Z/\epsilon^2$, for any list L of rectangles with dimensions at most Z.*

We note that for the problem 2SP$^{\mathrm{r}}$, the bound 2 of the algorithms BLDW and NFDH$^{(s)}$ is also valid, since the proofs of the bounds are based only in area arguments.

The algorithm we present for 2SP$^{\mathrm{r}}$ in this section is called SPR (*Strip Packing with Rotation*). Before presenting this algorithm, we describe an algorithm used as subroutine called COLUMN$^{(s)}$. This algorithms builds a packing consisting of two columns, each of which is a stack of rectangles packed one on top of the other (see figure 1). Each column is associated with a set of items (called *critical sets*), and the two sets are chosen carefully so as to guarantee a 'good' packing. This algorithm combines items of types that, if considered separately, do not lead to good space filling.

The algorithm COLUMN$^{(s)}$ 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. Each sublist is associated with one of the columns. We call the *height of the column* the sum of the heights of the rectangles in the corresponding column.

To pack a rectangle, the algorithm chooses a column with the smallest height. Let h be the height of this column and L_i the list associated with this column. In case not all rectangles of L_i have been packed, the next rectangle is packed in the position (x_i, h). Then, the algorithm updates the list L_i by removing the rectangle r. This process is repeated until one of the lists, L_1 or L_2 is totally packed. The algorithm returns a pair $(\mathcal{P}', L')$, where $\mathcal{P}'$ is the packing generated and L' is the set of the rectangles packed in $\mathcal{P}'$. We assume that the positions x_1 and x_2 and the lists L_1 and L_2 are such that they do not generate infeasible packings.

The following result holds for this algorithm.

Lemma 3.2 *Let $\mathcal{P}'$ be a packing of $L' \subseteq L_1 \cup L_2$ generated by the algorithm* COLUMN$^{(s)}$ *when applied to lists L_1 and L_2 for* 2SP(a)*. If $x(r) \geq l_i\, a$ and no rectangle $r \in L_i$, $i = 1, 2$ has height greater than Z, then $H(\mathcal{P}') \leq \frac{1}{l_1+l_2} \frac{S(L')}{a} + Z$.*

Proof. Consider the two columns of the packing $\mathcal{P}'$. Note that each column has height at least $H(\mathcal{P}') - Z$. Since each rectangle of L_i has width at least $l_i\, a$ the total area of the rectangles is at least $S(L) \geq (l_1 + l_2)\, a\, (H(\mathcal{P}') - Z)$. Therefore, $H(\mathcal{P}') \leq \frac{1}{l_1+l_2} \frac{S(L')}{a} + Z$. □

We say that the value s in inequalities of the form $H(\mathcal{P}) \leq \frac{1}{s} \frac{S(L)}{a} + Z$ is an *area guarantee* of the packing $\mathcal{P}$. The idea used in the algorithm SPR is to generate a packing consisting of two parts. One part is associated with a partial optimum packing generated with 'large' rectangles (those with width greater than $a/2$) and the other part is associated with a packing with better area guarantee.

The algorithm COLUMN$^{(s)}$ packs together some large rectangles with some 'thin' rectangles. After this step, the packing of the remaining non-packed rectangles is done using NFDH$^{(s)}$ strategy: first the rectangles are sorted in decreasing order of their height and then they are packed in this order, side by side generating levels. When a rectangle cannot be packed in a level, then a new level is created parallel to the last one (and starting at the top of the highest rectangle of the previous level), and the process is repeated. The following result holds for the algorithm NFDH$^{(s)}$ [7].

Lemma 3.3 *For a list L, let $N_1, \ldots, N_v$ be the levels generated (in this order) by* NFDH$^{(s)}(L)$*. 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 $w(N_i) \geq s\, a$, for $1 \leq i \leq v - 1$, then we have* NFDH$^{(s)}(L) \leq \frac{1}{s} \frac{S(L)}{a} + Z$.

Before we present the algorithm SPR, we state a result that will be useful in this section and in the others. The proof of this result can be found in [18].

Lemma 3.4 *Suppose X, Y, x, y are real numbers such that $x > 0$ and $0 < X < Y < 1$. Then*

$$\frac{x+y}{\max\{x,\, X\,x + Y\,y\}} \le 1 + \frac{1-X}{Y}.$$

Algorithm SPR(L)

Input: List of rectangles L and a bin $B = (a, \infty)$ (instance of $2\mathrm{SP}^{\mathrm{r}}(a)$))
Output: Packing $\mathcal{P}$ of L into B.
Subroutines: $\mathrm{COLUMN}^{(s)}$ and $\mathrm{NFDH}^{(s)}$.

1 Rotate all rectangles $r \in L$ with $x(r) > \frac{a}{2}$ and $y(r) \le \frac{a}{2}$.

2 Rotate all rectangles $r \in L$ with $x(r) > \frac{a}{2}$ and $x(r) < y(r) \le a$.
/* each large rectangle is rotated so as to have the lowest possible height. */

3 Let $p \leftarrow 1/\sqrt{6}$.

4 Let $L_A \leftarrow L \cap \mathcal{X}[\frac{1}{2}, (1-p)]$, $L_B \leftarrow L \cap \mathcal{X}[1-2p, p]$.

5 $(\mathcal{P}_{AB}, L_{AB}) \leftarrow \mathrm{COLUMN}^{(s)}(L_A, L_B, 0, 1-p)$.

6 Let $L \leftarrow L \setminus L_{AB}$, $L_1 \leftarrow L \cap \mathcal{X}[\frac{1}{2}, 1]$, $L_2 \leftarrow L \cap \mathcal{X}[\frac{1}{3}, \frac{1}{2}]$, $L_3 \leftarrow L \cap \mathcal{X}[0, \frac{1}{3}]$.

7 $\mathcal{P}_i \leftarrow \mathrm{NFDH}^{(s)}(L_i)$, $i = 1, 2, 3$.

8 $\mathcal{P}_{opt} \leftarrow \mathcal{P}_1 \| \mathcal{P}_{AB}$.

9 $\mathcal{P}_{aux} \leftarrow \mathcal{P}_2 \| \mathcal{P}_3$.

10 Return $\mathcal{P}_{opt} \| \mathcal{P}_{aux}$.

End Algorithm.

Theorem 3.5 *For any input list L for the $2\mathrm{SP}^{\mathrm{r}}(a)$, where the rectangles of L have dimensions at most Z, we have*

$$\mathrm{SPR}(L) \le \alpha_{\mathrm{SPR}}\, \mathrm{OPT}(L) + 3Z,$$

where $\alpha_{\mathrm{SPR}} = 1 + \frac{\sqrt{6}}{4} = 1.613\ldots$.

Proof. Let us first show that the packing $\mathcal{P}_{AB}$ of the sublist L_{AB} has area guarantee $s = 1/2 + (1-2p)$. 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 3.2 we have the following inequality.

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

Since $1/2 + (1-2p) \ge 1 - p$, we have

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

Let L_{opt} be 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 $L_1 \cup L_2 \cup L_3 \cup L_{AB}$ cannot be rotated, or if they can, they remain in the set defined as L_1. Moreover, the large rectangles of L_{opt} are packed with the smallest height possible. Therefore, we have

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

Now, we analyse two cases.

Case 1. $L_A \subseteq L_{AB}$. (All rectangles of L_A are totally packed in $\mathcal{P}_{AB}$.)

In this case, all rectangles of L_1 have width greater than $(1-p)a$. Thus, $H(\mathcal{P}_1) \leq \frac{1}{1-p}\frac{S(L_1)}{a}$. Since $H(\mathcal{P}_{opt}) = H(\mathcal{P}_1) + H(\mathcal{P}_{AB})$, using (1), we obtain

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

For the lists L_2 and L_3 the occupied area in each level of $\mathcal{P}_i$, except perhaps in the last, is at least $\frac{2}{3}\,a$. So, from Lemma 3.3 we have

$$H(\mathcal{P}_i) \leq \frac{1}{2/3}\frac{S(L_i)}{a} + Z, \quad i = 2, 3. \tag{4}$$

Since the packing $\mathcal{P}_{aux}$ is the concatenation of packings $\mathcal{P}_2$ and $\mathcal{P}_3$, setting $L_{aux} := L_1 \cup L_2$, we have

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

Defining h_1 and h_2 as $h_1 := H(\mathcal{P}_{opt}) - Z$ and $h_2 := H(\mathcal{P}_{aux}) - 2\,Z$, we have

$$\mathrm{OPT}(L) \geq \frac{S(L)}{a} \geq \frac{S(L_{opt})}{a} + \frac{S(L_{aux})}{a} \geq (1-p)\,h_1 + \frac{2}{3}\,h_2\ . \tag{6}$$

From (6) and (2) we have $\mathrm{OPT}(L) \geq \max\{h_1,\, (1-p)\,h_1 + \frac{2}{3}\,h_2\}$, and therefore, we obtain

$$H(\mathcal{P}) \leq \alpha_1\, \mathrm{OPT}(L) + 3\,Z,$$

where $\alpha_1 = (h_1 + h_2)/\max\{h_1, (1-p)\,h_1 + \frac{2}{3}\,h_2\}$.

Case 2. $L_B \subseteq L_{AB}$. (All rectangles of L_B are totally packed in $\mathcal{P}_{AB}$.)

The analysis of this case is based on the same arguments used in Case 1; therefore, we present only the inequalities that can be obtained.

$$\begin{aligned} H(\mathcal{P}_{opt}) &\leq \frac{1}{1/2}\frac{S(L_{opt})}{a} + Z, \\ H(\mathcal{P}_{aux}) &\leq \frac{1}{2\,p}\frac{S(L_{aux})}{a} + 2\,Z, \\ \mathrm{OPT}(L) &\geq \max\{h_1,\, \frac{1}{2}\,h_1 + 2\,p\,h_2\}. \end{aligned}$$

From these inequalities and proceeding as in the previous case, we have

$$H(\mathcal{P}) \leq \alpha_2\, \mathrm{OPT}(L) + 3\,Z,$$

where $\alpha_2 = (h_1 + h_2)/\max\{h_1, \frac{1}{2} h_1 + 2 p h_2\}$.

Now consider α_1 and α_2 obtained in Case 1 and Case 2, respectively. Using Lemma 3.4 we can conclude that $\alpha_1 \leq 1+\frac{3}{2} p$ and $\alpha_2 \leq 1+\frac{1}{4p}$. Since $p = \frac{1}{\sqrt{6}}$, we have that $1+\frac{3p}{2} = 1+\frac{1}{4p}$ (in fact, the value of p defined in step 3 of the algorithm was obtained by imposing this equality). Thus combining cases 1 and 2 we can conclude that $H(\mathcal{P}) \leq \alpha\,\mathrm{OPT}(L) + 3Z$, where $\alpha = 1 + \frac{3}{2} p = 1.6123\ldots$ This completes the proof of the theorem. □

4 Two-Dimensional Bin Packing Problem

As we mentioned in the introduction, Caprara [3] presented a 1.691-approximation algorithm for 2BP; to our knowledge, this is the algorithm with the best asymptotic performance bound that is known for this problem. In this section, we present an algorithm for the problem $2\mathrm{BP}^{\mathrm{r}}$, denoted by $\mathrm{BI}_{k,\epsilon}$. We show that the asymptotic performance bound of this algorithm does not exceed a value that can be made as close to 2.64 as desired.

The techniques we used to design algorithm $\mathrm{BI}_{k,\epsilon}$ are very similar to the ones we used for the algorithm SPR: they are based on critical sets and combination of them.

Before presenting the algorithm, we describe four algorithms which will be used as subroutines: $\mathrm{NFDH}^{(2)}$, BI_m, FFC, and COMBINE-AB_k^{xy}. The algorithm BI_m is used to pack small rectangles (and is based on $\mathrm{NFDH}^{(2)}$). The algorithm FFC is a specific routine for $2\mathrm{BP}^{\mathrm{r}}$ used to combine critical sets. The algorithm COMBINE-AB_k^{xy} is also used to combine critical sets, but as it is used in other algorithms for three-dimensional packing problems, it will be described in a more general way.

The algorithm $\mathrm{NFDH}^{(2)}$ (Next Fit Decreasing Height), first sorts the input list L in decreasing order of height, then packs the rectangles side by side generating levels. For more details on this algorithm the reader may refer to [16], where the proof of the following result may be found.

Lemma 4.1 *If L is a list of rectangles $L \subseteq \mathcal{C}_m^{xy}$ and $S(L) \leq \left(1 - \frac{1}{m}\right)^2 ab$, then* $\mathrm{NFDH}^{(2)}$ *packs L into one bin (a, b).*

We denote by $\mathrm{NFDH}^{(2x)}$, respectively $\mathrm{NFDH}^{(2y)}$, the variant of $\mathrm{NFDH}^{(2)}$ that creates levels parallel to x-axis, repectively y-axis.

The design of the next algorithm, BI_m, is based on the above lemma, and a partitioning of the input list into sublists for which we can guarantee an area occupation of at least $\left(\frac{m}{m+1}\right)^2 ab$ in each bin, except perhaps in a constant number of them.

Algorithm $\mathrm{BI}_\mathbf{m}$
Input: List of rectangles $L \subseteq \mathcal{C}_m^{xy}$.
Output: Packing of L into bins $R = (a, b)$.

1 Partition the list L into sublists $L_1, \ldots, L_6$ as follows:

$$L_1 \leftarrow L \cap \mathcal{C}^{xy}\left[\tfrac{1}{m+1},\tfrac{1}{m}\ ;\ \tfrac{1}{m+1},\tfrac{1}{m}\right], \qquad L_2 \leftarrow L \cap \mathcal{C}^{xy}\left[0,\tfrac{1}{m+1}\ ;\ \tfrac{1}{m+1},\tfrac{1}{m}\right],$$
$$L_3 \leftarrow L \cap \mathcal{C}^{xy}\left[\tfrac{1}{m+1},\tfrac{1}{m}\ ;\ 0,\tfrac{1}{m+1}\right], \qquad L_4 \leftarrow L \cap \mathcal{C}^{xy}\left[0,\tfrac{1}{m+1}\ ;\ \tfrac{1}{m+2},\tfrac{1}{m+1}\right] \cap \mathcal{C}^x_y,$$
$$L_5 \leftarrow L \cap \mathcal{C}^{xy}\left[\tfrac{1}{m+2},\tfrac{1}{m+1}\ ;\ 0,\tfrac{1}{m+1}\right] \cap \mathcal{C}^y_x, \quad L_6 \leftarrow L \cap \mathcal{C}^{xy}\left[0,\tfrac{1}{m+2}\ ;\ 0,\tfrac{1}{m+2}\right].$$

3 $\mathcal{P}_i \leftarrow \mathrm{NFDH}^{(2y)}(L_i), \quad i = 3,5.$

4 Generate a packing $\mathcal{P}_6$ of L_6 as follows:

Partition the list L_6 into sublists $L_6^1,\ldots,L_6^v$ such that

$$\begin{aligned} S(L_6^i) &\le \left[\left(\tfrac{m}{m+2}\right)+\left(\tfrac{1}{m+2}\right)^2\right]ab, \quad \text{for} \quad i=1,\ldots,v; \\ S(L_6^i)+S(\mathrm{first}(L_6^{i+1})) &> \left[\left(\tfrac{m}{m+2}\right)+\left(\tfrac{1}{m+2}\right)^2\right]ab, \quad \text{for} \quad i=1,\ldots,v-1; \end{aligned}$$

where $\mathrm{first}(L_6^{i+1})$ returns the first item of the list L_6^{i+1}.

$\mathcal{P}_6^i \leftarrow \mathrm{NFDH}^{(2)}(L_6^i) \quad$ for $\quad i=1,\ldots,v;$

$\mathcal{P}_6 \leftarrow \mathcal{P}_6^1\|\ldots\|\mathcal{P}_6^v.$

5 $\mathcal{P} \leftarrow \mathcal{P}_1\|\ldots\|\mathcal{P}_6.$

6 Return $\mathcal{P}$.

end algorithm.

It is not difficult to see that, for $i = 1,\ldots,6$ the area occupied by the rectangles of L_i in the bins is at least $\left(\frac{m+1}{m}\right)^2$, except perhaps in 1 bin. Thus, the following result follows.

Lemma 4.2 *For any list of rectangles $L \subseteq \mathcal{C}^{xy}_m$ we have*

$$\mathrm{BI}_m(L) \le \left(\frac{m+1}{m}\right)^2 \frac{S(L)}{ab} + 6.$$

In fact, the following generalization also holds.

Lemma 4.3 *Let $\mathcal{A}$ be an algorithm for $2\mathrm{BP}^{\mathrm{r}}$ to pack rectangles into a bin $B = (a,b)$. If $\mathcal{A}$ guarantees an area occupation of at least $s\,a\,b$, $s > 0$, in each bin, except perhaps in a constant number C of them, then the following holds for any input list L.*

$$\mathcal{A}(L) \le \frac{1}{s}\frac{S(L)}{a\,b} + C \le \frac{1}{s}\mathrm{OPT}(L) + C. \tag{7}$$

We call the value s in the above lemma an *area guarantee* of the packing generated by $\mathcal{A}$.

Now, let us describe the algorithm FFC, *First Fit Column*. The input parameters of this algorithm are a list of rectangles L, two sets (types) of rectangles $\mathcal{T}_1$ and $\mathcal{T}_2$ and two lists of coordinates p_1 and p_2 (corresponding to positions in the bin B). The list p_i is associated with $\mathcal{T}_i$ and is composed by n_i points, $i = 1, 2$. This algorithm generates a packing where each bin has $n_1 + n_2$ rectangles, except perhaps the last.

Algorithm FFC.

Input: $(L, \mathcal{T}_1, \mathcal{T}_2, p_1, p_2)$

Output: A pair $(\mathcal{P}, L')$, where $\mathcal{P}$ is a packing such that all rectangles in L of type $\mathcal{T}_1$ or all rectangles of type $\mathcal{T}_2$ are totally packed; and L' is the set of rectangles in the packing $\mathcal{P}$.

1 Repeat

1.1 Let $q \in p_i$, for some $i \in \{1, 2\}$, be a free position in the current bin (if there is no free position, consider a position in a new bin, which becomes the current bin). The positions in the bin where no rectangles were packed are considered free.

1.2 Take an unpacked rectangle r' of type $\mathcal{T}_i$ (without loss of generality, consider $r' \in \mathcal{T}_i$, otherwise, rotate r' previously).

1.3 If there is no such a rectangle in step 1.2 go to step 2.

1.4 Pack r' at the position q.

2 Let L' be the set of rectangles packed in $\mathcal{P}$.

3 Return $(\mathcal{P}, L')$.

End algorithm.

In what follows we present the algorithm COMBINE-AB$_k^{xy}$. To describe this algorithm we have to define some numbers which are used to define special critical sets. These numbers have already been used in [19, 17]. For completeness, we present them, as well as the critical sets and related results.

Definition 4.4 *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 defined as follows:*

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

The following result can be proved using a continuity argument.

Claim 4.5 *The numbers $r_1^{(k)}, r_2^{(k)}, \ldots, r_k^{(k)}$ are such that $r_1^{(k)} > r_2^{(k)} > \cdots > r_k^{(k)} > \frac{1}{3}$ and $r_1^{(k)} \to \frac{4}{9}$ as $k \to \infty$.*

For simplicity, we omit the superscripts $^{(k)}$ of the notation $r_i^{(k)}, s_i^{(k)}$ when k is clear from the context.

Using the numbers in Definition 4.4, we define the following critical sets (see figure 2).

$$\mathcal{A}_i^{xy} = \mathcal{C}^{xy}\left[r_{i+1}, r_i \ ; \ \frac{1}{2}, s_i\right], \quad \mathcal{B}_i^{xy} = \mathcal{C}^{xy}\left[\frac{1}{2}, s_i \ ; \ r_{i+1}, r_i\right],$$

$$\mathcal{A}^{xy} = \bigcup_{i=1}^{k+14} \mathcal{A}_i^{xy}, \quad \mathcal{B}^{xy} = \bigcup_{i=1}^{k+14} \mathcal{B}_i^{xy}, \quad \mathcal{A}_{[1-k]}^{xy} = \bigcup_{i=1}^{k} \mathcal{A}_i^{xy}, \quad \mathcal{B}_{[1-k]}^{xy} = \bigcup_{i=1}^{k} \mathcal{B}_i^{xy}.$$

The next definition refers to the lists of positions $p_{i,j}, q_{i,j}, p'_j, q'_j, p''_j$ and q''_j to be considered when applying the algorithm FFC. To use in this context, we have to consider a proportional scaling for a bin $B = (a, b)$.

Definition 4.6 Positions to combine sublists of $\mathcal{A}_i^{xy}$ and $\mathcal{B}_j^{xy}$. *We define here the positions only for $i \leq j$ (the case $i > j$ is symmetric); and these are relative to a bin $(1, 1)$.*

- *To combine the lists $\mathcal{A}_i^{xy}$ $(1 \leq i \leq k)$ and $\mathcal{B}_j^{xy}$ $(i \leq j \leq k)$, take*

  $p_{i,j} \ = \left[(0,0), (\frac{1}{2}, 0)\right]$ *and* $q_{i,j} \ = \left[(0, s_i)\right]$.

  (In this case we can obtain a packing with an area guarantee of at least $\frac{1}{2}$.)

- *To combine the list $\mathcal{A}_{[1-k]}^{xy} := \mathcal{A}_1^{xy} \cup \ldots \cup \mathcal{A}_k^{xy}$ with $\mathcal{B}_j^{xy}$ $(k+1 \leq j \leq k+14)$, we consider two phases. We divide $\mathcal{A}_{[1-k]}^{xy}$ into $\mathcal{A}_j'^{xy}$ and $\mathcal{A}_j''^{xy}$ taking $\mathcal{A}_j'^{xy} := \{b \in \mathcal{A}_{[1-k]}^{xy} : \ x(b) \leq 1 - s_j\}$ and $\mathcal{A}_j''^{xy} := \mathcal{A}_{[1-k]}^{xy} \setminus \mathcal{A}'^{xy}$.*

  - ⋆ *To combine $\mathcal{A}_j'^{xy}$ with $\mathcal{B}_j^{xy}$ take*

    $p'_j \ = [(s_j, 0)]$ *and*

    $q'_j \ = \left[(0,0)\, , \left(0, \frac{1}{j-k+2}\right), \left(0, \frac{2}{j-k+2}\right), \ldots, \left(0, \frac{j-k+1}{j-k+2}\right)\right]$.

    (In this case we can obtain a packing with an area guarantee of at least $\frac{13}{24}$. This minimum is attained when $j = k + 1$.)

  - ⋆ *To combine $\mathcal{A}_j''^{xy}$ with $\mathcal{B}_j^{xy}$ take*

    $p''_j \ = \left[(0,0), (\frac{1}{2}, 0)\right]$ *and*

    $q''_j \ = \left[\left(0, \frac{2}{3}\right), \left(0, \frac{2}{3} + \frac{1}{j-k+2}\right), \left(0, \frac{2}{3} + \frac{2}{j-k+2}\right), \ldots, \left(0, \frac{2}{3} + \left(\lfloor \frac{j-k+2}{3} \rfloor - 1\right) \frac{1}{j-k+2}\right)\right]$.

    (Here we can obtain a packing with an area guarantee of at least $\frac{27}{56}$.)

- *To combine the lists $\mathcal{A}_i^{xy}$ $(k+1 \leq i \leq k+14)$ and $\mathcal{B}_j^{xy}$ $(i \leq j \leq k+14)$, take*

  $p_{i,j} = \left[(s_j, 0)\, , \left(s_j + \frac{1}{i-k+2}, 0\right), \left(s_j + \frac{2}{i-k+2}, 0\right), \ldots, \left(s_j + (\lfloor (1-s_j) \cdot (i-k+2) \rfloor - 1) \frac{1}{i-k+2}, 0\right)\right]$ and

  $q_{i,j} \ = \left[(0,0)\, , \left(0, \frac{1}{j-k+2}\right), \left(0, \frac{2}{j-k+2}\right), \ldots, \left(0, \frac{j-k+1}{j-k+2}\right)\right]$.

  (In this case we can also obtain an area guarantee of at least $\frac{27}{56}$.)

The proof of the next result follows from the areas guarantee we mentioned in the definition given above.

Lemma 4.7 *The following statements hold for the list of positions $p_{i,j}, q_{i,j}, p'_j, q'_j, p''_j$ and q''_j:*

(a) If $\mathcal{P}$ is a packing into a bin $B = (a, b)$ generated by the algorithm FFC *with parameters $(L, \mathcal{A}_i^{xy}, \mathcal{B}_j^{xy}, p_{i,j}, q_{i,j})$, $1 \le i, j \le k$ or $k+1 \le i, j \le k+14$, then $\#(\mathcal{P}) \le \frac{56}{27} \frac{S(L')}{a\, b} + Z$, where L' is the set of rectangles packed in $\mathcal{P}$.*

(b) There is a partition of $\mathcal{A}_{[1-k]}^{xy}$ into sets $\mathcal{A}_j^{\prime xy}$ and $\mathcal{A}_j^{\prime\prime xy}$ such that a packing $\mathcal{P}'$ generated by the algorithm FFC *with parameters $(L, \mathcal{A}_j^{\prime xy}, \mathcal{B}_j^{xy}, p'_j, q'_j)$, $k+1 \le j \le k+14$, is such that $\#(\mathcal{P}') \le \frac{56}{27} \frac{S(\mathcal{P}'')}{a\, b} + Z$; and a packing $\mathcal{P}''$ generated by the algorithm* FFC *with parameters $(L, \mathcal{A}_j^{\prime\prime xy}, \mathcal{B}_j^{xy}, p''_j, q''_j)$, $k+1 \le j \le k+14$, is such that $\#(\mathcal{P}'') \le \frac{56}{27} \frac{S(\mathcal{P}'')}{a\, b} + Z$.*

(c) Defining positions symmetric to $p_{i,j}, q_{i,j}, p'_j, q'_j, p''_j$ and q''_j, analogous results hold when the sets $\mathcal{A}^{xy}$ and $\mathcal{B}^{xy}$ are exchanged in the items above.

Now, let us give an idea on how the combination of items of types $\mathcal{A}_k^{xy}$ and $\mathcal{B}_k^{xy}$ is done by the algorithm COMBINE-AB$_k^{xy}$ (see figure 2). This algorithm is called with five parameters: $(L, \text{ftype}, \mathcal{T}_A, \mathcal{T}_B, \text{COMBINE})$. The first parameter L is a list of items, ftype is a function that is either rtype or xy-type; $\mathcal{T}_A$ and $\mathcal{T}_B$ are sets used to restrict the input items; and COMBINE is a subroutine called to generate partial packings by combining two types of items. In the present section, COMBINE will be the algorithm FFC and ftype will be the function xy-type.

For the informal description, we refer to the description in steps given in the sequel. The routine UPDATE receives a sequence of lists as input parameters, $L_1, \ldots, L_m$. This function removes from the lists L_i, $1 \le i \le m$, all items that have already been packed up to the moment of its call, and returns the updated lists.

In step 3, the algorithm COMBINE-AB$_k^{xy}$ calls the subroutine COMBINE to pack all items of ftype$(L, \mathcal{T}_A \cap \mathcal{A}_k^{xy})$ or all items of ftype$(L, \mathcal{T}_B \cap \mathcal{B}_k^{xy})$. For that, it combines the items of ftype$(L, \mathcal{T}_A \cap \mathcal{A}_{k,i}^{xy})$ with ftype$(L, \mathcal{T}_B \cap \mathcal{B}_{k,j}^{xy})$, $1 \le i, j \le k$. It starts combining items of ftype$(L, \mathcal{T}_A \cap \mathcal{A}_{k,1}^{xy})$ with ftype$(L, \mathcal{T}_B \cap \mathcal{B}_{k,1}^{xy})$. If all items of ftype$(L, \mathcal{T}_A \cap \mathcal{A}_{k,i}^{xy})$ have been packed, then the algorithm proceeds combining items of ftype$(L, \mathcal{T}_A \cap \mathcal{A}_{k,i+1}^{xy})$ with ftype$(L, \mathcal{T}_B \cap \mathcal{B}_{k,j}^{xy})$; otherwise, of ftype$(L, \mathcal{T}_A \cap \mathcal{A}_{k,i}^{xy})$ with ftype$(L, \mathcal{T}_B \cap \mathcal{B}_{k,j+1}^{xy})$.

After performing step 3, either all items of ftype$(L, \mathcal{T}_A \cap \mathcal{A}_{[1-k]}^{xy})$ are packed (go to step 4) or all items of ftype$(L, \mathcal{T}_B \cap \mathcal{B}_{[1-k]}^{xy})$ are packed (go to step 5). The steps 4 and 5 are symmetric, therefore, w.l.o.g., consider that all items of ftype$(L, \mathcal{T}_B \cap \mathcal{B}_{[1-k]}^{xy})$ were packed (step 4). In this case, the set ftype$(L, \mathcal{T}_A \cap \mathcal{A}_{[1-k]}^{xy})$ is divided in two parts: the sets ftype$(L, \mathcal{T}_A \cap \mathcal{A}_j^{xy'})$ and ftype$(L, \mathcal{T}_A \cap \mathcal{A}_j^{xy''})$. After this, the items of each part are combined with the items of ftype$(L, \mathcal{T}_B \cap \mathcal{B}_{k,j}^{xy})$. At the end of step 4, we have that all items of ftype$(L, \mathcal{T}_A \cap \mathcal{A}_{[1-k]}^{xy})$ or all items of ftype$(L, \mathcal{T}_B \cap \mathcal{B}_k^{xy})$ were packed. In case there are unpacked items of ftype $(L, \mathcal{T}_B \cap \mathcal{B}_k^{xy})$, they are packed with the remaining items of ftype $(L, \mathcal{T}_A \cap \mathcal{A}_k^{xy})$ in the same way as done in step 3.

Algorithm COMBINE-AB$_k^{xy}(L, \text{ftype}, \mathcal{T}_A, \mathcal{T}_B, \text{COMBINE})$

Input: A list of items L; sets $\mathcal{T}_A$ and $\mathcal{T}_B$ and a subroutine COMBINE.

Output: Partial packing $\mathcal{P}_{AB}$ of L, such that all items of ftype $(L, \mathcal{T}_A \cap \mathcal{A}_k^{xy})$ or all items of ftype $(L, \mathcal{T}_B \cap \mathcal{B}_k^{xy})$ are packed.

1 Let $p_{i,j}, q_{i,j}$, $(1 \leq i, j \leq k+14)$, and p'_j, p''_j, q'_j, q''_j, $(k+1 \leq j \leq k+14)$, be the positions given in Definition 4.6.

2 $i \leftarrow 1$; $j \leftarrow 1$; $\mathcal{P}_{AB} \leftarrow \emptyset$.

3 While $(i \leq k$ and $j \leq k)$ do

- **3.1** $\mathcal{P}' \leftarrow \text{COMBINE}\,(L, \mathcal{T}_A \cap \mathcal{A}_{k,i}^{xy}, \mathcal{T}_B \cap \mathcal{B}_{k,j}^{xy}, p_{ij}, q_{ij})$.
- **3.2** UPDATE (L).
- **3.3** $\mathcal{P}_{AB} \leftarrow \mathcal{P}_{AB} \cup \mathcal{P}'$.
- **3.4** if ftype $(L, \mathcal{T}_A \cap \mathcal{A}_{k,i}^{xy}) = \emptyset$ then $i \leftarrow i+1$ else $j \leftarrow j+1$.

4 if ftype $(L, \mathcal{T}_B \cap \mathcal{B}_{[1-k]}^{xy}) = \emptyset$ then

- **4.1** While $(j \leq k+14$ and ftype $(L, \mathcal{T}_A \cap \mathcal{A}_{k,i}^{xy}) \neq \emptyset)$ do
  - **4.1.1** $\mathcal{P}' \leftarrow \text{COMBINE}\,(L, \mathcal{T}_A \cap \mathcal{A}_j'^{xy}, \mathcal{T}_B \cap \mathcal{B}_{k,j}^{xy}, p'_j, q'_j)$;
  - **4.1.2** $\mathcal{P}'' \leftarrow \text{COMBINE}\,(L, \mathcal{T}_A \cap \mathcal{A}_j''^{xy}, \mathcal{T}_B \cap \mathcal{B}_{k,j}^{xy}, p''_j, q''_j)$;
  - **4.1.3** $\mathcal{P}_{AB} \leftarrow \mathcal{P}_{AB} \| \mathcal{P}' \| \mathcal{P}''$.
  - **4.1.4** UPDATE (L).
  - **4.1.5** if ftype $(L, \mathcal{T}_B \cap \mathcal{B}_{k,j}^{xy}) = \emptyset$ then $j \leftarrow j+1$.
- **4.2** $i \leftarrow k+1$.

5 else (all items of ftype $(L, \mathcal{T}_A \cap \mathcal{A}_{[1-k]}^{xy})$ were packed) proceed as in step 4, in a symmetrical way.

6 While $(i \leq k+14$ and $j \leq k+14)$ do

- **6.1** $\mathcal{P}' \leftarrow \text{COMBINE}\,(L, \mathcal{T}_A \cap \mathcal{A}_{k,i}^{xy}, \mathcal{T}_B \cap \mathcal{B}_{k,j}^{xy}, p_{ij}, q_{ij})$.
- **6.2** UPDATE (L).
- **6.3** $\mathcal{P}_{AB} \leftarrow \mathcal{P}_{AB} \cup \mathcal{P}'$.
- **6.4** if ftype $(L, \mathcal{T}_A \cap \mathcal{A}_{k,i}^{xy}) = \emptyset$ then $i \leftarrow i+1$ else $j \leftarrow j+1$.

7 Return $\mathcal{P}_{AB}$.

End algorithm.

The algorithms we described above is for the plane xy. We denote by COMBINE-AB$_k^{yz}$ and COMBINE-AB$_k^{zx}$ the analogous algorithms using planes yz and zx, respectively. These algorithms are also used in the algorithm we designed for the problem 3BP$^{\text{r}}$ (see Section 7).

Now, we are ready to describe the algorithm $\text{BI}_{k,\epsilon}$. This algorithm performs two combination steps. One for combining critical items of types $\mathcal{A}_k^{xy}$ and $\mathcal{B}_k^{xy}$, and the other for combining critical items of types $\mathcal{T}_C$ and $\mathcal{T}_D$. The set $\mathcal{A}_k^{xy}$ is defined as the union of the sets $\mathcal{A}_{k,1}^{xy}, \ldots, \mathcal{A}_{k,14}^{xy}$ and the set $\mathcal{B}_k^{xy}$ as the union of the sets $\mathcal{B}_{k,1}^{xy}, \ldots, \mathcal{B}_{k,14}^{xy}$ (see figure 2). The combination is generated by the algorithm COMBINE-AB$_k^{xy}$ and the items of one of these types are totally packed (see figures 2 and 3). Denote by $\mathcal{P}_{AB}$ the packing generated by the algorithm COMBINE-AB$_k^{xy}$ and by L_{AB} the items packed in $\mathcal{P}_{AB}$. The packing $\mathcal{P}_{AB}$ has an area guarantee of at least 17/36.

After step 3, suppose that all rectangles of type $\mathcal{B}$ were packed. Consider the lists $L_1, \ldots, L_{23}$, defined in step 4.5 of the algorithm 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 the sublists $L_2, \ldots, L_{17}, L_{19}, \ldots, L_{23}$ the NFDH strategy generates packings with area guarantee at least 17/36. The packings with this area guarantee are packings with good area guarantee. The set of rectangles in L_1 and L_{18} give packings with area guarantee $\frac{4}{9}$, if packed by Next Fit Decreasing strategy. Therefore, we use the set $\mathcal{T}_D := \mathcal{T}_D' \cup \mathcal{T}_D''$ (see step 4.3) to define the critical rectangles in $L_1 \cup L_{18}$ that give packings with area guarantee close to $\frac{4}{9}$ (the rectangles in $(L_1 \cup L_{18}) \setminus \mathcal{T}_D$ give packings with good area guarantee).

Another critical set comes from the items in $L \cap (\wp_2 \cup \wp_4) \setminus L_{AB}$. These items give packings with area guarantee $\frac{1}{4}$. So, we use the set $\mathcal{T}_C$ to obtain the critical rectangles of $L \cap (\wp_2 \cup \wp_4) \setminus L_{AB}$ that give packings with area guarantee close to $\frac{1}{4}$.

We use the algorithm FFC to combine the items of type $\mathcal{T}_C$ and $\mathcal{T}_D$. The packing that is generated has bins with one rectangle of type $\mathcal{T}_C$ and one rectangle of type $\mathcal{T}_D'$, or one rectangle of type $\mathcal{T}_C$ and two rectangles of type $\mathcal{T}_D''$. If $\mathcal{P}_{CD}$ is a packing generated by FFC, and L_{CD} is the set of rectangles packed in $\mathcal{P}_{CD}$, then all rectangles of type $\mathcal{T}_C$ or all rectangles of type $\mathcal{T}_D$ are totally packed in the packing $\mathcal{P}_{CD}$. The packing $\mathcal{P}_{CD}$ also has 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.

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

Input: List of rectangles L (instance of 2BP$^{\mathrm{r}}(a, b)$).

Output: Packing $\mathcal{P}$ of L into bins $R = (a, b)$.

1 Rotate all rectangles $r \in L \cap \wp_4$ where $\rho(r) \in \wp_1 \cup \wp_2 \cup \wp_3$.
/* Let $R_1 \leftarrow \{r \in L \cap \wp_4 : \rho(r) \in \wp_1 \cup \wp_2 \cup \wp_3\}$; $L \leftarrow (L \setminus R_1) \bigcup \rho(R_1)$. */

2 $t \leftarrow (\sqrt{33} - 3)/6$.

3 $\mathcal{P}_{AB} \leftarrow$COMBINE-AB$_k^{xy}(L, xy\text{-type}, \wp_2, \wp_3, \text{FFC})$.
UPDATE(L).

4 If all rectangles of xy-type$(L, \mathcal{B}_k^{xy})$ were packed then

4.1 Rotate the rectangles of $L \cap \wp_2$ that fits in $\wp_1 \cup \wp_3$.

4.2 Rotate the rectangles b of $L \cap (\wp_2 \cup \wp_4)$ so as to have $b \in \mathcal{C}_1$ and $x(b)$ minimum.
/* The items in $L \cap (\wp_2 \cup \wp_4)$ will be considered in step 4.8. */

4.3 Let

$$\mathcal{T}_C = \mathcal{C}^{xy}\left[\tfrac{1}{2},1\ ;\ \tfrac{1}{2},1-t\right], \quad \mathcal{T}'_D = \mathcal{C}^{xy}[0,t\ ;\ 0,1],$$
$$\mathcal{T}''_D = \mathcal{C}^{xy}[0,t\ ;\ 0,1], \quad \mathcal{T}_D = \mathcal{T}'_D \bigcup \mathcal{T}''_D,$$
$$L_C \leftarrow xy\text{-type}(L,\mathcal{T}_C), \quad L_D \leftarrow xy\text{-type}(L,\mathcal{T}_D).$$

4.4 Generate packing $\mathcal{P}_{CD}$ as follows.

$(\mathcal{P}_{CD'}, L_{CD'}) \leftarrow \text{FFC}(L, \mathcal{T}_C, \mathcal{T}'_D, [(0,0)], [(0,1-t)])$.
$(\mathcal{P}_{CD''}, L_{CD''}) \leftarrow \text{FFC}(L, \mathcal{T}_C \setminus L_{CD'}, \mathcal{T}''_D, [(0,0)], [(0,1-t), (\frac{1}{2}, 1-t)])$.
$\mathcal{P}_{CD} \leftarrow \mathcal{P}_{CD'} \| \mathcal{P}_{CD''}$;
$L_{CD} \leftarrow L_{CD'} \bigcup L_{CD''}$. /* L_C or L_D is totally packed in L_{CD}. */

4.5 Partition the list $L \cap (\wp_1 \cup \wp_3)$ into sublists $L_1, \ldots, L_{23}$ as follows (see figure 3).

$$L_i \leftarrow L \bigcap \mathcal{C}^{xy}\left[\tfrac{1}{2},1\ ;\ \tfrac{1}{i+2},\tfrac{1}{i+1}\right],\ i=1,\ldots,16, \qquad L_{17} \leftarrow L \bigcap \mathcal{C}^{xy}\left[\tfrac{1}{2},1\ ;\ 0,\tfrac{1}{18}\right],$$
$$L_{18} \leftarrow L \bigcap \mathcal{C}^{xy}\left[\tfrac{1}{3},\tfrac{1}{2}\ ;\ \tfrac{1}{3},\tfrac{1}{2}\right], \qquad L_{19} \leftarrow L \bigcap \mathcal{C}^{xy}\left[\tfrac{1}{3},\tfrac{1}{2}\ ;\ \tfrac{1}{4},\tfrac{1}{3}\right],$$
$$L_{20} \leftarrow L \bigcap \mathcal{C}^{xy}\left[\tfrac{1}{3},\tfrac{1}{2}\ ;\ 0,\tfrac{1}{4}\right], \qquad L_{21} \leftarrow L \bigcap \mathcal{C}^{xy}\left[\tfrac{1}{4},\tfrac{1}{3}\ ;\ \tfrac{1}{3},\tfrac{1}{2}\right],$$
$$L_{22} \leftarrow L \bigcap \mathcal{C}^{xy}\left[0,\tfrac{1}{4}\ ;\ \tfrac{1}{3},\tfrac{1}{2}\right], \qquad L_{23} \leftarrow L \bigcap \mathcal{C}^{xy}\left[0,\tfrac{1}{3}\ ;\ 0,\tfrac{1}{3}\right].$$

4.6 Generate packings $\mathcal{P}_1, \ldots, \mathcal{P}_{23}$ as follows.

$\mathcal{P}_i \leftarrow \text{NFDH}^{(2y)}(L_i)$ for $i = 1, \ldots, 21$;
$\mathcal{P}_i \leftarrow \text{NFDH}^{(2x)}(L_i)$ for $i = 22$;
$\mathcal{P}_{23} \leftarrow \text{BI}_3(L_{23})$.

4.7 UPDATE (L). /* Note that $L \subseteq \wp_2 \cup \wp_4$. */

4.8 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.
Apply the algorithm FL_ϵ into L; let $\mathcal{P}_{\text{FL}_\epsilon}$ be this packing.
Let $\mathcal{P}_{\text{FFD}}$ be the packing $\text{FFD}(L \cap \mathcal{X}[0,\frac{1}{3}]) \| \text{FFD}(L \cap \mathcal{X}[\frac{1}{3},\frac{1}{2}]) \| \text{FFD}(L \cap \mathcal{X}[\frac{1}{2},1])$.
Let $\mathcal{P}_{\text{UNI}}$ be the smallest packing in $\{\mathcal{P}_{\text{FL}_\epsilon}, \mathcal{P}_{\text{FFD}}\}$.

4.9 $\mathcal{P}_{aux} \leftarrow \mathcal{P}_{AB} \| \mathcal{P}_{CD} \| \mathcal{P}_1 \| \ldots \| \mathcal{P}_{23}$;

4.10 $\mathcal{P} \leftarrow \mathcal{P}_{\text{UNI}} \| \mathcal{P}_{aux}$.

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

6 Return $\mathcal{P}$.

End algorithm.

The next theorem gives an asymptotic performance bound of the algorithm $\text{BI}_{k,\epsilon}$.

Theorem 4.8 *For any instance L of 2BP^{r}, we have*

$$\text{BI}_{k,\epsilon}(L) \leq \alpha_{k,\epsilon}\,\text{OPT}(L) + \mathcal{O}\left(k + \frac{1}{\epsilon}\right),$$

where $\alpha_{k,\epsilon} \to (25 + 3\sqrt{33})/16 = 2.639\ldots$ as $k \to \infty$ and $\epsilon \to 0$.

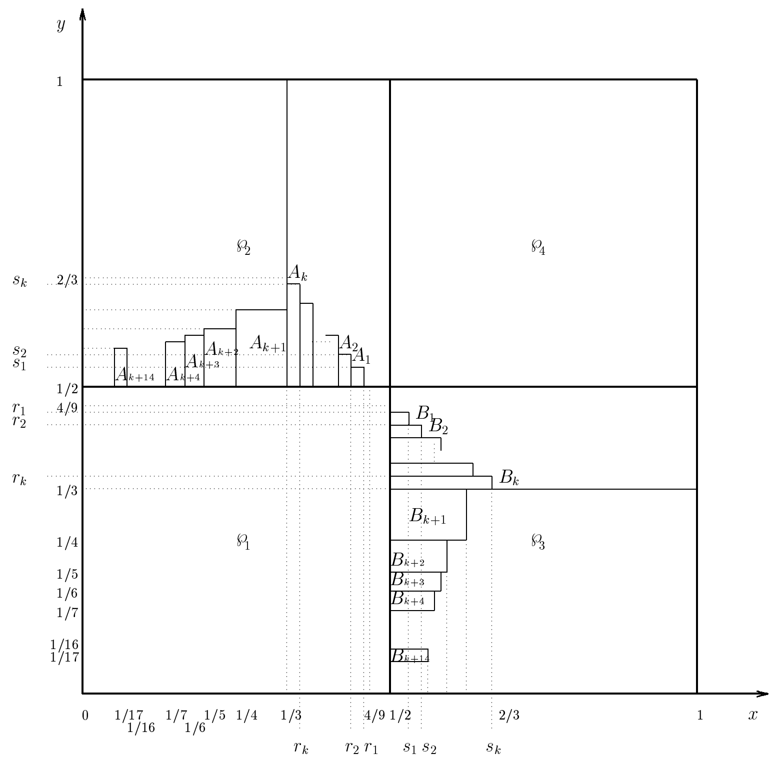

Figure 2: Sublists $A_i := \mathcal{A}_i^{xy}$ and $B_j := \mathcal{B}_j^{xy}$.

Proof. We present the proof for the case in which all rectangles of xy-type $(L, \mathcal{B}_k^{xy})$ were packed in step 4. The proof for the other case (step 5) is analogous. This proof is divided in two cases, according to step 4.4 ($L_C \subseteq L_{CD}$).

Each packing $\mathcal{P}_i$, $i \in \{1, \ldots, 23\} \setminus \{1, 18\}$, has an area guarantee of at least $\frac{17}{36}\, a\, b$, this minimum being attained when $i \in \{16, 17\}$. Therefore, applying Lemma 4.3 and 4.2 we can conclude that

$$\#(\mathcal{P}_i) \leq \frac{36}{17} \frac{S(L_i)}{ab} + 1, \quad \text{for} \quad i \in \{1, \ldots, 23\} \setminus \{1, 18\}. \tag{8}$$

Now, for each partial packing $\mathcal{Q}$ of $\mathcal{P}_{AB}$ generated by the algorithm COMBINE-AB$_k^{xy}$, we have $\#(\mathcal{Q}) \leq \frac{56}{27} \frac{S(\mathcal{Q})}{ab} + 1$. To see this, use Lemma 4.7 to conclude that in each one of these packings, the area guarantee in each bin, except perhaps in the last, is at least $\frac{27}{56} ab$. Since there exists a maximum of $(2k-1) + 28 + 14 \ = \ 2k + 41$ packings generated by L_A and L_B, we have $\#(\mathcal{P}_{AB}) \leq \frac{56}{27} \frac{S(L_{AB})}{ab} + (2k + 41)$. So, the following holds.

$$\#(\mathcal{P}_{AB}) \leq \frac{36}{17} \frac{S(L_{AB})}{ab} + (2k + 41). \tag{9}$$

For packings $\mathcal{P}_{CD'}$ and $\mathcal{P}_{CD''}$ (in step 4.4), the combined area in each bin is at least

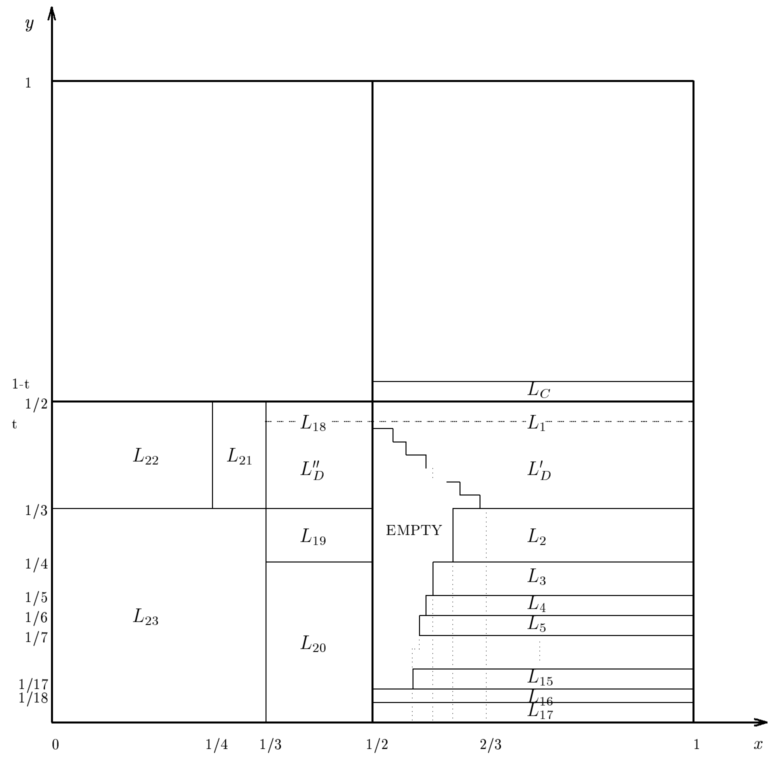

Figure 3: Sublist after the packing of list $L_B = (B_1 \cup \ldots \cup B_{k+14})$.

$(\frac{1}{4} + \frac{r_1}{2})ab$, then it follows from Lemma 4.3 that

$$\#(\mathcal{P}_{CD}) \leq \frac{1}{\left(\frac{1}{4} + \frac{r_1}{2}\right)} \frac{S(L_{CD})}{ab} + 2. \tag{10}$$

Let us analyze the two possibilities: $L_C \subseteq L_{CD}$ or $L_D \subseteq L_{CD}$ (see step 4.4).

Case 1. $L_C \subseteq L_{CD}$.

For packings $\mathcal{P}_1$ and $\mathcal{P}_{18}$ we have:

$$\#(\mathcal{P}_1) \leq \frac{1}{r_1} \frac{S(L_1)}{ab} + 1, \tag{11}$$

$$\#(\mathcal{P}_{18}) \leq \frac{1}{\frac{4}{9}} \frac{S(L_{18})}{ab} + 1. \tag{12}$$

By Theorem 2.2,

$$\#(\mathcal{P}_{\mathrm{UNI}}) \leq \#(\mathcal{P}_{\mathrm{FL}_\epsilon}) \leq (1+\epsilon)\mathrm{OPT}(L_{\mathrm{UNI}}) + \beta_\epsilon, \tag{13}$$

where L_{UNI} is the set of items packed in $\mathcal{P}_{\mathrm{UNI}}$. Note that to derive the last inequality we used the fact the items in L_{UNI} cannot be packed side by side in the y-direction. Also note that with the algorithm FFD applied to list L_{UNI} we obtain the following inequality

$$\#(\mathcal{P}_{\mathrm{UNI}}) \leq \#(\mathcal{P}_{\mathrm{FFD}}) \leq \frac{1}{(1-t)\frac{1}{2}} \frac{S(L_{\mathrm{UNI}})}{ab} + 3. \tag{14}$$

Now, for the packing $\mathcal{P}_{aux} = \mathcal{P}_{AB} \| \mathcal{P}_1 \| \ldots \| \mathcal{P}_{23}$, using the inequalities (9),...,(12) and the fact that $r_1 \leq \min\left\{\frac{17}{36}, \frac{1}{4} + \frac{r_1}{2}, \frac{4}{9}\right\}$, we obtain

$$\#(\mathcal{P}_{aux}) \leq \frac{1}{r_1} \frac{S(L_{aux})}{ab} + (2k + 68), \tag{15}$$

where L_{aux} denotes the set of rectangles in the packing $\mathcal{P}_{aux}$.

Let

$$n_1 := \#(\mathcal{P}_{\mathrm{UNI}}) - \beta_\epsilon, \tag{16}$$

$$n_2 := \#(\mathcal{P}_{aux}) - (2k + 68). \tag{17}$$

From inequality (13) we have $n_1 \leq (1+\epsilon)\mathrm{OPT}(L_{\mathrm{UNI}})$ and therefore,

$$\mathrm{OPT}(L) \geq \mathrm{OPT}(L_{\mathrm{UNI}}) \geq \frac{n_1}{(1+\epsilon)}. \tag{18}$$

From (15) and (17) we can conclude that

$$\frac{S(L_{aux})}{ab} \geq r_1 n_2, \tag{19}$$

and from (14) and (16), we have

$$\frac{S(L_{\mathrm{UNI}})}{ab} \geq \frac{(1-t)}{2} n_1. \tag{20}$$

Since $S(L) = S(L_{aux}) + S(L_{\mathrm{UNI}})$, using (19) and (20) we obtain $\frac{S(L)}{ab} \geq r_1 n_2 + \frac{(1-t)}{2} n_1$. So,

$$\mathrm{OPT}(L) \geq \frac{S(L)}{ab} \geq r_1 n_2 + \frac{(1-t)}{2} n_1.$$

Combining (18) and the above inequality, it follows that

$$\mathrm{OPT}(L) \geq \max\left\{\frac{1}{1+\epsilon} n_1, \frac{1-t}{2} n_1 + r_1 n_2\right\}.$$

Since $\#(\mathcal{P}) = \#(\mathcal{P}_{aux}) + \#(\mathcal{P}_{\mathrm{UNI}})$; using (16) and (17), we have

$$\#(\mathcal{P}) = (n_2 + (2k + 68) + n_1 + \beta_\epsilon) = n_1 + n_2 + \left(2k + \beta'_\epsilon\right),$$

where $\beta'_\epsilon = \beta_\epsilon + 68$. Therefore,

$$\mathrm{BI}_{k,\epsilon}(L) \leq \alpha'_{k,\epsilon}(r_1)\mathrm{OPT}(L) + \left(2k + \beta'_\epsilon\right),$$

where $\alpha'_{k,\epsilon}(r_1) = (n_1+n_2)/\max\left\{\frac{1}{1+\epsilon}n_1, \frac{(1-t)}{2}n_1 + r_1 n_2\right\}$. Now using Lemma 3.4, we can conclude that $\alpha'_{k,\epsilon}(r_1) \leq \left[\frac{1}{r_1} - \frac{(1-t)(1+\epsilon)}{2r_1} + (1+\epsilon)\right]$.

Case 2. $L_D \subseteq L_{CD}$.

The proof of this case is analogous. Therefore, the proof is shortened. Since all rectangles of L'_D were packed in $\mathcal{P}_{CD}$, we have an area guarantee of at least t for bins of $\mathcal{P}_1$, except perhaps the last. The same can be verified for the packing $\mathcal{P}_{18}$. Thus, the following holds.

$$\#(\mathcal{P}_i) \leq \frac{1}{t}\frac{S(L_i)}{ab} + 1 \quad \text{for} \quad i \in \{1, 18\}. \tag{21}$$

Since $t \leq \min\left\{\frac{1}{4} + \frac{r_1}{2}, \frac{17}{36}\right\}$, from (21), (9) and (10) we have

$$\#(\mathcal{P}_{aux}) \leq \frac{1}{t}\frac{S(L_{aux})}{ab} + (2k+68). \tag{22}$$

By Theorem 2.2,

$$\#(\mathcal{P}_{\text{UNI}}) \leq \#(\mathcal{P}_{\text{FL}_\epsilon}) \leq (1+\epsilon)\text{OPT}(L_{\text{UNI}}) + \beta_\epsilon. \tag{23}$$

The packing $\mathcal{P}_{\text{FFD}}$ has an area guarantee of at least $\frac{1}{4}$ in all bins, except perhaps in three of them; and since $\#(\mathcal{P}_{\text{UNI}}) \leq \#(\mathcal{P}_{\text{FFD}})$, we have

$$\#(\mathcal{P}_{\text{UNI}}) \quad \leq \quad \frac{1}{1/4}\frac{S(L_{\text{UNI}})}{ab} + 3. \tag{24}$$

Let

$$\begin{aligned} n_1 &:= \#(\mathcal{P}_{\text{UNI}}) - \beta_\epsilon, \quad \text{and} \\ n_2 &:= \#(\mathcal{P}_{aux}) - (2k+68). \end{aligned}$$

Then, from (23) we can conclude that

$$\text{OPT}(L) \geq \text{OPT}(L_{\text{UNI}}) \geq \frac{1}{1+\epsilon}n_1.$$

Now, from (22) and (24), we have

$$\frac{S(L_{aux})}{ab} \geq t\,n_2 \quad \text{and} \quad \frac{S(L_{\text{UNI}})}{ab} \geq \frac{1}{4}n_1,$$

and therefore,

$$\text{OPT}(L) \geq \frac{S(L)}{ab} \geq t\,n_2 + \frac{1}{4}n_1\ .$$

So,

$$\text{OPT}(L) \geq \max\left\{\frac{1}{1+\epsilon}n_1, \frac{1}{4}n_1 + t\,n_2\right\}.$$

Thus, $\text{BI}_{k,\epsilon}(L) \leq \alpha''_{k,\epsilon}(r_1)\text{OPT}(L) + (2k+\beta'_\epsilon)$, where $\alpha''_{k,\epsilon}(r_1) \leq \left[\frac{1}{t} - \frac{(1+\epsilon)}{4t} + (1+\epsilon)\right]$. The last inequality follows by taking $\alpha''_{k,\epsilon}(r_1) = (n_1+n_2)/\max\left\{\frac{1}{1+\epsilon}n_1, \frac{1}{4}n_1 + tn_2\right\}$ and using Lemma 3.4.

From both cases above, we can conclude that for $k \to \infty$ and $\epsilon \to 0$ the statement of the theorem holds. □

5 Three-Dimensional Strip Packing

In this section, we present an algorithm for $3SP^r$, called TRI_k, with asymptotic performance bound close to 2.75. This algorithm uses many other algorithms as subroutines, which we describe in what follows. For them, the following result will be useful (see [17, 19]).

Lemma 5.1 *Let L be an instance of $3SP^r$ and $\mathcal{P}$ be a packing of L consisting of levels $N_1, \ldots, N_v$ such that $\min\{z(b) : b \in N_i\} \geq \max\{z(b) : b \in N_{i+1}\}$, and $S(N_i) \geq s\,a\,b$ for a given constant $s > 0$, $i = 1, \ldots, v-1$. Then $H(\mathcal{P}) \leq \frac{1}{s}\frac{V(L)}{a\,b} + Z$.*

The value s in the above lemma is called *volume guarantee* of the packing $\mathcal{P}$.

First we describe the algorithm NFDH (Next Fit Decreasing Height) presented by Li and Cheng in [15]. The algorithm has two variants: $NFDH^x$ and $NFDH^y$. The notation NFDH is used to refer to any of these variants.

The algorithm $NFDH^x$ first sorts the boxes of L in decreasing order of their height, say $L = (e_1, e_2, \ldots, e_n)$. The first box e_1 is packed in the position $(0, 0, 0)$, the next one is packed in the position $(x(e_1), 0, 0)$ and so on, side by side, until a box is found that does not fit in this layer. At this moment the next box e_k is packed in the position $(0, y(e^*), 0)$, where $y(e^*) = \max\{y(e_i), i = 1, \ldots, k-1\}$. The process continues in this way until a box e_l is found that does not fit in the first level. Then the algorithm packs this box in a new level at the height $z(e_1)$. The algorithm proceeds in this way until all boxes of L have been packed.

The algorithm $NFDH^y$ is analogous to the algorithm $NFDH^x$, except that it generates the layers in the y-axis direction (for a more detailed description see [15]).

Another algorithm we use is the algorithm LL_m, presented by Li and Cheng [15]. This algorithm generates a level oriented packing for lists $L \subset \mathcal{C}_m^{xy}$, satisfying the conditions of Lemma 5.1, and with volume guarantee at least $\frac{m-2}{m}$. So, the following holds for this algorithm (see [15]).

Lemma 5.2 *If $\mathcal{P}$ is a packing generated by the algorithm LL_m for an instance $L \subseteq \mathcal{C}_m^{xy}$, then $H(\mathcal{P}) \leq \left(\frac{m}{m-2}\right) V(L) + Z$.*

We describe now the algorithm $BI_m^{(t)}$, which is used to pack small items. This algorithm is a 3-dimensional version of the algorithm BI_m for $2BP^r$.

Algorithm $\mathrm{BI}_{\mathbf{m}}^{(\mathbf{t})}$

Input: List of boxes $L \subseteq \mathcal{C}_m^{xy}$.

Output: Packing of L into a box $B = (a, b, \infty)$.

1 Partition the list L into sublists $L_1, \ldots, L_6$ as follows.

$$L_1 \leftarrow L \cap \mathcal{C}^{xy}\left[\tfrac{1}{m+1}, \tfrac{1}{m}\ ;\ \tfrac{1}{m+1}, \tfrac{1}{m}\right], \qquad L_2 \leftarrow L \cap \mathcal{C}^{xy}\left[0, \tfrac{1}{m+1}\ ;\ \tfrac{1}{m+1}, \tfrac{1}{m}\right],$$
$$L_3 \leftarrow L \cap \mathcal{C}^{xy}\left[\tfrac{1}{m+1}, \tfrac{1}{m}\ ;\ 0, \tfrac{1}{m+1}\right], \qquad L_4 \leftarrow L \cap \mathcal{C}^{xy}\left[0, \tfrac{1}{m+1}\ ;\ \tfrac{1}{m+2}, \tfrac{1}{m+1}\right] \cap \mathcal{C}_y^x,$$
$$L_5 \leftarrow L \cap \mathcal{C}^{xy}\left[\tfrac{1}{m+2}, \tfrac{1}{m+1}\ ;\ 0, \tfrac{1}{m+1}\right] \cap \mathcal{C}_x^y, \qquad L_6 \leftarrow L \cap \mathcal{C}^{xy}\left[0, \tfrac{1}{m+2}\ ;\ 0, \tfrac{1}{m+2}\right].$$

2 $\mathcal{P}_i \leftarrow \mathrm{NFDH}^x(L_i), \quad i = 1, 2, 4;$

3 $\mathcal{P}_i \leftarrow \mathrm{NFDH}^y(L_i), \quad i = 3, 5.$

4 $\mathcal{P}_6 \leftarrow \mathrm{LL}_m(L_6).$

5 $\mathcal{P} \leftarrow \mathcal{P}_1 \| \ldots \| \mathcal{P}_6.$

6 return $\mathcal{P}$.

end algorithm.

Lemma 5.3 *For any list of boxes $L \subset \mathcal{C}_m^{xy}$, $m \geq 2$, where each box has height at most Z, we have*

$$\mathrm{BI}_m^{(t)}(L) \leq \left(\frac{m+1}{m}\right)^2 \frac{V(L)}{ab} + 6Z.$$

Proof. The proof is similar to the one given for Lemma 4.2. It follows from the fact that each partial packing $\mathcal{P}_i$ satisfies the conditions of Lemma 5.1 with volume guarantee at least $\left(\frac{m+1}{m}\right)^2$. □

Finally, we describe the algorithm COL, which is similar to the algorithm FFC for the problem $2\mathrm{BP}^{\mathrm{r}}$. This algorithm has parameters $(L, \mathcal{T}_1, \mathcal{T}_2, p_1, p_2)$, where each $\mathcal{T}_i$ is a set of boxes and p_i is a list of coordinates in the plane xy, where the boxes of type $\mathcal{T}_i$ can be packed, $i = 1, 2$. Boxes in the same coordinate $p \in p_i$ in the plane xy, are placed one on top of the other, generating a column at position p. At each iteration, the algorithm chooses a column with the smallest height, say a coordinate in p_i and packs the next box $b \in L_i$, of type $\mathcal{T}_i$, on the top of that column (if needed, rotate box b so as to have $b \in \mathcal{T}_i$). The process terminates when all boxes in L of type $\mathcal{T}_1$ or of type $\mathcal{T}_2$ are packed. The algorithm returns a pair $(\mathcal{P}, L')$ where $\mathcal{P}$ is the packing generated by the algorithm, and L' is the set of boxes in $L_1 \cup L_2$ that were packed in $\mathcal{P}$.

Algorithm TRI$_k$

Input: List of boxes L (an instance of $3\mathrm{SP}^{\mathrm{r}}(a, b)$).
Output: Packing $\mathcal{P}$ of L into $B = (a, b, \infty)$.

1 Rotate all possible boxes $e \in L \cap \wp_4$ to an orientation $e' \in \wp_2 \cup \wp_3$.

2 Rotate all possible boxes $e \in L \cap (\wp_2 \cup \wp_3)$ to an orientation $e' \in \wp_1$.

3 $\mathcal{P}_{AB} \leftarrow \mathrm{COMBINE\!-\!AB}_k^{xy}(L, \mathrm{rtype}\,, \mathcal{C}_1^{xy}, \mathcal{C}_1^{xy}, \mathrm{COL})$.

4 UPDATE (L).

5 If all boxes of rtype$(L, \mathcal{B}_k^{xy})$ have been packed then

5.1 Rotate all possible boxes $e \in L \cap \wp_2$ to a box $e' \in \wp_3$.

5.2 Rotate each box $e \in \wp_4$ so as to have $e \in \wp_4$ and $z(e)$ minimum.

5.3 Let

$$t \leftarrow 17/36\ ,$$
$$\mathcal{T}_C = \mathcal{C}^{xy}\left[\tfrac{1}{2}, 1\ ;\ \tfrac{1}{2}, 1-t\right], \quad \mathcal{T}'_D = \mathcal{C}^{xy}\left[0, t\ ;\ 0, 1\right],$$
$$\mathcal{T}''_D = \mathcal{C}^{xy}\left[0, t\ ;\ 0, 1\right], \quad \mathcal{T}_D = \mathcal{T}'_D \bigcup \mathcal{T}''_D,$$
$$L_C = \mathrm{rtype}(L, \mathcal{T}_C), \quad L_D = \mathrm{rtype}(L, \mathcal{T}_D).$$

5.4 Generate a packing $\mathcal{P}_{CD}$ as follows.

$(\mathcal{P}_{CD'}, L_{CD'}) \leftarrow \mathrm{COL}(L, \mathcal{T}_C, \mathcal{T}'_D, [(0,0)], [(0, 1-t)])$.
$(\mathcal{P}_{CD''}, L_{CD''}) \leftarrow \mathrm{COL}(L, \mathcal{T}_C \setminus L_{CD'}, \mathcal{T}''_D, [(0,0)], [(0, 1-t), (\frac{1}{2}, 1-t)])$.
$\mathcal{P}_{CD} \leftarrow \mathcal{P}_{CD'} \| \mathcal{P}_{CD''}$.
$L_{CD} \leftarrow L_{CD'} \bigcup L_{CD''}$. /* L_C or L_D is totally packed in L_{CD}. */
UPDATE(L).

5.5 Partition the list L into sublists $L_1, \ldots, L_{23}$ as follows (see figure 3).

$$L_i \leftarrow L \bigcap \mathcal{C}^{xy}\left[\tfrac{1}{2}, 1\ ;\ \tfrac{1}{i+2}, \tfrac{1}{i+1}\right],\ i = 1, \ldots, 16, \qquad L_{17} \leftarrow L \bigcap \mathcal{C}^{xy}\left[\tfrac{1}{2}, 1\ ;\ 0, \tfrac{1}{18}\right],$$
$$L_{18} \leftarrow L \bigcap \mathcal{C}^{xy}\left[\tfrac{1}{3}, \tfrac{1}{2}\ ;\ \tfrac{1}{3}, \tfrac{1}{2}\right], \qquad L_{19} \leftarrow L \bigcap \mathcal{C}^{xy}\left[\tfrac{1}{3}, \tfrac{1}{2}\ ;\ \tfrac{1}{4}, \tfrac{1}{3}\right],$$
$$L_{20} \leftarrow L \bigcap \mathcal{C}^{xy}\left[\tfrac{1}{3}, \tfrac{1}{2}\ ;\ 0, \tfrac{1}{4}\right], \qquad L_{21} \leftarrow L \bigcap \mathcal{C}^{xy}\left[\tfrac{1}{4}, \tfrac{1}{3}\ ;\ \tfrac{1}{3}, \tfrac{1}{2}\right],$$
$$L_{22} \leftarrow L \bigcap \mathcal{C}^{xy}\left[0, \tfrac{1}{4}\ ;\ \tfrac{1}{3}, \tfrac{1}{2}\right], \qquad L_{23} \leftarrow L \bigcap \mathcal{C}^{xy}\left[0, \tfrac{1}{3}\ ;\ 0, \tfrac{1}{3}\right].$$

5.6 Generate packings $\mathcal{P}_1, \ldots, \mathcal{P}_{23}$ as follows.

$\mathcal{P}_i \leftarrow \mathrm{NFDH}^y(L_i)$, for $i = 1, \ldots, 21$.
$\mathcal{P}_{22} \leftarrow \mathrm{NFDH}^x(L_{22})$.
$\mathcal{P}_{23} \leftarrow \mathrm{BI}_3^{(t)}(L_{23})$.

5.7 UPDATE (L). /* Note that $L \subseteq \wp_2 \cup \wp_4$. */

5.8 If $L_C \subseteq L_{CD}$

then /* (Case 1) L_C is totally packed, see figure 4, Cases 1.1 and 1.2.*/

$$p \leftarrow \left(-36 + 3\sqrt{6} + \sqrt{11078 + 2216\sqrt{6}}\right) / \left(38\ \left(4 + \sqrt{6}\right)\right)$$

else /* (Case 2) $L_D \subseteq L_{CD}$ is totally packed see figure 4, Cases 2.1 and 2.2.*/

$$p \leftarrow \left(-72 + \sqrt{6} + \sqrt{6\ \left(3313 + 588\sqrt{6}\right)}\right) / \left(36\ \left(4 + \sqrt{6}\right)\right).$$

5.9 Generate a packing $\mathcal{P}_{EF}$ as follows.

$$\mathcal{T}_E \leftarrow \mathcal{C}^{xy}\left[\tfrac{1}{2}, 1-p\ ;\ \tfrac{1}{2}, 1\right], \quad \mathcal{T}'_F \leftarrow \mathcal{C}^{xy}\left[\tfrac{1}{7}, p\ ;\ \tfrac{1}{2}, 1\right],$$
$$\mathcal{T}''_F \leftarrow \mathcal{C}^{xy}\left[\tfrac{1}{18}, \tfrac{1}{7}\ ;\ \tfrac{1}{2}, 1\right], \quad \mathcal{T}_F \leftarrow \mathcal{T}'_F \cup \mathcal{T}''_F,$$
$$L_E \leftarrow \mathrm{rtype}(L, \mathcal{T}_E), \quad L_F \leftarrow \mathrm{rtype}(L, \mathcal{T}_F).$$

$(\mathcal{P}_{EF'}, L_{EF'}) \leftarrow \mathrm{COL}(L, \mathcal{T}_E, \mathcal{T}'_F, [(0,0)], [(1-p, 0)])$.
$(\mathcal{P}_{EF''}, L_{EF''}) \leftarrow \mathrm{COL}(L, \mathcal{T}_E \setminus L_{EF'}, \mathcal{T}''_F, [(0,0)], [(0, 1-p), (0, 1-p+\frac{1}{7}), \ldots, (0, 1-p+(\lfloor\frac{p}{1/7}\rfloor - 1)\frac{1}{7})])$.
$\mathcal{P}_{EF} \leftarrow \mathcal{P}_{EF'} \| \mathcal{P}_{EF''}$.
$L_{EF} \leftarrow L_{EF'} \cup L_{EF''}$. /* L_E or L_F is totally packed in L_{EF}. */

5.10 If $L_E \subseteq L_{EF}$ /* (Subcase 1) L_E is totally packed */

then

Rotate (if possible) each box $e \in L \cap (\wp_2 \cup \wp_4)$ to a box $e' \in \mathcal{C}_1^{xy}$ so as to have $x(e')z(e')$ minimum.

Considering the face of each box in the plane xz as a rectangle, use the algorithm SPR to generate a packing $\mathcal{P}_{\mathrm{SPR}}$ of L.

$\mathcal{P}_{OC} \leftarrow \mathrm{NFDH}^x((L \setminus L_{EF}) \cap \wp_4)$.

$\mathcal{P}_{2e} \leftarrow \mathrm{NFDH}^x((L \setminus L_{EF}) \cap \mathcal{C}^{xy}\left[0, \frac{1}{3}\ ;\ \frac{1}{2}, 1\right])$.

$\mathcal{P}_{2d} \leftarrow \mathrm{NFDH}^x((L \setminus L_{EF}) \cap \mathcal{C}^{xy}\left[\frac{1}{3}, \frac{1}{2}\ ;\ \frac{1}{2}, 1\right])$.

$\mathcal{P}' \leftarrow \mathcal{P}_{OC} \| \mathcal{P}_{2e} \| \mathcal{P}_{2d} \| \mathcal{P}_{EF}$.

$\mathcal{P}'' \leftarrow$ (a packing $\mathcal{P} \in \{\mathcal{P}_{\mathrm{SPR}}, \mathcal{P}'\}$ such that $H(\mathcal{P})$ is minimum).

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

Let L'' and L_{aux} be the lists of boxes packed in $\mathcal{P}''$ and $\mathcal{P}_{aux}$, resp.

$\mathcal{P} \leftarrow \mathcal{P}_{aux} \| \mathcal{P}''$.

5.11 If $L_F \subseteq L_{EF}$ /* (Subcase 2) L_F is totally packed */

then

$\mathcal{P}_{OC} \leftarrow \mathrm{NFDH}^x((L \setminus L_{EF}) \cap \wp_4)$.

$\mathcal{P}_{2e} \leftarrow \mathrm{NFDH}^x((L \setminus L_{EF}) \cap \mathcal{C}^{xy}\left[0, \frac{1}{18}\ ;\ \frac{1}{2}, 1\right])$.

$\mathcal{P}_{2d} \leftarrow \mathrm{NFDH}^x((L \setminus L_{EF}) \cap \mathcal{C}^{xy}\left[p, \frac{1}{2}\ ;\ \frac{1}{2}, 1\right])$.

$\mathcal{P}' \leftarrow \mathcal{P}_{OC} \| \mathcal{P}_{EF}$.

$\mathcal{P}_{aux} \leftarrow \mathcal{P}_{AB} \| \mathcal{P}_{CD} \| \mathcal{P}_{2e} \| \mathcal{P}_{2d} \| \mathcal{P}_1 \| \ldots \| \mathcal{P}_{23}$.

Let L' and L_{aux} be the lists of boxes packed in $\mathcal{P}'$ and $\mathcal{P}_{aux}$, resp.

$\mathcal{P} \leftarrow \mathcal{P}_{aux} \| \mathcal{P}'$.

6 If all boxes of rtype$(L, \mathcal{A}_k^{xy})$ have been packed then generate a packing $\mathcal{P}$ of L as in step 5 (in a symmetric way).

7 Return $\mathcal{P}$.

end algorithm.

The next theorem gives an asymptotic performance bound of the algorithm TRI_k when $k \to \infty$.

Theorem 5.4 *For any instance L for the problem* $3\mathrm{SP}^{\mathrm{r}}$ *we have*

$$\mathrm{TRI}_k(L) \leq \alpha_k \mathrm{OPT}(L) + \beta_k Z\ ,$$

where $\lim_{k\to\infty} \alpha_k \leq 2.76$ *and* $\beta_k = \mathcal{O}(k)$.

Proof. We present the proof for the case in which all boxes of rtype$(L, \mathcal{B}_k^{xy})$ have been packed (see step 5). The proof for the other case (step 6) is analogous. We consider 4 cases, according to step 5.8 ($L_C \subseteq L_{CD}$) or ($L_D \subseteq L_{CD}$), step 5.10 ($L_E \subseteq L_{EF}$) and step 5.11 ($L_F \subseteq L_{EF}$).

Since many steps of the algorithm TRI_k are similar to the ones of the algorithm $\mathcal{A}_k$, presented in [17] for the problem 3SP, many of the inequalities obtained in the analysis of $\mathcal{A}_k$ are valid, mainly the ones that present volume guarantee. We present the inequalities we need, but omit the proofs, as they can be found in [17].

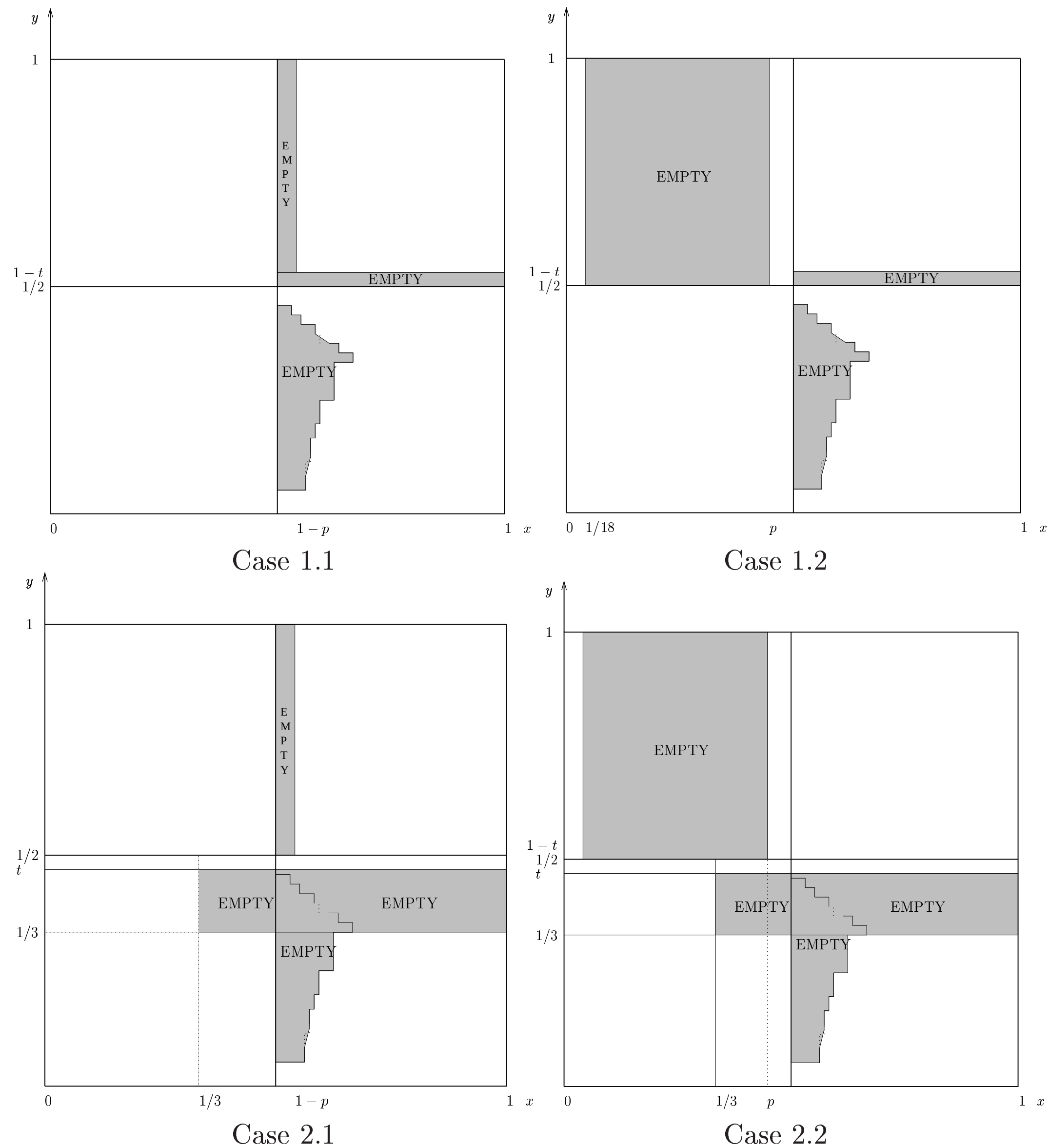

Figure 4: The four cases in the proof of Theorem 5.4, after packing the critical sets.

Claim:

$$H(\mathcal{P}_i) \quad \leq \quad \frac{1}{17/36}\frac{V(L_i)}{ab} + Z \quad \text{for} \quad i \in \{1, \ldots, 22\} \setminus \{1, 18\} \ , \tag{25}$$

$$H(\mathcal{P}_{23}) \quad \leq \quad \frac{1}{17/36}\frac{V(L_{23})}{ab} + 3Z \ , \tag{26}$$

$$H(\mathcal{P}_{AB}) \quad \leq \quad \frac{1}{17/36}\frac{V(L_{AB})}{ab} + (2k + 41)Z \ , \tag{27}$$

$$
\begin{aligned}
H(\mathcal{P}_{CD}) &\leq \frac{1}{\left(\frac{1}{4}+\frac{r_1}{2}\right)}\frac{V(L_{CD})}{ab}+2Z\ , && (28)\\
H(\mathcal{P}_{EF}) &\leq \frac{1}{1/4+1/14}\frac{V(L_{EF})}{ab}+2Z\ . && (29)
\end{aligned}
$$

To distinguish the value of p in cases 1 and 2, we denote by p_1 and p_2 the values of p in these cases, respectively.

Case 1.1 ($L_C \subseteq L_{CD}$) and ($L_E \subseteq L_{EF}$).

In this case, the following inequalities hold:

$$
\begin{aligned}
H(\mathcal{P}_{OC}) &\leq \frac{1}{(1-p_1)(1-t)}\frac{V(L_{OC})}{ab}, && (30)\\
H(\mathcal{P}_{2e}\|\mathcal{P}_{2d}) &\leq \frac{1}{1/3}\frac{V(L_{2e}\cup L_{2d})}{ab}+2Z, && (31)\\
H(\mathcal{P}_1) &\leq \frac{1}{r_1}\frac{V(L_1)}{ab}+Z, && (32)\\
H(\mathcal{P}_{18}) &\leq \frac{1}{4/9}\frac{V(L_{18})}{ab}+Z. && (33)
\end{aligned}
$$

Since $(1-p_1)(1-t) \leq \min\{1/3, 1/4+1/14\}$, and $\mathcal{P}' = \mathcal{P}_{OC}\|\mathcal{P}_{2e}\|\mathcal{P}_{2d}\|\mathcal{P}_{EF}$, we have

$$
H(\mathcal{P}'') \leq H(\mathcal{P}') \leq \frac{1}{(1-p_1)(1-t)}\frac{V(L')}{ab}+4Z\ . \quad (34)
$$

Considering packing $\mathcal{P}_{\mathrm{SPR}}$, from Theorem 3.5 and the fact that $H(\mathcal{P}'') = \min\{H(\mathcal{P}'), H(\mathcal{P}_{\mathrm{SPR}})\}$, we have

$$
H(\mathcal{P}'') \leq H(\mathcal{P}_{\mathrm{SPR}}) \leq \alpha_{\mathrm{SPR}}\,\mathrm{OPT}(L'')+3Z\ , \quad (35)
$$

where $\alpha_{\mathrm{SPR}} = 1+\frac{\sqrt{6}}{4}$.

Since $r_1 \leq \min\{4/9, 17/36, 1/4+r_1/2\}$, and $\mathcal{P}_{aux} = \mathcal{P}_{AB}\|\mathcal{P}_{CD}\|\mathcal{P}_1\|\ldots\|\mathcal{P}_{23}$, we have

$$
H(\mathcal{P}_{aux}) \leq \frac{1}{r_1}\frac{V(L_{aux})}{ab}+(2k+68)Z\ . \quad (36)
$$

From inequalities (34), (36) and (35) we have

$$
H(\mathcal{P}) \leq \alpha'_k(r_1)\,\mathrm{OPT}(L)+(2k+72)Z,
$$

where $\alpha'_k(r_1) = (h_1+h_2)/\max\{\frac{1}{\alpha_{\mathrm{SPR}}}h_1, (1-p)(1-t)h_1+r_1h_2\}$.

Case 1.2 ($L_C \subseteq L_{CD}$) and ($L_F \subseteq L_{EF}$).

In this case, the following inequalities hold:

$$
\begin{aligned}
H(\mathcal{P}_{OC}) &\leq \frac{1}{(1-t)/2}\frac{V(L_{OC})}{ab}, && (37)\\
H(\mathcal{P}_{2e}\|\mathcal{P}_{2d}) &\leq \frac{1}{p_1}\frac{V(L_{2e}\cup L_{2d})}{ab}+2Z, && (38)\\
H(\mathcal{P}_1) &\leq \frac{1}{r_1}\frac{V(L_1)}{ab}+Z, && (39)\\
H(\mathcal{P}_{18}) &\leq \frac{1}{4/9}\frac{V(L_{18})}{ab}+Z. && (40)
\end{aligned}
$$

Since $(1-p_1)/2 \leq 1/4+1/14$, and $\mathcal{P}' = \mathcal{P}_{OC} \| \mathcal{P}_{EF}$, we have

$$H(\mathcal{P}') \quad \leq \quad \frac{1}{(1-p_1)/2} \frac{V(L')}{ab} + 4Z \ . \tag{41}$$

Moreover, from steps 1, 2, 5.1 and 5.2 the packing of $\mathcal{P}'$ has a column of *big* boxes of $\wp_4$ for which the only possible packing is to place one box on top of the other. Therefore,

$$H(\mathcal{P}') \leq \mathrm{OPT}(L') + 2Z \ . \tag{42}$$

Since $p_1 \leq \min\{r_1, 4/9, 17/36, 1/4 + r_1/2\}$, and $\mathcal{P}_{aux} = \mathcal{P}_{AB} \| \mathcal{P}_{CD} \| \mathcal{P}_{2e} \| \mathcal{P}_{2d} \| \mathcal{P}_1 \| \ldots \| \mathcal{P}_{23}$, we have

$$H(\mathcal{P}_{aux}) \quad \leq \quad \frac{1}{p_1} \frac{V(L_{aux})}{ab} + (2k+70)Z \ . \tag{43}$$

From inequalities (41)–(43) we have

$$H(\mathcal{P}) \leq \alpha''_k(r_1)\,\mathrm{OPT}(L) + (2k+74)Z,$$

where $\alpha''_k(r_1) = (h_1+h_2)/\max\{h_1, (1-p_1)/2h_1 + p_1 h_2\}$.
Case 2.1 ($L_D \subseteq L_{CD}$) and ($L_E \subseteq L_{EF}$).

In this case, the following inequalities hold:

$$H(\mathcal{P}_{OC}) \quad \leq \quad \frac{1}{(1-p_2)/2} \frac{V(L_{OC})}{ab}, \tag{44}$$

$$H(\mathcal{P}_{2e} \| \mathcal{P}_{2d}) \quad \leq \quad \frac{1}{1/3} \frac{V(L_{2e} \cup L_{2d})}{ab} + 2Z, \tag{45}$$

$$H(\mathcal{P}_1) \quad \leq \quad \frac{1}{t} \frac{V(L_1)}{ab} + Z, \tag{46}$$

$$H(\mathcal{P}_{18}) \quad \leq \quad \frac{1}{4t/3} \frac{V(L_{18})}{ab} + Z. \tag{47}$$

Since $(1-p_2)/2 \leq \min\{1/3, 1/4+1/14\}$, and $\mathcal{P}' = \mathcal{P}_{OC} \| \mathcal{P}_{2e} \| \mathcal{P}_{2d} \| \mathcal{P}_{EF}$, we have

$$H(\mathcal{P}') \quad \leq \quad \frac{1}{(1-p_2)/2} \frac{V(L')}{ab} + 4Z \ . \tag{48}$$

Considering packing $\mathcal{P}_{\mathrm{SPR}}$, from Theorem 3.5 and the fact that $H(\mathcal{P}'') = \min\{H(\mathcal{P}'), H(\mathcal{P}_{\mathrm{SPR}})\}$, we have

$$H(\mathcal{P}'') \leq \alpha_{\mathrm{SPR}}\,\mathrm{OPT}(L'') + 3Z \ . \tag{49}$$

Since $1/4 + r_1/2 \leq \min\{4t/3, t, 17/36\}$, and $\mathcal{P}_{aux} = \mathcal{P}_{AB} \| \mathcal{P}_{CD} \| \mathcal{P}_1 \| \ldots \| \mathcal{P}_{23}$, we have

$$H(\mathcal{P}_{aux}) \quad \leq \quad \frac{1}{1/4 + r_1/2} \frac{V(L_{aux})}{ab} + (2k+68)Z. \tag{50}$$

From inequalities (48), (50) and (49) we have

$$H(\mathcal{P}) \leq \beta'_k(r_1)\mathrm{OPT}(L) + (2k+72)Z,$$

where $\beta'_k(r_1) = (h_1 + h_2)/\max\{\frac{1}{\alpha_{\text{SPR}}}\, h_1, (1-p_2)/2h_1 + (1/4 + r_1/2)h_2\}$.

Case 2.2 ($L_D \subseteq L_{CD}$) and ($L_F \subseteq L_{EF}$).

In this case, the following inequalities are valid:

$$H(\mathcal{P}_{OC}) \leq \frac{1}{1/4}\frac{V(L_{OC})}{ab}, \tag{51}$$

$$H(\mathcal{P}_{2e}\|\mathcal{P}_{2d}) \leq \frac{1}{p_2}\frac{V(L_{2e} \cup L_{2d})}{ab} + 2Z, \tag{52}$$

$$H(\mathcal{P}_1) \leq \frac{1}{t}\frac{V(L_1)}{ab} + Z, \tag{53}$$

$$H(\mathcal{P}_{18}) \leq \frac{1}{4t/3}\frac{V(L_{18})}{ab} + Z. \tag{54}$$

Since $1/4 \leq 1/4 + 1/14$, and $\mathcal{P}' = \mathcal{P}_{OC}\|\mathcal{P}_{EF}$, we have

$$H(\mathcal{P}') \leq \frac{1}{1/4}\frac{V(L')}{ab} + 4Z\ . \tag{55}$$

Moreover, from steps 1, 2, 5.1 and 5.2 the packing of $\mathcal{P}'$ has a column of *big* boxes of $\wp_4$, where the only possible packing is to place one box over the other. Therefore,

$$H(\mathcal{P}') \leq \text{OPT}(L') + 2Z\ . \tag{56}$$

Since $p_2 = \min\{4t/3, t, 17/36, 1/4 + r_1/2\}$, and $\mathcal{P}_{aux} = \mathcal{P}_{AB}\|\mathcal{P}_{CD}\|\mathcal{P}_{2e}\|\mathcal{P}_{2d}\|\mathcal{P}_1\| \ldots \|\mathcal{P}_{23}$, we have

$$H(\mathcal{P}_{aux}) \leq \frac{1}{p_2}\frac{V(L_{aux})}{ab} + (2k + 70)Z\ . \tag{57}$$

From inequalities (55)–(57) we have

$$H(\mathcal{P}) \leq \beta''_k(r_1)\,\text{OPT}(L) + (2k + 74)\ Z\ ,$$

where $\beta''_k(r_1) = (h_1 + h_2)/\max\{h_1, (1/4)h_1 + p_2h_2\}$.

Now let $\alpha_k(r_1) := \max\{\alpha'_k(r_1), \alpha''_k(r_1)\}$ and $\beta_k(r_1) := \max\{\beta'_k(r_1), \beta''_k(r_1)\}$, where $\alpha'_k(r_1)$, $\alpha''_k(r_1)$, $\beta'_k(r_1)$ and $\beta''_k(r_1)$ are the values obtained in the cases 1.1, 1.2, 2.1 and 2.2.

Since $r_1 \to \frac{4}{9}$ as $k \to \infty$, applying Lemma 3.4 we can conclude that $\alpha'_k(r_1) \to 2.727558\ldots$ and $\beta''_k(r_1) \to 2.753151\ldots$. This completes the proof of the theorem. □

6 z-Oriented Three-Dimensional Strip Packing Problem

In this section we consider the problem 3SP^z, for which we exhibit an algorithm with asymptotic performance bound 2.64 *.

*In [19] this problem is called "z-oriented three-dimensional packing problem". We considered convenient to add the term "strip" to differentiate it from the bin packing version.

The algorithm follows the algorithm $\mathrm{BI}_{k,\epsilon}$, presented for the two-dimensional bin packing problem. It uses the algorithm COL^z, which is similar to the algorithm COL, except that the boxes are considered with the function xy-type, instead of the function rtype.

Algorithm $\mathrm{TRI}^z_{k,\epsilon}(L)$

Input: List of boxes L (instance of $3\mathrm{SP}^z(a,b)$).

Output: Packing $\mathcal{P}$ of L into a bin $B=(a,b,\infty)$, allowing orthogonal rotations around the z axis.

1 Rotate all boxes $r \in \wp_4$ where $\rho(r) \in \wp_1 \cup \wp_2 \cup \wp_3$.
/* Let $R_1 \leftarrow \{r \in L \cap \wp_4 : \rho(r) \in \wp_1 \cup \wp_2 \cup \wp_3\}$; $L \leftarrow (L \setminus R_1) \bigcup \rho(R_1)$. */

2 $t \leftarrow (\sqrt{33}-3)/6$.

3 $\mathcal{P}_{AB} \leftarrow \mathrm{COMBINE\!-\!AB}^{xy}_k(L, xy\text{-type}, \wp_4, \wp_4, \mathrm{COL}^z)$.
UPDATE (L).

4 If all boxes of xy-type$(L, \mathcal{B}^{xy}_k)$ were packed then

4.1 Rotate the boxes of $L \cap \wp_2$ that fits in $\wp_1 \cup \wp_3$.

4.2 Rotate the boxes of $L \cap (\wp_2 \cup \wp_4)$ in such that if $e \in L \cap (\wp_2 \cup \wp_4)$ then $x(e) \le y(e)$ or $\rho(e) \notin \mathcal{C}_1$.

4.3 Let

$$\mathcal{T}_C = \mathcal{C}^{xy}\left[\tfrac{1}{2},1\ ;\ \tfrac{1}{2},1-t\right],\quad \mathcal{T}'_D = \mathcal{C}^{xy}\,[0,t\ ;\ 0,1],$$
$$\mathcal{T}''_D = \mathcal{C}^{xy}\,[0,t\ ;\ 0,1],\qquad \mathcal{T}_D = \mathcal{T}'_D \bigcup \mathcal{T}''_D.$$

4.4 Generate packing $\mathcal{P}_{CD}$ as follows.
$(\mathcal{P}_{CD'}, L_{CD'}) \leftarrow \mathrm{COL}^z(L, \mathcal{T}_C, \mathcal{T}'_D, [(0,0)], [(0,1-t)])$.
$(\mathcal{P}_{CD''}, L_{CD''}) \leftarrow \mathrm{COL}^z(L, \mathcal{T}_C \setminus L_{CD'}, \mathcal{T}''_D, [(0,0)], [(0,1-t), (\tfrac{1}{2}, 1-t)])$.
$\mathcal{P}_{CD} \leftarrow \mathcal{P}_{CD'} \| \mathcal{P}_{CD''}$;
$L_{CD} \leftarrow L_{CD'} \bigcup L_{CD''}$;

4.5 Subdivide the list L in sublists $L_1, \ldots, L_{23}$ as follows (see figure 3).

$$L_i \leftarrow L \bigcap \mathcal{C}^{xy}\left[\tfrac{1}{2},1\ ;\ \tfrac{1}{i+2},\tfrac{1}{i+1}\right],\ i=1,\ldots,16, \qquad L_{17} \leftarrow L \bigcap \mathcal{C}^{xy}\left[\tfrac{1}{2},1\ ;\ 0,\tfrac{1}{18}\right],$$
$$L_{18} \leftarrow L \bigcap \mathcal{C}^{xy}\left[\tfrac{1}{3},\tfrac{1}{2}\ ;\ \tfrac{1}{3},\tfrac{1}{2}\right], \qquad L_{19} \leftarrow L \bigcap \mathcal{C}^{xy}\left[\tfrac{1}{3},\tfrac{1}{2}\ ;\ \tfrac{1}{4},\tfrac{1}{3}\right],$$
$$L_{20} \leftarrow L \bigcap \mathcal{C}^{xy}\left[\tfrac{1}{3},\tfrac{1}{2}\ ;\ 0,\tfrac{1}{4}\right], \qquad L_{21} \leftarrow L \bigcap \mathcal{C}^{xy}\left[\tfrac{1}{4},\tfrac{1}{3}\ ;\ \tfrac{1}{3},\tfrac{1}{2}\right],$$
$$L_{22} \leftarrow L \bigcap \mathcal{C}^{xy}\left[0,\tfrac{1}{4}\ ;\ \tfrac{1}{3},\tfrac{1}{2}\right], \qquad L_{23} \leftarrow L \bigcap \mathcal{C}^{xy}\left[0,\tfrac{1}{3}\ ;\ 0,\tfrac{1}{3}\right].$$

4.6 Generate packings $\mathcal{P}_1, \ldots, \mathcal{P}_{23}$ as follows.
$\mathcal{P}_i \leftarrow \mathrm{NFDH}^y(L_i)$ for $i = 1, \ldots, 21$;
$\mathcal{P}_i \leftarrow \mathrm{NFDH}^x(L_i)$ for $i = 22$;
$\mathcal{P}_{23} \leftarrow \mathrm{BI}^{(t)}_3(L_{23})$.

4.7 UPDATE (L). /* Note that $L \subseteq \wp_2 \cup \wp_4$. */

4.8 Considering each box of L as an rectangle of length $x(r)$ and height $y(r)$ and the box $B=(a,b,\infty)$ as a rectangular strip of length a and unlimited height. Apply the algorithm of KR_ϵ in L; Let $\mathcal{P}_{\mathrm{KR}}$ be this packing. Let $\mathcal{P}_{\mathrm{NFDH}}$ be the packing

$\text{NFDH}^x(L \cap \mathcal{X}[0, \frac{1}{3}]) \| \text{NFDH}^x(L \cap \mathcal{X}[\frac{1}{3}, \frac{1}{2}]) \| \text{NFDH}^x(L \cap \mathcal{X}[\frac{1}{2}, 1])$.
Let $\mathcal{P}_{\text{STRIP}}$ be the smallest packing in $\{\mathcal{P}_{\text{KR}}, \mathcal{P}_{\text{NFDH}}\}$.

4.9 $\mathcal{P}_{aux} \leftarrow \mathcal{P}_{AB} \| \mathcal{P}_{CD} \| \mathcal{P}_1 \| \ldots \| \mathcal{P}_{23}$;

4.10 $\mathcal{P} \leftarrow \mathcal{P}_{\text{STRIP}} \| \mathcal{P}_{aux}$.

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

6 Return $\mathcal{P}$.

End algorithm.

Theorem 6.1 *For any instance L for the problem* 3SP^z, *we have*

$$\text{TRI}^z_{k,\epsilon}(L) \leq \alpha_{k,\epsilon} \text{OPT}(L) + \mathcal{O}\left(k + \frac{1}{\epsilon}\right) Z,$$

where $\alpha_{k,\epsilon} \to (25 + 3\sqrt{33})/16 = 2.639\ldots$ *as* $k \to \infty$ *and* $\epsilon \to 0$.

Proof. The proof follows the one given for Theorem 4.8 with minor alterations. The inequalities involving area are substituted by inequalities involving volume and the additive constants are multiplied by the value Z. The inequality analogous to (13), which followed from Theorem 2.2, follows now from Theorem 3.1. □

7 Three-Dimensional Bin Packing Problem

In this section, we consider the three-dimensional bin packing problem with rotation (3BP^r). For the problem 3BP, the best performance bound known is 4.84, of algorithms presented by Li and Cheng [14] and Csirik and van Vliet [9].

We present for 3BP^r an algorithm whose asymptotic performance bound may converge to a value smaller than 4.89. We denote the algorithm of this section by $\text{BOX}_{k,\epsilon}$. Before presenting the algorithm, we need some procedures used as subroutines. We use the same scheme of the algorithm FFC used for 2BP^r. For that, we first modify the algorithm FFC for the bin packing version. We denote this algorithm as FFC^{xy} (First Fit COLUMN for x and y axes). The algorithm FFC^{xy} combines the strategy of the algorithm FFC with the strategy of the algorithm FF (First Fit) to pack boxes into columns.

The input parameters are: a list of boxes L, two set of boxes $\mathcal{T}_1$ and $\mathcal{T}_2$ and two coordinate lists p_1 and p_2 associated with these sets. Each column starts at the bottom of a box B in a coordinate $p \in p_1 \cup p_2$. The columns located in coordinates of list $[p_i]$ have only boxes of type $\mathcal{T}_i$, $i = 1, 2$ and start in the plane xy growing in the direction of the z-axis.

Algorithm FFC^{xy}

Input: $(L, \mathcal{T}_1, \mathcal{T}_2, p_1, p_2)$ // each p_i is a list of coordinates in the plane xy.

Output: Partial packing of L into B, in which or all boxes of type $\mathcal{T}_1$ or all boxes of type $\mathcal{T}_2$ are totally packed.

1 While there are non-packed boxes of type $\mathcal{T}_1$ and $\mathcal{T}_2$ do

1.1 Let $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_i$ be the packings in the bins $B_1, \ldots, B_i$, respectively, generated so far.

1.2 Take the next box e' of type $\mathcal{T}_1$. If possible, pack e' in a column of boxes corresponding to $\mathcal{T}_1$ in $\mathcal{P}_1, \ldots, \mathcal{P}_i$, without violating the limits of the corresponding bin. If necessary, rotate the box e' so as to have $e' \in \mathcal{T}_1$.

1.3 If it is not possible to pack a box in step 1.2, pack (if possible) the next box e'', of type $\mathcal{T}_2$, using the same strategy used in step 1.2, but with columns of boxes of type $\mathcal{T}_2$.

1.4 If no packing was possible in steps 1.2 and 1.3, generate a new empty packing $\mathcal{P}_{i+1}$ (that starts with empty columns in positions $p_1 \cup p_2$) in a new bin B_{i+1}.

2 Return $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_i$.

end algorithm.

Another algorithm used as subroutine is the algorithm H3D *(Hybrid 3D)*. This algorithm uses the same strategy used in the algorithm HFF (*Hybrid First Fit*) presented by Chung, Garey and Johnson [4]. The algorithm H3D generate a packing in two main steps. First it generates a three-dimensional strip packing of L, subdivided in levels, and then packs the levels into bins, using a one-dimensional bin packing algorithm.

Algorithm $\mathrm{H3D}^z(L, \mathcal{A}_{\mathrm{TPP}}, \mathcal{A}_{\mathrm{UNI}})$

Input: List of boxes L (instance of $3\mathrm{BP}^{\mathrm{r}}(a, b, c)$).

Output: Packing of L into bins $B = (a, b, c)$.

Subroutines: An algorithm $\mathcal{A}_{\mathrm{TPP}}$ for the $3\mathrm{SP}^{\mathrm{r}}$, that generates a level oriented packing, and an algorithm $\mathcal{A}_{\mathrm{UNI}}$ for the one-dimensional bin packing problem.

1 $\mathcal{P} \leftarrow \mathcal{A}_{\mathrm{TPP}}(L)$.

2 Let $\mathcal{N}$ be the set of levels in $\mathcal{P}$.

3 Apply algorithm $\mathcal{A}_{\mathrm{UNI}}$ to pack the levels $\mathcal{N}$ into bins B. Each level $N \in \mathcal{N}$, of height z_N, is seen as an one-dimensional item of height z_N, and each bin B is seen as an one-dimensional bin of height c. Let $\mathcal{P}_{\mathrm{H3D}}$ be the resulting packing.

4 Return $\mathcal{P}_{\mathrm{H3D}}$.

end algorithm.

We denote by $\mathrm{H3D}^x$ and $\mathrm{H3D}^y$ the variants of this algorithm where the generation of levels is done in the x and y direction, respectively.

Depending on the algorithms used as subroutines, the resulting algorithm H3D can generate good packings for special instances of $3\mathrm{BP}^{\mathrm{r}}$. Two of these algorithms are the algorithm $\mathrm{BI}_m^{(t)}$, we have presented in section 5, and the algorithm NFDH, presented by Li and Cheng [15].

Denote by $\mathrm{H3D}_m$ the algorithm $\mathrm{H3D}^z$, where $\mathcal{A}_{\mathrm{TPP}} = \mathrm{BI}_m^{(t)}$ and $\mathcal{A}_{\mathrm{UNI}} = \mathrm{FFD}$. The following results can be easily proved for the algorithm $\mathrm{H3D}_m$.

Lemma 7.1 *If $L \subseteq \mathcal{C}_m$ then*

$$\mathrm{H3D}_m(L) \leq \left(\frac{m+1}{m}\right)^3 \frac{V(L)}{abc} + 14.$$

Proof. First consider the bins that have levels with heigth in $\mathcal{Z}[\frac{1}{m+1}, \frac{1}{m}]$. Let $\mathcal{P}'$ and L' be the packing and the list of boxes in these bins, respectively. The algorithm FFD packs m levels in each bin, except perhaps in the last. Since each level has an area guarantee of at least $\left(\frac{m}{m+1}\right)^2$, except perhaps in 6 levels, we have a volume guarantee in each bin of at least $\left(\frac{m}{m+1}\right)^3$, except perhaps in 7 bins. Therefore,

$$\#(\mathcal{P}') \leq \left(\frac{m+1}{m}\right)^3 \frac{V(L')}{abc} + 7. \tag{58}$$

Denote by L'' and $\mathcal{P}''$ the set of remaining boxes and the packing in the remaining bins. All bins in $\mathcal{P}''$ have been filled with levels up to the height $\left(1 - \frac{1}{m+1}\right) c$, except perhaps the last. Therefore,

$$\mathrm{BI}_m^{(t)}(L'') \geq (\#(\mathcal{P}'') - 1)\left(1 - \frac{1}{m+1}\right) c.$$

Since $\mathrm{BI}_m^{(t)}(L'') \leq \left(\frac{m+1}{m}\right)^2 \frac{V(L'')}{ab} + 6Z$, and $Z \leq \frac{c}{m+1}$, we have

$$\left(\frac{m+1}{m}\right)^2 \frac{V(L'')}{ab} + \frac{6c}{m+1} \geq \mathrm{BI}_m^{(t)}(L'') \geq (\#(\mathcal{P}'') - 1)\left(1 - \frac{1}{m+1}\right) c,$$

and therefore,

$$\begin{aligned}
\#(\mathcal{P}'') - 1 \quad &\leq \quad \left(\frac{m+1}{m}\right)\left[\left(\frac{m+1}{m}\right)^2 \frac{V(L'')}{abc} + \frac{6}{m+1}\right] \\
&= \quad \left(\frac{m+1}{m}\right)^3 \frac{V(L'')}{abc} + \frac{6}{m} \\
&\leq \quad \left(\frac{m+1}{m}\right)^3 \frac{V(L'')}{abc} + 6.
\end{aligned}$$

Thus,

$$\#(\mathcal{P}'') \leq \left(\frac{m+1}{m}\right)^3 \frac{V(L'')}{abc} + 7. \tag{59}$$

The result follows from inequalities (58) and (59). □

Now, we present the ideas behind the algorithm $\mathrm{BOX}_{k,\epsilon}$. To understand this algorithm, we first consider the volume guarantee one could obtain if only list partition and the next fit decreasing algorithms were used. Suppose we partition the items of the input list L into sets $S_{ijk} := L \cap \mathcal{T}_{ijk}$, for $i, j, k \in \{0, 1\}$, where $\mathcal{T}_{ijk}$ are defined in step 1 of the algorithm $\mathrm{BOX}_{k,\epsilon}$.

In the set S_{111} we have the larger items, which lead to the very poor volume guarantee of $\frac{1}{2}\frac{1}{2}\frac{1}{2} = \frac{1}{8} = 0.125$. For the boxes in S_{ijk}, with $i + j + k = 2$, we can obtain a volume guarantee of $\frac{1}{2}\frac{1}{2}\frac{2}{3} = \frac{1}{6} = 0.166\ldots$. For the sets S_{ijk}, with $i + j + k = 1$, we can obtain a volume guarantee of $\frac{1}{2}\frac{2}{3}\frac{2}{3} = \frac{2}{9} = 0.222\ldots$. For the set S_{000}, we can obtain a packing with a volume guarantee of $\frac{2}{3}\frac{2}{3}\frac{2}{3} = \frac{8}{27} = 0.296\ldots$.

The critical sets are defined for items which lead to packings with volume guarantee close to $\frac{1}{8}$ (in set S_{111}), $\frac{1}{6}$ (in sets S_{011}, S_{101} and S_{110}) and $\frac{2}{9}$ (in sets S_{001}, S_{010} and S_{100}).

The algorithm first combines critical sets defined for sets S_{ijk}, with $i+j+k = 2$. First it combines S_{011} and S_{101} using the algorithm COMBINE-AB$_k^{xy}$, obtaining a combined packing with a good volume guarantee (that is better than $\frac{2}{9}$). Suppose that all critical boxes from S_{011} are totally packed. The remaining boxes in S_{011} give packings with volume guarantee close to $\frac{2}{9}$. Now, we perform another combination step with the algorithm COMBINE-AB$_k^{yz}$ with the remaining boxes of S_{101} and the set S_{110}. Suppose that all critical items of the set S_{101} have been totally packed. In this case, the combined packing has a good volume guarantee and the remaining items of S_{101} give packings with volume guarantee close to $\frac{2}{9}$.

In this case, if we perform careful rotations before these combinations, the remaining items of S_{110} and the items of S_{111} cannot be packed side by side in the x and y axes. Therefore, we can pack these items with good algorithms for the problem 1BP. In this case, we can obtain almost optimum packings with a volume guarantee of $\frac{1}{8}$.

Now, the set of items can be partitioned into three parts. In one part we can obtain an almost optimum packing with volume guarantee of $\frac{1}{8}$ (remaining items of S_{110} and the set S_{111}). In a second part, with the remaining items of $S_{011} \cup S_{101} \cup S_{001} \cup S_{010} \cup S_{100}$, we can obtain packings with volume guarantee close to $\frac{2}{9}$. In the third part, for the items of the set S_{000} we can obtain a good volume guarantee.

At last, the algorithm performs various combinations of items that give packings with volume guarantee close to $\frac{1}{8}$ and $\frac{2}{9}$. In this case, we can improve the volume guarantee of $\frac{1}{8}$ or the volume guarantee of $\frac{2}{9}$.

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

Input: List of boxes L (instance of 3BP$^{\mathrm{r}}(a,b,c)$).

Output: Packing $\mathcal{P}$ of L into bins $B = (a,b,c)$.

1 Let

$$\begin{array}{ll}
\mathcal{X}_0 \leftarrow \mathcal{X}[0, \frac{1}{2}], & \mathcal{X}_1 \leftarrow \mathcal{X}[\frac{1}{2}, 1], \\
\mathcal{Y}_0 \leftarrow \mathcal{Y}[0, \frac{1}{2}], & \mathcal{Y}_1 \leftarrow \mathcal{Y}[\frac{1}{2}, 1], \\
\mathcal{Z}_0 \leftarrow \mathcal{Z}[0, \frac{1}{2}], & \mathcal{Z}_1 \leftarrow \mathcal{Z}[\frac{1}{2}, 1]; \\
\mathcal{T}_{ijk} \leftarrow \mathcal{X}_i \cap \mathcal{Y}_j \cap \mathcal{Z}_k, \quad ijk \in \{0,1\} \ ; &
\end{array}$$

2 $p \leftarrow \frac{\sqrt{137}-9}{6}$.

3 Rotate all possible boxes $e \in L \cap \mathcal{T}_{111}$ in such a way that e fits in one of the sets $\mathcal{T}_{ijk}$, $ijk \neq 111$. Ties can be decided arbitrarily.

4 $\mathcal{P}_{AB}^{xy} \leftarrow$ COMBINE-AB$_k^{xy}(L, \mathrm{rtype}\,, \mathcal{T}_{011}, \mathcal{T}_{101}, \mathrm{FFC}^{xy})$.

UPDATE (L);

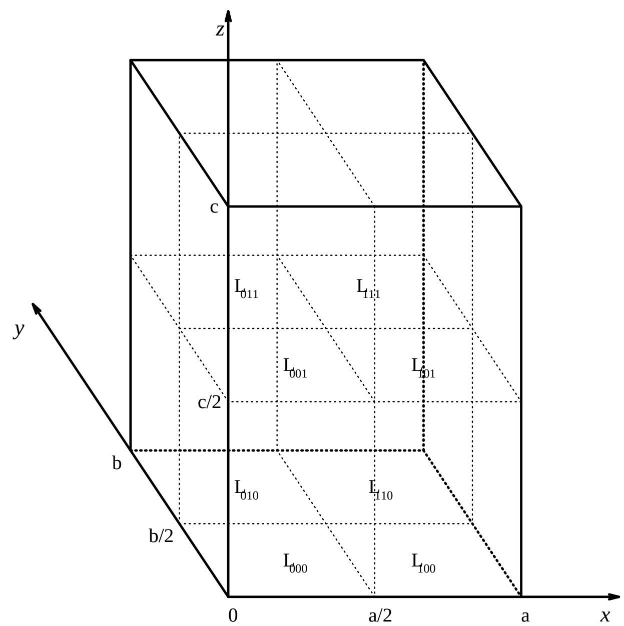

Figure 5: Lists L_{ijk}.

If rtype $(L, \mathcal{T}_{011} \cap \mathcal{A}_k^{xy}) = \emptyset$ then

$\mathcal{P}_{AB}^{yz} \leftarrow$ COMBINE-AB$_k^{yz}(L, \text{rtype}, \mathcal{T}_{101}, \mathcal{T}_{110}, \text{FFC}^{yz})$.
UPDATE (L);
$\mathcal{P}_{AB} \leftarrow \mathcal{P}_{AB}^{xy} \cup \mathcal{P}_{AB}^{yz}$.

Otherwise // all boxes of rtype $(\mathcal{T}_{101} \cap \mathcal{B}_k^{xy})$ were packed. //

$\mathcal{P}_{AB}^{xz} \leftarrow$ COMBINE-AB$_k^{xz}(L, \text{rtype}, \mathcal{T}_{110}, \mathcal{T}_{011}, \text{FFC}^{xz})$.
UPDATE (L).
$\mathcal{P}_{AB} \leftarrow \mathcal{P}_{AB}^{xy} \cup \mathcal{P}_{AB}^{xz}$.

5 Consider that all boxes of type $(\mathcal{T}_{011} \cap \mathcal{A}_k^{xy})$ and $(\mathcal{T}_{101} \cap \mathcal{A}_k^{yz})$ were totally packed.

5.1 Rotate all possible boxes $e \in L$ of type $\mathcal{T}_{110}$ in such a way that e fits in one of the sets $\mathcal{T}_{ijk}$, $ijk \notin \{111, 110\}$.

5.2 Rotate all possible boxes $e \in L$ of type $\mathcal{T}_{110} \cup \mathcal{T}_{111}$ in such a way that $z(b)$ is minimum.

5.3 Let $L_{ijk} \leftarrow L \cap \mathcal{T}_{ijk}$ for $i, j, k \in \{0, 1\}$ (see figure 5).

5.4 $\mathcal{P}_{000} \leftarrow \text{H3D}_2(L_{000})$.

5.5 Generate a packing $\mathcal{P}_{CD}$ in the following manner.

$\mathcal{P}_{CD}^{011} \leftarrow \text{FFC}^{xy}(L_{011} \cup L_{111}, L_{011} \cap \mathcal{X}[\frac{1}{3}, p], L_{111} \cap \mathcal{X}[\frac{1}{2}, 1-p], [(0,0)], [(0,p)])$;
UPDATE (L_{011}, L_{111});
$\mathcal{P}_{CD}^{101} \leftarrow \text{FFC}^{yz}(L_{101} \cup L_{111}, L_{101} \cap \mathcal{Y}[\frac{1}{3}, p], L_{111} \cap \mathcal{Y}[\frac{1}{2}, 1-p], [(0,0)], [(0,p)])$;

UPDATE (L_{101}, L_{111});
$\mathcal{P}_{CD}^{001} \leftarrow \text{FFC}^{xy}(L_{001} \cup L_{111}, L_{001} \cap \mathcal{X}[\frac{1}{3}, \frac{1}{2}] \cap \mathcal{Y}[\frac{1}{3}, p], L_{111} \cap \mathcal{Y}[\frac{1}{2}, 1-p], [(0,0), (0, \frac{1}{2})], [(0,p)])$;
UPDATE (L_{001}, L_{111});
$\mathcal{P}_{CD}^{010} \leftarrow \text{FFC}^{zx}(L_{010} \cup L_{111}, L_{010} \cap \mathcal{Z}[\frac{1}{3}, \frac{1}{2}] \cap \mathcal{X}[\frac{1}{3}, p], L_{111} \cap \mathcal{X}[\frac{1}{2}, 1-p], [(0,0), (0, \frac{1}{2})], [(0,p)])$;
UPDATE (L_{010}, L_{111});
$\mathcal{P}_{CD}^{100} \leftarrow \text{FFC}^{yz}(L_{100} \cup L_{111}, L_{100} \cap \mathcal{Y}[\frac{1}{3}, \frac{1}{2}] \cap \mathcal{Z}[\frac{1}{3}, p], L_{111} \cap \mathcal{Z}[\frac{1}{2}, 1-p], [(0,0), (0, \frac{1}{2})], [(0,p)])$;
UPDATE (L_{100}, L_{111});
$\mathcal{P}_{CD} \leftarrow \mathcal{P}_{CD}^{011} \cup \mathcal{P}_{CD}^{101} \cup \mathcal{P}_{CD}^{001} \cup \mathcal{P}_{CD}^{010} \cup \mathcal{P}_{CD}^{100}$.

5.6 Generate packings of the remaining boxes of the sublists L_{ijk} with $i + j + k = 1$.

5.6.1 Generate packing $\mathcal{P}_{001}$ of the remaining boxes in L_{001} in the following manner.
Let $L_{001}^{18}, \ldots, L_{001}^{23}$ be a partition of L_{001} such that (see figure 3).

$$\begin{array}{ll} L_{001}^{18} \leftarrow L_{001} \cap \mathcal{C}^{xy}\left[\frac{1}{3}, \frac{1}{2}\ ;\ \frac{1}{3}, \frac{1}{2}\right], & L_{001}^{19} \leftarrow L_{001} \cap \mathcal{C}^{xy}\left[\frac{1}{3}, \frac{1}{2}\ ;\ \frac{1}{4}, \frac{1}{3}\right], \\ L_{001}^{20} \leftarrow L_{001} \cap \mathcal{C}^{xy}\left[\frac{1}{3}, \frac{1}{2}\ ;\ 0, \frac{1}{4}\right], & L_{001}^{21} \leftarrow L_{001} \cap \mathcal{C}^{xy}\left[\frac{1}{4}, \frac{1}{3}\ ;\ \frac{1}{3}, \frac{1}{2}\right], \\ L_{001}^{22} \leftarrow L_{001} \cap \mathcal{C}^{xy}\left[0, \frac{1}{4}\ ;\ \frac{1}{3}, \frac{1}{2}\right], & L_{001}^{23} \leftarrow L_{001} \cap \mathcal{C}^{xy}\left[0, \frac{1}{3}\ ;\ 0, \frac{1}{3}\right]. \end{array}$$

$\mathcal{P}_{001}^{i} \leftarrow \text{H3D}^{z}(\text{NFDH}^{y}, L_{001}^{i}, \text{NF}), \quad i = 18, \ldots, 21$;
$\mathcal{P}_{001}^{22} \leftarrow \text{H3D}^{z}(\text{NFDH}^{x}, L_{001}^{22}, \text{NF})$;
$\mathcal{P}_{001}^{23} \leftarrow \text{H3D}^{z}(\text{BI}_3^{(t)}, L_{001}^{23}, \text{NF})$;
$\mathcal{P}_{001} \leftarrow \mathcal{P}_{001}^{18} \cup \ldots \cup \mathcal{P}_{001}^{25}$.

5.6.2 Generate a packing $\mathcal{P}_{010}$ of the remaining boxes of L_{010} in a way analogous to step 5.6.1, generating the levels in the y-axis direction.

5.6.3 Generate a packing $\mathcal{P}_{100}$ of the remaining boxes of L_{100} in a way analogous to step 5.6.1, generating the levels in the x-axis direction.

5.7 Generate a packing of the remaining boxes of L_{011} and L_{101}.

5.7.1 Generate packing $\mathcal{P}_{011}$ of the remaining boxes of L_{011}.
Let $(L_{011}^{1}, \ldots, L_{011}^{17})$ be a partition of L_{011} defined as follows (see figure 3).
$L_{011}^{i} \leftarrow L_{011} \cap \mathcal{Y}[\frac{1}{i+2}, \frac{1}{i+1}], \quad i = 1, \ldots, 16$;
$L_{011}^{17} \leftarrow L_{011} \cap \mathcal{Y}[0, \frac{1}{18}]$;
$\mathcal{P}_{011}^{i} \leftarrow \text{H3D}^{xy}(\text{NFDH}^{y}, L_{011}^{i}, \text{NF}), \quad i = 1, \ldots, 17$;
$\mathcal{P}_{011} \leftarrow \mathcal{P}_{011}^{1} \cup \ldots \cup \mathcal{P}_{011}^{17}$.

5.7.2 Generate packing $\mathcal{P}_{101}$ of the remaining boxes of L_{101} in a way analogous to step 5.7.1, considering the plane yz instead of xy.

5.8 Generate a packing of the remaining boxes of L_{110} and L_{111} as follows.

5.8.1 $L_{\text{UNI}} \leftarrow L_{110} \cup L_{111}$;

5.8.2 Consider each box e of L_{UNI} as a one-dimensional item of length $z(e)$ and each bin B as a one-dimensional bin with length c.

5.8.3 $\mathcal{P}'_{\text{UNI}} \leftarrow \text{FFD}^{z}(L_{\text{UNI}})$;

5.8.4 $\mathcal{P}''_{\text{UNI}} \leftarrow \text{FL}_{\epsilon}^{z}(L_{\text{UNI}})$;

5.8.5 $\mathcal{P}_{\text{UNI}} \leftarrow (\mathcal{P} \in \{\mathcal{P}'_{\text{UNI}}, \mathcal{P}''_{\text{UNI}}\} | \#(\mathcal{P}) \text{ is minimum})$.

5.9 $\mathcal{P}_{aux} \leftarrow \mathcal{P}_{AB} \cup \mathcal{P}_{CD} \cup \mathcal{P}_{000} \cup \mathcal{P}_{001} \cup \mathcal{P}_{010} \cup \mathcal{P}_{100} \cup \mathcal{P}_{011} \cup \mathcal{P}_{101}$;

5.10 $\mathcal{P} \leftarrow \mathcal{P}_{\text{UNI}} \cup \mathcal{P}_{aux}$.

6 For the other cases, the steps are analogous to step 5, differing only in the planes and directions the packing is generated.

7 Return $\mathcal{P}$.

end algorithm.

Theorem 7.2 *For any list of boxes L for* 3BP$^{\text{r}}$*, we have*

$$\text{BOX}_{k,\epsilon}(L) \leq \alpha_{k,\epsilon}\text{OPT}(L) + \beta_{k,\epsilon},$$

where $\alpha_{k,\epsilon} \to (43 + 3\sqrt{137})/16 = 4.882\ldots$ as $k \to \infty$ and $\epsilon \to 0$ and $\beta_{k,\epsilon}$ is constant for constant values of k and ϵ.

Proof. First, denote by L'_{ijk} the boxes packed in the packing $\mathcal{P}_{ijk}$, for $i, j, k \in \{0, 1\}$. We divide the proof in two cases, considering the set M, defined as

$$M := L_{111} \cap \mathcal{X}[\frac{1}{2}, 1-p] \cap \mathcal{Y}[\frac{1}{2}, 1-p] \cap \mathcal{Z}[\frac{1}{2}, 1-p],$$

after step 5.5.

In what follows, for a packing Q we denote by b_area(Q) the fraction of the bottom area of the bin B that is occupied by the packing Q.

Case 1. $M \neq \emptyset$ after step 5.5.

By Lemma 7.1 we have

$$\#(\mathcal{P}_{000}) \leq \frac{1}{8/27}\frac{V(L'_{000})}{abc} + 14. \tag{60}$$

For the packing $\mathcal{P}_{AB}$, note we have a b_area($\mathcal{P}_{AB}$) $\geq \frac{17}{36}$, except perhaps in $2(2k + 41)$ bins. Since each box of L_{AB} has height greater than $\frac{c}{2}$, we have

$$\#(\mathcal{P}_{AB}) \leq \frac{1}{17/72}\frac{V(L_{AB})}{abc} + 4k + 82. \tag{61}$$

For the packing $\mathcal{P}_{CD}$, note that for each bin B of $\mathcal{P}^i_{CD}$, $i \in \{011, 101, 001, 010, 100\}$, we have b_area($\mathcal{P}^i_{CD}$) $\geq \frac{1}{4} + \frac{r_1}{2}$, except perhaps for the last bin of each packing $\mathcal{P}^i_{CD}$. Also considering that each box has height greater than $\frac{c}{2}$, we have

$$\#(\mathcal{P}_{CD}) \leq \frac{1}{\frac{1}{8} + \frac{r_1}{4}}\frac{V(L_{CD})}{abc} + 6. \tag{62}$$

For the packing $\mathcal{P}_{001}$, for each bin B of $\mathcal{P}^i_{001}$, we have a b_area($\mathcal{P}^i_{001}$) $\geq \frac{17}{36}$ and each box has height at least $\frac{1}{2}$ of the corresponding dimension. Note that the boxes with small area guarantee in L^{18}_{001} were totally packed in $\mathcal{P}_{CD}$, otherwise we will not have $M \neq \emptyset$. Therefore, proceeding in the same way as before, we have

$$\#(\mathcal{P}_{001}) \leq \frac{1}{17/72}\frac{V(L'_{001})}{abc} + 8. \tag{63}$$

The same analysis we have made for packing $\mathcal{P}_{001}$ can be made for the packings $\mathcal{P}_{010}$ and $\mathcal{P}_{100}$. So, the following inequalities holds.

$$\#(\mathcal{P}_{010}) \leq \frac{1}{17/72}\frac{V(L'_{010})}{abc} + 8, \tag{64}$$

$$\#(\mathcal{P}_{100}) \leq \frac{1}{17/72}\frac{V(L'_{100})}{abc} + 8. \tag{65}$$

Now, consider the packing $\mathcal{P}_{011}$. Note that for each packing $\mathcal{P}^i_{011}$ we have b_area$(\mathcal{P}^i_{001}) \geq p$ (this minimum being attained for list L^1_{011}), except perhaps in the last bin of the packing $\mathcal{P}^i_{011}$. Therefore,

$$\#(\mathcal{P}_{011}) \leq \frac{1}{p/2}\frac{V(L'_{011})}{abc} + 17. \tag{66}$$

In the same way, we have the following inequality for packing $\mathcal{P}_{101}$.

$$\#(\mathcal{P}_{101}) \leq \frac{1}{p/2}\frac{V(L'_{101})}{abc} + 17. \tag{67}$$

From inequalities (60)—(67) and considering that $\frac{p}{2} = \min\{\frac{p}{2}, \frac{17}{72}, \frac{1}{8} + \frac{r_1}{4}\}$, we have

$$\#(\mathcal{P}_{aux}) \leq \frac{1}{p/2}\frac{V(L_{aux})}{abc} + C^k_{aux}. \tag{68}$$

Finally, consider the packing $\mathcal{P}_{\mathrm{UNI}}$ generated for boxes of L_{110} and L_{111} in step 5.8. The minimum volume in each bin B of $\mathcal{P}'_{\mathrm{UNI}}$, except perhaps in the last bin, is at least $\frac{abc}{8}$. Therefore,

$$\#(\mathcal{P}'_{\mathrm{UNI}}) \leq \frac{1}{1/8}\frac{V(L_{\mathrm{UNI}})}{abc} + 1.$$

Note that after the rotation of boxes made in step 5.1, there is no box e in L_{UNI} that can be rotated such that e fits in one of types $\mathcal{T}_{ijk}$, $ijk \notin \{110, 111\}$. So, after step 5.2, all boxes of L_{UNI} will have the smallest height possible, without leaving $\mathcal{T}_{110} \cup \mathcal{T}_{111}$. Therefore, after applying algorithm FL^z_ϵ in step 5.8.4, we have

$$\#(\mathcal{P}''_{\mathrm{UNI}}) \leq (1+\epsilon)\mathrm{OPT}(L_{\mathrm{UNI}}) + C^\epsilon_{\mathrm{UNI}}.$$

Since $\#(\mathcal{P}_{\mathrm{UNI}}) \leq \max\{\#(\mathcal{P}'_{\mathrm{UNI}}), \#(\mathcal{P}''_{\mathrm{UNI}})\}$, we have

$$\#(\mathcal{P}_{\mathrm{UNI}}) \leq \frac{1}{1/8}\frac{V(L_{\mathrm{UNI}})}{abc} + 1, \tag{69}$$

$$\#(\mathcal{P}_{\mathrm{UNI}}) \leq (1+\epsilon)\mathrm{OPT}(L_{\mathrm{UNI}}) + C^\epsilon_{\mathrm{UNI}}. \tag{70}$$

From inequalities (68)—(70) we can conclude that

$$\#(\mathcal{P}) \leq \alpha'_{k,\epsilon}\mathrm{OPT}(L) + \beta_{k,\epsilon},$$

where $\alpha'_{k,\epsilon} = (h_1 + h_2)/\max\{\frac{1}{1+\epsilon}h_1, \frac{1}{8}h_1 + \frac{p}{2}h_2\}$ and $\beta_{k,\epsilon} = C^k_{aux} + C^\epsilon_{\mathrm{UNI}}$.

Case 2. $M = \emptyset$ after step 5.5.

The analysis is analogous to Case 1, and the details will be omitted. We can conclude that

$$\begin{aligned}
\#(\mathcal{P}_{000}) &\leq \frac{1}{8/27}\frac{V(L_{000})}{abc} + 14, \\
\#(\mathcal{P}_{AB}) &\leq \frac{1}{17/72}\frac{V(L_{AB})}{abc} + 4k + 82, \\
\#(\mathcal{P}_{CD}) &\leq \frac{1}{\frac{1}{8} + \frac{r_1}{4}}\frac{V(L_{CD})}{abc} + 6.
\end{aligned}$$

Furthermore, for each packing $\mathcal{P}^i_{001}$ we have b_area$(\mathcal{P}^r_{001}) \geq \frac{4}{9}$. This also holds for the packings $\mathcal{P}_{010}$ and $\mathcal{P}_{100}$. Therefore, we have

$$\begin{aligned}
\#(\mathcal{P}_{001}) &\leq \frac{1}{2/9}\frac{V(L'_{001})}{abc} + 8, \\
\#(\mathcal{P}_{010}) &\leq \frac{1}{2/9}\frac{V(L'_{010})}{abc} + 8, \\
\#(\mathcal{P}_{100}) &\leq \frac{1}{2/9}\frac{V(L'_{100})}{abc} + 8.
\end{aligned}$$

For the packings $\mathcal{P}^i_{011}$ we have b_area$(\mathcal{P}^i_{011}) \geq r_1$. The same also holds for packing $\mathcal{P}_{101}$. Therefore,

$$\begin{aligned}
\#(\mathcal{P}_{011}) &\leq \frac{1}{r_1/2}\frac{V(L'_{011})}{abc} + 17, \\
\#(\mathcal{P}_{101}) &\leq \frac{1}{r_1/2}\frac{V(L'_{101})}{abc} + 17.
\end{aligned}$$

From the above inequalities, we have

$$\#(\mathcal{P}_{aux}) \leq \frac{1}{r_1/2}\frac{V(L_{aux})}{abc} + C^k_{aux}.$$

Since the boxes of M, $M \subseteq L_{111}$, were totaly packed, we have that the minimum volume of any box in L_{111} is at least $\frac{1-p}{4}$. Therefore, considering the packings of $\mathcal{P}_{110}$ and $\mathcal{P}_{111}$, we have

$$\begin{aligned}
\#(\mathcal{P}_{\mathrm{UNI}}) &\leq \frac{1}{(1-p)/4}\frac{V(L_{\mathrm{UNI}})}{abc} + 1. \\
\#(\mathcal{P}_{\mathrm{UNI}}) &\leq (1+\epsilon)\mathrm{OPT}(L_{\mathrm{UNI}}) + C^{\epsilon}_{\mathrm{UNI}}.
\end{aligned}$$

So, we obtain

$$\#(\mathcal{P}) \leq \alpha''_{k,\epsilon}\mathrm{OPT}(L) + \beta_{k,\epsilon},$$

where $\alpha''_{k,\epsilon} = (h_1 + h_2)/\max\{\frac{1}{1+\epsilon}h_1, \frac{1-p}{4}h_1 + \frac{r_1}{2}h_2\}$ and $\beta_{k,\epsilon} = C^k_{aux} + C^{\epsilon}_{\mathrm{UNI}}$.

Let $\alpha_{k,\epsilon} := \max\{\alpha'_{k,\epsilon}, \alpha''_{k,\epsilon}\}$. As for $k \to \infty$ we have $r_1 \to \frac{4}{9}$, we can conclude from both cases above that $\lim_{k\to\infty,\epsilon\to 0} \alpha_{k,\epsilon} \leq 4.8821\ldots$ □

8 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 here are the best known for each problem. All algorithms presented in this paper can be implemented to run in time polynomial in the number of items.

The algorithm presented for the problem $3SP^r$ uses as subroutine the algorithm for the problem $2SP^r$. We note that if one obtains a better algorithm for $2SP^r$, one can obtain a better algorithm for the problem $3SP^r$. We would like to thank David 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] N. Bansal and M. Sviridenko. New approximability and inapproximability results for 2-dimensional bin packing. In *Proc. 15th ACM-SIAM Symposium on Discrete Algorithms*, pages 189–196, 2004.

[3] A. Caprara. Packing 2-dimensional bins in harmony. In *Proc. of 43rd Symposium on Foundations of Computer Science*, pages 490–499, 2002.

[4] 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.

[5] 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.

[6] E. G. Coffman, Jr., M. R. Garey, and D. S. Johnson. Approximation algorithms for bin packing: a survey. In D. Hochbaum, editor, *Approximation Algorithms for NP-hard Problems*, chapter 2, pages 46–93. PWS, 1997.

[7] 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*, 9:808–826, 1980.

[8] J.R. Correa and C. Kenyon. Approximation schemes for multidimensional packing. In *Proc. 15th ACM-SIAM Symposium on Discrete Algorithms*, pages 179–188, 2004.

[9] J. Csirik and A. van Vliet. An on-line algorithm for multidimensional bin packing. *Operations Research Letters*, 13:149–158, 1993.

[10] L. Epstein. Two dimensional packing: The power of rotation. In *Proc. of the 28th International Symposium of Mathematical Foundations of Computer Science*, volume 2747 of *Lecture Notes on Computer Science – LNCS*, pages 398 – 407. Springer–Verlag, 2003.

[11] 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.

[12] D. S. Johnson. *Near-optimal bin packing algorithms.* PhD thesis, Massachusetts Institute of Technology, Cambridge, Mass., 1973.

[13] C. Kenyon and E. Rémila. A near-optimal solution to a two-dimensional cutting stock problem. *Mathematics of Operations Research*, 25:645–656, 2000.

[14] K. Li and K-H. Cheng. A generalized harmonic algorithm for on-line multidimensional bin packing. TR UH-CS-90-2, University of Houston, January 1990.

[15] K. Li and K-H. Cheng. On three-dimensional packing. *SIAM Journal on Computing*, 19:847–867, 1990.

[16] K. Li and K-H. Cheng. Static job scheduling in partitionable mesh connected systems. *Journal of Parallel and Distributed Computing*, 10:152–159, 1990.

[17] F. K. Miyazawa and Y. Wakabayashi. An algorithm for the three-dimensional packing problem with asymptotic performance analysis. *Algorithmica*, 18(1):122–144, 1997.

[18] F. K. Miyazawa and Y. Wakabayashi. Parametric on-line algorithms for packing rectangles and boxes. *European Journal on Operational Research*, 150:281–292, 2003.

[19] 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.

[20] F.K. Miyazawa and Y. Wakabayashi. Packing problems with orthogonal rotations. In *Proc. of Latin American Theoretical INformatics.*, volume 2976 of *Lecture Notes in Computer Science*, pages 359–368, Buenos Aires, Argentina, 2004. Springer-Verlag.