Three-dimensional packings with rotations*

F. K. Miyazawa† and Y. Wakabayashi‡

Abstract

We present approximation algorithms for the *three-dimensional strip packing problem*, and the *three-dimensional bin packing problem*. We consider orthogonal packings where ninety-degree rotations are allowed. The algorithms we show for these problems have asymptotic performance bounds 2.64, and 4.89, respectively. These algorithms are for the more general case in which the bounded dimensions of the bin given in the input are not necessarily equal (that is, we consider bins for which the length, the width and the height are not necessarily equal). Moreover, we show that these problems —in the general version— are as hard to approximate as the corresponding oriented version.

1 Introduction

We focus on orthogonal packing problems where ninety-degree rotations are allowed. These problems have many real-world applications [6, 19]: job scheduling, container loading, cutting of hardboard, glass, foam, etc.

We present approximation algorithms for 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 *d-dimensional 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 *d-dimensional 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 an 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 around any of the axes); to distinguish them we may refer to them as *r-packings* or *packings with rotation* (instead of saying non-oriented orthogonal packing). We also denote the corresponding problems by $d\mathrm{SP}^{\mathrm{r}}$ (d-dimensional strip packing problem with rotation) and $d\mathrm{BP}^{\mathrm{r}}$ (d-dimensional bin packing problem with rotation).

We present approximation algorithms with asymptotic performance bounds 2.64 and 4.89 for the problems $3\mathrm{SP}^{\mathrm{r}}$ and $3\mathrm{BP}^{\mathrm{r}}$, respectively.

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 [15] 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

*This research was partially supported by CNPq (Proc. 478470/06–1, 472504/07–0, 306624/07–9, 305702/07–6 and 490333/04–4) and ProNEx–FAPESP/CNPq (Proc. 03/09925–5).

†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.

paper, it follows that each problem considered here is NP-hard. Moreover, 1BP cannot be approximated —in the absolute sense— within $3/2 - \epsilon$; 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 [16] obtained an asymptotic polynomial time approximation scheme (APTAS). For the problem 2BP, Chung, Garey and Johnson [5] proved that the algorithm $\mathrm{HFF}^{(p)}$ (Hybrid First Fit) has asymptotic performance bound 2.125. In 2001, Caprara [4] 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, Correa, Kenyon and Sviridenko [1] proved that there is no APTAS for 2BP, unless $\mathrm{P} = \mathrm{NP}$. They also showed an APTAS for the problem dBP when the items and the bins are d-dimensional cubes. For the problem 3SP, in 1997 we presented a 2.67-approximation [20], then in 2006 Jansen and Solis-Oba [13] obtained a $(2 + \epsilon)$-approximation, and recently Bansal et al. [2] obtained a 1.69-approximation. For the problem 3BP, Li and Cheng [17] and Csirik and van Vliet [9] designed algorithms with asymptotic performance bound 4.84. Their algorithms generalize to the problem dBP, and achieve asymptotic performance bound close to 1.691^d.

The approximation bounds of some of the algorithms designed for the oriented version may also hold when these algorithms are used for the corresponding r-packing problem. This happens when the proofs are based only on area arguments. Except for these cases, not many results are known on approximation algorithms for r-packing problems. The interest on these problems is more recent and has increased lately.

For the problem $2\mathrm{SP}^{\mathrm{r}}$, the algorithms NFDH and BLDW have asymptotic performance bound 2 (see Coffman, Garey and Johnson [6]). In [24] we presented an algorithm with asymptotic performance bound 1.613; then Epstein and van Stee [11] obtained an algorithm with asymptotic bound 1.5, and Jansen and van Stee [14] presented an APTAS.

We considered a special case of the problem $3\mathrm{SP}^{\mathrm{r}}$, denoted by $3\mathrm{SP}^{z}$, in which the boxes can be rotated around the z-axis, but cannot be laid down. For this problem we obtained an algorithm with asymptotic performance bound 2.67 (see [21]). We also showed an algorithm with bound 2.55 for the special case of $3\mathrm{SP}^{z}$ in which the bin has square bottom, and also for a more specialized version in which the boxes have square bottom (see [24] for a result when the bin does not have a square bottom). To our knowledge, [21] is the first paper to present approximation algorithms for r-packing problems where rotations are exploited in a non-trivial way. It is easy to see that any algorithm for $3\mathrm{SP}^{z}$ leads to an algorithm to $2\mathrm{BP}^{\mathrm{r}}$ with the same bound. Therefore, the algorithms presented in [21] also lead to algorithms for the problem $2\mathrm{BP}^{\mathrm{r}}$. For the special case in which the bins are squares, Epstein [10] presented an on-line algorithm with asymptotic bound 2.45. Using the APTAS presented by Jansen and van Stee [14] for the problem $2\mathrm{SP}^{\mathrm{r}}$, it is possible to obtain a $(2 + \epsilon)$-asymptotic approximation algorithm for $2\mathrm{BP}^{\mathrm{r}}$, which is the best result known for this problem. Epstein and van Stee [11] obtained an algorithm with asymptotic bound 2.25 for the special case of $3\mathrm{SP}^{z}$ where the bin has square bottom. They also observed that this algorithm can be used to obtain an algorithm with asymptotic performance bound 4.5 for the special case when the bins are cubes.

Using the fact that there is no APTAS for 2BP, we may easily conclude that there is no APTAS for the problem 3SP or 3BP, unless $\mathrm{P} = \mathrm{NP}$.

For a survey on approximation algorithms for packing problems, we refer the reader to [6, 7].

This paper is organized as follows. In Section 2, we define the problems, give some basic definitions and state some results. Sections 3 and 4 are devoted to problems $3\mathrm{SP}^{\mathrm{r}}$ and $3\mathrm{BP}^{\mathrm{r}}$, respectively. In Section 5 we present some concluding remarks.

An extended abstract corresponding to an early version of this paper appeared in [24], presenting results for the two- and three-dimensional cases. In that paper, we only mentioned the bounds we have obtained, without any algorithm or proof for the three-dimensional case. In fact, in the present paper, for the 3SP problem we show an algorithm with performance bound 2.64, which is better than the bound 2.76, mentioned in that paper.

2 Preliminaries

In this section, we first define the packing problems that appear in this paper, then give some basic definitions, establish the notation, and mention some known results that we use. We also discuss some relations (reductions) between algorithms for the oriented version and the version with rotations. Since we use existing algorithms for subproblems of the $3\mathrm{SP}^{\mathrm{r}}$ and $3\mathrm{BP}^{\mathrm{r}}$, we also define these problems.

In the *bin packing problem*, 1BP, we are given a list of items $L = (s_1, \ldots, s_n)$, and bins B of capacity a, where $0 < s_i \leq a$, and we are asked to find a packing of L into a minimum number of bins B.

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 denote by $2\mathrm{SP}^{\mathrm{r}}(a)$, $2\mathrm{BP}^{\mathrm{r}}(a, b)$, $3\mathrm{SP}^{\mathrm{r}}(a, b)$, and $3\mathrm{BP}^{\mathrm{r}}(a, b, c)$ the corresponding problems versions with the bin sizes defined by values a, b and c. If $\mathcal{P}$ is a packing for the (three-dimensional) strip packing problem, we denote by $H(\mathcal{P})$ the *height* of packing $\mathcal{P}$, and if $\mathcal{P}$ is a packing for the (three-dimensional) bin packing problem, we denote by $\#(\mathcal{P})$ the *number of bins* used in $\mathcal{P}$.

2.1 Definitions and Notation

In all problems considered in this paper, the given list L of boxes must be packed orthogonally into bins B (3D strip or 3D bins) 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. We assume that the dimen-

sions of the input bin B is (a, b, c).

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

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

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

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

$$\mathcal{C}_{p,q} := \mathcal{X}[0,1/p]\cap\mathcal{Y}[0,1/q],$$

$$\mathcal{C}_{p,q,r} := \mathcal{X}[0,1/p]\cap\mathcal{Y}[0,1/q]\cap\mathcal{Z}[0,1/r].$$

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

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

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. If $\mathcal{P}$ is a packing, we denote by $\mathcal{L}(\mathcal{P})$ the set of boxes packed in $\mathcal{P}$.

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

2.2 Relations between algorithms for oriented packings and r-packings

One way to solve r-packing problems is to adapt algorithms designed for the oriented case. In [21] we mention that, for the problem $3\mathrm{SP}^z$, a simple algorithm that first rotates 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 [21]).

Most of the results concerning approximation results do not consider rotations. In the early 1980s, Coffman, Garey and Johnson [6] 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 [5] 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 [8, 16].

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^r *be constants and* $\mathcal{A}^r$ *an algorithm for* $\mathrm{PROB}^{\mathrm{r}}$ *such that* $\mathcal{A}^r(L) \leq \alpha\,\mathrm{OPT}(L) + \beta^r$ *for any input list* L *of* $\mathrm{PROB}^{\mathrm{r}}$*. Then, there is an algorithm* $\mathcal{A}$ *for* PROB *such that the following holds:*

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

where $\mathrm{OPT}'(L)$ *is the size of an optimum packing of* L *(w.r.t.* PROB*) and* β *is a constant. Moreover, the reduction is polynomial, if we consider a convenient representation for the instance.*

Proof. Consider the problem $2\mathrm{BP}^{\mathrm{r}}$ and an instance composed by an input list of rectangles L and bins of size $B = (a, b)$. Suppose that $x(e) \leq a$ and $y(e) \leq b$, for each item $e \in L$. Let $\mathcal{A}^r$ be an algorithm for $2\mathrm{BP}^{\mathrm{r}}$ such that $\mathcal{A}^r(L) \leq \alpha\mathrm{OPT}(L) + \beta^r$. Consider the following algorithm $\mathcal{A}'$ for the problem 2BP. First take a scaling of B to $B' = (a', b)$ and L to L' in the same proportion, in such a way that $\min\{x(e) : e \in L'\} > b$. In this case, all rectangles of L' can only be packed in the original orientation. At this point, we apply algorithm $\mathcal{A}^r$ to pack L' into the box B', obtaining a packing $\mathcal{P}'$. At last, reescale $\mathcal{P}'$ to the original sizes, obtaining a packing, say $\mathcal{P}$ (of the original list L into B). It is clear that $\mathcal{A}'(L) \leq \alpha \cdot \mathrm{OPT}'(L) + \beta$, where $\mathrm{OPT}'(L)$ is the size of an optimum packing for the problem 2BP and β is the additive constant obtained in the original scale.

For the three-dimensional case, we can first apply the strategy so that no box can be rotated around the z-axis. Then, we apply the same strategy scaling the height of all boxes and bins in such way that no item can be laid down. Denote the final input list by L'' and the final bins by B''. Obtain a packing $\mathcal{P}''$ applying the algorithm $\mathcal{A}'$ and then rescale the packing $\mathcal{P}''$ to the initial sizes.

We can apply the same strategy when rotations are allowed in only some axes, and design an algorithm $\mathcal{A}$ with the desired property. □

Using the fact that there is no APTAS for the problems 2BP (see [3]) we have the following negative result.

Corollary 2.2 *There is no APTAS for the problems* $2\mathrm{BP}^{\mathrm{r}}$*,* $3\mathrm{SP}^{\mathrm{r}}$ *and* $3\mathrm{BP}^{\mathrm{r}}$*, unless P=NP.*

One-Dimensional Bin Packing Problem: Some 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 [7]). Many algorithms have been designed for 1BP. 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 APTAS designed by Fernandez de la Vega and Lueker [12, 7], which we denote by FL_ϵ.

Theorem 2.3 *[12, 7] For any rational $\epsilon > 0$, there exists a polynomial-time algorithm FL_ϵ for the one-dimensional bin packing problem such that, for any input list L, $\mathrm{FL}_\epsilon(L) \leq (1+\epsilon)\,\mathrm{OPT}(L) + 1$.*

Two-dimensional Strip Packing Problem: Some of the algorithms we use as subroutine, are for the two-dimensional strip packing problem with rotation ($2\mathrm{SP}^{\mathrm{r}}$). For the problem 2SP, Coffman, Garey, Johnson and Tarjan [8] presented the algorithms $\mathrm{NFDH}^{(s)}$ (Next Fit Decreasing Height) and $\mathrm{FFDH}^{(s)}$ (First Fit Decreasing Height) and proved that their asymptotic performance bounds are 2 and 1.7, respectively. The algorithm $\mathrm{NFDH}^{(s)}$ first sorts the input list L in decreasing order of height, then packs the rectangles side by side generating levels. When an item cannot be packed in the current level, it is packed in a new level above the previous one. The algorithm $\mathrm{FFDH}^{(s)}$ also packs the items in decreasing order of height. Each item is packed in the first level with sufficiently space to accommodate it. If there is no such level, the item is packed in a new level above the previous one.

Recently, Jansen and van Stee [14] presented an asymptotic approximation scheme for problems in which all items can be packed in both ways. We note that the scheme presented can also be adapted to the case in which some of the items may not be rotated.

Theorem 2.4 *[14]. For any rational $\epsilon > 0$, there exists a polynomial-time algorithm JvS_ϵ for $2\mathrm{SP}^{\mathrm{r}}(a)$ such that $\mathrm{JvS}_\epsilon(L) \leq (1+\epsilon)\,\mathrm{OPT}(L) + O(Z/\epsilon^2)$, for any list L of rectangles with dimensions at most Z.*

3 Three-Dimensional Strip Packing Problem

In this section, we present an algorithm for $3\mathrm{SP}^{\mathrm{r}}$, called $\mathrm{TRI}_{k,\epsilon}$, with asymptotic performance bound close to 2.64. We observe that we consider the more general setting in which the bin may not have square bottom.

This algorithm uses the critical set combination strategy used in [20, 21]. The idea is to combine item types which do not lead to packings with good space filling, if considered independently.

In Section 3.1 we present some subroutines used by the main algorithm of this section. In sections 3.2–3.4 we present the ideas of the main algorithm and how they guide us to obtain the main algorithm. In Section 3.2 we show a first idea to obtain an approximation algorithm with asymptotic factor 3.25 and the points we need to improve to obtain a better bound. In Section 3.3, we present a first combination step to obtain an improved algorithm with bound 2.6875. In Section 3.4, we consider another combination step to obtain the final bound of 2.64. The use of the combination of critical sets is the key idea of algorithm $\mathrm{TRI}_{k,\epsilon}$. In Section 3.5 we present the main algorithm in details.

3.1 Subroutines

The algorithm $\mathrm{TRI}_{k,\epsilon}$ uses many algorithms as subroutines, which we describe in what follows.

First we describe the algorithm NFDH (Next Fit Decreasing Height) presented by Li and Cheng [18]. The algorithm has two variants: NFDH^x and NFDH^y. The notation NFDH is used to refer to any of these variants.

Algorithm NFDH: 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 [18]).

The following result will be useful (see [20, 21]).

Lemma 3.1 *Let L be an instance of $3\mathrm{SP}^{\mathrm{r}}$ and $\mathcal{P}$ be a packing of L consisting of levels $N_1, \ldots, N_v$ such that $\min\{z(e) : e \in N_i\} \geq \max\{z(e) : e \in N_{i+1}\}$, and $S(N_i) \geq s\,a\,b$ for a given constant $s > 0$, $i = 1, \ldots, v-1$. Then $H(\mathcal{P}) \leq \frac{1}{s}\frac{V(L)}{a\,b} + Z$.*

If a packing $\mathcal{P}$ satisfies the above inequality, we say that $\mathcal{P}$ has a *volume guarantee* of s.

Given a set of boxes S, we call these boxes as f-boxes if we can obtain a packing of S with volume guarantee of at least f. For example, the boxes $S \subset \mathcal{C}^{xy}\left[\frac{1}{3}, \frac{1}{2}\,;\, \frac{1}{2}, 1\right]$ are $\frac{1}{3}$-boxes, since we can sort the boxes in S in non-increasing order of height and place two boxes in each level, each box b with $S(b) \geq \frac{ab}{6}$, except perhaps in the last level. From Lemma 3.1, the obtained packing has volume guarantee $\frac{1}{3}$. Another way to obtain a packing $\mathcal{P}_S$ of S with volume guarantee $\frac{1}{3}$ is to iteratively pack the boxes generating two stacks, from the bottom of the bin, packing the next box of S at the top of the stack with smallest height. See Figure 1. When all boxes have been packed, the two stacks have almost the same height, except by a difference of Z. Since the bottom area of each box is at least $\frac{ab}{6}$ and the height difference of the two stacks is at most Z, we can conclude that $V(S) \geq \frac{ab}{3}(H(\mathcal{P}_S) - Z)$. Isolating $H(\mathcal{P}_S)$ we can see that packing $\mathcal{P}_S$ has a volume guarantee of $\frac{1}{3}$.

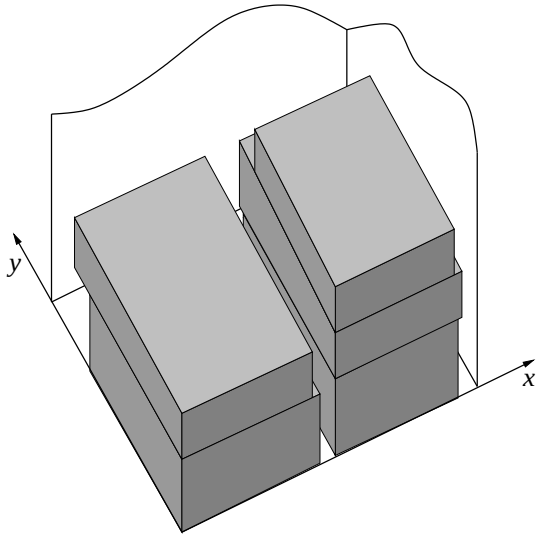

Figure 1: Two stacks of boxes.

Algorithm LL: Another algorithm we use is the algorithm LL_m, presented by Li and Cheng [19]. It is used to pack lists L such that $L \subset \mathcal{C}_m^{xy}$. The Algorithm LL first sorts the boxes in L in non-increasing order of their height and then divides L into sublists $L_1, \ldots, L_v$ such that $L = L_1 \| \ldots \| L_v$. Each sublist L_i has total bottom area $S(L_i)$ that is at least $\frac{m-2}{m}ab$ (except possibly for the last sublist) but not more than $\frac{m-2}{m}ab + \frac{ab}{m^2}$. Then, it uses a two-dimensional packing subroutine to pack each sublist into only one level (the subroutine is proved to pack in only one bin if the total area of rectangles is bounded in this way). Clearly, the area occupation in each level is at least $\frac{m-2}{m}ab$, and using Lemma 3.1 the following holds.

Lemma 3.2 *[19] If $\mathcal{P}$ is a packing generated by the algorithm LL_m for an instance $L \subset \mathcal{C}_m^{xy}$, then $\mathrm{LL}_m(L) \leq \frac{m}{m-2}\,V(L) + Z$.*

Algorithm A3S: Another algorithm we use is the algorithm $\mathrm{A3S}_{p,q}$, presented in [25]. This algorithm does not use any rotation. It divides the input list $L \subset \mathcal{C}_{p,q}$ into several sublists and apply specific algorithms for each one. Each packing is a level-oriented packing in the conditions presented for Lemma 3.1, with area occupation of at least $\frac{pq}{(p+1)(q+1)}ab$ in each level. The following holds for this algorithm.

Lemma 3.3 *If $\mathcal{P}$ is a packing generated by the algorithm $\mathrm{A3S}_{p,q}$ for an instance $L \subseteq \mathcal{C}_{p,q}$, then $\mathrm{A3S}_{p,q}(L) \leq \frac{(p+1)(q+1)}{pq}\ V(L) + 6Z$.*

Denote by $\mathrm{3S}_m$ the algorithm $\mathrm{A3S}_{\frac{1}{m},\frac{1}{m}}$. Using Lemma 3.3, the following holds.

Corollary 3.4 *If $\mathcal{P}$ is a packing generated by the algorithm $\mathrm{3S}_m$ for an instance $L \subseteq \mathcal{C}_{m,m}$, then $\mathrm{3S}_m(L) \leq \left(\frac{m+1}{m}\right)^2\ V(L) + 6Z$.*

3.2 Idea of the algorithm $\mathrm{TRI}_{k,\epsilon}$

First, let us give an idea of the algorithm $\mathrm{TRI}_{k,\epsilon}$. Consider the set of boxes defined by the sets $\wp_1, \ldots, \wp_4$ (mentioned in Section 2.1). Now, consider an input list of boxes for $\mathrm{3SP}^{\mathrm{r}}(a, b)$.

(i) First rotate each box $e \in \wp_1$ of the input list, if possible, to a box with orientation $e' \in \wp_2 \cup \wp_3 \cup \wp_4$.

(ii) Now, rotate each remaining box $e \in \wp_1$, if possible, to a box with orientation $e' \in \wp_1$ in such a way that $z(e')$ is minimum.

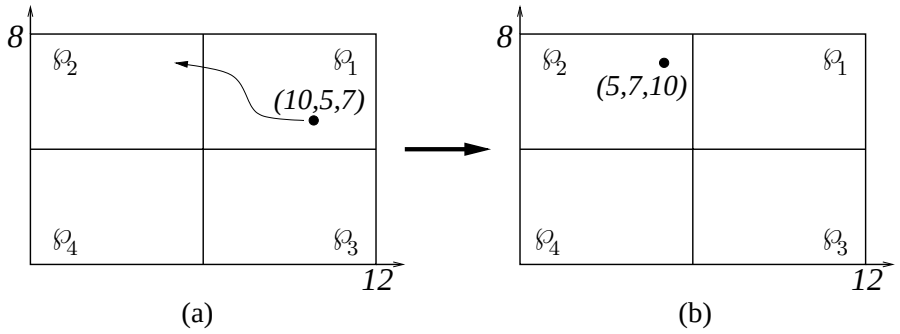

Figure 2: (a) Partition of boxes into parts (view of the xy-plane) and (b) rotation of a box with dimensions $(10, 5, 7)$ to a box $(5, 7, 10)$.

Figure 2 illustrates step *(i)*. Let L be the resulting input list, after these two steps and let L_i be the set $L \cap \wp_i$, for $i = 1, \ldots, 4$.

If we apply algorithm $\mathrm{A3S}_{p,q}$ for sublists $L_1, \ldots, L_4$ with appropriate values of p and q for each sublist, we obtain packings $\mathcal{P}_1, \ldots, \mathcal{P}_4$ for which the following holds:

$$H(\mathcal{P}_1) \leq \frac{1}{1/4}\frac{V(L_1)}{ab} + C_1 Z, \tag{1}$$

$$H(\mathcal{P}_2) \leq \frac{1}{1/3}\frac{V(L_2)}{ab} + C_2 Z, \tag{2}$$

$$H(\mathcal{P}_3) \leq \frac{1}{1/3}\frac{V(L_3)}{ab} + C_3 Z, \tag{3}$$

$$H(\mathcal{P}_4) \leq \frac{1}{4/9}\frac{V(L_4)}{ab} + C_4 Z, \tag{4}$$

where C_i, $i = 1, \ldots, 4$, are constants. That is, the boxes in the lists L_1, L_2, L_3 and L_4 are $\frac{1}{4}$-boxes, $\frac{1}{3}$-boxes, $\frac{1}{3}$-boxes and $\frac{4}{9}$-boxes, respectively.

Note that after steps *(i)* and *(ii)* the boxes of L_1 can only be packed one on top of the other, and therefore,

$$H(\mathcal{P}_1) \quad = \quad \mathrm{OPT}(L_1) \leq \mathrm{OPT}(L). \tag{5}$$

Let $n_1 = H(\mathcal{P}_1) - C_1 Z$ and $n_2 = \sum_{i=2}^{4} (H(\mathcal{P}_i) - C_i Z)$. Now we have two lower bounds for the height of an optimum packing: the height of packing $\mathcal{P}_1$ and the volume based lower bound $\frac{V(L)}{ab}$. That is,

$$\mathrm{OPT}(L) \quad \geq \quad H(\mathcal{P}_1) = n_1 \tag{6}$$

and

$$\begin{aligned} \mathrm{OPT}(L) \quad &\geq \quad \frac{V(L)}{ab} = \frac{V(L_1)}{ab} + \sum_{i=2}^{4} \frac{V(L_i)}{ab} \\ &\geq \quad \frac{1}{4} n_1 + \frac{1}{3} n_2. \end{aligned} \tag{7}$$

From (6) and (7), we have

$$\mathrm{OPT}(L) \quad \geq \quad \max\{n_1, \frac{1}{4} n_1 + \frac{1}{3} n_2\}. \tag{8}$$

Using the above relation, we can prove the following inequality for the final packing $\mathcal{P} = \mathcal{P}_1 \| \mathcal{P}_2 \| \mathcal{P}_3 \| \mathcal{P}_4$.

$$\begin{aligned} H(\mathcal{P}) \quad &= \quad H(\mathcal{P}_1) + H(\mathcal{P}_2) + H(\mathcal{P}_3) + H(\mathcal{P}_4) \\ &= \quad n_1 + n_2 + CZ \\ &\leq \quad \frac{n_1 + n_2}{\max\{n_1, \frac{1}{4} n_1 + \frac{1}{3} n_2\}} \mathrm{OPT}(L) + CZ \\ &= \quad \alpha' \mathrm{OPT}(L) + CZ, \end{aligned}$$

where $C = \sum_{i=1}^{4} C_i$ and $\alpha' := \frac{n_1 + n_2}{\max\{n_1, \frac{1}{4} n_1 + \frac{1}{3} n_2\}}$. The value of α' can be bounded by 3.25 using the following lemma, shown in [23].

Lemma 3.5 *Suppose X, Y, x, y are real numbers such that $x > 0$ and $0 < X < Y < 1$. Then*

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

In the analysis we considered that the packing that was generated consists of two parts: one optimum packing ($\mathcal{P}_1$) with "poor" volume guarantee (of $\frac{1}{4}$) and the other part (packing $\mathcal{P}_2 \| \mathcal{P}_3 \| \mathcal{P}_4$) with a "medium" volume guarantee (of at least $\frac{1}{3}$).

Note that if we could improve the volume guarantee of $\frac{1}{4}$ or $\frac{1}{3}$ that appear in the ratio α', then we obtain a bound that is better than 3.25. So, the first idea used in the algorithm $\mathrm{TRI}_{k,\epsilon}$ is to use a critical combination strategy to improve the volume guarantee of one of the parts. In each combination step, we combine two types of boxes, each type associated with a small volume guarantee. Although the boxes of each type may lead to packings with poor volume guarantee, if packed separately, the combined packing may have a good volume guarantee. The arrangement using two types of boxes may have a combination that leads to a better volume occupation than with only one type.

For each algorithm, we define the sublists that leads to packings with poor volume guarantee for each region, denoted as *critical boxes*, and make a combined packing with good volume guarantee, which we denote by *good packings*.

3.3 Combining critical $\frac{1}{3}$-boxes with critical $\frac{1}{3}$-boxes

In this section we present the algorithms that combine critical $\frac{1}{3}$-boxes: $\mathrm{COMBINE}^z$ and $\mathrm{COMBINE\text{-}AB}^z_k$. The packing obtained by the combination step has a good volume guarantee, of at least 0.457.

We consider the combination of boxes of type $\mathcal{A} = \mathcal{A}_1 \| \ldots \| \mathcal{A}_{k+14}$ with boxes of type $\mathcal{B} = \mathcal{B}_1 \| \ldots \| \mathcal{B}_{k+14}$ (throughout the paper, we denote the critical sets with lettered sets or lettered indexes). These sets are illustrated in Figure 3 and are produced by algorithm $\mathrm{COMBINE\text{-}AB}^z_k$. Since this algorithm is also used as a subroutine for problem $3\mathrm{BP}^{\mathrm{r}}$, it will be described in a more general way.

The combination is performed in steps by combining boxes of type $\mathcal{A}_i$ with boxes of type $\mathcal{B}_j$, for $1 \leq i, j \leq k+14$. At each combination step, all boxes of type $\mathcal{A}_i$ or all boxes of type $\mathcal{B}_j$ are totally packed. Figure 4 illustrates a packing that combines such boxes. The final combined packing of boxes of type $\mathcal{A}$ and $\mathcal{B}$ is the concatenation of all combined packings.

To describe algorithm $\mathrm{COMBINE\text{-}AB}^z_k$, we have to define some numbers which are used to define critical sets. For each critical subset $\mathcal{A}_i$ and $\mathcal{B}_j$, we can obtain positions and use the algorithm $\mathrm{COMBINE}$ to obtain a combined packing of items of $\mathcal{A}_i \cup \mathcal{B}_j$, such that the volume guarantee of the combined packing is at least $\frac{27}{56}$ and all items of one of these critical subsets are totally packed. These numbers have already been used in [21, 20]. For completeness, we present them and also the critical sets and related results.

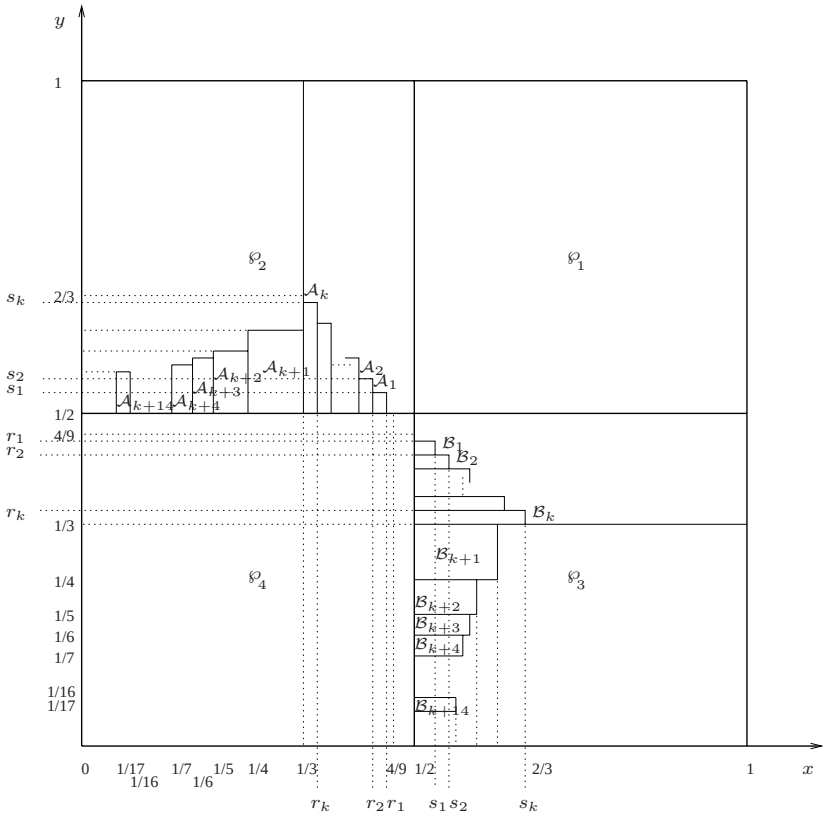

Figure 3: Sublists $\mathcal{A}_i := \mathcal{A}_i^{xy}$ and $\mathcal{B}_j := \mathcal{B}_j^{xy}$ when $a = b = 1$.

Definition 3.6 *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 3.7 *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 3.6, we define the following critical sets (see Figure 3).

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

We use basically the same procedure used in [21] with the algorithm COMBINE, with a small modification.
Algorithm COMBINE^z**:** This algorithm is called with the parameters $(L, \mathcal{T}^1, \mathcal{T}^2, p^1, p^2)$, where $p^1 = (p_1^1, p_2^1, \ldots, p_{n_1}^1)$ consists of the positions in the bottom of box B where the columns of boxes of type $\mathcal{T}^1$ should start and $p^2 = (p_1^2, p_2^2, \ldots, p_{n_2}^2)$ consists of the positions in the bottom of box B where the columns of boxes of type $\mathcal{T}^2$ should start. Each point $p_j^i = (x_j^i, y_j^i)$ represents the x-axis and the y-axis coordinates where the first box (if any) of each column of the respective type must be packed. Note that the z-axis coordinate need not be specified since it may always be assumed to be 0 (corresponding to the bottom of box B). Here we are assuming that the positions p^1, p^2 and the types $\mathcal{T}^1$, $\mathcal{T}^2$ are chosen in such a way that the defined packing can always be performed (a column will not intersect any other column). We call *height of a column* the sum of the height of all boxes in that column. Initially, all $n_1 + n_2$ columns are empty, starting at the bottom of box B. At each iteration, the algorithm chooses a column with the smallest height, say a column given by the position p_j^i, and packs the next box e of type $\mathcal{T}^i$, updating the list L after each iteration. If there is no such box e, then the algorithm teminates returning the partial packing $\mathcal{P}$ of L.

We denote by COMBINE^x and COMBINE^y the corresponding version of the algorithm COMBINE^z which generates columns in the x and y directions, respectively, or by COMBINE when considering any of these versions. The only modification of the algorithm COMBINE, from the version presented in [21], is that it may consider orthogonal rotations around any axis, to fit in the sets $\mathcal{A}_i^{xy}$ or $\mathcal{B}_j^{xy}$, or any set (type) given as a parameter. The next lemma is valid for this algorithm:

Lemma 3.8 *[21] Let* $\mathcal{P}$ *be the packing of* $L' \subseteq L$ *generated by the algorithm* COMBINE *when applied to lists of types* $\mathcal{T}^1$ *and* $\mathcal{T}^2$ *and list of positions* $p_1^i, p_2^i, \ldots, p_{n_i}^i$*,* $i = 1, 2$*. If* $S(e) \geq s_i ab$*, for all boxes* e *in* $\mathcal{T}^i$ $(i = 1, 2)$*, then* $H(\mathcal{P}) \leq \frac{1}{s_1 n_1 + s_2 n_2}\frac{V(L')}{ab} + Z$*.*

We also denote the sum $s_1 n_1 + s_2 n_2$, in Lemma 3.8, as the volume guarantee of the packing $\mathcal{P}$.

To combine all boxes of type $\mathcal{A}^{xy}$ or $\mathcal{B}^{xy}$, we call the algorithm COMBINE^z with pairs of types $\mathcal{A}_i^{xy}$ and $\mathcal{B}_j^{xy}$. Since each run of algorithm COMBINE packs all boxes of one type, it is sufficient to call COMBINE $2(k+14)$

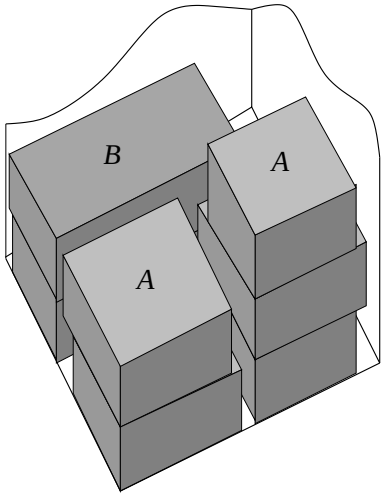

Figure 4: Example of a packing produced by algorithm COMBINE with 3 columns.

times. We denote this algorithm as COMBINE-AB$_k^z(L, \text{COMBINE})$. The algorithm COMBINE is given as a parameter, since in the next section, we use the algorithm COMBINE-AB$_k^z$ to pack boxes into bins for problem 3BP$^{\text{r}}$ with another subroutine. Figure 4 illustrates a packing produced by algorithm COMBINE-AB.

The following lemma is obtained from Lemma 3.8, using the fact that the volume guarantee of packing $\mathcal{P}_{AB}$ is at least $\frac{17}{36}$.

Lemma 3.9 *[21] If $\mathcal{P}_{AB}$ is a packing of a list L_{AB} generated by the algorithm* COMBINE-AB$_k^z$ *with parameters* $(L, \text{COMBINE})$*, then*

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

Figure 5 presents the volume guarantee one can obtain for each region using only list partition without any combination. As the algorithm COMBINE-AB$_k^z$ packs all boxes of type $\mathcal{A}$ or type $\mathcal{B}$, assume that all boxes of type $\mathcal{B}$ have been packed by this routine. Figure 6 illustrates the volume guarantee for the remaining boxes, in each region, and the situation when all boxes of type $\mathcal{B}$ have been packed. Now, if possible rotate each box in parts $\wp_1 \cup \wp_2$ that fits in the set $\wp_3 \cup \wp_4$. Clearly, after this step there will be no box which, if rotated, becomes a box in the set $\mathcal{B}$, because the construction of packing $\mathcal{P}_{AB}$ considers any possible rotation of boxes in the input list L.

Since the remaining items in $\wp_1 \cup \wp_2$ cannot be packed side by side in the y-dimension, we have now an instance of the two-dimensional strip packing problem, for which we can obtain an almost optimum packing, (according to Theorem 2.4), and volume guarantee at least $\frac{1}{4}$ (minimum of $\frac{1}{4}$ and $\frac{1}{3}$). To this end, rotate each box $e \in \wp_1 \cup \wp_2$, if possible, to a box $e' \in \wp_1 \cup \wp_2$ so that $y(e')$ is maximum (this allows to consider each box as a smallest possible rectangle in the xz plane). Therefore, at this point, we can obtain a final packing consisting of two parts: one almost optimum packing with volume guarantee $\frac{1}{4}$ and another (for the remaining boxes) with volume guarantee $\frac{4}{9}$. In this case, when $\epsilon \to 0$, we can use Lemma 3.5 to obtain a packing with asymptotic performance bound

$$\frac{h_1' + h_2'}{\max\{\frac{1}{1+\epsilon}h_1', \frac{1}{4}h_1 + \frac{4}{9}h_2'\}} \approx 2.6875.$$

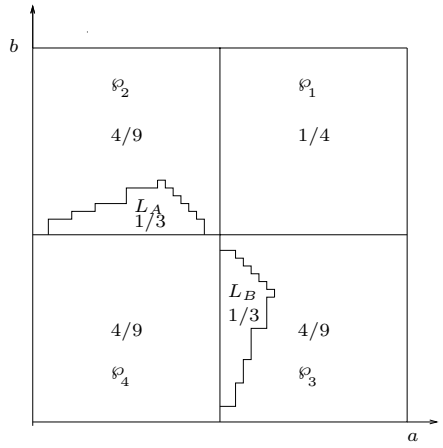

Figure 5: Critical sets $\mathcal{A}$ and $\mathcal{B}$.

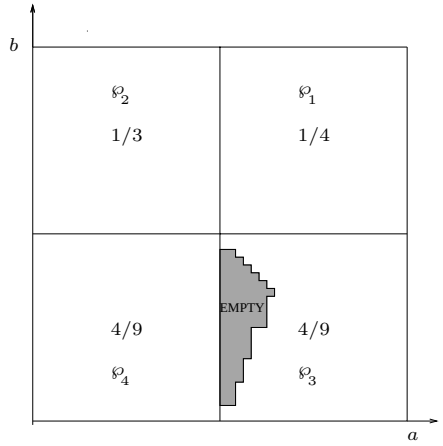

Figure 6: After combination of boxes of type $\mathcal{A}$ and $\mathcal{B}$: All boxes of type $\mathcal{B}$ have been packed.

3.4 Combining critical $\frac{1}{4}$-boxes with critical $\frac{4}{9}$-boxes

At this point, we continue with a new combination step, just after the generation of the combination of boxes of type $\mathcal{A}$ and $\mathcal{B}$ that produced $\mathcal{P}_{AB}$. Now, we combine critical $\frac{1}{4}$-boxes, which are given by the set L_C, and critical $\frac{4}{9}$-boxes, which are given by the set $L_D = L'_D \| L''_D$. These critical sets can be seen in Figure 9. The precise definition of sets L_C and L_D can be found in the description of Algorithm $\mathrm{TRI}_{k,\epsilon}$: these are the set of boxes of types $\mathcal{T}_C$ and $\mathcal{T}_D$, defined in step 4.3. The combined packing has also a good volume guarantee (which is also at least $\frac{1}{4} + \frac{2}{9} = \frac{17}{36} = 0.472\ldots$). Figures 7 and 8 present the regions of the critical sets L_C and $L'_D \| L''_D$ and the status when the boxes of each critical set are totally packed. The fractions indicate the minimum volume guarantee we can obtain for each region. The combined packing $\mathcal{P}_{CD}$ is the concatenation of two packings: $\mathcal{P}'_{CD}$, which combines L_C and L'_D, and $\mathcal{P}''_{CD}$, which combines the remaining items of L_C and L''_D. The packing $\mathcal{P}'_{CD}$ has one column consisting of items in L_C and one column consisting of items in L'_D. The packing $\mathcal{P}''_{CD}$ has one column consisting of items in L_C and two columns consisting of items in L''_D.

If all critical $\frac{1}{4}$-boxes have been packed in $\mathcal{P}_{CD}$, we obtain the following bound:

$$\frac{h'_1 + h'_2}{\max\{\frac{h'_1}{1+\epsilon}, 0.271 h_1 + \frac{4}{9} h'_2\}}.$$

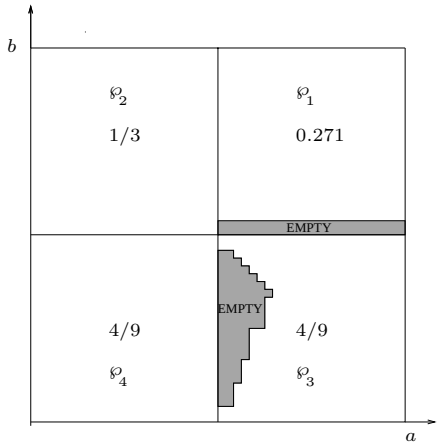

Figure 7: After combination of L_C and L_D: $L_C \subseteq \mathcal{P}_{CD}$.

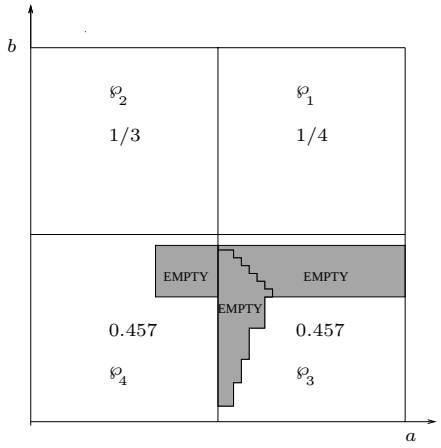

Figure 8: After combination of L_C and L_D: $L_D \subseteq \mathcal{P}_{CD}$.

Otherwise, if all critical $\frac{4}{9}$-boxes have been packed in $\mathcal{P}_{CD}$ we obtain the following bound:

$$\frac{h_1'' + h_2''}{\max\{\frac{h_1''}{1+\epsilon}, \frac{1}{4}h_1'' + 0.457h_2''\}}.$$

In both cases, a simple calculation shows that the asymptotic performance bound is at most 2.64.

3.5 Description and analysis of algorithm $\mathrm{TRI}_{k,\epsilon}$

In this section we present a formal descritption of algorithm $\mathrm{TRI}_{k,\epsilon}$ and analyse its asympototic performance bound. We observe that the value of t defined in step 2 was obtained by imposing equality for the bounds obtained in both cases analysed in the proof of Theorem 3.10.

The algorithm $\mathrm{TRI}_{k,\epsilon}$ and the proof of its approximation factor follows very closely the ideas presented in sections 3.2, 3.3 and 3.4. The packings obtained with combinations of critical sets have lettered indexes: $\mathcal{P}_{AB}$ and $\mathcal{P}_{CD}$. Both packings have good volume guarantee (at least 0.47). The remaining boxes are divided into many sublists, but basically, the final packing is divided into two parts (see Section 3.4): one with poor volume guarantee (in Case 1 it is 0.271 and in Case 2 it is 0.25) and the other part has a good volume guarantee (in Case 1 it is 0.444

and in Case 2 it is 0.457). For the part with poor volume guarantee, we could obtain an almost optimum packing (one within $(1+\epsilon)$ of the optimum).

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

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

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

1 Rotate each box $e \in \wp_1$, if possible, to a box $e' \in \wp_2 \cup \wp_3 \cup \wp_4$.

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

3 $\mathcal{P}_{AB} \leftarrow \mathrm{COMBINE\text{-}AB}^z_k(L, \mathrm{COMBINE})$.
$L \leftarrow L \setminus \mathcal{L}(\mathcal{P}_{AB})$.

4 If all boxes of type $\mathcal{B}^{xy}_k$ were packed in $\mathcal{P}_{AB}$ then

4.1 Rotate each box $e \in L \cap \wp_2$, if possible, to a box $e' \in \wp_3 \cup \wp_4$.

4.2 Rotate each box $e \in L \cap (\wp_1 \cup \wp_2)$, if possible, to a box $e' \in \wp_1 \cup \wp_2$ such that $y(e)$ is maximum.

4.3 Let

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

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

$\mathcal{P}_{CD'} \leftarrow \mathrm{COMBINE}(L, \mathcal{T}_C, \mathcal{T}'_D, [(0,0)], [(0,1-t)])$.
$\mathcal{P}_{CD''} \leftarrow \mathrm{COMBINE}(L \setminus L_{CD'}, \mathcal{T}_C, \mathcal{T}''_D, [(0,0)], [(0,1-t),(\tfrac{1}{2},1-t)])$.
$\mathcal{P}_{CD} \leftarrow \mathcal{P}_{CD'} \| \mathcal{P}_{CD''}$;
$L_{CD} \leftarrow L_{CD'} \bigcup L_{CD''}$;

4.5 Subdivide the list L into sublists $L_1, \ldots, L_{23}$ as follows (see Figure 9).

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

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 $i=22$;
$\mathcal{P}_{23} \leftarrow \mathrm{LL}_3(L_{23})$.

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

4.8 $L \leftarrow L \setminus \mathcal{L}(\mathcal{P}_{aux})$. /* Note that $L \subseteq \wp_1 \cup \wp_2$. */

4.9 Consider each box $e \in L$ as a rectangle of length $x(e)$ and height $y(e)$ and the box $B=(a,b,\infty)$ as a rectangular strip of length a and unlimited height. Apply algorithm JvS_ϵ to L (see Theorem 2.4) and let $\mathcal{P}_{\mathrm{JvS}}$ be the resulting packing.
Let $\mathcal{P}_{\mathrm{NFDH}}$ be the packing $\mathrm{NFDH}^x(L \cap \mathcal{X}[0,\tfrac{1}{3}]) \| \mathrm{NFDH}^x(L \cap \mathcal{X}[\tfrac{1}{3},\tfrac{1}{2}]) \| \mathrm{NFDH}^x(L \cap \mathcal{X}[\tfrac{1}{2},1])$.
Let $\mathcal{P}_{\mathrm{STRIP}}$ be the smallest packing in $\{\mathcal{P}_{\mathrm{JvS}}, \mathcal{P}_{\mathrm{NFDH}}\}$.

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

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

6 Return $\mathcal{P}$.

End algorithm.

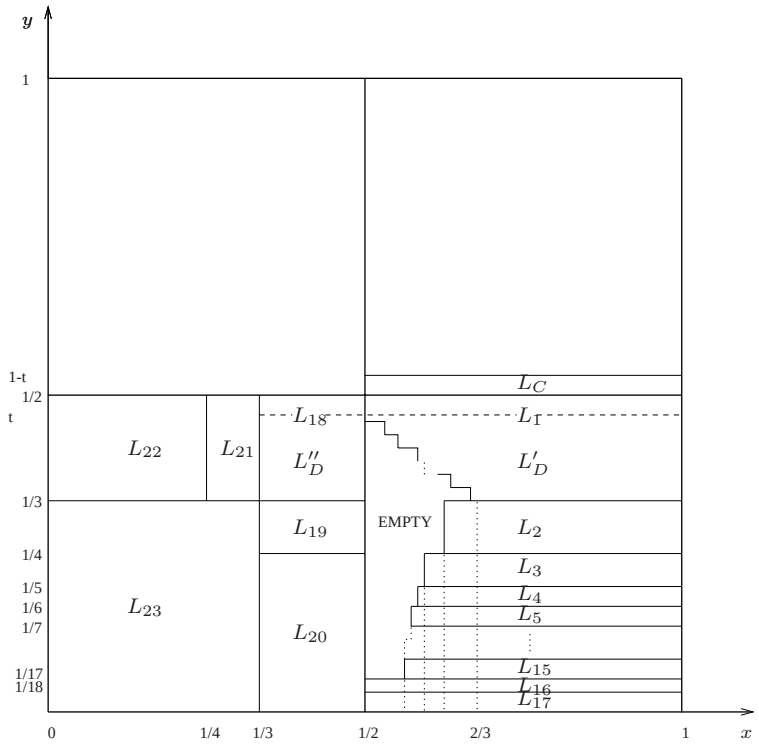

Figure 9: Sublist after the packing of list $L_B = (B_1 \cup \ldots \cup B_{k+14})$.

Theorem 3.10 *For any instance L for the problem $3\mathrm{SP}^{\mathrm{r}}$, we have*

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

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

Proof. We present the proof for the case in which all boxes of type B_k^{xy} were packed in step 4. The proof for the other case (step 5) is analogous. We analyse two subcases, according to step 4.4 ($I_C \subseteq L_{CD}$).

Each packing $\mathcal{P}_i$, $i \in \{1, \ldots, 23\} \setminus \{1, 18\}$, has a volume guarantee of at least $\frac{17}{36}$. Furthermore, this minimum value is attained when $i \in \{16, 17\}$. Therefore, applying Lemma 3.1 and 3.2 we can conclude that

$$H(\mathcal{P}_i) \leq \frac{36}{17}\frac{V(L_i)}{ab} + Z, \text{ for } i \in \{1, \ldots, 23\} \setminus \{1, 18\}. \tag{9}$$

From Lemma 3.9, we have

$$H(\mathcal{P}_{AB}) \leq \frac{36}{17}\frac{V(L_{AB})}{ab} + (2k + 41)Z. \tag{10}$$

For the packings $\mathcal{P}_{CD'}$ and $\mathcal{P}_{CD''}$ (step 4.4), the volume guarantee (by Lemma 3.8) is at least $\frac{1}{4}+\frac{r_1}{2}$, and therefore

$$H(\mathcal{P}_{CD}) \leq \frac{1}{\frac{1}{4}+\frac{r_1}{2}} \frac{V(L_{CD})}{ab} + 2Z. \tag{11}$$

Let us analyse 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 the packings $\mathcal{P}_1$ and $\mathcal{P}_{18}$ we have

$$H(\mathcal{P}_1) \leq \frac{1}{r_1}\frac{V(L_1)}{ab} + Z, \tag{12}$$

$$H(\mathcal{P}_{18}) \leq \frac{1}{\frac{4}{9}}\frac{V(L_{18})}{ab} + Z. \tag{13}$$

By Theorem 2.4,

$$H(\mathcal{P}_{\mathrm{STRIP}}) \leq H(\mathcal{P}_{\mathrm{JvS}}) \leq (1+\epsilon)\mathrm{OPT}(L_{\mathrm{STRIP}}) + \beta_\epsilon Z, \tag{14}$$

where L_{STRIP} is the set of items packed in $\mathcal{P}_{\mathrm{STRIP}}$. Note that to derive the last inequality we used the fact that the items in L_{STRIP} cannot be packed side by side in the y-direction. Moreover, any item $e \in L_{\mathrm{STRIP}}$ has been previously rotated to have maximum (possible) value of $y(e)$. So, applying algorithm NFDH to L_{STRIP} we obtain

$$H(\mathcal{P}_{\mathrm{STRIP}}) \leq H(\mathcal{P}_{\mathrm{NFDH}}) \leq \frac{1}{(1-t)\frac{1}{2}}\frac{V(L_{\mathrm{STRIP}})}{ab} + 3Z. \tag{15}$$

Now, for the packing $\mathcal{P}_{aux} = \mathcal{P}_{AB}\|\mathcal{P}_1\|\ldots\|\mathcal{P}_{23}$, using the inequalities (10),...,(13) 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

$$H(\mathcal{P}_{aux}) \leq \frac{1}{r_1}\frac{V(L_{aux})}{ab} + (2k+68)Z, \tag{16}$$

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

Let

$$n_1 := H(\mathcal{P}_{\mathrm{STRIP}}) - \beta_\epsilon Z, \tag{17}$$

$$n_2 := H(\mathcal{P}_{aux}) - (2k+68)Z. \tag{18}$$

From inequality (14) we have $n_1 \leq (1+\epsilon)\mathrm{OPT}(L_{\mathrm{STRIP}})$ and therefore,

$$\mathrm{OPT}(L) \geq \mathrm{OPT}(L_{\mathrm{STRIP}}) \geq \frac{n_1}{(1+\epsilon)}. \tag{19}$$

From (16) and (18) we can conclude that

$$\frac{V(L_{aux})}{ab} \geq r_1 n_2, \tag{20}$$

and from (15) and (17), we have

$$\frac{V(L_{\mathrm{STRIP}})}{ab} \geq \frac{(1-t)}{2} n_1. \tag{21}$$

Since $V(L) = V(L_{aux}) + V(L_{\mathrm{STRIP}})$, using (20) and (21) we obtain $\frac{V(L)}{ab} \geq r_1 n_2 + \frac{(1-t)}{2} n_1$.

So,

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

Combining (19) and the inequality above, we get

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

Since $H(\mathcal{P}) = H(\mathcal{P}_{aux}) + H(\mathcal{P}_{\mathrm{STRIP}})$; using (17) and (18), we have

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

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

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

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 3.5, 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. $L_D \subseteq L_{CD}$.

As the proof of this case is analogous, we omit some details. Since all rectangles of L'_D were packed in $\mathcal{P}_{CD}$, we have no critical $\frac{1}{4}$-boxes in L_1. More precisely, we have a volume guarantee of at least t for the packing $\mathcal{P}_1$. The same can be verified for the packing $\mathcal{P}_{18}$. Thus, the following holds.

$$H(\mathcal{P}_i) \leq \frac{1}{t}\frac{V(L_i)}{ab} + Z \quad \text{for} \quad i \in \{1, 18\}. \tag{22}$$

Since $t \leq \min\left\{\frac{1}{4} + \frac{r_1}{2}, \frac{17}{36}\right\}$, from (22), (10) and (11) we have

$$H(\mathcal{P}_{aux}) \leq \frac{1}{t}\frac{V(L_{aux})}{ab} + (2k+68)Z. \tag{23}$$

By Theorem 2.4,

$$H(\mathcal{P}_{\mathrm{STRIP}}) \leq H(\mathcal{P}_{\mathrm{JvS}}) \leq (1+\epsilon)\mathrm{OPT}(L_{\mathrm{STRIP}}) + \beta_\epsilon. \tag{24}$$

The packing $\mathcal{P}_{\mathrm{NFDH}}$ has a volume guarantee of at least $\frac{1}{4}$ and since $H(\mathcal{P}_{\mathrm{STRIP}}) \leq H(\mathcal{P}_{\mathrm{NFDH}})$, we have

$$H(\mathcal{P}_{\mathrm{STRIP}}) \leq \frac{1}{1/4}\frac{V(L_{\mathrm{STRIP}})}{ab} + 3Z. \tag{25}$$

Let

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

Then, from (24) we can conclude that

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

Now, from (23) and (25), we have

$$\frac{V(L_{aux})}{ab} \geq t\, n_2 \ \text{ and } \ \frac{V(L_{\mathrm{STRIP}})}{ab} \geq \frac{1}{4}n_1,$$

and therefore,

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

So,

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

Thus, $\mathrm{TRI}_{k,\epsilon}(L) \leq \alpha''_{k,\epsilon}(r_1)\mathrm{OPT}(L) + (2k+\beta'_\epsilon)Z$, 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 3.5.

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

4 Three-Dimensional Bin Packing Problem

In this section, we consider the three-dimensional bin packing problem with rotation (3BP). We present an algorithm with an asymptotic performance bound that may converge to a value smaller than 4.89. We denote the algorithm of this section by $\mathrm{BOX}_{k,\epsilon}$.

4.1 Subroutines

Before presenting the main algorithm, we describe some algorithms used as subroutines.

Algorithm H3B: This algorithm uses the same strategy used in the algorithm HFF (*Hybrid First Fit*) presented by Chung, Garey and Johnson [5]. The algorithm $\mathrm{H3B}^z$ generates a packing in two steps. First it generates a three-dimensional strip packing of L, subdivided in levels and using the z-axis as a height dimension, and then packs the levels into bins, using a one-dimensional bin packing algorithm. The algorithms for the problems 3SP and 1BP used in these steps must be given as subroutines. We denote by $\mathrm{H3B}_{p,q,r}$ the algorithm that uses the algorithms $\mathrm{A3S}_{p,q}$ and FFD as subroutines and by $\mathrm{H3B}_m$ the algorithm $\mathrm{H3B}_{\frac{1}{m},\frac{1}{m},\frac{1}{m}}$. The following holds for this algorithm.

Lemma 4.1 *[22] If $\mathcal{P}$ is a packing generated by the algorithm $\mathrm{H3B}_{p,q,r}$ for an instance $L \subseteq \mathcal{C}_{p,q,r}$, then $\mathrm{H3B}_{p,q,r}(L) \leq \frac{(p+1)(q+1)(r+1)}{pqr}\,V(L) + 14$.*

Lemma 4.2 *If $\mathcal{P}$ is a packing generated by the algorithm $\mathrm{H3B}_m$ for an instance $L \subseteq \mathcal{C}_m$, then $\mathrm{H3B}_m(L) \leq \left(\frac{m+1}{m}\right)^3\,V(L) + 14$.*

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.

Algorithm FFDC: We use the same scheme of the algorithm COMBINE used for $3\mathrm{SP}^r$. For that, we first modify the algorithm COMBINE to the bin packing version. We denote this algorithm as FFDC^z (First Fit Decreasing Combine for z-axis). The algorithm FFDC^z combines the strategy of the algorithm COMBINE with the strategy of the algorithm FFD 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 FFDC^z

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 such that 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 in L 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 a non-packed box e' of type $\mathcal{T}_1$ with $z(e')$ maximum. 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 it was not possible to pack an item by steps 1.2 and 1.3, let $i \leftarrow i + 1$; generate a new empty packing $\mathcal{P}_i$ (that starts with empty columns in positions $p_1 \cup p_2$) in a new bin B_i.

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

end algorithm.

We denote by FFDC^x and FFDC^y the corresponding versions of the algorithm FFDC^z that generates columns in the x and y directions, respectively.

4.2 Main algorithm for $3\mathrm{BP}^{\mathrm{r}}$

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 region of boxes in types $\mathcal{T}_{ijk}$, for $i, j, k \in \{0, 1\}$ as follows:

$$\begin{array}{ll} \mathcal{X}_0 \leftarrow \mathcal{X}[0, \frac{1}{2}], & \mathcal{X}_1 \leftarrow \mathcal{X}[\frac{1}{2}, 1], \\ \mathcal{Y}_0 \leftarrow \mathcal{Y}[0, \frac{1}{2}], & \mathcal{Y}_1 \leftarrow \mathcal{Y}[\frac{1}{2}, 1], \\ \mathcal{Z}_0 \leftarrow \mathcal{Z}[0, \frac{1}{2}], & \mathcal{Z}_1 \leftarrow \mathcal{Z}[\frac{1}{2}, 1]. \end{array}$$

$$\mathcal{T}_{ijk} \leftarrow \mathcal{X}_i \cap \mathcal{Y}_j \cap \mathcal{Z}_k, \quad i, j, k \in \{0, 1\}\ .$$

First, rotate each box $e \in L$ of type $\mathcal{T}_{111}$, if possible, to a box $e' \in \cup_{ijk \neq 111} \mathcal{T}_{ijk}$. Now, consider the volume guarantee one can obtain with Lemma 4.1, only with list partition and algorithm H3B. Partition L into sets $S_{ijk} := L \cap \mathcal{T}_{ijk}$, for $i, j, k \in \{0, 1\}$. We have the following volume guarantees for each sublist:

- In the set S_{111} we have the larger items. Since $S_{111} = L \cap \mathcal{X}[\frac{1}{2}, 1] \cap \mathcal{Y}[\frac{1}{2}, 1] \cap \mathcal{Z}[\frac{1}{2}, 1]$, we have that $S_{ijk} \subseteq \mathcal{C}_{1,1,1}$. From Lemma 4.1 we have that

$$\begin{aligned} \mathrm{H3B}_{1,1,1}(S_{111}) &\leq \frac{(1+1)(1+1)(1+1)}{1 \cdot 1 \cdot 1} V(S_{111}) + 14 \\ &= \frac{1}{1/8} V(S_{111}) + 14. \end{aligned}$$

  Therefore, the set S_{111} leads to the very poor volume guarantee of $\frac{1}{8}$. The analysis for other sublists are similar.

- 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 defined for algorithm $\mathrm{BOX}_{k,\epsilon}$ consider regions for which we obtain 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}). Since no two boxes of S_{111} can be packed in a same bin, placing one box of S_{111} in each bin leads to an optimum packing of S_{111}. Therefore, using Lemma 3.5, we can obtain a packing with asymptotic performance that is bounded by

$$\frac{n_1 + n_2}{\max\left\{n_1, \frac{1}{8}n_1 + \frac{1}{6}n_2\right\}} \leq 6.25.$$

In the next sections, we define and combine critical sets to obtain packings with better bounds.

4.2.1 Combining critical $\frac{1}{6}$-boxes with critical $\frac{1}{6}$-boxes

The algorithm first combines critical sets of type $\mathcal{T}_{ijk}$, with $i + j + k = 2$, using the algorithm $\mathrm{COMBINE\text{-}AB}_k$ with subroutine FFDC.

First, it combines critical $\frac{1}{6}$-boxes of type $\mathcal{T}_{011}$ and $\mathcal{T}_{101}$ obtaining a combined packing with a good volume guarantee. If all critical boxes of type $\mathcal{T}_{011}$ have been packed, it combines critical boxes of type $\mathcal{T}_{101}$ with critical boxes of type $\mathcal{T}_{110}$; otherwise, it combines critical boxes of type $\mathcal{T}_{011}$ with critical boxes of type $\mathcal{T}_{110}$. In each combination step, it defines the corresponding critical sets. To combine critical boxes of type $\mathcal{T}_{011}$ and $\mathcal{T}_{101}$, it uses the routine $\mathrm{COMBINE\text{-}AB}_k^z$ with subroutine FFDC^z. See Figure 11(a). Note that all boxes $e \in \mathcal{T}_{011} \cup \mathcal{T}_{101}$ have $z(e) > \frac{c}{2}$ and each bin produced in the combined packing has three boxes, except perhaps in the last, with total bottom area at least $\frac{17}{36}ab$. Therefore, the volume guarantee of this combined packing is at least $\frac{17}{36}\,\frac{1}{2} = \frac{17}{72} = 0.231\ldots$. The same volume guarantee can be obtained when critical $\frac{1}{6}$-boxes of types $\mathcal{T}_{011}$ and $\mathcal{T}_{110}$ are combined. Denote by $\mathcal{P}_{AB}$ the packing produced by calling algorithm $\mathrm{COMBINE\text{-}AB}_k$

Before continuing with new combination steps, let us give the idea behind the direction of these combinations. After the generation of packing $\mathcal{P}_{AB}$, we can obtain packings with volume guarantee close to $\frac{2}{9} = 0.222\ldots$ for the remaining boxes, except for those in $\mathcal{T}_{110} \cup \mathcal{T}_{111}$. Fortunately, if each box $e \in \mathcal{T}_{110} \cup \mathcal{T}_{111}$ is previously rotated to a box $e' \in \mathcal{T}_{ijk}$, for $ijk \notin \{110, 111\}$, the remaining items in $L \cap (\mathcal{T}_{110} \cup \mathcal{T}_{111})$ can only be packed side by side in the z-axis. Therefore, it is an instance of a one-dimensional bin packing problem, for which we can obtain an almost optimum packing (see Theorem 2.3) and volume guarantee of at least $\frac{1}{8}$. This leads to a final packing with asymptotic performance bound

$$\frac{n_1 + n_2}{\max\left\{\frac{1}{1+\epsilon}n_1, \frac{1}{8}n_1 + \frac{2}{9}n_2\right\}} \leq 4.9375.$$

We can obtain a further improvement with some more combinations.

4.2.2 Combining critical $\frac{1}{8}$-boxes with critical $\frac{2}{9}$-boxes

Assume that all critical $\frac{1}{6}$-boxes of types $\mathcal{T}_{011}$ and $\mathcal{T}_{101}$ have been totally packed. So, the current situation is the following: The packing $\mathcal{P}_{AB}$ has volume guarantee at least $\frac{17}{72}$ and the remaining boxes in the set $\mathcal{T}_{011}$ and $\mathcal{T}_{101}$ lead to packings with volume guarantee close to $\frac{17}{72}$. The last combination steps consider the critical $\frac{1}{8}$-boxes, in the set $\mathcal{T}_{111}$, and the critical $\frac{2}{9}$-boxes in $\bigcup_{ijk \notin \{111,000\}} \mathcal{T}_{ijk}$. The combined packing of this step, denoted by $\mathcal{P}_{CD}$, has good volume guarantee (that is at least $\frac{p}{2} = 0.225\ldots$) and totally packs one of the critical sets. See Figures 11(b) and (c). We have two cases, depending on which critical set is totally packed:

Case 1. All critical $\frac{2}{9}$-boxes are totally packed. In this case, the asymptotic performance bound is at most

$$\frac{n_1 + n_2}{\max\left\{\frac{1}{1+\epsilon}n_1, \frac{1}{8}n_1 + \frac{p}{2}n_2\right\}}.$$

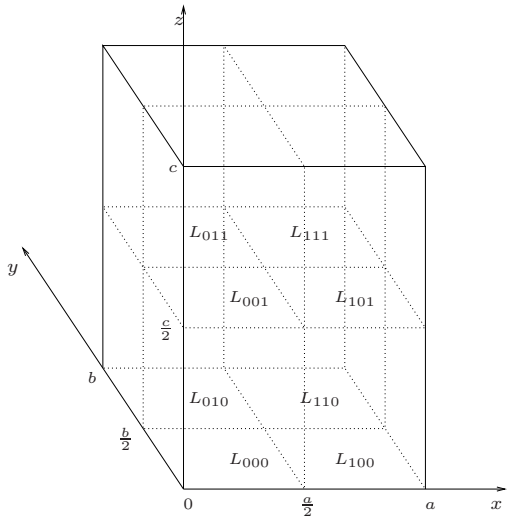

Figure 10: Lists L_{ijk}.

Case 2. All critical $\frac{1}{8}$-boxes are totally packed. In this case, the asymptotic performance bound is at most

$$\frac{n_1 + n_2}{\max\left\{\frac{1}{1+\epsilon}n_1, \frac{1-p}{4}n_1 + \frac{2}{9}n_2\right\}}.$$

In both cases, the performance bounds is close to 4.882. The value of p was obtained in such a way that the two cases give the same bound.

4.2.3 Algorithm $\mathrm{BOX}_{k,\epsilon}$

The Algorithm $\mathrm{BOX}_{k,\epsilon}$ and the proof of its approximation factor follows the ideas presented in Section 4.2. As before, the packings obtained with combination of critical sets have lettered indexes: $\mathcal{P}_{AB}$ and $\mathcal{P}_{CD}$, since the ideas used are close to the ones used in algorithm $\mathrm{TRI}_{k,\epsilon}$. The volume guarantees we can obtain for the bin packing case are worse than those for the strip packing version. So, in this case we say that a good volume guarantee, obtained for packings $\mathcal{P}_{AB}$ and $\mathcal{P}_{CD}$, is close to 0.236. The final packing is divided into two parts, one with poor volume guarantee (in Case 1 it is 0.125 and in Case 2 it is 0.137) and the other part with good volume guarantee (in Case 1 it is 0.225 and in Case 2 it is 0.222). Here we could also obtain an almost optimum packing (within $(1+\epsilon)$ of the optimum) for the items in the part with poor volume guarantee.

We present now a formal description of the algorithm we explained previously.

Algorithm $\mathrm{BOX}_{k,\epsilon}(L)$
Input: List of boxes L (instance of $3\mathrm{BP}^{\mathrm{r}}(a,b,c)$).
Output: Packing $\mathcal{P}$ of L into bins $B = (a,b,c)$.

1 Let

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

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

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

4 $\mathcal{P}'_{AB} \leftarrow \text{COMBINE-AB}^z_k(L, \mathcal{T}_{011}, \mathcal{T}_{101}, \text{FFDC}^z)$; $L \leftarrow L \setminus \mathcal{L}(\mathcal{P}'_{AB})$;

If all boxes of type $\mathcal{T}_{011} \cap \mathcal{A}^{xy}_k$ were packed then

$$\mathcal{P}''_{AB} \leftarrow \text{COMBINE-AB}^x_k(L, \mathcal{T}_{101}, \mathcal{T}_{110}, \text{FFDC}^x);$$

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

$$\mathcal{P}''_{AB} \leftarrow \text{COMBINE-AB}^y_k(L, \mathcal{T}_{110}, \mathcal{T}_{011}, \text{FFDC}^y);$$

$\mathcal{P}_{AB} \leftarrow \mathcal{P}'_{AB} \cup \mathcal{P}''_{AB}$; $L \leftarrow L \setminus \mathcal{L}(\mathcal{P}''_{AB})$;

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

5.1 Rotate each box $e \in L$ of type $\mathcal{T}_{110}$, if possible, to a box $e' \in \mathcal{T}_{ijk}$, for some $ijk \notin \{111, 110\}$.

5.2 Rotate each box $e \in L$ of type $\mathcal{T}_{110} \cup \mathcal{T}_{111}$, if possible, to a box $e' \in \mathcal{T}_{110} \cup \mathcal{T}_{111}$ such 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 10).

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

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

$\mathcal{P}^{011}_{CD} \leftarrow \text{FFDC}^z(L_{011} \cup L_{111}, \mathcal{T}_{011} \cap \mathcal{X}[\frac{1}{3}, p], \mathcal{T}_{111} \cap \mathcal{X}[\frac{1}{2}, 1-p], [(0,0)], [(0,p)])$;
$L_{111} \leftarrow L_{111} \setminus \mathcal{L}(\mathcal{P}^{011}_{CD})$; (See Figure 11(b))
$\mathcal{P}^{101}_{CD} \leftarrow \text{FFDC}^x(L_{101} \cup L_{111}, \mathcal{T}_{101} \cap \mathcal{Y}[\frac{1}{3}, p], \mathcal{T}_{111} \cap \mathcal{Y}[\frac{1}{2}, 1-p], [(0,0)], [(0,p)])$;
$L_{111} \leftarrow L_{111} \setminus \mathcal{L}(\mathcal{P}^{101}_{CD})$;
$\mathcal{P}^{001}_{CD} \leftarrow \text{FFDC}^z(L_{001} \cup L_{111}, \mathcal{T}_{001} \cap \mathcal{X}[\frac{1}{3}, \frac{1}{2}] \cap \mathcal{Y}[\frac{1}{3}, p], \mathcal{T}_{111} \cap \mathcal{Y}[\frac{1}{2}, 1-p], [(0,0), (0, \frac{1}{2})], [(0,p)])$;
$L_{111} \leftarrow L_{111} \setminus \mathcal{L}(\mathcal{P}^{001}_{CD})$;
$\mathcal{P}^{010}_{CD} \leftarrow \text{FFDC}^y(L_{010} \cup L_{111}, \mathcal{T}_{010} \cap \mathcal{Z}[\frac{1}{3}, \frac{1}{2}] \cap \mathcal{X}[\frac{1}{3}, p], \mathcal{T}_{111} \cap \mathcal{X}[\frac{1}{2}, 1-p], [(0,0), (0, \frac{1}{2})], [(0,p)])$;
$L_{111} \leftarrow L_{111} \setminus \mathcal{L}(\mathcal{P}^{010}_{CD})$; (See Figure 11(c))
$\mathcal{P}^{100}_{CD} \leftarrow \text{FFDC}^x(L_{100} \cup L_{111}, \mathcal{T}_{100} \cap \mathcal{Y}[\frac{1}{3}, \frac{1}{2}] \cap \mathcal{Z}[\frac{1}{3}, p], \mathcal{T}_{111} \cap \mathcal{Z}[\frac{1}{2}, 1-p], [(0,0), (0, \frac{1}{2})], [(0,p)])$;
$L_{111} \leftarrow L_{111} \setminus \mathcal{L}(\mathcal{P}^{100}_{CD})$;
$\mathcal{P}_{CD} \leftarrow \mathcal{P}^{011}_{CD} \cup \mathcal{P}^{101}_{CD} \cup \mathcal{P}^{001}_{CD} \cup \mathcal{P}^{010}_{CD} \cup \mathcal{P}^{100}_{CD}$.

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

5.6.1 Generate a packing $\mathcal{P}_{001}$ of the remaining boxes in L_{001} in the following manner.
Let $L^{18}_{001}, \ldots, L^{23}_{001}$ be a partition of L_{001} such that (see Figure 9).

$$L^{18}_{001} \leftarrow L_{001} \cap \mathcal{C}^{xy}\left[\tfrac{1}{3}, \tfrac{1}{2}\,;\, \tfrac{1}{3}, \tfrac{1}{2}\right], \quad L^{19}_{001} \leftarrow L_{001} \cap \mathcal{C}^{xy}\left[\tfrac{1}{3}, \tfrac{1}{2}\,;\, \tfrac{1}{4}, \tfrac{1}{3}\right],$$
$$L^{20}_{001} \leftarrow L_{001} \cap \mathcal{C}^{xy}\left[\tfrac{1}{3}, \tfrac{1}{2}\,;\, 0, \tfrac{1}{4}\right], \quad L^{21}_{001} \leftarrow L_{001} \cap \mathcal{C}^{xy}\left[\tfrac{1}{4}, \tfrac{1}{3}\,;\, \tfrac{1}{3}, \tfrac{1}{2}\right],$$
$$L^{22}_{001} \leftarrow L_{001} \cap \mathcal{C}^{xy}\left[0, \tfrac{1}{4}\,;\, \tfrac{1}{3}, \tfrac{1}{2}\right], \quad L^{23}_{001} \leftarrow L_{001} \cap \mathcal{C}^{xy}\left[0, \tfrac{1}{3}\,;\, 0, \tfrac{1}{3}\right].$$

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

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 a packing $\mathcal{P}_{011}$ of the remaining boxes of L_{011}.
Let $(L^1_{011}, \ldots, L^{17}_{011})$ be a partition of L_{011} defined as follows (see Figure 9).
$L^i_{011} \leftarrow L_{011} \cap \mathcal{Y}[\frac{1}{i+2}, \frac{1}{i+1}], \quad i = 1, \ldots, 16$;
$L^{17}_{011} \leftarrow L_{011} \cap \mathcal{Y}[0, \frac{1}{18}]$;
$\mathcal{P}^i_{011} \leftarrow \text{H3D}^{xy}(\text{NFDH}^y, L^i_{011}, \text{NF}), \quad i = 1, \ldots, 17$;
$\mathcal{P}_{011} \leftarrow \mathcal{P}^1_{011} \cup \ldots \cup \mathcal{P}^{17}_{011}$.

5.7.2 Generate a 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 $\mathcal{P}''_{\text{UNI}} \leftarrow \text{FL}^z_\epsilon(L_{\text{UNI}})$;

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

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.

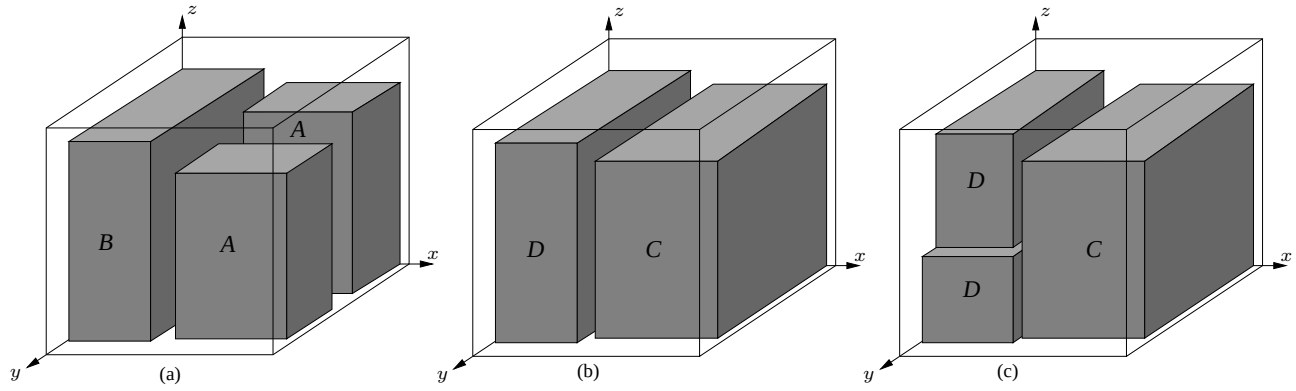

Figure 11: Example of bins: (a) Bin of packing $\mathcal{P}_{AB}$. (b) Bin with boxes of types $\mathcal{T}_{011} \cap \mathcal{X}[\frac{1}{3}, p]$ and $\mathcal{T}_{111} \cap \mathcal{X}[\frac{1}{2}, 1-p]$. (c) Bin with boxes of types $\mathcal{T}_{010} \cap \mathcal{Z}[\frac{1}{3}, \frac{1}{2}] \cap \mathcal{X}[\frac{1}{3}, p]$ and $\mathcal{T}_{111} \cap \mathcal{X}[\frac{1}{2}, 1-p]$.

The next result shows a bound for the asymptotic performance of algorithm $\text{BOX}_{k,\epsilon}$.

Theorem 4.3 *For any list of boxes L for $3\mathrm{BP}^{\mathrm{r}}$, we have*

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

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 a constant that depends on 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 analyse 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 4.2 we have

$$\#(\mathcal{P}_{000}) \leq \frac{1}{8/27}\frac{V(L'_{000})}{abc} + 14. \tag{26}$$

For the packing $\mathcal{P}_{AB}$, 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

$$\#(\mathcal{P}_{AB}) \leq \frac{1}{17/72}\frac{V(L_{AB})}{abc} + 4k + 82. \tag{27}$$

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

$$\#(\mathcal{P}_{CD}) \leq \frac{1}{\frac{1}{8}+\frac{r_1}{4}}\frac{V(L_{CD})}{abc} + 6. \tag{28}$$

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

$$\#(\mathcal{P}_{001}) \leq \frac{1}{17/72}\frac{V(L'_{001})}{abc} + 8. \tag{29}$$

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 hold.

$$\#(\mathcal{P}_{010}) \leq \frac{1}{17/72}\frac{V(L'_{010})}{abc} + 8, \tag{30}$$

$$\#(\mathcal{P}_{100}) \leq \frac{1}{17/72}\frac{V(L'_{100})}{abc} + 8. \tag{31}$$

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 is attained for list L^1_{011}), except perhaps in the last bin of the packing $\mathcal{P}^i_{011}$. Therefore,

$$\#(\mathcal{P}_{011}) \leq \frac{1}{p/2}\frac{V(L'_{011})}{abc} + 17. \tag{32}$$

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

$$\#(\mathcal{P}_{101}) \leq \frac{1}{p/2}\frac{V(L'_{101})}{abc} + 17. \tag{33}$$

From inequalities (26)–(33) and considering that $\frac{p}{2} = \min\{\frac{p}{2}, \frac{17}{72}, \frac{1}{8} + \frac{r_1}{4}\}$, we have

$$\#(\mathcal{P}_{aux}) \leq \frac{1}{p/2}\frac{V(L_{aux})}{abc} + C^k_{aux}. \tag{34}$$

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

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

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 the 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_ϵ in step 5.8.4, we have

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

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

$$\#(\mathcal{P}_{\text{UNI}}) \leq \frac{1}{1/8}\frac{V(L_{\text{UNI}})}{abc} + 1, \tag{35}$$

$$\#(\mathcal{P}_{\text{UNI}}) \leq (1+\epsilon)\text{OPT}(L_{\text{UNI}}) + C^\epsilon_{\text{UNI}}. \tag{36}$$

From inequalities (34)–(36), we can conclude that

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

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 $\beta_{k,\epsilon} = C^k_{aux} + C^\epsilon_{\text{UNI}}$.

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 $\text{b\_area}(\mathcal{P}^r_{001}) \geq \frac{4}{9}$. 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 packing $\mathcal{P}^i_{011}$, 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 $M \subseteq L_{111}$ and the boxes of M 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

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

So, we obtain

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

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 $\beta_{k,\epsilon} = C^k_{aux} + C^\epsilon_{\rm UNI}$.

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 $\lim_{k\to\infty,\epsilon\to 0} \alpha_{k,\epsilon} \leq 4.882\ldots$. □

5 Concluding Remarks

We presented approximation algorithms for three-dimensional packing problems where orthogonal rotations are allowed. We showed that these problems —in their general version— are as hard to approximate as the oriented case. In this case, any approximation algorithm for the case with rotations can be used to obtain an approximation algorithm for the oriented case with the same approximation bound. The two approximation algorithms for three-dimensional packing problems presented in this paper can be implemented to run in time polynomial in the number of items.

References

[1] N. Bansal, J.R. Correa, C. Kenyon, and M. Sviridenko. Bin packing in multiple dimensions: Inapproximability results and approximation schemes. *Mathematics of Operations Research*, 31(1):31–49, 2006.

[2] N. Bansal, X. Han, K. Iwama, M. Sviridenko, and G. Zhang. Harmonic algorithm for 3-dimensional strip packing problem. In *Proc. of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1197–1206, 2007.

[3] N. Bansal and M. Sviridenko. New approximability and inapproximability results for 2-dimensional bin packing. In *Proc. of the 15th ACM-SIAM Symposium on Discrete Algorithms*, pages 189–196, 2004.

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

[5] 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.

[6] 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. Springer-Verlag, New York, 1984.

[7] 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.

[8] 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.

[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] Leah Epstein and Rob van Stee. This side up! *ACM Transactions on Algorithms*, 2(2):228–243, 2006.

[12] 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.

[13] K. Jansen and R. Solis-Oba. An asymptotic approximation algorithm for 3D-strip packing. In *Proc. of the 17th annual ACM-SIAM Symposium on Discrete Algorithms*, pages 143–152, 2006.

[14] Klaus Jansen and Rob van Stee. On strip packing with rotations. In *Proc. of the 37th ACM Symposium on Theory of Computing*, 2005.

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

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

[17] 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.

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

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

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

[21] 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.

[22] F. K. Miyazawa and Y. Wakabayashi. Cube packing. *Theoretical Computer Science*, 297:355–366, 2003.

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

[24] F. K. Miyazawa and Y. Wakabayashi. Packing problems with orthogonal rotations. In *Proc. of the 6th Latin American Theoretical INformatics.*, volume 2976 of *Lecture Notes in Computer Science*, pages 359–368, Buenos Aires, Argentina, 2004. Springer-Verlag.

[25] F. K. Miyazawa and Y. Wakabayashi. Two- and three-dimensional parametric packing. *Computers and Operations Research*, 34:2589–2603, 2007.