Two- and Three-dimensional Parametric Packing[§]

F. K. Miyazawa[*] Y. Wakabayashi[‡]

July 28, 2005

Abstract

We present approximation algorithms for the *two-* and *three-dimensional bin packing* problems and the *three-dimensional strip packing* problem. We consider the special case of these problems in which a parameter m (a positive integer) is given, indicating that each of the dimensions of the items to be packed is at most $\frac{1}{m}$ of the corresponding dimension of the recipient. We analyze the asymptotic performance of these algorithms and exhibit bounds that, to our knowledge, are the best known for this special case.

Key Words: Approximation algorithms, asymptotic performance, packing.

1 Introduction

We present fast asymptotic approximation algorithms for special packing problems, parameterized by a positive integer m. This parameter indicates that the input list consists of items (rectangles, boxes) whose each of its dimension is at most $1/m$ of the respective dimension of the recipient. These problems have many applications, specially in job scheduling.

We consider the following problems:

1. *Two-dimensional Bin Packing* ($2\mathrm{BP}_m$) problem: given a list L of rectangles, each rectangle with dimensions at most $1/m$, and rectangles of unit dimensions $(1, 1)$, called bins, pack the rectangles of L into a minimum number of bins.

2. *Three-dimensional Strip Packing* ($3\mathrm{SP}_m$) problem: given a list L of boxes, with bottom dimensions at most $1/m$, and a box $B = (1, 1, \infty)$, pack the boxes of L into B such that the height of the packing is minimized.

3. *Three-dimensional Bin Packing* ($3\mathrm{BP}_m$) problem: given a list L of boxes with dimensions at most $1/m$, $m \geq 2$, and boxes of dimensions $B = (1, 1, 1)$, also called bins, pack the boxes of L into a minimum number of bins.

[§]This research was partially supported by CNPq (Proc. 478818/03-3, 306526/04-2, 308138/04-0 and 490333/04-4), and ProNEx–FAPESP/CNPq (Proc. 2003/09925-5).

[*]Instituto de Computação — Universidade Estadual de Campinas, Caixa Postal 6176 — 13084–971 — Campinas–SP — Brazil. Corresponding author. fkm@ic.unicamp.br. Fax Number: +55 19 3788-5847.

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

We denote by $1\mathrm{BP}_m$, respectively $2\mathrm{SP}_m$, the One-dimensional Bin Packing and the Two-dimensional Strip Packing problems, both parameterized by m and defined analogously to the above problems.

Given an algorithm $\mathcal{A}$ for one of the previous problems and a list of items L for the respective problem, we denote by $\mathcal{A}(L)$ the height or the number of recipients (depending on which problem is considered) of the packing generated by the algorithm $\mathcal{A}$ applied to the list L. We denote by $\mathrm{OPT}(L)$ the corresponding value of an optimum packing. We say that an algorithm $\mathcal{A}$ has *asymptotic performance bound* α if there exists a constant γ such that for any instance L we have $\mathcal{A}(L) \leq \alpha \ \mathrm{OPT}(L) + \gamma$. When $\gamma = 0$, we say that $\mathcal{A}$ has *performance bound* α.

Although it seems easier to deal with packing of small items (that is, the case m is large), Li and Cheng [11] showed that: *"for any $m \geq 1$, there is no polynomial time algorithm for the $2\mathrm{BP}_m$ problem with performance bound $\alpha < 2$, unless* P=NP". This result shows that, unless we consider the asymptotic case, performance bounds close to 1 are not achievable under the hypothesis that P$\neq$ NP. We show that, as most of the parametric approximation algorithms for packing problems, the algorithms for the problems we focus here have asymptotic performance bounds that tend to 1 as m increases.

Parametric packing problems have been investigated by many authors. For the $1\mathrm{BP}_m$ problem, Johnson *et al.* [9] proved that the asymptotic performance bound of the First Fit (FF) algorithm is $(m+1)/m$, $m \geq 2$. Johnson [7, 8] also presented other algorithms with asymptotic performance bound $(m+3)/(m+2)$. Csirik [5] proved that the First Fit Decreasing (FFD) algorithm has asymptotic performance bound $(m+3)/(m+2)-1/(m(m+1)(m+2))$, when m is odd, and $(m+3)/(m+2)-2/(m(m+1)(m+2))$, when m is even, $m \geq 5$. For the $2\mathrm{SP}_m$ problem, Coffman *et al.* [4] obtained an algorithm with asymptotic performance bound $(m+2)/(m+1)$. For the $2\mathrm{BP}_m$ problem, Frenk and Galambos [6] analyzed the parametric behavior of the HNF (Hybrid Next Fit) algorithm (they did not give an explicit formula). For this problem, we presented in [19] an on-line algorithm with asymptotic performance bound that can be made as close to $(m+2)/m+1/(m+1)^2$, as desired. For the $3\mathrm{SP}_m$ problem, Li and Cheng [10] designed an algorithm with asymptotic performance bound $(m+1)/(m-1)$, $m \geq 2$, and in [19], we presented an on-line algorithm with asymptotic performance bound close to $(m+2)/m+1/(m+1)^2$. In the same paper, we also presented an on-line algorithm for the $3\mathrm{BP}_m$ problem with asymptotic performance bound close to $(m+3)/m+2/m^2+1/(m+1)^2$.

In this paper, we present approximation algorithms for the $2\mathrm{BP}_m$, $3\mathrm{SP}_m$ and $3\mathrm{BP}_m$ problems. The algorithms we describe for the first two problems have asymptotic performance bound $\alpha_m \leq \frac{2m^3+5m^2+5m+2+\sqrt{9m^4+34m^3+41m^2+20m+4}}{2m(m+1)^2}$. For the $3\mathrm{BP}_m$ problem, we show an algorithm with asymptotic performance bound $\beta_m \leq \frac{2m^4+6m^3+9m^2+7m+2+\sqrt{16m^6+76m^5+141m^4+142m^3+85m^2+28m+4}}{2m^2(m+1)^2}$. Both α_m and β_m are decreasing functions of m. These results improve the bound obtained by Li and Cheng [10] and our previous results in [19].

The ideas presented in this paper are general in the sense that they can be extended to other problems or dimensions. In fact, they can be applied to the $1\mathrm{BP}_m$ and $2\mathrm{SP}_m$ problems, but they do not lead to bounds that are better than those given by the specific algorithms that have been designed for these two problems.

For a survey on approximation algorithms for packing problems and some classic algorithms we mention here, the reader is referred to Coffman, Garey and Johnson [2, 3]. Other recent surveys on two-dimensional packing problems have been presented by Lodi, Martello and Monaci [12] and also Lodi, Martello and Vigo [13]. Exact algorithms for the strip packing problem and the two-dimensional bin packing problem have been proposed by Martello, Monaci and Vigo [14] and Martello and Vigo [16], respectively. Martello, Pisinger and Vigo [15] also showed lower bounds for the three-dimensional bin packing problem. For a recent improved typology of cutting and packing problems, the reader is referred to Wäscher, Haußner and Schumann [21].

An extended abstract mentioning the results of this paper has appeared in [20].

2 Notation

All packings considered here are orthogonal and oriented. This means that the items are packed in such a way that the edges of the items are orthogonal or parallel to the edges of the recipient; furthermore, each item is oriented with respect to the x, y, z coordinates and the packing into the recipient agrees with this orientation. We denote by $x(e)$ (respectively $y(e)$, $z(e)$) the length (respectively width, height) of the item e. Since the packings are orthogonal and oriented, we assume, without loss of generality, that the limited dimensions of the recipients have value 1. For the $3\mathrm{SP}_m$ problem, we assume that all items have height not greater than a constant Z.

If L is a list of items consisting of rectangles (respectively boxes), given as an input for one of the problems, then we denote by $S(L)$ the total area of the rectangles (respectively the total bottom area of the boxes) in L.

If $\mathcal{P}$ is a packing, then we denote by $\#(\mathcal{P})$ the number of recipients used by $\mathcal{P}$. Given two packings $\mathcal{P}'$ and $\mathcal{P}''$ for the $3\mathrm{SP}_m$ problem, we denote by $\mathcal{P}'\|\mathcal{P}''$ the *concatenation* of the packings $\mathcal{P}'$ and $\mathcal{P}''$.

We denote by $\mathcal{X}[a,b]$ (respectively $\mathcal{Y}[a,b]$ and $\mathcal{Z}[a,b]$) the set of items e with $a < x(e) \le b$ (respectively $a < y(e) \le b$ and $a < z(e) \le b$). We also use the following notation.

$$
\begin{aligned}
\mathcal{C}\,[a',b'\,;\,a'',b''] &= \mathcal{X}[a',b'] \cap \mathcal{Y}[a'',b''],\\
\mathcal{C}\,[a',b'\,;\,a'',b''\,;\,a''',b'''] &= \mathcal{X}[a',b'] \cap \mathcal{Y}[a'',b''] \cap \mathcal{Z}[a''',b'''],\\
\mathcal{C}_{p,q} &= \mathcal{X}[0,1/p] \cap \mathcal{Y}[0,1/q],\\
\mathcal{C}_{p,q,r} &= \mathcal{X}[0,1/p] \cap \mathcal{Y}[0,1/q] \cap \mathcal{Z}[0,1/r].
\end{aligned}
$$

We use the symbol $\mathcal{X}$ to denote the set of rectangles $r = (x,y)$ such that $x > y$, and the symbol $\mathcal{Y}$ for the set of rectangles $r = (x,y)$ such that $x \le y$.

3 Two-dimensional bin packing problem

In this section we describe an algorithm, which we call $\mathrm{A2B}_m$, for the two-dimensional bin packing problem. Before that, we present three other algorithms used as subroutines: $\mathrm{C2B}_m$, HNF and $\mathrm{A2B}_{p,q}$.

Let us consider first the algorithm $\mathrm{C2B}_m$ (Combine items in a rectangle). This algorithm is called with two parameters: a list L_A and a list L_B. The list L_A consists of rectangles in $\mathcal{C}\left[\frac{1}{m+1}, q_m\,;\,\frac{1}{m+1}, q_m\right]$ and L_B is a list that can be partitioned into two lists L'_B and L''_B, defined as follows: L'_B contains rectangles in $\mathcal{C}\left[\frac{1}{3m}, p_m\,;\,\frac{1}{3m}, \frac{1}{m}\right] \cap \mathcal{Y}$, and L''_B contains rectangles in $\mathcal{C}\left[\frac{1}{3m}, \frac{1}{m}\,;\,\frac{1}{3m}, p_m\right] \cap \mathcal{X}$, where q_m, p_m are such that (*i*) $\frac{1}{m+1} < q_m < \frac{1}{m}$, (*ii*) $p_m = 1 - mq_m$. See Figure 1 (p and q correspond to p_m and q_m).

The algorithm $\mathrm{C2B}_m$ first generates a packing combining items of L_A and of L'_B. At each iteration, the algorithm $\mathrm{C2B}_m$ packs items into a new bin. The packing in each bin is obtained by dividing it into two smaller bins, B_A and B_B, the first with dimensions $(m\,q_m, 1)$ and the second with dimensions $(1 - m\,q_m, 1)$. The algorithm packs up to m^2 items of L_A into the bin B_A. Then, it packs the items of L'_B in the bin B'_B using the algorithm NFD^y (Next Fit Decreasing), that packs the items side by side in the y-direction in decreasing order of width, until an item cannot be packed anymore. The packing of the lists L_A and L'_B continues until all items of one of these lists are totally packed (see Figure 2).

When this happens, the algorithm $\mathrm{C2B}_m$ starts packing the remaining items of L_A (if any) and the items of L''_B in an analogous way. In this case, a bin is partitioned into bins B_A and B'_B, the first with

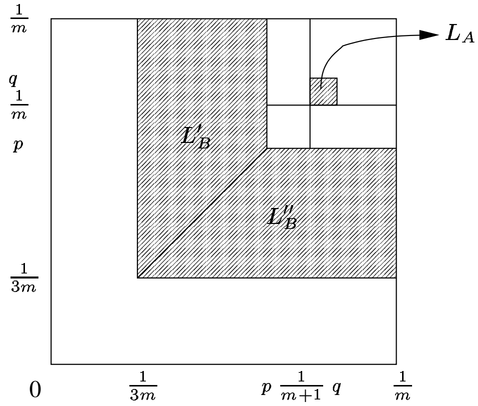

Figure 1: Sets L_A, L'_B and L''_B.

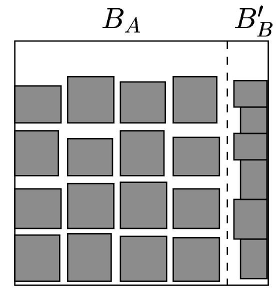

Figure 2: Combining items in L_A and L'_B.

dimensions $(1, m\,q_m)$ and the second with dimensions $(1, 1 - m\,p_m)$. The items of L''_B are packed in the bin B''_B by the algorithm NFD^x (the items are packed side by side in the x-direction in decreasing order of length). When all items of L_A or all items of $L'_B \cup L''_B$ have been totally packed, the algorithm C2B$_m$ returns a pair $(\mathcal{P}_{AB}, L_{AB})$, where $\mathcal{P}_{AB}$ is the packing produced and the list L_{AB} is the set of items packed in $\mathcal{P}_{AB}$.

Before we present a result concerning the area occupied by the packing generated by the algorithm C2B$_m$, we mention a result that will be useful in what follows.

Lemma 3.1 *Let $\mathcal{P}$ be a packing of a list L for the* 2BP$_m$ *problem, such that in each bin, except perhaps in C bins, the rectangles have total area at least s. Then $\#(\mathcal{P}) \le S(L)/s + C$.*

We call the value s, in the lemma above an *area guarantee* of the packing $\mathcal{P}$. Using this lemma, it is easy to prove the next result.

Lemma 3.2 *If $\mathcal{P}_{AB}$ is a packing of $L_{AB} \subseteq L_A \cup L_B$ generated by the algorithm* C2B$_m$*, applied to the sublists L_A and L_B, then $\#(\mathcal{P}_{AB}) \le S(L_{AB}) \Big/ \left(\left(\frac{m}{m+1}\right)^2 + \frac{1}{3\,(m+1)} \right) + 4$.*

Proof. We consider separately the area occupied by the items of L_A and the items of L_B, in each bin. In the packing $\mathcal{P}_{AB}$, each of the bins that were used, contains at least m^2 items of L_A, except perhaps the last bin. Thus, we can guarantee an area occupation of at least $\left(\frac{m}{m+1}\right)^2$ in these bins. Now consider the area occupied by the items of L'_B in one bin of type B'_B. Since the algorithm NFD^y sorts by non-increasing order of width, the items of width in $\left(\frac{1}{m+1}, \frac{1}{m}\right]$ are packed before the items with width in $\left(0, \frac{1}{m+1}\right]$. The packing of the items with width in $\left(\frac{1}{m+1}, \frac{1}{m}\right]$ occupies an area of at least $m\frac{1}{3m}\frac{1}{m+1} = \frac{1}{3(m+1)}$ in each bin of type B'_B (except perhaps one). The packing of the items with width in $\left(0, \frac{1}{m+1}\right]$, occupies an area of at least $\left(1 - \frac{1}{m+1}\right)\frac{1}{3m} = \frac{1}{3(m+1)}$ in each bin of type B'_B (except possibly one). So, the area occupation in each bin is at least $\left(\frac{m}{m+1}\right)^2 + \frac{1}{3(m+1)}$, except perhaps in two bins. The analysis for bins with items of L_A and items of L''_B is analogous. The proof follows from Lemma 3.1. □

Another subroutine used by the algorithm A2B$_m$ is the algorithm HNF (Hybrid Next Fit). Given an input list L, the algorithm HNF sorts the items in L in non-increasing order of width. Then, it applies the algorithm NF^x (Next Fit) for the 1BP$_m$ problem, to pack each item of L into bins of unit length, considering only the length dimension. Each bin B generated by the algorithm NF is considered as a level of width $W(B) = \max\{y(e) : e \in B\}$. Then, the algorithm HNF uses the algorithm NFD^y to pack the levels into bins of unit width, now considering only the width dimension. We denote by HNF^x (HNF^y) the HNF algorithm which generates the application of the NF algorithm in the length (width) dimension and the application of the NFD algorithm in the width (length) dimension.

The following result holds for the algorithm HNF.

Lemma 3.3 *For any list of rectangles $L \subseteq \mathcal{C}_{p,q}$, where $p, q \geq 2$, we have*

$$\mathrm{HNF}(L) \leq \frac{pq}{(p-1)(q-1)} S(L) + 2.$$

Proof. Consider the algorithm HNF^x. The analysis for the algorithm HNF^y is analogous. Let $L_1, \ldots, L_k$ be the levels generated by the application of the NF^x algorithm, where $W(L_i) \geq W(L_{i+1})$. Since the length of an item is at least $1/p$, each level has a length occupation of at least $1 - 1/p$, except perhaps the last. Therefore,

$$\begin{aligned} S(L) &\geq S(L_1) + S(L_2) + \cdots + S(L_{k-1}) \\ &\geq (1 - 1/p)W(L_2) + (1 - 1/p)W(L_2) + \cdots + (1 - 1/p)W(L_k) \\ &= (1 - 1/p)\left(\sum_{i=1}^{k} W(L_i) - W(L_1)\right). \end{aligned} \qquad (1)$$

The levels are packed into two-dimensional bins using the algorithm NFD^y. Since each level has width at most $1/q$, each bin (except perhaps the last) has a width occupation of at least $1 - 1/q$. Therefore,

$$(1 - 1/q)(\mathrm{HNF}^x(L) - 1) \leq \sum_{i=1}^{k} W(L_i). \qquad (2)$$

From inequalities (1) and (2), we have

$$\mathrm{HNF}^x(L) \leq \frac{pq}{(p-1)(q-1)} S(L) + \frac{1}{q-1} + 1.$$

□

In what follows, we present an algorithm which leads to packings with better area guarantee for the $2\mathrm{BP}_m$ problem. It uses list partition and the algorithm HNF.

ALGORITHM $\mathrm{A2B}_{p,q}$
Input: List of items $L \subseteq \mathcal{C}_{p,q}$.
Output: Packing of L into unit bins.

1. Partition the list L into sublists

$$\begin{aligned} L_1 &\leftarrow L \cap \mathcal{C}\left[\tfrac{1}{p+1}, \tfrac{1}{p}\,;\, \tfrac{1}{q+1}, \tfrac{1}{q}\right]; \quad L_2 \leftarrow L \cap \mathcal{C}\left[0, \tfrac{1}{p+1}\,;\, \tfrac{1}{q+1}, \tfrac{1}{q}\right]; \\ L_3 &\leftarrow L \cap \mathcal{C}\left[\tfrac{1}{p+1}, \tfrac{1}{p}\,;\, 0, \tfrac{1}{q+1}\right]; \quad L_4 \leftarrow L \cap \mathcal{C}\left[0, \tfrac{1}{p+1}\,;\, 0, \tfrac{1}{q+1}\right]. \end{aligned}$$

2. $\mathcal{P}_i \leftarrow \mathrm{HNF}(L_i)$ for $i = 1, 4$.
3. $\mathcal{P}_2 \leftarrow \mathrm{HNF}^x(L_2)$; $\mathcal{P}_3 \leftarrow \mathrm{HNF}^y(L_3)$; $\mathcal{P} \leftarrow \mathcal{P}_1 \| \ldots \| \mathcal{P}_4$.
4. Return $\mathcal{P}$.

We are now ready to describe the algorithm $\mathrm{A2B}_m$.

ALGORITHM A2B$_m$
Input: List of items $L \subseteq \mathcal{C}_{m,m}$.
Output: Packing $\mathcal{P}$ of L into unit bins.

1. $p = p(m) \leftarrow \frac{\sqrt{9m^4+34m^3+41m^2+20m+4}-m^2-3m-2}{2m(m^2+3m+2)}$; $\quad q = q(m) \leftarrow \frac{1-p}{m}$;

2. $L_A \leftarrow L \cap \mathcal{C}\left[\frac{1}{m+1}, q\,;\ \frac{1}{m+1}, q\right]$;

$L_B \leftarrow L \cap \left(\mathcal{C}\left[\frac{1}{3m}, p\,;\ \frac{1}{3m}, \frac{1}{m}\right] \cup \mathcal{C}\left[\frac{1}{3m}, \frac{1}{m}\,;\ \frac{1}{3m}, p\right]\right)$.

3. $(\mathcal{P}_{AB}, L_{AB}) \leftarrow \text{C2B}_m(L_A, L_B)$; $L \leftarrow L \setminus L_{AB}$;

4. If $L_A \subseteq L_{AB}$ then

$$L_1 = L \cap \mathcal{C}\left[\tfrac{1}{m+1}, \tfrac{1}{m}\,;\ \tfrac{1}{m+1}, \tfrac{1}{m}\right]; \qquad \mathcal{P}_1 \leftarrow \text{HNF}(L_1);$$
$$L_2 = L \cap \mathcal{C}\left[0, \tfrac{1}{m+1}\,;\ 0, \tfrac{1}{m}\right] \cap \mathcal{Y}; \qquad \mathcal{P}_2 \leftarrow \text{A2B}_{m+1,m}(L_2);$$
$$L_3 = L \cap \mathcal{C}\left[0, \tfrac{1}{m}\,;\ 0, \tfrac{1}{m+1}\right] \cap \mathcal{X}; \qquad \mathcal{P}_3 \leftarrow \text{A2B}_{m,m+1}(L_3);$$
$$\mathcal{P}_{opt} \leftarrow \mathcal{P}_1 \| \mathcal{P}_{AB}; \quad \mathcal{P}_{aux} \leftarrow \mathcal{P}_2 \| \mathcal{P}_3;$$

5. else

$$L_1 = L \cap \mathcal{C}\left[\tfrac{1}{m+1}, \tfrac{1}{m}\,;\ \tfrac{1}{m+1}, \tfrac{1}{m}\right]; \qquad L_2 = L \cap \mathcal{C}\left[p, \tfrac{1}{m+1}\,;\ p, \tfrac{1}{m+1}\right];$$
$$L_3 = L \cap \mathcal{C}\left[p, \tfrac{1}{m+1}\,;\ \tfrac{1}{m+1}, \tfrac{1}{m}\right]; \qquad L_4 = L \cap \mathcal{C}\left[\tfrac{1}{m+1}, \tfrac{1}{m}\,;\ p, \tfrac{1}{m+1}\right];$$
$$L_5 = L \cap \mathcal{C}\left[0, \tfrac{1}{3m}\,;\ 0, \tfrac{1}{m}\right] \cap \mathcal{Y}; \qquad L_6 = L \cap \mathcal{C}\left[0, \tfrac{1}{m}\,;\ 0, \tfrac{1}{3m}\right] \cap \mathcal{X};$$
$$\mathcal{P}_i \leftarrow \text{HNF}(L_i), \quad i = 1, \ldots, 4; \qquad \mathcal{P}_5 \leftarrow \text{A2B}_{3m,m}(L_5);$$
$$\mathcal{P}_6 \leftarrow \text{A2B}_{m,3m}(L_6); \ \mathcal{P}_{opt} \leftarrow \mathcal{P}_1 \| \mathcal{P}_{AB}; \quad \mathcal{P}_{aux} \leftarrow \mathcal{P}_2 \| \ldots \| \mathcal{P}_6.$$

6. $\mathcal{P} \leftarrow \mathcal{P}_{opt} \| \mathcal{P}_{aux}$.

7. Return $\mathcal{P}$.

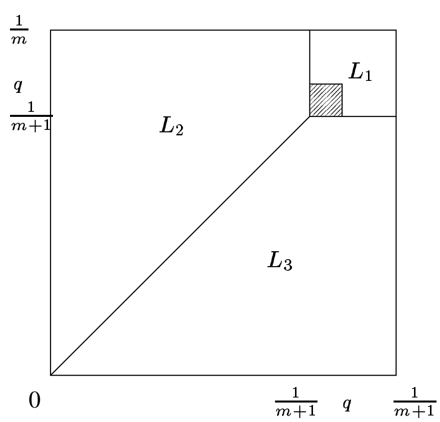

Figure 3: List subdivision when $L_A \subseteq L_{AB}$.

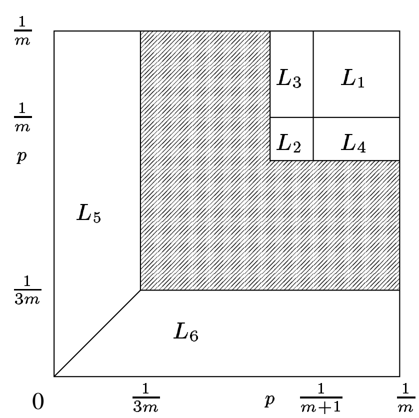

Figure 4: List subdivision when $L_B \subseteq L_{AB}$.

Lemma 3.4 *For any list of rectangles $L \subseteq \mathcal{C}_{p,q}$, we have*

$$\text{A2B}_{p,q}(L) \leq \frac{(p+1)(q+1)}{pq} S(L) + 5.$$

Proof. The proof follows easily from the fact that for each sublist L_i ($i = 1, \dots, 4$) we have an area guarantee of at least $pq/(p+1)(q+1)$. □

The next result will be useful in this section and in the others (see [19]).

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

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

The following result holds for the algorithm A2B_m.

Theorem 3.6 *For any list $L \subseteq \mathcal{C}_{m,m}$, we have*

$$\text{A2B}_m(L) \leq \alpha_m \ \text{OPT}(L) + 18,$$

where $\alpha_m = (2m^3 + 5m^2 + 5m + 2 + \sqrt{9m^4 + 34m^3 + 41m^2 + 20m + 4})/(2m(m+1)^2)$.

Proof. First, note that the packing $\mathcal{P}_1 \| \mathcal{P}_{AB}$ is an asymptotic optimum packing for the list $L_{opt} := L_1 \cup L_{AB}$. It suffices to note that in all bins of $\mathcal{P}_1 \| \mathcal{P}_{AB}$, except perhaps in 2 bins, there are at least m^2 rectangles of $L \cap \mathcal{C}\left[\frac{1}{m+1}, \frac{1}{m} \,;\, \frac{1}{m+1}, \frac{1}{m}\right]$ in each bin. Moreover, we cannot have more than m^2 rectangles of this type in each bin. That is,

$$\#(\mathcal{P}_{opt}) \leq \text{OPT}(L) + 2. \tag{3}$$

From Lemma 3.2, we have that the following inequality holds.

$$\#(\mathcal{P}_{AB}) \leq S(L_{AB}) \Big/ \left(\left(\frac{m}{m+1} \right)^2 + \frac{1}{3(m+1)} \right) + 4. \tag{4}$$

Now we analyse two cases, according to steps 4 and 5 in the description of the algorithm.
Case 1: $L_A \subseteq L_{AB}$ (all rectangles of L_A have been packed in $\mathcal{P}_{AB}$).

We analyze each of the packings $\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, \mathcal{P}_{opt}$ and $\mathcal{P}_{aux}$. The packing $\mathcal{P}_1$ is generated by applying the algorithm HNF to the list L_1. Since each bin of $\mathcal{P}_1$ has m^2 rectangles, except perhaps the last, and the area of each rectangle is at least $q/(m+1)$, the occupied area is at least $m^2\,q/(m+1)$. Thus, by Lemma 3.1 we can conclude that

$$\#(\mathcal{P}_1) \leq \frac{m+1}{m^2 q} S(L_1) + 1. \tag{5}$$

Since $\frac{m^2 q}{m+1} \leq \left(\frac{m}{m+1}\right)^2 + \frac{1}{3(m+1)}$, from (4) and (5) we have

$$\#(\mathcal{P}_{opt}) \leq \frac{m+1}{m^2 q} S(L_{opt}) + 5. \tag{6}$$

For the packings $\mathcal{P}_2$ and $\mathcal{P}_3$, the analysis is also based on the area guarantee obtained in each packing. From Lemma 3.4, we have

$$\#(\mathcal{P}_i) \leq \frac{m+2}{m} S(L_i) + 5, \quad \text{for} \quad i = 2, 3.$$

Therefore,

$$\#(\mathcal{P}_{aux}) \leq \frac{m+2}{m} S(L_{aux}) + 10. \tag{7}$$

Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be defined as $\mathcal{H}_1 := \#(\mathcal{P}_{opt}) - 5$ and $\mathcal{H}_2 := \#(\mathcal{P}_{aux}) - 10$.

Since $\mathrm{OPT}(L) \geq S(L)$, from (6), (7) and the definition of $\mathcal{H}_1$ and $\mathcal{H}_2$, we have

$$\begin{aligned} \mathrm{OPT}(L) &\geq S(L) = S(L_{opt}) + S(L_{aux}) \\ &\geq \frac{m^2 q}{m+1}\,\mathcal{H}_1 + \frac{m}{m+2}\,\mathcal{H}_2. \end{aligned} \tag{8}$$

From (8) and using the definition of $\mathcal{H}_1$ in (3) we obtain

$$\mathrm{OPT}(L) \geq \max\left\{\mathcal{H}_1,\ \frac{m^2 q}{m+1}\,\mathcal{H}_1 + \frac{m}{m+2}\mathcal{H}_2\right\}. \tag{9}$$

As $\#(\mathcal{P}) = \#(\mathcal{P}_{opt}) + \#(\mathcal{P}_{aux}) = (\mathcal{H}_1 + \mathcal{H}_2) + 15$, we have

$$\#(\mathcal{P}) \leq \alpha'_m\ \mathrm{OPT}(L) + 15, \tag{10}$$

where $\alpha'_m = (\mathcal{H}_1 + \mathcal{H}_2)/\max\left\{\mathcal{H}_1, \frac{m^2 q}{m+1}\mathcal{H}_1 + \frac{m}{m+2}\mathcal{H}_2\right\}$.

Case 2: $L_B \subseteq L_{AB}$ (all rectangles of L_B have been packed in $\mathcal{P}_{AB}$).

In this case, the proof is analogous to the previous case. Thus we omit the details and simply mention the inequalities that can be obtained.

$$\begin{aligned} \#(\mathcal{P}_1) &\leq \left(\frac{m+1}{m}\right)^2 S(L_1) + 1; && (11) \\ \#(\mathcal{P}_{opt}) &\leq \left(\frac{m+1}{m}\right)^2 S(L_{opt}) + 5; && (12) \\ \#(\mathcal{P}_i) &\leq \frac{1}{m} S(L_i) + 1, \quad \text{for} \quad i = 2,3,4; && (13) \\ \#(\mathcal{P}_i) &\leq \frac{(3m+1)(m+1)}{3m^2} S(L_i) + 5; \quad \text{for} \quad i = 5,6. && (14) \\ & && (15) \end{aligned}$$

Since $mp \leq \frac{3m^2}{(3m+1)(m+1)}$, from (13) and (14) we obtain

$$\#(\mathcal{P}_{aux}) \leq \frac{1}{mp} S(L_{aux}) + 13. \tag{16}$$

Defining $\mathcal{H}_1$ and $\mathcal{H}_2$ as $\mathcal{H}_1 := \#(\mathcal{P}_{opt}) - 5$ and $\mathcal{H}_2 := \#(\mathcal{P}_{aux}) - 13$, and proceeding as in case 1, we have

$$H(\mathcal{P}) \leq \alpha''_m\ \mathrm{OPT}(L) + 18, \tag{17}$$

where $\alpha''_m = (\mathcal{H}_1 + \mathcal{H}_2)/\max\left\{\mathcal{H}_1, \left(\frac{m}{m+1}\right)^2 \mathcal{H}_1 + m\mathcal{H}_2\right\}$.

Now using Lemma 3.5 we can obtain bounds for α_m and α''_m, and conclude that both are at most α_m. This completes the proof of the theorem. We observe that the values of p and q were defined in such a way that in both cases (1 and 2) we obtain the same asymptotic bound. □

4 Three-dimensional strip packing problem

In this section we use basically the same list subdivisions used in the algorithm $\mathrm{A2B}_m$ to obtain an algorithm for the $3\mathrm{SP}_m$ problem with the same asymptotic performance bound.

We first present the subroutines used by the main algorithm. One of the subroutines is the well-known algorithm NFDH (Next Fit Decreasing Height), described in [11]. We denote by NFDH^x the version of the NFDH that packs the boxes side by side in the x-direction, and by NFDH^y the version that generates strips in the y-direction.

Now, let us describe the subroutine to pack lists $L \subseteq \mathcal{C}_{p,q}$, which we call $\mathrm{PQ}_{p,q}$. This algorithm sorts the list L in decreasing order of height and partition it into lists $L_1, \ldots, L_k$, such that $L = L_1 \| \ldots \| L_k$ and $S(L_i) \leq \left(\frac{p-1}{p}\right)\left(\frac{q-1}{q}\right)$, for $i = 1, \ldots, k$ and $S(L_i) + S(\mathit{first}(L_{i+1})) > \left(\frac{p-1}{p}\right)\left(\frac{q-1}{q}\right)$, for $i = 1, \ldots, k-1$. Then it packs each list L_i into only one level, using the algorithm HNF. Finally, it returns a packing that is the concatenation of all the levels that were generated.

In what follows we shall use the next result, that holds for level-oriented packings (see [17, 18]).

Lemma 4.1 *Let L be an instance of $3\mathrm{SP}_m$ 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 $S(N_i) \geq s$ for a given constant $s > 0$, $i = 1, \ldots, v-1$. Then $H(\mathcal{P}) \leq V(L)/s + Z$.*

The value s in the above lemma is called *volume guarantee* of the packing $\mathcal{P}$. We are now interested in the volume guarantee of the packing produced by the algorithm $\mathrm{PQ}_{p,q}$. To obtain this, we need the next result, that is an extension of the result presented in [10]. We leave the proof to the reader.

Lemma 4.2 *If $L \subseteq \mathcal{C}_{p,q}$ is a list with $0 < S(L) \leq \left(\frac{p-1}{p}\right)\left(\frac{q-1}{q}\right)$, then $\mathrm{HNF}(L) = 1$.*

Lemma 4.3 *If $L \subseteq \mathcal{C}_{p,q}$ is an instance for the $3\mathrm{SP}_m$ problem, then*

$$\mathrm{PQ}_{p,q}(L) \leq \frac{pq}{pq - p - q} V(L) + Z.$$

Proof. Since the packing is level-oriented, each level with area occupation of at least $\left(\frac{p-1}{p}\right)\left(\frac{q-1}{q}\right) - \frac{1}{pq}$, the result follows by applying Lemma 4.1. □

We describe now an algorithm, called COL, that produces an asymptotic optimum packing for items in a list $L' \subseteq \mathcal{C}\left[\frac{1}{m+1}, \frac{1}{m} ; \frac{1}{m+1}, \frac{1}{m}\right]$. This algorithm generates a packing consisting of m^2 columns. Initially, all these columns are empty. The items are then considered in the order given by L' and packed in a column of smallest height. The following holds for this algorithm.

Lemma 4.4 *If $L \subseteq \mathcal{C}\left[\frac{1}{m+1}, \frac{1}{m} ; \frac{1}{m+1}, \frac{1}{m}\right]$ then*

$$\mathrm{COL}(L) \leq \left(\frac{m+1}{m}\right)^2 S(L) + Z \quad \textit{and} \quad \mathrm{COL}(L) \leq \mathrm{OPT}(L) + Z.$$

We describe now the algorithm $\mathrm{A3S}_{p,q}$ to pack lists $L \subseteq \mathcal{C}_{p,q}$.

ALGORITHM A3S$_{p,q}$
Input: List of items $L \subseteq \mathcal{C}_{p,q}$.
Output: Packing $\mathcal{P}$ of L into a box with unit bottom.

1. Partition the list L into sublists (see Figure 5):

$$L_1 \leftarrow L \cap \mathcal{C}\left[\tfrac{1}{p+1}, \tfrac{1}{p}\ ;\ \tfrac{1}{q+1}, \tfrac{1}{q}\right]; \qquad L_2 \leftarrow L \cap \mathcal{C}\left[0, \tfrac{1}{p+1}\ ;\ \tfrac{1}{q+1}, \tfrac{1}{q}\right];$$
$$L_3 \leftarrow L \cap \mathcal{C}\left[\tfrac{1}{p+1}, \tfrac{1}{p}\ ;\ 0, \tfrac{1}{q+1}\right]; \qquad L_4 \leftarrow L \cap \mathcal{C}\left[0, \tfrac{1}{p+1}\ ;\ \tfrac{1}{q+2}, \tfrac{1}{q+1}\right] \cap \mathcal{X}\,;$$
$$L_5 \leftarrow L \cap \mathcal{C}\left[\tfrac{1}{p+2}, \tfrac{1}{p+1}\ ;\ 0, \tfrac{1}{q+1}\right] \cap \mathcal{Y}\,; \qquad L_6 \leftarrow L \cap \mathcal{C}\left[0, \tfrac{1}{p+2}\ ;\ 0, \tfrac{1}{q+2}\right].$$

2. $\mathcal{P}_i \leftarrow \mathrm{NFDH}^x(L_i)$ for $i = 1, 2, 4$.
3. $\mathcal{P}_i \leftarrow \mathrm{NFDH}^y(L_i)$ for $i = 3, 5$.
4. $\mathcal{P}_6 \leftarrow \mathrm{PQ}_{p+2,q+2}(L_6)$.
5. $\mathcal{P} \leftarrow L_1 \| \dots \| \mathcal{P}_6$.
6. Return $\mathcal{P}$.

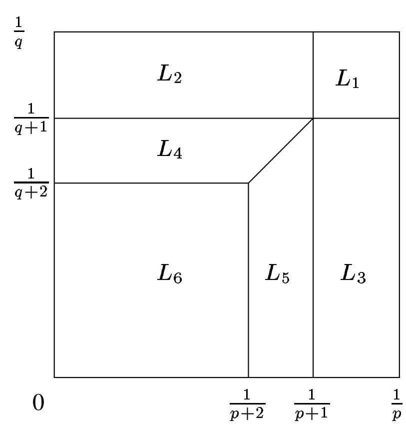

Figure 5: Partition of the input list performed by algorithm A3S$_{p,q}$.

Lemma 4.5 *If* $L \subseteq \mathcal{C}_{p,q}$ *then* $\mathrm{A3S}_{p,q}(L) \leq \frac{(p+1)(q+1)}{pq} V(L) + 6Z$.

Proof. It suffices to note that the packing of each sublist has a volume guarantee of at least $pq/((p+1)(q+1))$. The additive term $6Z$ comes from the additive terms of the inequalities obtained for the lists $L_1, \dots, L_6$. □

The equivalent result for Lemma 3.2 can be obtained using a similar algorithm for C2B$_m$, which we denote by C3S$_m$. To generate a combined packing, the bin $B = (1, 1, \infty)$ is divided into smaller bins. In one case the bin B is divided into bins B'_A and B'_B and in the other case, the bin is divided in bins B''_A and B''_B. The dimensions of these bins are the following

$$B'_A = (mq, 1, \infty)\,, \quad B'_B = (1 - mq, 1, \infty)\,,$$
$$B''_A = (1, mq, \infty)\,, \quad B''_B = (1, 1 - mq, \infty).$$

The algorithm $\mathrm{C3S}_m$ receives as input two sublists: a list $L_A \subset \mathcal{C}\left[\frac{1}{m+1}, q\, ;\, \frac{1}{m+1}, q\right]$ and a list $L_B = L'_B \| L''_B$, where $L'_B \subset \mathcal{C}\left[\frac{1}{3m}, p\, ;\, \frac{1}{3m}, \frac{1}{m}\right] \cap \mathcal{Y}$ and $L''_B \subseteq \mathcal{C}\left[\frac{1}{3m}, \frac{1}{m}\, ;\, \frac{1}{3m}, p\right] \cap \mathcal{X}$. The algorithm first generates a packing $\mathcal{P}'_{AB}$ combining items of L_A and L'_B and then, a packing $\mathcal{P}''_{AB}$ combining remaining items of L_A and items of L''_B. The final packing is the concatenation of $\mathcal{P}'_{AB}$ and $\mathcal{P}''_{AB}$.

The packing $\mathcal{P}'_{AB}$ is produced as follows: Start empty packings into boxes B'_A and B'_B. Note that B'_A and B'_B are parts of the same bin B. The packing in the bin B'_A is performed by the algorithm COL. To generate the packing in the bin B'_B, the list L'_B is first divided into two sublists L'_{B1} and L'_{B2}. The list L'_{B1} contains the items with width in $\left(\frac{1}{m+1}, \frac{1}{m}\right]$ and the list L'_{B2} contains the items with width in $\left(0, \frac{1}{m+1}\right]$. Both lists are sorted in decreasing order of height. The list $L'_B = L'_{B1} \| L'_{B2}$ is packed by the algorithm NF^y, for the $1\mathrm{BP}_m$ problem, considering only the width dimension. In this case, each bin leads to a level of the packing in the bin B'_B. At each iteration the algorithm choose the packing with smallest height. If it is the packing in the box B'_A, it packs the next box of L_A into the bin B'_A using the algorithm COL. If the chosen packing is in the box B'_B, the algorithm packs a new level of items in L'_B using the algorithm NF^y. The algorithm stops when the list L_A or the list L'_B is totally packed. Then, it continues to pack the remaining items of L_A and the items in L''_B in analogous way, in this case with bins B'_y and B''_y.

Lemma 4.6 *Let $\mathcal{P}_{AB}$ be a packing of $L_{AB} \subseteq L_A \cup L_B$ generated by algorithm $\mathrm{C3S}_m$ applied to sublists L_A and L_B. Then $\#(\mathcal{P}_{AB}) \leq \frac{1}{\left(\frac{m}{m+1}\right)^2 + \frac{1}{3(m+1)}} S(L_{AB}) + 4Z$.*

We are now ready to describe the algorithm $\mathrm{A3S}_m$ for the $3\mathrm{SP}_m$ problem. It uses the same list subdivision performed by the algorithm $\mathrm{A2B}_m$, generates partial packings for each sublist, and produces a final packing that is the concatenation of these partial packings.

ALGORITHM $\mathrm{A3S}_m(L)$
Input: List of boxes $L \subseteq \mathcal{C}_{m,m}$.
Output: Packing $\mathcal{P}$ of L into a bin $B = (1, 1, \infty)$.

1. $p = p(m) \leftarrow \frac{\sqrt{9m^4+34m^3+41m^2+20m+4} - m^2 - 3m - 2}{2m(m^2+3m+2)}$; $\quad q = q(m) \leftarrow \frac{1-p}{m}$;

2. $L_A \leftarrow L \cap \mathcal{C}\left[\frac{1}{m+1}, q\, ;\, \frac{1}{m+1}, q\right]$;
$L_B \leftarrow L \cap \left(\mathcal{C}\left[\frac{1}{3m}, p\, ;\, \frac{1}{3m}, \frac{1}{m}\right] \cup \mathcal{C}\left[\frac{1}{3m}, \frac{1}{m}\, ;\, \frac{1}{3m}, p\right]\right)$.

3. $(\mathcal{P}_{AB}, L_{AB}) \leftarrow \mathrm{C3S}_m(L_A, L_B)$;
$L \leftarrow L \setminus L_{AB}$.

4. If $L_A \subseteq L_{AB}$ then (see the subdivision in Figure 3).

$$L_1 = L \cap \mathcal{C}\left[\frac{1}{m+1}, \frac{1}{m}\, ;\, \frac{1}{m+1}, \frac{1}{m}\right], \quad \mathcal{P}_1 \leftarrow \mathrm{COL}(L_1)\, ;$$
$$L_2 = L \cap \mathcal{C}\left[0, \frac{1}{m+1}\, ;\, 0, \frac{1}{m}\right] \cap \mathcal{Y}, \quad \mathcal{P}_2 \leftarrow \mathrm{A3S}_{m+1,m}(L_2)\, ;$$
$$L_3 = L \cap \mathcal{C}\left[0, \frac{1}{m}\, ;\, 0, \frac{1}{m+1}\right] \cap \mathcal{X}, \quad \mathcal{P}_3 \leftarrow \mathrm{A3S}_{m,m+1}(L_3)\, ;$$
$$\mathcal{P}_{opt} \leftarrow \mathcal{P}_1 \| \mathcal{P}_{AB};$$
$$\mathcal{P}_{aux} \leftarrow \mathcal{P}_2 \| \mathcal{P}_3.$$

5. else (see the subdivison in Figure 4)

$$L_1 = L \cap \mathcal{C}\left[\tfrac{1}{m+1}, \tfrac{1}{m}\ ;\ \tfrac{1}{m+1}, \tfrac{1}{m}\right]; \qquad L_2 = L \cap \mathcal{C}\left[p, \tfrac{1}{m+1}\ ;\ p, \tfrac{1}{m+1}\right];$$
$$L_3 = L \cap \mathcal{C}\left[p, \tfrac{1}{m+1}\ ;\ \tfrac{1}{m+1}, \tfrac{1}{m}\right]; \qquad L_4 = L \cap \mathcal{C}\left[\tfrac{1}{m+1}, \tfrac{1}{m}\ ;\ p, \tfrac{1}{m+1}\right];$$
$$L_5 = L \cap \mathcal{C}\left[0, \tfrac{1}{3m}\ ;\ 0, \tfrac{1}{m}\right] \cap \mathcal{Y}; \qquad L_6 = L \cap \mathcal{C}\left[0, \tfrac{1}{m}\ ;\ 0, \tfrac{1}{3m}\right] \cap \mathcal{X};$$
$$\mathcal{P}_1 \leftarrow \mathrm{COL}(L_1); \qquad \mathcal{P}_i \leftarrow \mathrm{NFDH}^x(L_i), \quad i = 2,3,4;$$
$$\mathcal{P}_5 \leftarrow \mathrm{A3S}_{3m,m}(L_5); \qquad \mathcal{P}_6 \leftarrow \mathrm{A3S}_{m,3m}(L_6);$$
$$\mathcal{P}_{opt} \leftarrow \mathcal{P}_1 \| \mathcal{P}_{AB}; \qquad \mathcal{P}_{aux} \leftarrow \mathcal{P}_2 \| \ldots \| \mathcal{P}_6;$$

6. $\mathcal{P} \leftarrow \mathcal{P}_{opt} \| \mathcal{P}_{aux}$;
7. Return $\mathcal{P}$.

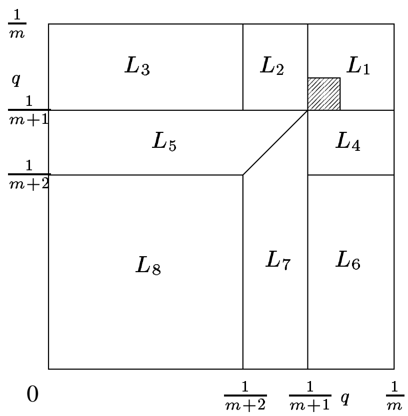

Figure 6: List subdivision when $L_A \subseteq L_{AB}$.

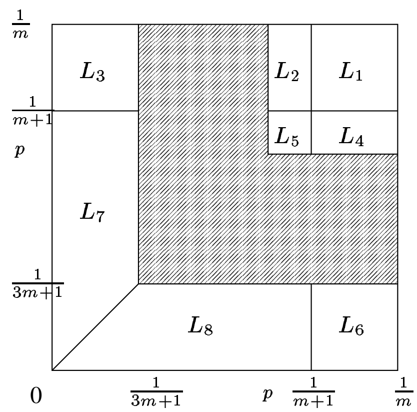

Figure 7: List subdivision when $L_B \subseteq L_{AB}$.

To prove the next result on the algorithm $\mathrm{A3S}_m$, we can use basically the inequalities presented in the proof of Theorem 3.6 with the additive term multiplied by Z. The only difference is for the two packings obtained with the algorithm $\mathrm{A3S}_{p,q}$. In this case, the additive constant is $6Z$ (see Lemma 4.5). The corresponding additive constant for the algorithm $\mathrm{A2B}_{p,q}$ is 5 (see Lemma 3.4). We leave to the reader the proof of the next result.

Theorem 4.7 *For any list L of boxes, $L \subset \mathcal{C}_{m,m}$ we have*

$$\mathrm{A3S}_m(L) \le \alpha_m\ \mathrm{OPT}(L) + 20\,Z,$$

where $\alpha_m = \left(2m^3 + 5m^2 + 5m + 2 + \sqrt{9m^4 + 34m^3 + 41m^2 + 20m + 4}\right) \big/ \left(2m(m+1)^2\right)$.

5 Three-dimensional bin packing problem

We describe in this section the algorithm $\mathrm{A3B}_m$ for the $3\mathrm{BP}_m$ problem. This algorithm uses two other algorithms as subroutines: $\mathrm{H3B}_{p,q,r}$ and $\mathrm{C3B}_m$.

The algorithm $\mathrm{H3B}_{p,q,r}$ is basically the algorithm H3D (Hybrid 3D), described in [19], that uses the same strategy used by the algorithm HFF (Hybrid First Fit) presented by Chung, Garey and Johnson [1]. The algorithm H3D calls two other algorithms: $\mathcal{A}_{3sp}$ and $\mathcal{A}_{1bp}$. The algorithm $\mathcal{A}_{3sp}$ can be any level-oriented algorithm for the $3\mathrm{SP}_m$ problem and $\mathcal{A}_{1bp}$ can be any algorithm for the $1\mathrm{BP}_m$ problem. First, it generates a packing of L divided into levels using the algorithm $\mathcal{A}_{3sp}$; then it uses the algorithm $\mathcal{A}_{1bp}$ to pack into bins the levels that were generated. For each choice of the algorithms $\mathcal{A}_{3sp}$ and $\mathcal{A}_{1bp}$, we obtain a different algorithm. The algorithm $\mathrm{H3B}_{p,q,r}$ corresponds to the algorithm H3D, where $\mathcal{A}_{3sp} = \mathrm{A3S}_{p,q}$ and $\mathcal{A}_{1bp} = \mathrm{FFD}$. The following result can be proved using arguments based on minimum area occupation in each level guaranteed by the algorithm FFD and volume guarantee of the algorithm $\mathrm{A3S}_{p,q}$.

Lemma 5.1 *If $L \subseteq \mathcal{C}_{p,q,r}$ then*

$$\mathrm{H3B}_{p,q,r}(L) \leq \frac{(p+1)(q+1)(r+1)}{pqr}\, V(L) + 14.$$

The algorithm $\mathrm{C3B}_m$ combines two lists, say L_A and L_B. The list L_A consists of items in the set $\mathcal{C}\left[\frac{1}{m+1}, q\,;\ \frac{1}{m+1}, q\,;\ \frac{1}{m+1}, q\right]$. The list $L_B = L'_B \| L''_B \| L'''_B$ contains boxes $b \in L$ such that $\frac{1}{3m} < \min\{x(b), y(b), z(b)\} \leq p_m$, where $\frac{1}{m+2} < p_m < \frac{1}{m}$. The list L'_B contains the boxes $b \in L_B$ with $x(b) \leq p_m$. The list L''_B contains the boxes $b \in L_B \setminus L'_B$ with $y(b) \leq p_m$, and the list L'''_B is the list $L_B \setminus (L''_B \cup L'''_B)$. The algorithm $\mathrm{C3B}_m$ generates combined packings, each one with items of L_A and items of one of the sublists of L_B.

To pack items of L_A and L'_B, the algorithm subdivides each bin $B = (1, 1, 1)$ into two smaller bins: $B'_A = (m\,q, 1, 1)$ and $B'_B = (1 - m\,q, 1, 1)$. At each iteration, it packs m^3 boxes of L_A into a bin B'_A and uses the algorithm $\mathrm{A2B}_{m,m}$ to pack items of L'_B into a bin B'_B. In this case, it considers each item of L'_B and the bin B'_B as a two-dimensional item with y- and z-dimensions. This step is repeated until all items of L_A or all items of L'_B are totally packed. The algorithm performs analogous steps to combine the remaining items of L_A (if any) with items in L''_B and in L'''_B. The algorithm $\mathrm{C3B}_m$ halts when all items of L_A or of L_B are packed. It returns a pair $(\mathcal{P}_{AB}, L_{AB})$, where $\mathcal{P}_{AB}$ is the packing produced and the list L_{AB} is the set of items packed in $\mathcal{P}_{AB}$. The following result holds for this algorithm.

Lemma 5.2 *If $\mathcal{P}_{AB}$ is a packing of $L_{AB} \subseteq L_A \cup L_B$ generated by the algorithm $\mathrm{C3B}_m$ applied to the sublists L_A and L_B, then $\#(\mathcal{P}_{AB}) \leq V(L_{AB}) \big/ \left(\left(\frac{m}{m+1}\right)^3 + \frac{m}{3(m+1)^2}\right) + 18$.*

Proof. Consider the packing obtained by combining items of L_A and L'_B (into bins of type B'_A and B'_B). From Lemma 3.4, each bin of type B'_B has volume guarantee of at least $\left(\frac{m}{m+1}\right)^2$ (considering only the y and z-dimension), except perhaps 5 of these bins. Thus, each bin of the combined packing (of L_A and L'_B) has a volume guarantee of at least $\frac{1}{3m}\left(\frac{m}{m+1}\right)^2 = \frac{m}{3(m+1)^2}$, except perhaps 6 of them. The same analysis holds for the packing that combines items of L_A with items in L''_B and in L'''_B, and this gives us the desired inequality. ∎

ALGORITHM $\mathrm{A3B}_m$

Input: List of boxes $L \subset \mathcal{C}_{m,m,m}$

Output: Packing $\mathcal{P}$ of L into bins $B = (1, 1, 1)$.

1. Let $p = p(m) \leftarrow \frac{\sqrt{16m^6+76m^5+141m^4+142m^3+85m^2+28m+4}-2m^3-7m^2-7m-2}{2m^2(2+m^2+3m)}$;
$q = q(m) \leftarrow \frac{1-p}{m}$.

2. $L_A \leftarrow L \cap \mathcal{X}[\frac{1}{m+1}, q] \cap \mathcal{Y}[\frac{1}{m+1}, q] \cap \mathcal{Z}[\frac{1}{m+1}, q]$.

3. $L_B \leftarrow \{b \in L : \frac{1}{3m} < \min\{x(b), y(b), z(b)\} \leq p\}$.

4. $(\mathcal{P}_{AB}, L_{AB}) \leftarrow \mathrm{C3B}_m(L_A, L_B)$.

5. $L \leftarrow L \setminus L_{AB}$.

6. If $L_A \subseteq L_{AB}$ then

$$L_1 = L \cap \mathcal{C}\left[\tfrac{1}{m+1}, \tfrac{1}{m}\, ;\ \tfrac{1}{m+1}, \tfrac{1}{m}\, ;\ \tfrac{1}{m+1}, \tfrac{1}{m}\right] ; \quad L_2 = L \cap \mathcal{C}_{m+1,m,m} ;$$
$$L_3 = L \cap \mathcal{C}_{m,m+1,m} \setminus L_1 ; \quad L_4 = L \cap \mathcal{C}_{m,m,m+1} \setminus (L_1 \cup L_2) ;$$
$$\mathcal{P}_1 \leftarrow \mathrm{HNF}(L_1) ; \quad \mathcal{P}_2 \leftarrow \mathrm{H3B}_{m+1,m,m}(L_2) ;$$
$$\mathcal{P}_3 \leftarrow \mathrm{H3B}_{m,m+1,m}(L_3) ; \quad \mathcal{P}_4 \leftarrow \mathrm{H3B}_{m,m,m+1}(L_4) ,$$
$$\mathcal{P}_{opt} \leftarrow \mathcal{P}_1 \| \mathcal{P}_{AB} ; \quad \mathcal{P}_{aux} \leftarrow \mathcal{P}_2 \| \dots \| \mathcal{P}_4 .$$

7. else

$$L_1 = L \cap \mathcal{C}\left[\tfrac{1}{m+1}, \tfrac{1}{m}\, ;\ \tfrac{1}{m+1}, \tfrac{1}{m}\, ;\ \tfrac{1}{m+1}, \tfrac{1}{m}\right] ,$$
$$L_2 = L \cap \mathcal{X}[p, \tfrac{1}{m+1}] \cap \mathcal{Y}[\tfrac{1}{m+1}, \tfrac{1}{m}] \cap \mathcal{Z}[\tfrac{1}{m+1}, \tfrac{1}{m}] ;$$
$$L_3 = L \cap \mathcal{X}[\tfrac{1}{m+1}, \tfrac{1}{m}] \cap \mathcal{Y}[p, \tfrac{1}{m+1}] \cap \mathcal{Z}[\tfrac{1}{m+1}, \tfrac{1}{m}] ;$$
$$L_4 = L \cap \mathcal{X}[\tfrac{1}{m+1}, \tfrac{1}{m}] \cap \mathcal{Y}[\tfrac{1}{m+1}, \tfrac{1}{m}] \cap \mathcal{Z}[p, \tfrac{1}{m+1}] ;$$
$$L_5 = L \cap \mathcal{X}[\tfrac{1}{m+1}, \tfrac{1}{m}] \cap \mathcal{Y}[p, \tfrac{1}{m+1}] \cap \mathcal{Z}[p, \tfrac{1}{m+1}] ;$$
$$L_6 = L \cap \mathcal{X}[p, \tfrac{1}{m+1}] \cap \mathcal{Y}[\tfrac{1}{m+1}, \tfrac{1}{m}] \cap \mathcal{Z}[p, \tfrac{1}{m+1}] ;$$
$$L_7 = L \cap \mathcal{X}[p, \tfrac{1}{m+1}] \cap \mathcal{Y}[p, \tfrac{1}{m+1}] \cap \mathcal{Z}[\tfrac{1}{m+1}, \tfrac{1}{m}] ;$$
$$L_8 = L \cap \mathcal{X}[p, \tfrac{1}{m+1}] \cap \mathcal{Y}[p, \tfrac{1}{m+1}] \cap \mathcal{Z}[p, \tfrac{1}{m+1}] ;$$
$$L_9 = L \cap \mathcal{C}_{3m,m,m} ; \quad L_{10} = L \cap \mathcal{C}_{m,3m,m} \setminus L_9 ;$$
$$L_{11} = L \cap \mathcal{C}_{m,m,3m} \setminus (L_9 \cup L_{10}) ; \quad \mathcal{P}_i \leftarrow \mathrm{HNF}(L_i), \quad i = 1, \dots, 8;$$
$$\mathcal{P}_9 \leftarrow \mathrm{H3B}_{3m,m,m}(L_9) ; \quad \mathcal{P}_{10} \leftarrow \mathrm{H3B}_{m,3m,m}(L_{10}) ;$$
$$\mathcal{P}_{11} \leftarrow \mathrm{H3B}_{m,m,3m}(L_{11}) ; \quad \mathcal{P}_{opt} \leftarrow \mathcal{P}_1 \| \mathcal{P}_{AB} ;$$
$$\mathcal{P}_{aux} \leftarrow \mathcal{P}_2 \| \dots \| \mathcal{P}_{11} ;$$

8. Let $\mathcal{P} \leftarrow \mathcal{P}_{opt} \| \mathcal{P}_{aux}$.

9. Return $\mathcal{P}$.

The proof of the next result is analogous to the proof for the algorithm $\mathrm{A2B}_m$. Therefore, we omit the details and present the inequalities with which we can prove the desired result. We note that $\frac{1}{m+2} < p(m) < \frac{1}{m+1}$ and that the values of $p(m)$ and $q(m)$ in step 1 are chosen so as to obtain the same asymptotic performance bound for both the cases 1 and 2 analysed in the proof (corresponding to steps 6 and 7 of the algorithm).

Theorem 5.3 *For any list L of boxes, where $L \subset \mathcal{C}_{m,m,m}$, we have*

$$\mathrm{A3B}_m(L) \leq \beta_m\ \mathrm{OPT}(L) + 70,$$

where $\beta_m = \frac{2m^4+6m^3+9m^2+7m+2+\sqrt{16m^6+76m^5+141m^4+142m^3+85m^2+28m+4}}{2m^2(m+1)^2}$.

Proof. Let us consider the two possibilities corresponding to steps 6 and 7 of the algorithm.

Case 1: $L_A \subseteq L_{AB}$

In this case, the packings $\mathcal{P}_1$ and $\mathcal{P}_{AB}$ have volume guarantee of at least $\frac{m^3}{(m+1)^2}q$. Note that $\mathcal{P}_1$ has m^3 boxes in each bin (each box with a volume of at least $q\frac{1}{m+1}\frac{1}{m+1}$, except perhaps the last). From Lemma 5.2 we can conclude that the packing $\mathcal{P}_{AB}$ has a volume guarantee of $1 \Big/ \left(\left(\frac{m}{m+1}\right)^3 + \frac{m}{3(m+1)^2}\right)$. Thus,

$$\#(\mathcal{P}_{opt}) \leq \mathrm{OPT}(L_{opt}) + Z,$$

and

$$\#(\mathcal{P}_{opt}) \le \frac{(m+1)^2}{m^3 q} V(L_{opt}) + 19Z.$$

The other packings ($\mathcal{P}_2$, $\mathcal{P}_3$ and $\mathcal{P}_4$) have volume guarantee of at least $\frac{m^2(m+1)}{(m+1)^2(m+2)}$. Hence, the following inequality can be proved for the packing $\mathcal{P}_{aux}$:

$$\#(\mathcal{P}_{aux}) \le \frac{(m+1)^2(m+2)}{m^2(m+1)} V(L_{aux}) + 42Z.$$

Proceeding as in the proof of Theorem 3.6, we can show that

$$\#(\mathcal{P}) \le \beta'_m \ \mathrm{OPT}(L) + 61Z, \tag{18}$$

where $\beta'_m = \left(\mathcal{H}_1 + \mathcal{H}_2\right) / \max \left\{ \mathcal{H}_1, \left(\frac{m^3}{(m+1)^2} q\right)^2 \mathcal{H}_1 + \frac{m^2(m+1)}{(m+1)^2(m+2)} \mathcal{H}_2 \right\}$.

Case 2: $L_B \subseteq L_{AB}$.

In this case, we have a volume guarantee of at least $\frac{m^3}{(m+1)^3}$ for the packing $\mathcal{P}_1 \| \mathcal{P}_{AB}$. That is,

$$\#(\mathcal{P}_{opt}) \le \mathrm{OPT}(L_{opt}) + Z,$$

and

$$\#(\mathcal{P}_{opt}) \le \frac{(m+1)^3}{m^3} V(L_{opt}) + 19\,Z.$$

For the remaining packings, we obtain a volume guarantee of $\frac{m^2(m+1)p}{(m+1)^2}$. Therefore,

$$\#(\mathcal{P}_{aux}) \le \frac{(m+1)^2}{m^2(m+1)p} V(L_{aux}) + 49Z.$$

Analogously to the previous case, we can prove that

$$\#(\mathcal{P}) \le \beta''_m \ \mathrm{OPT}(L) + 61\,Z, \tag{19}$$

where $\beta''_m = \left(\mathcal{H}_1 + \mathcal{H}_2\right) / \max \left\{ \mathcal{H}_1, \left(\frac{m^3}{(m+1)^3}\right)^2 \mathcal{H}_1 + \frac{m^2(m+1)p}{(m+1)^2} \mathcal{H}_2 \right\}$.

Using Lemma 3.5, we can obtain bounds for β'_m and β''_m and conclude that both are at most β_m. Using this, the result follows from the inequalities (18) and (19). ∎

6 Final Remarks

Table 1 shows the asymptotic performance bounds (correspondly α_m or β_m) of each algorithm, for $m = 1, \ldots, 9$. The algorithms presented in this paper are marked with a $\star$. As a final remark we observe that all ideas applied for the problems presented here can be extended for packing problems of higher dimensions.

References

[1] 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.

Prob.	Algorithm	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$	$m=7$	$m=8$	$m=9$
$2BP_m$	α(HNF) [6]	3.383	2.847	1.954	1.646	1.489	1.394	1.329	1.283	1.248
	$\alpha(2D_m)$ [19]	3.250	2.112	1.730	1.540	1.428	1.354	1.302	1.263	1.233
	$\alpha(A2B_m)$ $\star$	3.050	2.028	1.684	1.512	1.409	1.340	1.291	1.254	1.226
$3SP_m$	$\alpha(TP_m)$ [19]	3.250	2.112	1.730	1.540	1.428	1.354	1.302	1.263	1.233
	$\alpha(A3S_m)$ $\star$	3.050	2.028	1.684	1.512	1.409	1.340	1.291	1.254	1.226
$3BP_m$	$\alpha(3D_m)$ [19]	6.25	3.112	2.285	1.915	1.708	1.576	1.486	1.419	1.369
	$\alpha(A3B_m)$ $\star$	6.023	3.016	2.233	1.883	1.686	1.560	1.473	1.409	1.361

Table 1: Asymptotic performance bounds of the algorithms for $m = 1, \ldots, 9$.

[2] 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.

[3] 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.

[4] 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.

[5] J. Csirik. The parametric behavior of the first-fit decreasing bin packing algorithm. *Journal of Algorithms*, 15:1–28, 1993.

[6] J. B. Frenk and G. Galambos. Hybrid next fit algorithm for the two-dimensional rectangle bin packing problem. *Computing*, 39:201–217, 1987.

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

[8] D. S. Johnson. Fast algorithms for bin packing. *Journal of Computer and System Sciences*, 8:272–314, 1974.

[9] D. S. Johnson, A. Demers, J. D. Ullman, M. R. Garey, and R. L. Graham. Worst-case performance bounds for simple one-dimensional packing algorithms. *SIAM Journal on Computing*, 3:299–325, 1974.

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

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

[12] Andrea Lodi, Silvano Martello, and Michele Monaci. Two-dimensional packing problems: a survey. *European J. Operational Research*, 141(2):241–252, 2002.

[13] Andrea Lodi, Silvano Martello, and Daniele Vigo. Recent advances on two-dimensional bin packing problems. *Discrete and Applied Mathematics*, 123:379–396, 2002.

[14] Silvano Martello, Michele Monaci, and Daniele Vigo. An exact approach to the strip-packing problem. *INFORMS Journal on Computing*, 15(3):310–319, 2003.

[15] Silvano Martello, David Pisinger, and Daniele Vigo. The three-dimensional bin packing problem. *Oper. Res.*, 48(2):256–267, 2000.

[16] Silvano Martello and Daniele Vigo. Exact solution of the finite two-dimensional bin packing problem. *Management Science*, 44(2):388–399, 1998.

[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. Approximation algorithms for the orthogonal z-oriented 3-D packing problem. *SIAM Journal on Computing*, 29(3):1008–1029, 2000.

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

[20] F. K. Miyazawa and Y. Wakabayashi. Two- and three-dimensional parametric packing problems. *Electronic Notes in Discrete Mathematics*, 19:313–319, 2005. Proceedings of GRACO: Brazilian Symposium on Graphs, Algorithms and Combinatorics.

[21] G. Wäscher, H. Haußner, and H. Schumann. An improved typology of cutting and packing problems. Technical Report Working Paper 24, Otto von Guericke University Magdeburg, 2005.