An algorithm for the three-dimensional packing problem with asymptotic performance analysis

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

Abstract

The three-dimensional packing problem can be stated as follows. Given a list of boxes, each with a given length, width and height, the problem is to pack these boxes into a rectangular box of fixed size bottom and unbounded height, so that the height of this packing is minimized. The boxes have to be packed orthogonally and oriented in all three dimensions. We present an approximation algorithm for this problem and show that its asymptotic performance bound is between 2.5 and 2.67. This result answers a question raised by Li and Cheng [5] about the existence of an algorithm for this problem with an asymptotic performance bound less than 2.89.

1 Introduction

In this paper we present an approximation algorithm for the **Three-dimensional Packing Problem**. This problem is defined as follows: Given a rectangular box B with a fixed size bottom and unbounded height and a list $L = (b_1, \ldots, b_n)$ of rectangular boxes, find an orthogonal oriented packing of the boxes $b_1, \ldots, b_n$ into B that minimizes the total height. The boxes are required to be packed into B orthogonally and oriented in all three dimensions.

We denote each box b_i as a triplet $b_i = (x_i, y_i, z_i)$, where x_i, y_i and z_i are its **length**, **width** and **height**, respectively. We assume here that the box B has dimensions $(1, 1, \infty)$, since if this were not the case and we had $B = (l, w, \infty)$, $l > 0, w > 0$, we could divide the length l_i and the width w_i of each box b_i by l and w, respectively. This problem will be denoted by **TPP**. Since one can reduce the uni-dimensional packing problem [2] to this problem, it follows that it is an $\mathcal{NP}$-hard problem.

If $\mathcal{A}$ is an algorithm for TPP and L is a list of boxes, then $\mathcal{A}(\mathbf{L})$ denotes the height of the packing generated by the algorithm $\mathcal{A}$ when applied to the list L; and $\mathbf{OPT(L)}$ denotes the height of an optimal packing of L. We say that α is *an* **asymptotic performance bound** of an algorithm $\mathcal{A}$ if there exists a constant β such that for all lists L, in which all boxes have height at most Z, the following holds: $\mathcal{A}(L) \leq \alpha \cdot \mathrm{OPT}(L) + \beta \cdot Z$. Furthermore, if for any small ϵ and any large M, both positive, there is an instance L such that $\mathcal{A}(L) > (\alpha - \epsilon)\mathrm{OPT}(L)$ and $\mathrm{OPT}(L) > M$, then we say that α is *the* **asymptotic performance bound** of the algorithm $\mathcal{A}$.

[1] Instituto de Matemática e Estatística — Universidade de São Paulo — Caixa Postal 66281 — 05389-970 — São Paulo, SP — Brazil (e-mail: keidi@ime.usp.br, yw@ime.usp.br).

In 1990, Li and Cheng [3] presented several algorithms for TPP: for the general case, an algorithm whose asymptotic performance bound is 3.25; and for the special case in which all boxes have square bottom, an algorithm whose asymptotic performance bound is 2.6875. In 1992, these authors [5] also presented an on-line algorithm with asymptotic performance bound that can be made as close to 2.89 as desired. The algorithm to be presented here has an asymptotic performance bound less than 2.67. This result answers a question raised by Li and Cheng [5] about the existence of an algorithm for TPP with an asymptotic performance bound less than 2.89.

We show that *the* asymptotic performance bound of our algorithm is between 2.5 and 2.67.

This paper is organized as follows. In Section 2 we establish the notation, mention some basic results and describe algorithms that are used as subroutines of the main algorithm. In Section 3 we first explain results on the ideas of the main algorithm, and then give a formal description of it. In the sequel, we prove results on the asymptotic performance bound of the algorithm and in Section 4 we discuss its time complexity.

2 Notation and Basic Results

Most of the concepts and notation used here can be found in [3]. Given a list of boxes $L = (b_1, \ldots, b_n)$, we assume that each box b_i is of the form $b_i = (x_i, y_i, z_i)$, with $x_i \leq 1$ and $y_i \leq 1$. Given a triplet $t = (a, b, c)$, we also refer to each of its elements a, b and c as $x(t)$, $y(t)$ and $z(t)$, respectively. We denote by $\mathbf{S(b)}$ and $\mathbf{V(b)}$ the **bottom area** (*i.e.* $S(b) := x(b)y(b)$) and the **volume** of the box b, respectively. Given a function $f : C \to I\!R$, and a subset $C' \subseteq C$, we denote by $f(C')$ the sum $\sum_{e \in C'} f(e)$.

Although a list is given as an ordered n-tuple of boxes, when the order of the boxes is irrelevant the corresponding list may be viewed as a set (*e.g.* if L is a list of boxes then we may refer to $S(L)$ and $V(L)$ as the sum $\sum_{b \in L} S(b)$ and $\sum_{b \in L} V(b)$, respectively).

Note that, by using a three-dimensional coordinate system, the box $B = (1, 1, \infty)$ can be seen as the region $[0, 1) \times [0, 1) \times [0, \infty)$, and we may define a **packing** $\mathcal{P}$ of a list of boxes $L = (b_1, \ldots, b_n)$ into B as a mapping $\mathcal{P} : L \to [0, 1) \times [0, 1) \times [0, \infty)$, such that

$$\mathcal{P}^x(b_i) + x_i \leq 1 \quad \text{and} \quad \mathcal{P}^y(b_i) + y_i \leq 1 \ ,$$

where $\mathcal{P}(b_i) = (\mathcal{P}^x(b_i), \mathcal{P}^y(b_i), \mathcal{P}^z(b_i))$, $i = 1, \ldots, n$.

Furthermore, if $\mathcal{R}(b_i)$ is defined as

$$\mathcal{R}(b_i) = [\mathcal{P}^x(b_i), \mathcal{P}^x(b_i) + x_i) \times [\mathcal{P}^y(b_i), \mathcal{P}^y(b_i) + y_i) \times [\mathcal{P}^z(b_i), \mathcal{P}^z(b_i) + z_i),$$

then

$$\mathcal{R}(b_i) \cap \mathcal{R}(b_j) = \emptyset \quad \forall i, j, \ 1 \leq i \neq j \leq n \ .$$

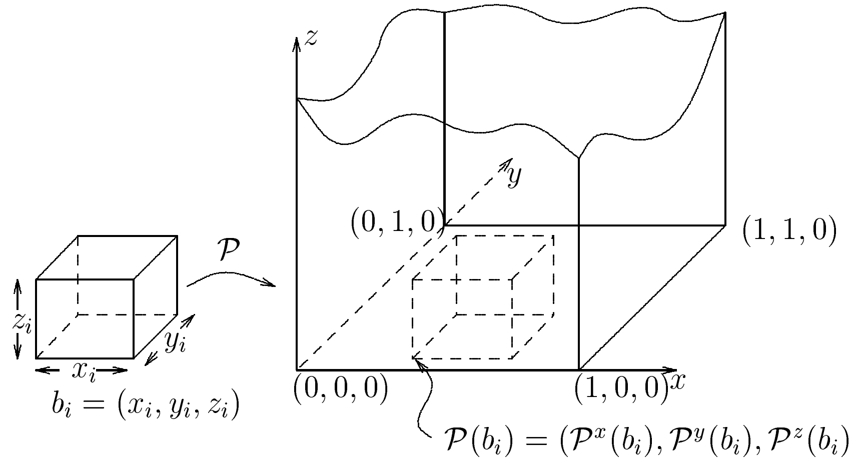

Figure 1: Packing of a box $b_i = (x_i, y_i, z_i)$ into the box $B = (1, 1, \infty)$.

The above conditions mean that each box in L must be entirely enclosed in the box B and must be packed orthogonally and oriented in all three dimensions. Furthermore, no two boxes can overlap in the packing $\mathcal{P}$ (see figure 1).

Given a packing $\mathcal{P}$ of L, we denote by $\mathbf{H}(\mathcal{P})$ the **height** of the packing $\mathcal{P}$, *i.e.*, $H(\mathcal{P}) := \max\{\mathcal{P}^z(b) + z(b) : \ b \in L\}$.

All packings will be denoted by the letter $\mathcal{P}$, with or without a subscript and/or superscript (for example, $\mathcal{P}'$, $\mathcal{P}_{OC}$, $\mathcal{P}'_{AB}$).

If $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_v$ are packings of disjoint lists $L_1, L_2, \ldots, L_v$, respectively, we define the **concatenation** of these packings as a packing $\mathcal{P} = \mathcal{P}_1 \| \mathcal{P}_2 \| \ldots \| \mathcal{P}_v$ of $L = L_1 \cup L_2 \cup \ldots \cup L_v$, where $\mathcal{P}(b) = (\mathcal{P}_i^x(b), \mathcal{P}_i^y(b), \sum_{j=1}^{i-1} H(\mathcal{P}_j) + \mathcal{P}_i^z(b))$, for all $b \in L_i$, $1 \le i \le v$.

The other notations to be used here are the following.

- $\mathcal{C}\,[\mathbf{p}'', \mathbf{p}' \ ; \ \mathbf{q}'', \mathbf{q}'] := \{b_i = (x_i, y_i, z_i) : \ p'' < x_i \le p' \ , \quad q'' < y_i \le q'\}$,
  for $0 \le p'' < p' \le 1$, $0 \le q'' < q' \le 1$.
- $\mathcal{C}_{\mathbf{m}} := \mathcal{C}\left[0, \frac{1}{m} \ ; \ 0, \frac{1}{m}\right]$, for $m > 0$.

A **level** N in a packing $\mathcal{P}$ is a region $[0,1) \times [0,1) \times [Z_1, Z_2)$ in which there is a set L' of boxes such that for all $b \in L'$, $\mathcal{P}^z(b) = Z_1$ and $Z_2 - Z_1 = \max\{z(b) : \ b \in L'\}$. We denote by $\mathbf{S}(\mathbf{N})$ the sum $\sum_{b \in L'} S(b)$. Sometimes we shall consider the level N as a packing of the list L'.

A **layer** (in the x-axis direction) in a level is a region $[0,1) \times [Y_1, Y_2) \times [Z_1, Z_2)$ in which there is a set L' of boxes such that for all $b \in L'$, $\mathcal{P}^y(b) = Y_1$ and $\mathcal{P}^z(b) = Z_1$ and $Y_2 - Y_1 = \max\{y(b) : \ b \in L'\}$ and $Z_2 - Z_1 = \max\{z(b) : \ b \in L'\}$.

Throughout this paper we consider **Z** as the height of the highest box in the list L (or in the list under consideration).

Some of the algorithms that will be used in the main algorithm generate packings consisting of levels satisfying certain properties. We prove in the sequel a result concerning these packings and derive special cases of it which will be used in the proof of the main theorem.

Proposition 2.1 *Let L be an instance of* TPP *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 $H(\mathcal{P}) \leq \frac{1}{s}V(L) + Z$.*

Proof. Let h_i be the height of level N_i, $i = 1, \ldots, v$.

$$\begin{aligned} V(L) &\geq S(N_1) \cdot h_2 + S(N_2) \cdot h_3 + \cdots S(N_{v-1}) \cdot h_v \\ &\geq s \cdot h_2 + s \cdot h_3 + \cdots s \cdot h_v \\ &= s \cdot (\sum_{i=1}^{v} h_i - h_1) \\ &= s \cdot (H(\mathcal{P}) - Z). \end{aligned}$$

□

The constant s mentioned in the above proposition will be called an **area guarantee** of the packing $\mathcal{P}$.

We describe in the sequel two algorithms for which Proposition 2.1 can be applied. First we describe an algorithm called **NFDH** (Next Fit Decreasing Height) that was presented by Li and Cheng in [3]. This algorithm has two variants: NFDH^x and NFDH^y. The notation NFDH is used to refer to any of these variants.

The Algorithm NFDH^x first sorts the boxes of L in non-increasing order of their height: $b_1, b_2, \ldots, b_n$. The first box b_1 is packed in the position $(0, 0, 0)$, the next one is packed in the position $(x(b_1), 0, 0)$ and so on, side by side, until a box is found that does not fit in this layer. At this moment the next box b_k is packed in the position $(0, y(b^*), 0)$, where $y(b^*) = \max\{y(b_i), i = 1, \ldots, k-1\}$. The process continues in this way until a box b_l is found that does not fit in the first level. Then, the algorithm packs this box in a new level at the height $z(b_1)$. 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 [3]).

The following result proved by Li and Cheng [3] can be derived as a corollary of Proposition 2.1. In fact, the proof of Proposition 2.1 is similar to the proof of this lemma.

Lemma 2.2 *If* $L \in \mathcal{C}\left[\frac{1}{m+1}, \frac{1}{m} \;;\; 0, \frac{1}{m}\right]$ *then* $\mathrm{NFDH}^y(L) \leq \left(\frac{m+1}{m-1}\right) V(L) + Z$. *The same result also holds for the Algorithm* NFDH^x *when applied to a list* $L \in \mathcal{C}\left[0, \frac{1}{m} \;;\; \frac{1}{m+1}, \frac{1}{m}\right]$.

Another algorithm that generates a packing as mentioned in Proposition 2.1 is the algorithm called **LL**, presented by Li and Cheng in [4]. This algorithm is used to pack a list of boxes $L \in \mathcal{C}\left[0, \frac{1}{m} \;;\; 0, \frac{1}{m}\right]$, $m \geq 3$. We indicate by (L, m) the parameters that must be specified to call this algorithm.

Let us give an idea of the Algorithm $\mathrm{LL}(L, m)$, as we shall refer to it in the sequel. Initially it sorts the boxes in L in non-increasing order of their height. Then it divides L into sublists $L_1, \ldots, L_v$, such that $L = L_1 \| L_2 \| \ldots \| L_v$, each sublist preserving the (non-increasing) order of the boxes, and

$$\begin{array}{ll} S(L_i) \leq \left(\frac{m-2}{m}\right) + \left(\frac{1}{m}\right)^2 & \text{for } \; i = 1, \ldots, v \;, \\ S(L_i) + S(first(L_{i+1})) > \left(\frac{m-2}{m}\right) + \left(\frac{1}{m}\right)^2 & \text{for } \; i = 1, \ldots, v-1 \;; \end{array}$$

where $first(L')$ is the first box in L'. Then, the Algorithm LL uses a two-dimensional packing algorithm to pack each list L_i in only one level, say N_i. The final packing is the concatenation of each of these levels. As each level N_i (except perhaps the last) is such that $S(N_i) \geq \frac{m-2}{m}$, the following result (given in [4]) can be obtained by applying Proposition 2.1.

Lemma 2.3 *If* $\mathcal{P}$ *is the packing generated by the Algorithm* LL *for an instance* $L \in \mathcal{C}\left[0, \frac{1}{m} \;;\; 0, \frac{1}{m}\right]$, *then* $H(\mathcal{P}) \leq \left(\frac{m}{m-2}\right) V(L) + Z$.

Another algorithm that will play an important role in the main algorithm is the Algorithm **COLUMN**. This algorithm generates a partial packing of two lists, say L_1 and L_2. The packing consists of several stacks of boxes, referred to as **columns**. Each column is built by putting the boxes one on top of the other, and each column consists only of boxes in either L_1 or L_2.

The Algorithm COLUMN is called with the parameters $(L_1, [p^1], L_2, [p^2])$, where $p^1 = p^1_1, p^1_2, \ldots, p^1_{n_1}$ consists of the positions in the bottom of box B where the columns of boxes in L_1 should start and $p^2 = p^2_1, p^2_2, \ldots, p^2_{n_2}$ consists of the positions in the bottom of box B where the columns of boxes in L_2 should start. Each point $p^i_j = (x^i_j, y^i_j)$ represents the x-axis and the y-axis coordinates where the first box (if any) of each column of the respective list 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 lists L_1, L_2 are chosen in such a way that they do not lead to an infeasible packing.

We call **height of a column** the sum of the height of all boxes in that column.

The positions p_j^i for $j = 1, \ldots, n_i$ and $i = 1, 2$ must be given. 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 and packs the next box from the respective list on the top of that column. The process terminates when all the boxes in L_1 or L_2 are packed. At this point, the algorithm returns the pair $(\mathcal{P}, L')$ where L' consists of the boxes in $L_1 \cup L_2$ that were packed, and $\mathcal{P}$ is the packing of L' generated by the algorithm. We also say that $\mathcal{P}$ **combines** the lists L_1 and L_2.

If each box of L_i has bottom area at least s_i, $i = 1, 2$, the sum $n_1 s_1 + n_2 s_2$ is called the **combined area** of the packing generated by the Algorithm COLUMN.

The following lemma about this algorithm holds.

Lemma 2.4 *Let $\mathcal{P}$ be the packing of $L' \subseteq L_1 \cup L_2$ generated by the Algorithm* COLUMN *when applied to lists L_1 and L_2 and list of positions $p_1^i, p_2^i, \ldots, p_{n_i}^i$, $i = 1, 2$. If $S(b) \geq s_i$, for all boxes b in L_i, $i = 1, 2$, then $H(\mathcal{P}) \leq \frac{1}{s_1 n_1 + s_2 n_2} V(L') + Z$.*

Proof. Note that the difference between the height of any two columns is not greater than Z. Thus, $V(L') \geq (H(\mathcal{P}) - Z)(s_1 n_1 + s_2 n_2)$.

□

Another simple algorithm that we shall use is the Algorithm **OC** (One Column). Given a list of boxes, say $L = (b_1, \ldots, b_n)$, this algorithm packs each box b_{i+1} on top of box b_i , for $i = 1, \ldots, n-1$. Thus, the first box is packed in the position $(0, 0, 0)$, the second box is packed in the position $(0, 0, z(b_1))$, and so on. It is easy to verify the following results.

Lemma 2.5 *If $\mathcal{P}$ is the packing generated by the Algorithm* OC *when applied to a list L and s is a constant such that $S(b) \geq s$ for all boxes b in L, then $H(\mathcal{P}) \leq \frac{V(L)}{s}$.*

Lemma 2.6 *If $\mathcal{P}$ is the packing generated by the Algorithm* OC *when applied to a list L such that $x(b) > \frac{1}{2}$ and $y(b) > \frac{1}{2}$ for all boxes b in L, then $H(\mathcal{P}) = \mathrm{OPT}(L)$.*

Two other algorithms that we shall need in the main algorithm are based on the algorithm **UD**, developed by Baker, Brown and Kattseff [1] for the strip packing problem. This problem consists in packing a list of rectangles $R = (r_1, r_2, \ldots, r_n)$ in a rectangle of unit length and infinite height, and the objective is to minimize the height of the packing. The following result concerning the algorithm UD is presented in [1].

Lemma 2.7 *Let $R = (r_1, \ldots, r_n)$ be an instance for the strip packing problem, in which no rectangle has height greater than Z'. Then the height of the packing $\mathcal{B}$ generated by the algorithm* UD *when applied to the list R is such that $H(\mathcal{B}) \leq \frac{5}{4}\mathrm{OPT}(L) + \frac{53}{8} Z'$.*

Based on Algorithm UD, we define the algorithms UD^x and UD^y for TPP as follows. Given a list $L = (b_1, b_2, \ldots, b_n)$ of rectangular boxes $b_i = (x_i, y_i, z_i)$, for $i = 1, \ldots, n$, the Algorithm UD^x first uses the Algorithm UD to generate a packing $\mathcal{B}$, applying it to a list of rectangles $R = (r_1, r_2, \ldots, r_n)$, where $r_i = (x_i, z_i)$, $i = 1, \ldots, n$. Then it generates a packing of L by packing the corresponding boxes in the position 0 in the y-axis and using the same coordinates of the two-dimensional packing $\mathcal{B}$ for the x- and z-axis. The algorithm UD^y is symmetric to the Algorithm UD^x. Using Lemma 2.7 it is immediate that the following holds:

Lemma 2.8 *Let L be an instance for* TPP *such that $y(b) > \frac{1}{2}$ (resp. $x(b) > \frac{1}{2}$) for all boxes b in L. Then the packing $\mathcal{P}$ generated by the algorithm* UD^x *(resp.* UD^y*) is such that*

$$H(\mathcal{P}) \leq \frac{5}{4}\mathrm{OPT}(L) + \frac{53}{8}Z \ .$$

3 The main algorithm

The algorithm to be described here will be called Algorithm $\mathcal{A}_{\mathbf{k}}$. It depends on a parameter k, an integer greater than 5. Before giving its formal description, let us first explain the idea behind it.

This algorithm divides the given instance L into sublists and applies an appropriate algorithm to each (or a combination) of these sublists. The final packing is obtained as a concatenation of these packings.

Initially, the boxes of L are divided into four parts, P_1, P_2, P_3 and P_4, as follows.

$$\begin{aligned} P_1 &= \{b \in L : x(b) \leq \tfrac{1}{2},\ y(b) \leq \tfrac{1}{2}\}, & P_2 &= \{b \in L : x(b) \leq \tfrac{1}{2},\ y(b) > \tfrac{1}{2}\}, \\ P_3 &= \{b \in L : x(b) > \tfrac{1}{2},\ y(b) \leq \tfrac{1}{2}\}, & P_4 &= \{b \in L : x(b) > \tfrac{1}{2},\ y(b) > \tfrac{1}{2}\}. \end{aligned}$$

Suppose for each of these parts we generate packings consisting of levels. Li and Cheng [3] have shown that one can get a packing of part P_1 with area guarantee $\frac{1}{3}$, and the same for P_2 and P_3. Note that for part P_4 the best one can guarantee is $\frac{1}{4}$. They proved the statements for P_1, P_2 and P_3 by considering the subdivision indicated in figure 2.

As we shall see later, considering another subdivision of P_1, one can have a packing of P_1 with area guarantee $\frac{4}{9}$. Thus, with respect to area guarantee, we can classify the packings of part P_1 as being *good*, P_2 and P_3 as *regular* and P_4 as *bad*. We call a packing of a sublist as being *good* if it has an area guarantee close to that of part P_1. The idea of our algorithm is to refine the subdivision of these parts in such a way that the obtained sublists allow a better combined area or a better area guarantee. For that, we have to detect the boxes that do not yield packings with good area guarantee. These boxes will be called *critical*.

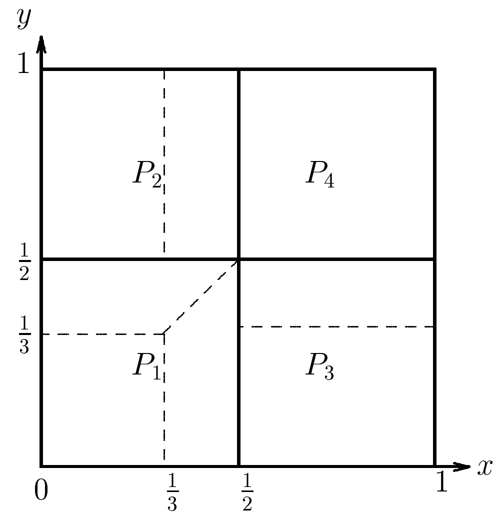

Figure 2: Subdivision of P_1, P_2 and P_3.

The Algorithm $\mathcal{A}_k$ uses the Algorithm COLUMN to combine the critical boxes in P_2 and P_3 (these are the boxes in the sets $L_A = (A_1 \cup \ldots \cup A_{k+14})$ and $L_B = (B_1 \cup \ldots \cup B_{k+14})$, illustrated in figure 3) in such a manner that the resulting partial packing is a good one and the critical boxes of P_2 and P_3 that could not be packed remain in only one of these parts. Furthermore, the other part —now without the critical boxes— allows a good packing.

Suppose that the critical boxes in P_3 (boxes in the set L_B) could all be packed (see figure 4). The way the set L_B is defined guarantees that the remaining part of P_3 (sublists L_1 to L_{17}, see figure 5) has a good area guarantee. Now we apply the same process for parts $P_1 \cup P_3$ and P_4. That is, we combine critical boxes of $P_1 \cup P_3$ with the critical boxes of P_4 (these are the boxes in $L'_D \cup L''_D$ and L_C, see figure 4). Note that the choices of the sublists to be combined have to be carefully done so that they allow good combinations, and once one of the sublists is packed the remaining boxes in the corresponding part also allow a good packing.

Suppose L_C is totally packed (see figure 5). Now we define new critical boxes in P_2 and P_4 (these are the boxes in $L'_F \cup L''_F$ and L_E, see figure 6) and apply the algorithm COLUMN to the corresponding sublists. The resulting packing $\mathcal{P}_{EF}$ has an area guarantee better than that when only boxes of P_4 are considered.

In both cases, considering the boxes not packed yet we can obtain packings which can be compared with an optimum packing of the corresponding sublist. The details of this process will be clear in the description of the Algorithm $\mathcal{A}_k$.

The sublists A_i and B_i we have mentioned above are constructed using values r_i and s_i, $i = 1, \ldots, k + 14$, defined in the sequel. These sublists are illustrated in figure 3, and are formally defined in step 2 of the algorithm.

Definition 3.1 *Let $r_1^{(k)}, r_2^{(k)}, \ldots, r_{k+15}^{(k)}$ and $s_1^{(k)}, s_2^{(k)}, \ldots, s_{k+14}^{(k)}$ be real numbers 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 existence of the numbers $r_1^{(k)}, r_2^{(k)}, \ldots, r_k^{(k)}$ can be shown using a continuity argument. Furthermore, one can show 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 shall omit the superscripts $^{(k)}$ of the notation $r_i^{(k)}, s_i^{(k)}$ when k is clear from the context.

As we are going to apply Algorithm COLUMN combining sublists A_i and B_j, we have to specify the coordinates where the columns of A_i and B_j are to be built. To this end we define lists of positions, $p_{i,j}, q_{i,j}, p'_j, q'_j, p''_j$ and q''_j.

REMARK: The positions $p_{i,j}, q_{i,j}, p'_j, q'_j, p''_j$ and q''_j are defined in such a manner that the combined area of the packings generated by the Algorithm COLUMN (in step 5 of the Algorithm $\mathcal{A}_k$) is at least $\frac{27}{56}$.

Positions to combine sublists $\mathbf{A_i}$ and $\mathbf{B_j}$.

We define these positions only for $i \leq j$. The case in which $i > j$ is symmetric (see figure 3 to visualize these positions).

- To combine the lists A_i, $1 \le i \le k$, and B_j, $i \le j \le k$, take

  $$p_{i,j} \;= \Big[(0,0), \big(\tfrac{1}{2}, 0\big)\Big] \quad \text{and} \quad q_{i,j} \;= [(0, s_i)] \ .$$

  Note that in this case we have an area guarantee of at least $\frac{1}{2}$.

- To combine the list $A_{[1-k]} = (A_1 \cup \ldots \cup A_k)$ with B_j, $k+1 \le j \le k+14$, we consider two phases. We divide $A_{[1-k]}$ into A' and A'' taking $A' = \{b \in A_{[1-k]} : \ x(b) \le 1 - s_j\}$ and $A'' = A_{[1-k]} \setminus A'$.

  ⋆ To combine A' with B_j take

  $$p'_j \;= [(s_j, 0)] \quad \text{and}$$

  $$q'_j \;= \Big[(0,0)\,, \big(0, \tfrac{1}{j-k+2}\big)\,, \big(0, \tfrac{2}{j-k+2}\big)\,, \ldots, \big(0, \tfrac{j-k+1}{j-k+2}\big)\Big] \ .$$

  In this case we have an area guarantee of at least $\frac{13}{24}$. This minimum is attained when $j = k+1$.

  ⋆ To combine A'' with B_j take

  $$p''_j \;= \Big[(0,0), (\tfrac{1}{2}, 0)\Big] \quad \text{and}$$

  $$q''_j \;= \Big[\big(0, \tfrac{2}{3}\big)\,, \big(0, \tfrac{2}{3} + \tfrac{1}{j-k+2}\big)\,, \big(0, \tfrac{2}{3} + \tfrac{2}{j-k+2}\big)\,, \ldots, \big(0, \tfrac{2}{3} + \big(\lfloor \tfrac{j-k+2}{3} \rfloor - 1\big)\,\tfrac{1}{j-k+2}\big)\Big] \ .$$

  Here we obtain an area guarantee of at least $\frac{27}{56}$. In fact, the values of s_j (which determine A' and A''), $k+1 \le j \le k+14$, were chosen in such a way that for the boxes in P_3 not in L_B we also have a good area guarantee. The value $\frac{27}{56}$ is attained when $j = k+1$ (one box from B_{k+1} with bottom area $\frac{1}{8}$ and two boxes from A'', each with bottom area $\frac{5}{28}$).

- To combine the lists A_i, $k+1 \le i \le k+14$, and B_j, $i \le j \le k+14$, take

  $$\begin{aligned} p_{i,j} = \; & \Big[(s_j, 0)\,, \big(s_j + \tfrac{1}{i-k+2}, 0\big)\,, \big(s_j + \tfrac{2}{i-k+2}, 0\big)\,, \ldots, \\ & \big(s_j + (\lfloor (1-s_j)\cdot(i-k+2)\rfloor - 1)\,\tfrac{1}{i-k+2}, 0\big)\Big] \ \text{and} \end{aligned}$$

  $$q_{i,j} \;= \Big[(0,0)\,, \big(0, \tfrac{1}{j-k+2}\big)\,, \big(0, \tfrac{2}{j-k+2}\big)\,, \ldots, \big(0, \tfrac{j-k+1}{j-k+2}\big)\Big] \ .$$

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

We are now ready to describe the Algorithm $\mathcal{A}_k$.

Algorithm $\mathcal{A}_k$

Input: List of boxes L.

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

1 Let $P_1 = \{b \in L : x(b) \le \frac{1}{2},\ y(b) \le \frac{1}{2}\}$, $\quad P_2 = \{b \in L : x(b) \le \frac{1}{2},\ y(b) > \frac{1}{2}\}$,
$P_3 = \{b \in L : x(b) > \frac{1}{2},\ y(b) \le \frac{1}{2}\}$, $\quad P_4 = \{b \in L : x(b) > \frac{1}{2},\ y(b) > \frac{1}{2}\}$.

2 Let $r_1, r_2, \ldots, r_{k+16}$ and $s_1, s_2, \ldots, s_{k+14}$ be given as in Definition 3.1. Define the sets A_i and B_i, for $i = 1, \ldots, k+14$, in the following way (see figure 3).

$A_i = \left\{b \in L : x(b) \in (r_{i+1}, r_i]\,, y(b) \in \left(\frac{1}{2}, s_i\right]\right\}$,

$B_i = \left\{b \in L : x(b) \in \left(\frac{1}{2}, s_i\right],\ y(b) \in (r_{i+1}, r_i]\right\}$,

$L_A \leftarrow A_1 \cup \ldots \cup A_{k+14}, \quad L_B \leftarrow B_1 \cup \ldots \cup B_{k+14}$.

3 $i \leftarrow 1$; $j \leftarrow 1$; $\mathcal{P}_{AB} \leftarrow \emptyset$;

4 Let $p_{i,j}, q_{i,j}$, $1 \le i, j \le k+14$, and p'_j, p''_j, q'_j, q''_j, $k+1 \le j \le k+14$, be as defined previously.

5 Combine sets L_A and L_B as follows.

5.1 While ($i \le k$ and $j \le k$) do

$(\mathcal{P}_{i,j}, L_{i,j}) \leftarrow \mathrm{COLUMN}(A_i, p_{i,j}, B_j, q_{i,j})$;

$A_i \leftarrow A_i \setminus L_{i,j}$; $B_i \leftarrow B_i \setminus L_{i,j}$; $\mathcal{P}_{AB} \leftarrow \mathcal{P}_{AB} \| \mathcal{P}_{i,j}$;

If $A_i = \emptyset$ then $i \leftarrow i+1$ else $j \leftarrow j+1$;

5.2 If j=k+1

then

5.2.1 /* all boxes $B_1, \ldots, B_k$ have been packed */

$A_{[1-k]} \leftarrow A_1 \cup \ldots \cup A_k$;

While ($j \le k+14$ and $A_{[1-k]} \ne \emptyset$) do

$t \leftarrow 1 - s_{k+j}$;

$A' \leftarrow \{b \in A_{[1-k]} : x(b) \le t\}$; $\quad A'' \leftarrow A_{[1-k]} \setminus A'$;

$(\tilde{\mathcal{P}}'_j, \tilde{L}'_j) \leftarrow \mathrm{COLUMN}(A', p'_j, B_j, q'_j)$;

$(\tilde{\mathcal{P}}''_j, \tilde{L}''_j) \leftarrow \mathrm{COLUMN}(A'', p''_j, B_j \setminus \tilde{L}'_j, q''_j)$;

$\mathcal{P}_{AB} \leftarrow \mathcal{P}_{AB} \| \tilde{\mathcal{P}}'_j \| \tilde{\mathcal{P}}''_j$;

$B_j \leftarrow B_j \setminus \tilde{L}'_j \cup \tilde{L}''_j$; $\quad A_{[1-k]} \leftarrow A_{[1-k]} \setminus \tilde{L}'_j \cup \tilde{L}''_j$;

if $B_j = \emptyset$ then $j \leftarrow j+1$;

$i \leftarrow k+1$

else

5.2.2 /* All boxes $A_1, \ldots, A_k$ have been packed */

Perform steps symmetric to the ones given in the case 5.2.1;

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

$(\mathcal{P}_{i,j}, L_{i,j}) \leftarrow \mathrm{COLUMN}(A_i, p_{i,j}, B_j, q_{i,j})$;
$A_i \leftarrow A_i \setminus L_{i,j}$; $B_i \leftarrow B_i \setminus L_{i,j}$; $\mathcal{P}_{AB} \leftarrow \mathcal{P}_{AB} \| \mathcal{P}_{i,j}$;
If $A_i = \emptyset$ then $i \leftarrow i+1$ else $j \leftarrow j+1$;

6 If $j > k+14$ then /* all boxes in L_B have been packed */

6.1 Let L_{AB} be the set of boxes packed in $\mathcal{P}_{AB}$; $L \leftarrow L \setminus L_{AB}$;

6.2 Subdivide the list L in $L_1, \ldots, L_{25}$ as follows (see figure 4).

$$
\begin{array}{ll}
L_i = L \bigcap \mathcal{C}\left[\frac{1}{2}, 1 \ ; \ \frac{1}{i+2}, \frac{1}{i+1}\right], \text{ for } i = 1, \ldots, 16 & L_{17} = L \bigcap \mathcal{C}\left[\frac{1}{2}, 1 \ ; \ 0, \frac{1}{18}\right], \\
L_{18} = L \bigcap \mathcal{C}\left[\frac{1}{3}, \frac{1}{2} \ ; \ \frac{1}{3}, \frac{1}{2}\right], & L_{19} = L \bigcap \mathcal{C}\left[\frac{1}{3}, \frac{1}{2} \ ; \ \frac{1}{4}, \frac{1}{3}\right], \\
L_{20} = L \bigcap \mathcal{C}\left[\frac{1}{3}, \frac{1}{2} \ ; \ 0, \frac{1}{4}\right], & L_{21} = L \bigcap \mathcal{C}\left[\frac{1}{4}, \frac{1}{3} \ ; \ \frac{1}{3}, \frac{1}{2}\right], \\
L_{22} = L \bigcap \mathcal{C}\left[\frac{1}{4}, \frac{1}{3} \ ; \ 0, \frac{1}{3}\right], & L_{23} = L \bigcap \mathcal{C}\left[0, \frac{1}{4} \ ; \ \frac{1}{3}, \frac{1}{2}\right], \\
L_{24} = L \bigcap \mathcal{C}\left[0, \frac{1}{4} \ ; \ \frac{1}{4}, \frac{1}{3}\right], & L_{25} = L \bigcap \mathcal{C}_4, \\
L_C = L \bigcap \mathcal{C}\left[\frac{1}{2}, 1 \ ; \ \frac{1}{2}, \frac{19}{36}\right] & L'_D = \left\{b \in L_1 : \ y(b) \leq \frac{17}{36}\right\}, \\
L''_D = \left\{b \in L_{18} : \ y(b) \leq \frac{17}{36}\right\}. & L_D = L'_D \bigcup L''_D.
\end{array}
$$

6.3 Generate packings $\mathcal{P}_1, \ldots, \mathcal{P}_{25}$ as follows.

$(\mathcal{P}_{CD'}, L_{CD'}) \leftarrow \mathrm{COLUMN}(L_C, [(0,0)], L'_D, [(0, \frac{19}{36})])$;
$(\mathcal{P}_{CD''}, L_{CD''}) \leftarrow \mathrm{COLUMN}(L_C \setminus L_{CD'}, [(0,0)], L''_D, [(0, \frac{19}{36}), (\frac{1}{2}, \frac{19}{36})])$;
$\mathcal{P}_{CD} \leftarrow \mathcal{P}_{CD'} \| \mathcal{P}_{CD''}$;
$L_{CD} \leftarrow L_{CD'} \bigcup L_{CD''}$;
$L_1 \leftarrow L_1 \setminus L_{CD}$;
$L_{18} \leftarrow L_{18} \setminus L_{CD}$;

6.4 $\mathcal{P}_i \leftarrow \mathrm{NFDH}^y(L_i)$ for $i = 1, \ldots, 22$;
$\mathcal{P}_i \leftarrow \mathrm{NFDH}^x(L_i)$ for $i = 23, 24$;
$\mathcal{P}_{25} \leftarrow \mathrm{LL}(L_{25}, 4)$;

6.5 $P'_1 \leftarrow P_1 \setminus L_{CD}$;
$P'_2 \leftarrow P_2 \setminus L_{AB}$;
$P'_3 \leftarrow P_3 \setminus (L_{AB} \cup L_{CD})$;
$P'_4 \leftarrow P_4 \setminus L_{CD}$;

6.6 If $L_C \subseteq L_{CD}$
then (Case 1) $p \leftarrow \frac{\sqrt{199145}-195}{570} = 0.440\ldots$ /$*L_C$ is packed $*$/ (see figure 5)
else (Case 2) $p \leftarrow \frac{\sqrt{23401}-71}{180} = 0.455\ldots$; /$*L_D$ is packed $*$/ (see figure 6)

6.7 $L_E \leftarrow \{b \in P'_4 : \; x(b) \le 1-p\}$; $\quad L'_F \leftarrow \left\{b \in P'_2 : \; \frac{1}{9} < x(b) \le p\right\}$;
$L''_F \leftarrow \left\{b \in P'_2 : \; \frac{1}{18} < x(b) \le \frac{1}{9}\right\}$; $\;L_F \leftarrow L'_F \cup L''_F$;

6.8 $(\mathcal{P}_{EF'}, L_{EF'}) \leftarrow \mathrm{COLUMN}(L_E, [(0,0)], L'_F, [(1-p,0)])$;
$(\mathcal{P}_{EF''}, L_{EF''}) \leftarrow \mathrm{COLUMN}(L_E \setminus L_{EF'}, [(0,0)], L''_F, [(1-p,0), (1-p+\frac{1}{9}, 0), \ldots, (1-p+(\lfloor 9p \rfloor - 1)\frac{1}{9}, 0)])$;
$\mathcal{P}_{EF} \leftarrow \mathcal{P}_{EF'} \| \mathcal{P}_{EF''}$;
$L_{EF} \leftarrow L_{EF'} \cup L_{EF''}$;
$P''_2 \leftarrow P'_2 \setminus L_{EF}$;
$P''_4 \leftarrow P'_4 \setminus L_{EF}$;

6.9 (Subcase 1) If $L_E \subseteq L_{EF}$ then /* L_E is totally packed */
$\mathcal{P}_{UD} \leftarrow \mathrm{UD}^x(P'_2 \cup P'_4)$;
$\mathcal{P}_{OC} \leftarrow \mathrm{OC}(P''_4)$;
$P''_{2e} \leftarrow \left\{b \in P''_2 : \; x(b) \le \frac{1}{3}\right\}$;
$P''_{2d} \leftarrow \left\{b \in P''_2 : \; x(b) > \frac{1}{3}\right\}$;
$\mathcal{P}_{2e} \leftarrow \mathrm{NFDH}^x(P''_{2e})$;
$\mathcal{P}_{2d} \leftarrow \mathrm{NFDH}^x(P''_{2d})$;
$\mathcal{P}' \leftarrow \mathcal{P}_{OC} \| \mathcal{P}_{2e} \| \mathcal{P}_{2d} \| \mathcal{P}_{EF}$;
$\mathcal{P}'' \leftarrow \{\mathcal{P} \in \{\mathcal{P}_{UD}, \mathcal{P}'\} : \; H(\mathcal{P})$ is minimum $\}$;
$\mathcal{P}_{aux} \leftarrow \mathcal{P}_{AB} \| \mathcal{P}_{CD} \| \mathcal{P}_1 \| \ldots \| \mathcal{P}_{25}$;
$\mathcal{P} \leftarrow \mathcal{P}_{aux} \| \mathcal{P}''$;

6.9 (Subcase 2) If $L_F \subseteq L_{EF}$ then /* L_F is totally packed */
$\mathcal{P}_{OC} \leftarrow \mathrm{OC}(P''_4)$;
$P''_{2e} \leftarrow \left\{b \in P''_2 : \; x(b) \le \frac{1}{18}\right\}$;
$P''_{2d} \leftarrow \{b \in P''_2 : \; x(b) > p\}$;
$\mathcal{P}_{2e} \leftarrow \mathrm{NFDH}^x(P''_{2e})$;
$\mathcal{P}_{2d} \leftarrow \mathrm{NFDH}^x(P''_{2d})$;
$\mathcal{P}' \leftarrow \mathcal{P}_{OC} \| \mathcal{P}_{EF}$;
$\mathcal{P}_{aux} \leftarrow \mathcal{P}_{AB} \| \mathcal{P}_{CD} \| \mathcal{P}_{2e} \| \mathcal{P}_{2d} \| \mathcal{P}_1 \| \ldots \| \mathcal{P}_{25}$;
$\mathcal{P} \leftarrow \mathcal{P}_{aux} \| \mathcal{P}'$;

6.10 Return $\mathcal{P}$;

7 If $i > k + 14$ then generate a packing $\mathcal{P}$ of L as in step 6 (in a symmetric way);

8 Return $\mathcal{P}$;

end algorithm.

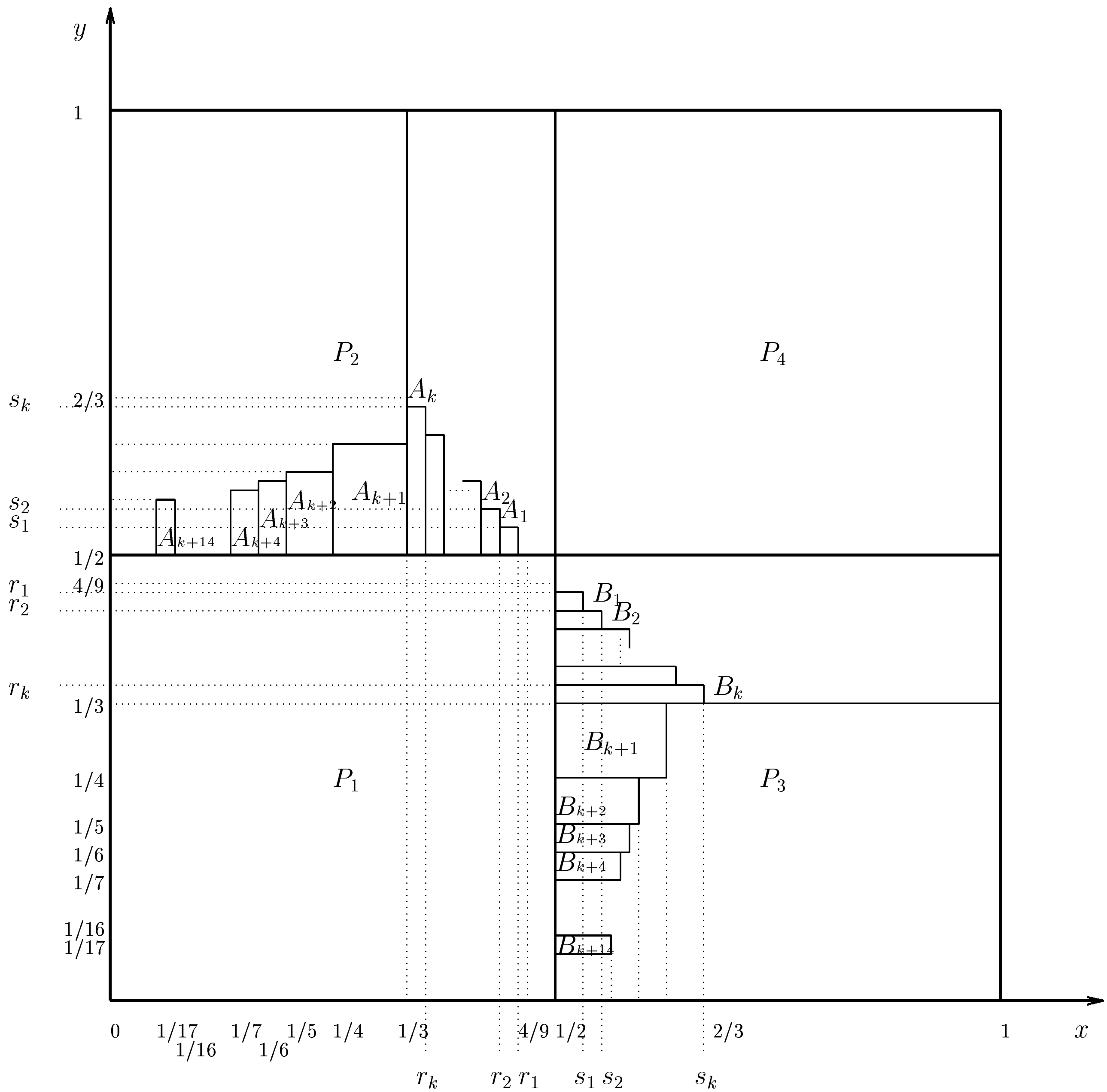

Figure 3: Sublists A_i and B_j.

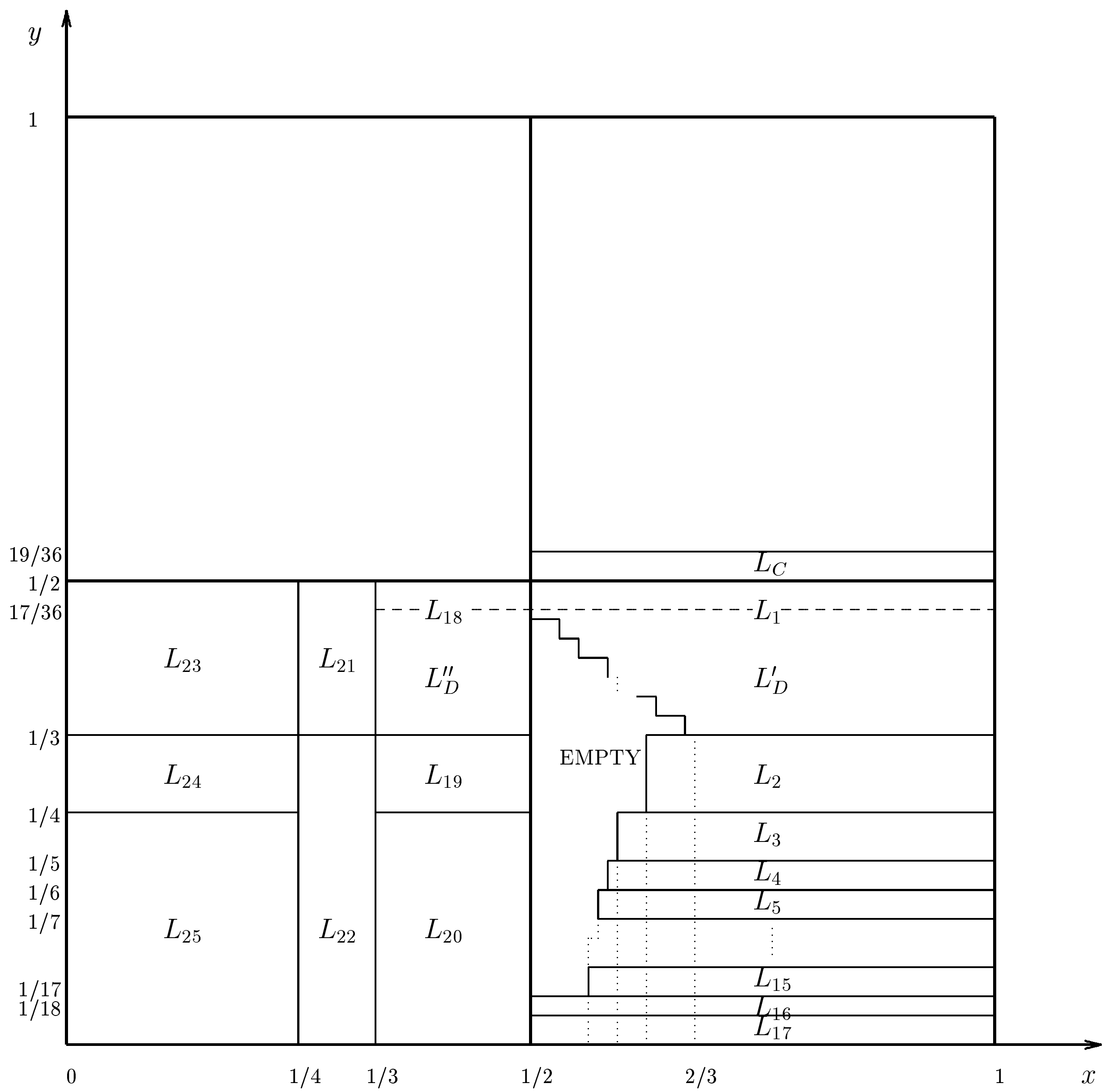

Figure 4: Sublists after the list $L_B = (B_1 \cup \ldots \cup B_{k+14})$ is totally packed.

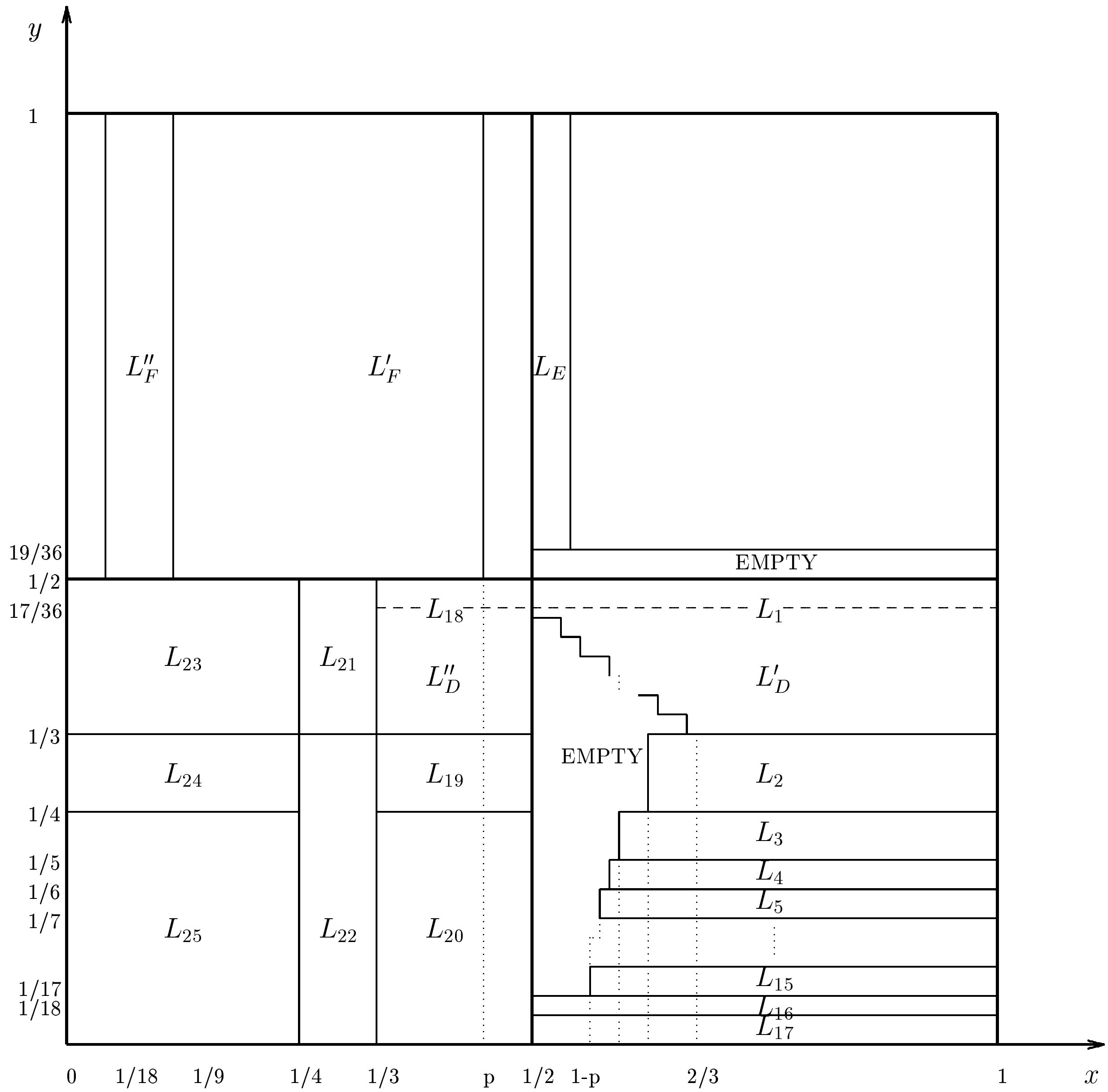

Figure 5: Combination of L_C and $L'_D \cup L''_D$: L_C is totally packed.

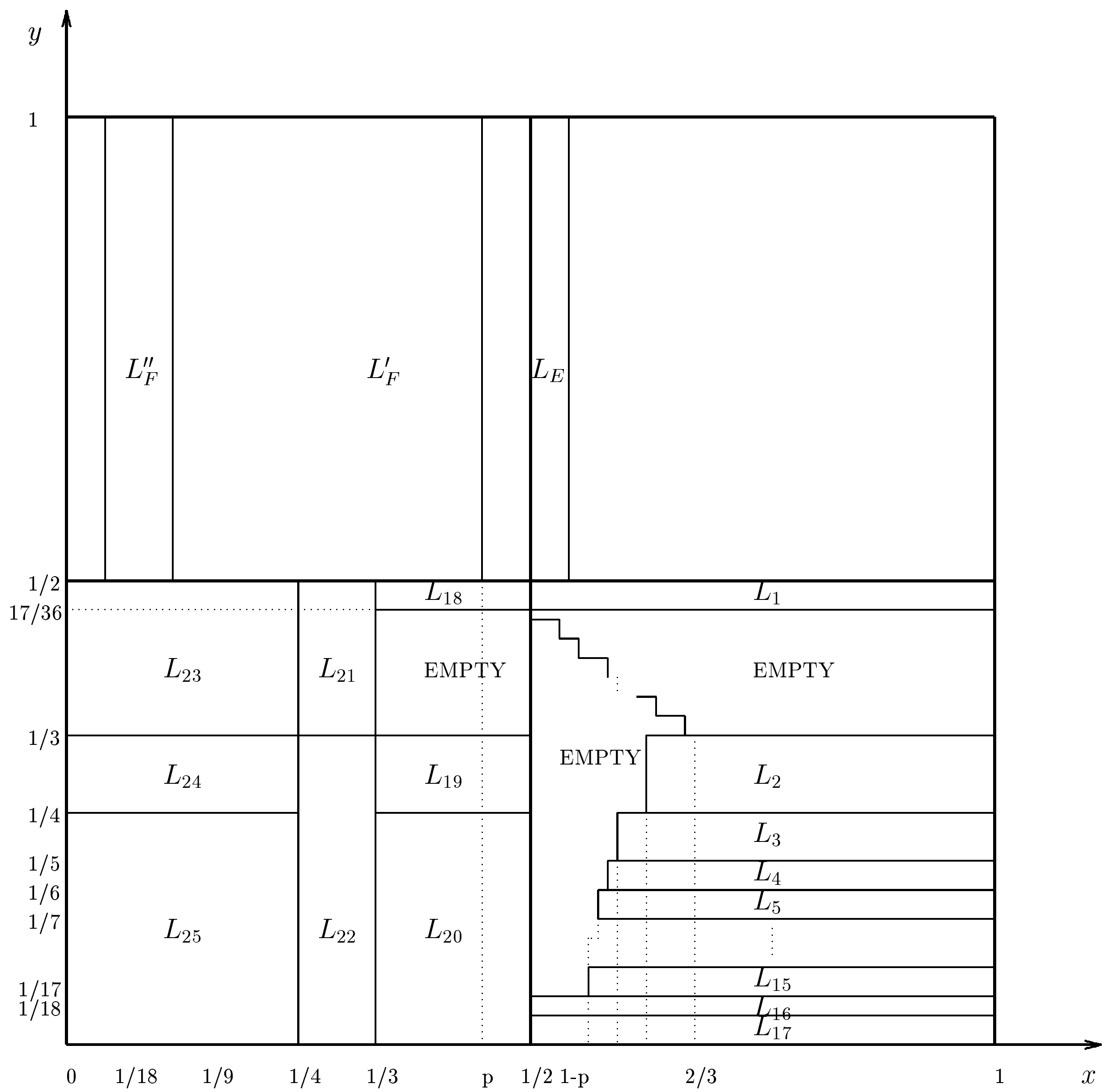

Figure 6: Combination of L_C and $L'_D \cup L''_D$: $L'_D \cup L''_D$ is totally packed.

The next theorem gives an asymptotic performance bound of the Algorithm $\mathcal{A}_k$ when $k \to \infty$. After the proof of this result we show that for relatively small value of k ($k = 13$) the Algorithm $\mathcal{A}_k$ has already an asymptotic performance bound that is very close to the value shown for $k \to \infty$. This conclusion will follow from the proof of the next theorem.

Theorem 3.2 *For any instance L of* TPP *we have*

$$\mathcal{A}_k(L) \leq \alpha_k \cdot \mathrm{OPT}(L) + \left(2k + \frac{597}{8}\right) Z \ ,$$

where $\alpha_k \to \frac{579+\sqrt{199145}}{384} = 2.669\ldots$ as $k \to \infty$.

Proof. Let us recall that when $k \to \infty$ the value of $r_1^{(k)}$ tends to $\frac{4}{9}$, $r_1 = r_1^{(k)} < \frac{4}{9}$ (see Definition 3.1). Each of the packings $\mathcal{P}_i$, $i \in \{1, \ldots, 25\} \setminus \{1, 18\}$, has an area guarantee that is at least $\frac{17}{36}$, this minimum being attained when $i \in \{16, 17\}$. Thus applying Lemma 2.2 and Lemma 2.3 we can conclude that

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

Now, for each of the packings $\mathcal{Q} \in \left\{\mathcal{P}_{i,j}, \tilde{\mathcal{P}'_{i,j}}, \tilde{\mathcal{P}''_{i,j}}\right\}$ that are used to generate the packing $\mathcal{P}_{AB}$ at the end of step 5, $H(\mathcal{Q}) \leq \frac{56}{27} V(\mathcal{Q}) + Z$. To see this, apply Lemma 2.4 together with the fact that for each packing $\mathcal{Q}$ that combines sets of L_A and L_B, the combined area is at least $\frac{27}{56}$. As there is a maximum of $(2k-1) + 28 + 14 \ = \ 2k + 41$ packings generated from combinations of sets in L_A and L_B, we can see that $H(\mathcal{P}_{AB}) \leq \frac{56}{27} V(L_{AB}) + (2k + 41)Z$. Thus the following inequality holds:

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

For the packings $\mathcal{P}_{CD'}$ and $\mathcal{P}_{CD''}$ (in step 6.3), since the combined area is at least $(\frac{1}{4} + \frac{r_1}{2})$, it follows by Lemma 2.4 that

$$H(\mathcal{P}_{CD}) \leq \frac{1}{\left(\frac{1}{4} + \frac{r_1}{2}\right)} V(L_{CD}) + 2Z \ . \tag{3}$$

Let us now analyse the two possible cases (*cf.* step 6.6).

Case 1: $L_C \subseteq L_{CD}$ and $p = \frac{\sqrt{199145}-195}{570} = 0.440\ldots$.

For the packings $\mathcal{P}_1$ and $\mathcal{P}_{18}$ the following inequalities hold:

$$H(\mathcal{P}_1) \ \leq \ \frac{1}{r_1} V(L_1) + Z \ , \tag{4}$$

$$H(\mathcal{P}_{18}) \ \leq \ \frac{1}{\frac{4}{9}} V(L_{18}) + Z \ . \tag{5}$$

Since each of the packings $\mathcal{P}_{EF'}$ and $\mathcal{P}_{EF''}$ has an area guarantee that is at least $\frac{3}{10}$, we can conclude that

$$H(\mathcal{P}_{EF}) \leq \frac{10}{3}V(L_{EF}) + 2Z \ . \tag{6}$$

Subcase 1.1: $L_E \subseteq L_{EF}$

By Lemma 2.8,

$$H(\mathcal{P}_{UD}) \leq \frac{5}{4}\mathrm{OPT}(P_2' \cup P_4') + \frac{53}{8}Z \ . \tag{7}$$

Applying Lemma 2.5, since $S(b) \geq (1-p)\frac{19}{36}$ for $b \in P_4''$, it follows that

$$H(\mathcal{P}_{OC}) \leq \frac{1}{(1-p)\frac{19}{36}}V(P_4'') \ . \tag{8}$$

For the packings $\mathcal{P}_{2e}$ and $\mathcal{P}_{2d}$, using Lemma 2.2, we can conclude that

$$H(\mathcal{P}_{2e}\|\mathcal{P}_{2d}) \leq \frac{1}{\frac{1}{3}}V(P_{2e}'' \cup P_{2d}'') + 2Z \ . \tag{9}$$

From (6), (8), (9) and the fact that $(1-p)\frac{19}{36} = \min\{\frac{3}{10}, (1-p)\frac{19}{36}, \frac{1}{3}\}$ it follows that

$$\begin{aligned} H(\mathcal{P}') &= H(\mathcal{P}_{OC}\|\mathcal{P}_{2e}\|\mathcal{P}_{2d}\|\mathcal{P}_{EF}) \\ &\leq \frac{1}{(1-p)\frac{19}{36}}V\left(P_4'' \cup P_{2e}'' \cup P_{2d}'' \cup L_{EF}\right) + 4Z \\ &= \frac{1}{(1-p)\frac{19}{36}}V\left(P_2' \cup P_4'\right) + 4Z \ . \end{aligned} \tag{10}$$

Since $\mathcal{P}'' = \{\mathcal{P} \in \{\mathcal{P}_{UD}, \mathcal{P}'\} : \ H(\mathcal{P}) \ \text{ is } \ \text{minimum}\}$, we have

$$H(\mathcal{P}'') = \min\{H(\mathcal{P}_{UD}), H(\mathcal{P}')\} \ . \tag{11}$$

Now for the packing $\mathcal{P}_{aux} = \mathcal{P}_{AB}\|\mathcal{P}_{CD}\|\mathcal{P}_1\|\ldots\|\mathcal{P}_{25}$, using the inequalities (2),...,(5) and the fact that $r_1 = \min\left\{\frac{17}{36}, r_1, \frac{1}{4} + \frac{r_1}{2}, \frac{4}{9}\right\}$, we obtain

$$H(\mathcal{P}_{aux}) \leq \frac{1}{r_1}V(L_{aux}) + (2k+68)Z \ , \tag{12}$$

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

Let

$$\mathcal{H}_1 \ := \ H(\mathcal{P}'') - \frac{53}{8}Z \ , \tag{13}$$

$$\mathcal{H}_2 \ := \ H(\mathcal{P}_{aux}) - (2k+68)Z \ . \tag{14}$$

From (7) and (11), in particular we have

$$\mathcal{H}_1 \le \frac{5}{4}\mathrm{OPT}(P_2' \cup P_4') \ ,$$

and therefore,

$$\mathrm{OPT}(P_2' \cup P_4') \ge \frac{4}{5}\mathcal{H}_1 \ .$$

Thus,

$$\mathrm{OPT}(L) \ge \mathrm{OPT}(P_2' \cup P_4') \ge \frac{4}{5}\mathcal{H}_1 \ . \tag{15}$$

Note that from (12) and (14) we can conclude that

$$V(L_{aux}) \ge r_1\mathcal{H}_2 \ . \tag{16}$$

On the other hand, from (11) and (10) we have

$$\begin{aligned} H(\mathcal{P}'') &\le H(\mathcal{P}') \\ &\le \frac{1}{(1-p)\frac{19}{36}}V(P_2' \cup P_4') + 4Z \\ &\le \frac{1}{(1-p)\frac{19}{36}}V(P_2' \cup P_4') + \frac{53}{8}Z \ , \end{aligned}$$

and thus,

$$\mathcal{H}_1 = H(\mathcal{P}'') - \frac{53}{8}Z \le \frac{1}{(1-p)\frac{19}{36}}V(P_2' \cup P_4') \ ,$$

i.e.,

$$V(P_2' \cup P_4') \ge (1-p)\frac{19}{36}\mathcal{H}_1 \ . \tag{17}$$

Since $V(L) = V(L_{aux}) + V(P_2' \cup P_4')$, using (16) and (17) we get

$$V(L) \ge r_1\mathcal{H}_2 + (1-p)\frac{19}{36}\mathcal{H}_1 \ .$$

Thus,

$$\mathrm{OPT}(L) \ge V(L) \ge r_1\mathcal{H}_2 + (1-p)\frac{19}{36}\mathcal{H}_1 \ .$$

Combining (15) and the inequality above, it follows that

$$\mathrm{OPT}(L) \ge \max\left\{\frac{4}{5}\mathcal{H}_1, (1-p)\,\frac{19}{36}\mathcal{H}_1 + r_1\mathcal{H}_2\right\} \ .$$

Since $H(\mathcal{P}) = H(\mathcal{P}_{aux}) + H(\mathcal{P}'')$; using (13) and (14), we have

$$\begin{aligned} H(\mathcal{P}) &= \left(\mathcal{H}_2 + (2k+68)\,Z + \mathcal{H}_1 + \frac{53}{8}Z\right) \\ &= \mathcal{H}_1 + \mathcal{H}_2 + \left(2k + \frac{597}{8}\right) Z\ . \end{aligned}$$

Thus, $\mathcal{A}_k(L) \le \alpha'_k(r_1) \cdot \mathrm{OPT}(L) + \left(2k + \frac{597}{8}\right) Z$, where $\alpha'_k(r_1) = \frac{49+95p+180r_1}{144r_1}$. To prove this, we show that $\frac{\mathcal{H}_1+\mathcal{H}_2}{\max\left\{\frac{4}{5}\mathcal{H}_1, (1-p)\frac{19}{36}\mathcal{H}_1 + r_1\mathcal{H}_2\right\}} \le \alpha'_k(r_1)$, by analysing two cases:

Case (a): $\max\left\{\frac{4}{5}\mathcal{H}_1, (1-p)\,\frac{19}{36}\mathcal{H}_1 + r_1\mathcal{H}_2\right\} = \frac{4}{5}\mathcal{H}_1$.

In this case, $\mathcal{H}_1 \ge \frac{180r_1}{49+95p}\mathcal{H}_2$, and thus

$$\begin{aligned} \frac{\mathcal{H}_1+\mathcal{H}_2}{\max\left\{\frac{4}{5}\mathcal{H}_1, (1-p)\,\frac{19}{36}\mathcal{H}_1 + r_1\mathcal{H}_2\right\}} &= \frac{\mathcal{H}_1+\mathcal{H}_2}{\frac{4}{5}\mathcal{H}_1} \\ &= \frac{5}{4} + \frac{5}{4}\frac{\mathcal{H}_2}{\mathcal{H}_1} \\ &\le \frac{49+95p+180r_1}{144r_1}\ . \end{aligned}$$

Case (b): $\max\left\{\frac{4}{5}\mathcal{H}_1, (1-p)\,\frac{19}{36}\mathcal{H}_1 + r_1\mathcal{H}_2\right\} = (1-p)\,\frac{19}{36}\mathcal{H}_1 + r_1\mathcal{H}_2$.

Then $\mathcal{H}_1 \le \frac{180r_1}{49+95p}\mathcal{H}_2$.

In this case, note that $\frac{\mathcal{H}_1+\mathcal{H}_2}{(1-p)\frac{19}{36}\mathcal{H}_1+r_1\mathcal{H}_2}$ is a strictly increasing function of $\mathcal{H}_1$, and hence when $\mathcal{H}_1 = \frac{180r_1}{49+95p}\mathcal{H}_2$ it attains its maximum value. Thus,

$$\begin{aligned} \frac{\mathcal{H}_1+\mathcal{H}_2}{\max\left\{\frac{4}{5}\mathcal{H}_1, (1-p)\,\frac{19}{36}\mathcal{H}_1 + r_1\mathcal{H}_2\right\}} &= \frac{\mathcal{H}_1+\mathcal{H}_2}{(1-p)\,\frac{19}{36}\mathcal{H}_1 + r_1\mathcal{H}_2} \\ &\le \frac{49+95p+180r_1}{144r_1}\ . \end{aligned}$$

Subcase 1.2: $L_F \subseteq L_{EF}$

In this case,

$$H(\mathcal{P}_{OC}) \le \frac{1}{\frac{19}{72}} V(P''_4)\ . \tag{18}$$

Since $\mathcal{P}' = \mathcal{P}_{OC} \| \mathcal{P}_{EF'} \| \mathcal{P}_{EF''}$ and all these packings combine boxes in P'_4, it follows that

$$\mathrm{OPT}(L) \ge \mathrm{OPT}(P''_4 \cup L_{EF}) \ge H(\mathcal{P}_{OC}) + H(\mathcal{P}_{EF}) - 2Z = H(\mathcal{P}') - 2Z\ . \tag{19}$$

Recalling that $\mathcal{P}' = \mathcal{P}_{OC} \| \mathcal{P}_{EF}$, and using (6) and (18) we have

$$H(\mathcal{P}') \leq \frac{1}{\frac{19}{72}} V(L_{EF} \cup \mathcal{P}_4'') + 2Z \ . \tag{20}$$

Now using Lemma 2.2 for the packings $\mathcal{P}_{2e}$ and $\mathcal{P}_{2d}$ we can conclude that

$$H(\mathcal{P}_{2e} \| \mathcal{P}_{2d}) \leq \frac{1}{p} V(P_{2e}'' \cup P_{2d}'') + 2Z \ . \tag{21}$$

From (2),..., (5) and (21) and the fact that $p = \min\{\frac{17}{36}, \frac{1}{4} + \frac{r_1}{2}, r_1, \frac{4}{9}, p\}$ we have

$$H(\mathcal{P}_{aux}) \leq \frac{1}{p} V(L_{aux}) + (2k + 70)Z \ . \tag{22}$$

Let

$$\begin{aligned} \mathcal{H}_1 &:= H(\mathcal{P}') - 2Z \ , & (23) \\ \mathcal{H}_2 &:= H(\mathcal{P}_{aux}) - (2k + 70)Z \ . & (24) \end{aligned}$$

From (19) and (23), it follows that

$$\mathrm{OPT}(L) \geq \mathcal{H}_1 \ . \tag{25}$$

Using (20) and (23), resp. (22) and (24), we have

$$\begin{aligned} V(L_{EF} \cup P_4'') &\geq \frac{19}{72} \mathcal{H}_1 \ , \\ V(L_{aux}) &\geq p\mathcal{H}_2 \ . \end{aligned}$$

Since $V(L) = V(L_{EF} \cup P_4'' \cup L_{aux})$, adding up the above inequalities, we get

$$V(L) \geq \frac{19}{72} \mathcal{H}_1 + p\mathcal{H}_2 \ ,$$

and thus

$$\mathrm{OPT}(L) \geq \frac{19}{72} \mathcal{H}_1 + p\mathcal{H}_2 \ .$$

Combining the inequality above with (25) we can prove that

$$\mathcal{A}_k(L) \leq \alpha_k'' \cdot \mathrm{OPT}(L) + (2k + 72)Z \ ,$$

where $\alpha_k'' = \frac{53+72p}{72p}$. This can be shown by proving that $\frac{\mathcal{H}_1 + \mathcal{H}_2}{\max\{\mathcal{H}_1, \frac{19}{72}\mathcal{H}_1 + p\mathcal{H}_2\}} \leq \alpha_k''$. The proof can be done analogously to the previous case, and therefore will be omitted.

The value of p ($p = \frac{\sqrt{199145}-195}{570}$) that we considered in the algorithm was in fact obtained by setting $\alpha'_k(\frac{4}{9}) = \alpha''_k$. We leave to the reader the verification of this fact.

Thus, from the analysis of both subcases we can conclude that

$$\mathcal{A}_k(L) \leq \alpha_k \cdot \mathrm{OPT}(L) + \left(2k + \frac{597}{8}\right) Z \ ,$$

where $\alpha_k \to \alpha'_k(\frac{4}{9}) = \alpha''_k = \frac{\sqrt{199145}+579}{384} = 2.669\ldots$ as $k \to \infty$.

Case 2: $L_D \subseteq L_{CD}$ and $p = \frac{\sqrt{23401}-71}{180} = 0.455\ldots$

In this case the proof is similar to the one presented in Case 1, therefore we omit the details and simply mention the inequalities that can be obtained.

$$\begin{aligned} H(\mathcal{P}_i) &\leq \frac{36}{17} V(L_i) + Z \text{ for } i \in \{1, 18\} \ . \\ H(\mathcal{P}_{EF}) &\leq \frac{36}{11} V(L_{EF}) + 2Z \ . \end{aligned}$$

Subcase 2.1: $L_E \subseteq L_{EF}$

$$\begin{aligned} H(\mathcal{P}_{UD}) &\leq \frac{5}{4}\mathrm{OPT}(P'_2 \cup P'_4) + \frac{53}{8} Z \ . \\ H(\mathcal{P}_{OC}) &\leq \frac{1}{(1-p)\frac{1}{2}} V(P''_4) \ . \\ H(\mathcal{P}_{2e} \| \mathcal{P}_{2d}) &\leq \frac{1}{\frac{1}{3}} V(P''_{2e} \cup P''_{2d}) + 2Z \ . \\ H(\mathcal{P}') &= H(\mathcal{P}_{OC} \| \mathcal{P}_{2e} \| \mathcal{P}_{2d} \| \mathcal{P}_{EF}) \\ &\leq \frac{1}{(1-p)\frac{1}{2}} V\left(P'_2 \cup P'_4\right) + 4Z \ . \end{aligned}$$

Since $\frac{1}{4} + \frac{r_1}{2} = \min\left\{\frac{17}{36}, \frac{1}{4} + \frac{r_1}{2}\right\}$, we have

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

Let

$$\begin{aligned} \mathcal{H}_1 &:= H(\mathcal{P}'') - \frac{53}{8} Z \ , \\ \mathcal{H}_2 &:= H(\mathcal{P}_{aux}) - (2k + 68)Z \ . \end{aligned}$$

Then

$$\mathrm{OPT}(L) \geq \frac{4}{5}\mathcal{H}_1 \ .$$

On the other hand,

$$\begin{aligned} V(L_{aux}) &\geq \left(\frac{1}{4}+\frac{r_1}{2}\right)\mathcal{H}_2 \ \text{ and} \\ V(P_2' \cup P_4') &\geq (1-p)\frac{1}{2}\mathcal{H}_1 \ , \end{aligned}$$

and therefore

$$\mathrm{OPT}(L) \geq V(L) \geq \left(\frac{1}{4}+\frac{r_1}{2}\right) H_2 + (1-p)\frac{1}{2}\mathcal{H}_1 \ .$$

Thus,

$$\mathrm{OPT}(L) \geq \max\left\{\frac{4}{5}\mathcal{H}_1, (1-p)\,\frac{1}{2}\mathcal{H}_1 + \left(\frac{1}{4}+\frac{r_1}{2}\right)\mathcal{H}_2\right\} \ .$$

Therefore, $\mathcal{A}_k(L) \leq \beta_k'(r_1)\cdot \mathrm{OPT}(L) + \left(2k+\frac{597}{8}\right)Z$, where $\beta_k'(r_1) = \frac{11+10p+10r_1}{4+8r_1}$. The last inequality follows by showing that $\frac{\mathcal{H}_1+\mathcal{H}_2}{\max\left\{\frac{4}{5}\mathcal{H}_1,(1-p)\frac{1}{2}\mathcal{H}_1+\left(\frac{1}{4}+\frac{r_1}{2}\right)\mathcal{H}_2\right\}} \leq \beta_k'(r_1)$.

Subcase 2.2: $L_F \subseteq L_{EF}$

$$\begin{aligned} H(\mathcal{P}_{OC}) &\leq \frac{1}{\frac{1}{4}}V(P_4'') \ . \\ \mathrm{OPT}(L) &\geq H(\mathcal{P}') - 2Z \ . \\ H(\mathcal{P}') &\leq \frac{1}{\frac{1}{4}}V(L_{EF} \cup \mathcal{P}_4'') + 2Z \ . \\ H(\mathcal{P}_{2e} \| \mathcal{P}_{2d}) &\leq \frac{1}{p}V(P_{2e}'' \cup P_{2d}'') + 2Z \ . \end{aligned}$$

Since $p = \min\left\{\frac{1}{4}+\frac{r_1}{2}, \frac{17}{36}, p\right\}$ we have

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

Let

$$\begin{aligned} \mathcal{H}_1 &:= H(\mathcal{P}') - 2Z \ , \\ \mathcal{H}_2 &:= H(\mathcal{P}_{aux}) - (2k+70)Z \ . \end{aligned}$$

Then

$$\begin{aligned} \mathrm{OPT}(L) &\geq \mathcal{H}_1 \ , \\ V(L_{EF} \cup P_4'') &\geq \frac{1}{4}\mathcal{H}_1 \ , \\ V(L_{aux}) &\geq p\mathcal{H}_2 \ . \end{aligned}$$

Thus

$$\mathrm{OPT}(L) \geq V(L) \geq \frac{1}{4}\mathcal{H}_1 + p\mathcal{H}_2 \ .$$

Therefore,

$$\mathcal{A}_k(L) \leq \beta_k'' \cdot \mathrm{OPT}(L) + (2k+72)Z \ ,$$

where $\beta_k'' = \frac{3+4p}{4p}$. The last inequality is proved by showing that $\frac{\mathcal{H}_1+\mathcal{H}_2}{\max\{\mathcal{H}_1, \frac{1}{4}\mathcal{H}_1+p\mathcal{H}_2\}} \leq \frac{3+4p}{4p}$.

Here again, the value of p ($p = \frac{\sqrt{23401}-71}{180}$) that we considered in the algorithm was obtained by setting $\beta_k'(\frac{4}{9}) = \beta_k''$. Thus, for the given value of p, as in the previous case we can conclude that

$$\mathcal{A}_k(L) \leq \beta_k \cdot \mathrm{OPT}(L) + \left(2k + \frac{597}{8}\right) Z \ ,$$

where $\beta_k \to \beta_k'(\frac{4}{9}) = \beta_k'' = \frac{\sqrt{23401}+207}{136} = 2.64\ldots$ as $k \to \infty$.

The theorem follows from the conclusions obtained in the cases 1 and 2.

□

Corollary 3.3 *For any instance L of* TPP *and $k \geq 13$ we have*

$$\mathcal{A}_k(L) \leq \gamma_k \cdot \mathrm{OPT}(L) + \left(2k + \frac{597}{8}\right) Z \ ,$$

where $\gamma_k = \frac{99+1080r_1^{(k)}+\sqrt{199145}}{864r_1^{(k)}} < 2.67$.

Proof. The result follows from the proof of the previous theorem. It is sufficient to observe that for $k \geq 13$ we have $r_1^{(k)} \geq 0.444430896$, and therefore all arguments used in the proof remain valid. Note that the statement of the corollary holds taking

$$\begin{aligned}
\gamma_k &= \max\left\{\frac{49+95p_1+180r_1^{(k)}}{144r_1^{(k)}}, \frac{53+72p_1}{72p_1}, \frac{11+10p_2+10r_1^{(k)}}{4+8r_1^{(k)}}, \frac{3+4p_2}{4p_2}\right\} \\
&= \frac{49+95p_1+180r_1^{(k)}}{144r_1^{(k)}} = \frac{99+1080r_1^{(k)}+\sqrt{199145}}{864r_1^{(k)}} \ ,
\end{aligned}$$

where p_i corresponds to the value of p in the Case i, $i = 1, 2$. That is, $p_1 = \frac{\sqrt{199145}-195}{570}$ and $p_2 = \frac{\sqrt{23401}-71}{180}$.

□

Proposition 3.4 *The asymptotic performance bound of the Algorithm $\mathcal{A}_k$, $k \geq 13$, is between* 2.5 *and* 2.67.

Proof. By the Theorem 3.2 it is sufficient to prove that 2.5 is a lower bound for the asymptotic performance bound of the Algorithm $\mathcal{A}_k$.

Let L be an instance for TPP, $L = L' \cup L''$, where $L' = (b'_1, b'_2, \ldots, b'_{2N})$ and $L'' = (b''_1, b''_2, \ldots, b''_{27N})$, and N is a large positive integer.

Each box b'_i in L', $i = 1, \ldots, 2N$, is defined as

$$b'_i = \left(\frac{1}{2} + \epsilon, \frac{1}{2} + \epsilon, 1\right) .$$

Each box b''_i in L'', $i = 1, \ldots, 27N$, is defined as

$$b''_i = \begin{cases} (\delta, \delta, 1 - (i-1)\xi_N) & \text{if } \; i \bmod 9 = 0 \\ \left(\frac{1}{4} - \epsilon, \frac{1}{4} - \epsilon, 1 - (i-1)\xi_N\right) & \text{otherwise .} \end{cases}$$

The values for ϵ, ξ_N and δ must be positive and very small, and furthermore the following must hold: $8\left(\frac{1}{4} - \epsilon\right)^2 + \delta^2 \leq \frac{1}{2} + \left(\frac{1}{4}\right)^2$ and $9\left(\frac{1}{4} - \epsilon\right)^2 + \delta^2 > \frac{1}{2} + \left(\frac{1}{4}\right)^2$. This can be achieved by fixing a small δ and taking $\epsilon = \frac{\delta^2}{8}$.

The Algorithm $\mathcal{A}_k$ applied to the list L generates a packing $\mathcal{P} = \mathcal{P}' \| \mathcal{P}''$ where $\mathcal{P}'$ (resp. $\mathcal{P}''$) is the packing generated by the Algorithm OC (resp. LL$(L'', 4)$) applied to the list L' (resp. L'').

It is clear that $H(\mathcal{P}') = 2N$. As for the packing $\mathcal{P}''$, it is generated as follows : $\mathcal{P}''$ consists of $3N$ levels, each consisting of 8 boxes of type $\left(\frac{1}{4} - \epsilon, \frac{1}{4} - \epsilon, 1 - (i-1)\xi_N\right)$ and one box of type $(\delta, \delta, 1 - (i-1)\xi_N)$. Therefore, $H(\mathcal{P}'') = 3N - h(\xi_N)$, where $h(\xi_N) = 9\xi_N \left(\frac{9N^2 - 3N}{2}\right)$; and thus

$$H(\mathcal{P}) = 2N + 3N - h(\xi_N) = 5N - h(\xi_N) .$$

A better packing $\mathcal{P}^*$ of the list L can be obtained by generating:

- $2N$ levels, each consisting of one box of type $\left(\frac{1}{2} + \epsilon, \frac{1}{2} + \epsilon, 1\right)$ and 12 boxes of type $\left(\frac{1}{4} - \epsilon, \frac{1}{4} - \epsilon, 1 - (i-1)\xi_N\right)$;
- one level consisting of all boxes of the form $(\delta, \delta, 1 - (i-1)\xi_N)$. We note that this is possible by choosing δ conveniently.

Thus $H(\mathcal{P}^*) \leq 2N + 1$.

Therefore, by choosing ξ_N such that $h(\xi_N)$ tends to 0 when $N \to \infty$, we have

$$\lim_{N\to\infty} \frac{H(\mathcal{P})}{\mathrm{OPT}(L)} \geq \lim_{N\to\infty} \frac{5N - h(\xi_N)}{2N+1} = \frac{5}{2} . \qquad \square$$

4 Concluding remarks

It is easy to see that all algorithms we have used in the Algorithm $\mathcal{A}_k$ —except for the Algorithm UD and LL— have time complexity $\mathcal{O}(m \log m)$, where m is the number of boxes in the input list. It can be seen that the Algorithm LL also has the same complexity [4]. As for the Algorithm UD, the authors claim (*cf.* [1]) that it can be implemented to run in time $\mathcal{O}(m \log m)$. Thus, the Algorithm $\mathcal{A}_k$ has time complexity $\mathcal{O}(n \log n)$, where n is the number of boxes in the input list.

In the special case of TPP in which the input list consists of boxes with square bottom we have developed an algorithm with an asymptotic performance bound close to 2.36. This result appears in a forthcoming paper where another variant of TPP is discussed [6].

Acknowledgement. We thank the referee whose suggestions improved the presentation of this paper

References

[1] B. S. Baker, D. J. Brown, and H. P. Katseff. A $\frac{5}{4}$ algorithm for two-dimensional packing. *J. of Algorithms*, 2:348–368, 1981.

[2] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of $\mathcal{NP}$-Completeness.* Freeman, San Francisco, 1979.

[3] K. Li and K-H. Cheng. On three-dimensional packing. *SIAM J. Comput.*, 19:847–867, 1990.

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

[5] K. Li and K-H. Cheng. Heuristic algorithms for on-line packing in three dimensions. *J. of Algorithms*, 13:589–605, 1992.

[6] F. K. Miyazawa and Y. Wakabayashi. Polynomial approximation algorithms for the orthogonal z-oriented 3-D packing problem. Technical Report RT–MAC–9512, Instituto de Matemática e Estatística – Universidade de São Paulo – Brasil, 1995.