Approximation Algorithms for 3D Orthogonal Knapsack*

Florian Diedrich[†] Rolf Harren[‡] Klaus Jansen[†] Ralf Thöle[†] Henning Thomas[†]

We study non-overlapping axis-parallel packings of 3D boxes with profits into a dedicated bigger box where rotation is either forbidden or permitted; we wish to maximize the total profit. Since this optimization problem is NP-hard, we focus on approximation algorithms. We obtain fast and simple algorithms for the non-rotational scenario with approximation ratios $9 + \epsilon$ and $8 + \epsilon$ as well as an algorithm with approximation ratio $7 + \epsilon$ that uses more sophisticated techniques; these are the smallest approximation ratios known for this problem. Furthermore, we show how the used techniques can be adapted to the case where rotation by 90° either around the z-axis or around all axes is permitted, where we obtain algorithms with approximation ratios $6 + \epsilon$ and $5 + \epsilon$, respectively. Finally our methods yield a 3D generalization of a packability criterion and a strip packing algorithm with absolute approximation ratio 29/4, improving the previously best known result of 45/4.

Topics: approximation algorithms, computational and structural complexity, geometric configurations.

1 Introduction

Given a list $L = \{R_1, \ldots, R_n\}$ of boxes with sizes $R_i = (x_i, y_i, z_i)$ and positive profits p_i for each $i \in \{1, \ldots, n\}$ and a dedicated box $Q = (a, b, c)$, we study non-overlapping axis-parallel packings of sublists of L into Q which we call *feasible*. For simplicity we

*Research supported in part by DFG Project, Entwicklung und Analyse von Approximativen Algorithmen für Gemischte und Verallgemeinerte Packungs- und Überdeckungsprobleme, JA 612/10-1, in part by the German Academic Exchange Service DAAD, in part by project AEOLUS, EU contract number 015964, and in part by a grant “DAAD Doktorandenstipendium” of the German Academic Exchange Service DAAD. Part of this work was done while visiting the LIG, Grenoble University.

[†]Institut für Informatik, Christian-Albrechts-Universität zu Kiel, Olshausenstr. 40, 24098 Kiel, Germany

[‡]Max-Planck-Institut für Informatik, Campus E1 4, 66123 Saarbrücken, Germany

Journal of Computer Science and Technology, 23(5): 749–762, 2008.
http://www.springer.de/comp/lncs/index.html

call Q a *bin*. We wish to select a sublist which permits a packing and maximizes the profit. This problem will be called the *orthogonal three-dimensional knapsack problem* or *OKP-3* for short and we denote the optimal profit by OPT. It is a natural generalization of the knapsack problem (KP) which is known to be NP-hard. This makes an exact algorithm with a polynomial worst-case runtime bound impossible unless P = NP holds. For this reason, we concentrate on approximation algorithms; we refer the reader to [31] for a detailed description of the approach and common notions. W.l.o.g. we assume $a = b = c = 1$ and that each $R_i \in L$ can be packed by otherwise removing infeasible boxes and scaling in $\mathcal{O}(n)$ time. Note that the scaling is only possible in the non-rotational case. In the rotational case the fixed size of the bin is an explicit assumption.

Related problems. Different geometrically constrained two- and three-dimensional packing problems were studied, resulting in three main directions.

In *strip packing* the target area is a strip of infinite height; the objective is to minimize the height of the packing. For the 2D case, Sleator [29] proved an approximation ratio of 5/2; Baker et al. [1] obtained an asymptotic approximation ratio of 5/4. The best known absolute approximation ratio of 2 was obtained independently with different techniques by Schiermeyer [28] and Steinberg [30]. Kenyon & Rémila [18] found an AFPTAS (asymptotic fully polynomial time approximation scheme) for the problem; they obtained an additive error of $\mathcal{O}(1/\epsilon^2)$. This additive error was later improved by Jansen & Solis-Oba in [15], where the authors presented an APTAS (asymptotic polynomial time approximation scheme) with additive error 1. For the 3D case, research has focused mainly on the *asymptotic* approximation ratio, where Miyazawa & Wakabayashi [26] found an algorithm with asymptotic approximation ratio of 27/10 at most. An asymptotic ratio of $2+\epsilon$ was obtained by Jansen & Solis-Oba [14]; this was improved to 1.691 by Bansal et al. [4]. The best known *absolute* approximation ratio of 45/4 follows from an asymptotic approximation ratio of 13/4 from [21]. Li & Cheng [22] studied the *on-line* version, resulting in a competitive ratio of 29/10.

In *bin packing* the objective is to minimize the number of identical bins. For the 1D case, an APTAS was presented by Fernandez de la Vega & Lueker [12], while Li & Yue [23] proved the bound of $11/9\text{OPT}+7/9$ for the popular FFD algorithm. The study of FFD was settled recently by Dósa [8] who proved that the bound $11/9\text{OPT} + 6/9$ is tight. For the 2D case, an asymptotic approximation ratio of 1.691 was obtained by Caprara [5]; this result was improved to an asymptotic approximation ratio of 1.525 by Bansal et al. [2]. Furthermore, Bansal et al. [3] proved that 2D bin packing does not admit an APTAS (asymptotic polynomial time approximation scheme) and therefore no FPTAS (fully polynomial time approximation scheme) if P $\neq$ NP. They also presented an APTAS for packing d-dimensional cubes into the minimum number of unit cubes in the same paper.

In the *knapsack* scenario the number of bins is a fixed constant [7], usually 1. For the 2D case, Jansen & Zhang [16] obtained an approximation ratio of $2+\epsilon$. Classical 1D knapsack problems are relatively well understood, see [17, 25] for surveys.

Although these problems are closely related, the results cannot be transferred directly. One main contrast between bin and strip packing on the one hand and knapsack on the other hand is that in the first setting all boxes of the instance must be packed but in the latter a feasible selection of items must be found.

Previous results and applications. Harren [13] obtained a ratio of $9/8+\epsilon$ for the special case of packing cubes and proved the APX-completeness of the general case [6]. A *cutting stock* application is cutting blocks with given profits from larger pieces of material to maximize the profit; another application is the problem of selecting boxes to be transported in a container. Besides these, the problem is motivated from multiprocessor scheduling on grid topology. In this perspective, for a time slice of fixed duration, a set of jobs to be executed must be chosen and each job requires a subgrid of prespecified rectangular shape. For a special case of this application, Ye & Zhang [32] presented an on-line algorithm–see [11] for a study of similar problems.

New results. Our contribution is a fast and simple $(9+\epsilon)$-approximation algorithm based on strip packing which is refined to an $(8+\epsilon)$-approximation algorithm. Both of these have practical running times. With more sophisticated techniques we obtain a $(7+\epsilon)$-approximation algorithm. We also study the case where rotation by 90° either around the z-axis or around all axes is permitted, where we significantly improve upon the approximation ratios of the algorithms presented for OKP-3. We derive approximation ratios of $6+\epsilon$ for the former and $5+\epsilon$ for the latter case. Finally our methods yield a three-dimensional generalization of a packability criterion and a strip packing algorithm with absolute approximation ratio 29/4.

This paper is organized as follows. In Section 2 we present a fast algorithm for non-rotational packing which is improved in Section 3. In Section 4 we generalize a two-dimensional packability criterion for boxes and improve a known result on strip packing in Section 5 before turning back to the knapsack problem in Section 6 where we obtain a better yet more costly algorithm. Finally, we discuss the cases of rotational packing in Section 7 before concluding with open problems in Section 8.

2 An Algorithm Based on Strip Packing

We approximately solve a relaxation by selecting $L' \subseteq L$ that is at least near-optimal and has a total volume of at most 1. This relaxed solution is partitioned into 9 sublists. For each of these a packing into the bin will be generated. Out of these one with maximum profit is chosen, resulting in a $(9+\epsilon)$-approximation algorithm. More precisely L' will be packed into a strip $[0,1]\times[0,1]\times[0,\infty)$ by a level-oriented algorithm, i.e. an algorithm which packs all boxes into disjoint levels and stacks these levels on top of one another into the strip. We partition the strip into packings of sublists of L' and among these return one with maximum profit.

For each box R_i the rectangle (x_i, y_i) is called the *base rectangle* of R_i, denoted as $br(R_i)$. Such a rectangle (x_i, y_i) is called

$$\begin{array}{rclcl} \textit{big} & :\Leftrightarrow & x_i \in (1/2,1] & \text{and} & y_i \in (1/2,1], \\ \textit{long} & :\Leftrightarrow & x_i \in (1/2,1] & \text{and} & y_i \in (0,1/2], \\ \textit{wide} & :\Leftrightarrow & x_i \in (0,1/2] & \text{and} & y_i \in (1/2,1], \\ \textit{small} & :\Leftrightarrow & x_i \in (0,1/2] & \text{and} & y_i \in (0,1/2]. \end{array}$$

For each list L of boxes we use $V(L)$ to denote the total volume of L and for each list $\hat{L}$ of rectangles use $A(\hat{L})$ to denote the total area of $\hat{L}$. Furthermore, $P(L)$ denotes the total profit of L. Finally, for each list L of boxes use we $H(L)$ to denote the height of a packing of L where the packing itself will be clear from the context. We use the following theorem from [16] which is a refinement of the main result from [30].

Theorem 1. [16] *Let L be a list of n rectangles such that $A(L) \leq 1/2$ holds and no long rectangles or no wide rectangles occur in L, i.e., there might be either long or wide rectangles but not both. Then L permits a feasible packing into the unit square which can be generated in time $\mathcal{O}(n \log^2 n / \log \log n)$.*

Note that rectangles which are long or wide are not big. It is therefore possible in Theorem 1 that there is a big rectangle in L. First we apply the modified strip packing algorithm, then we construct the partition of the strip. The strip packing algorithm uses Theorem 1 to obtain an *area guarantee* for each but the last level, improving a result from [21].

Algorithm A.

1. Partition L into two sublists $L_1 := \{R_i | br(R_i) \text{ is long}\}$ and $L_2 := L \setminus L_1$. W.l.o.g. let $L_1 = \{R_1, \ldots, R_m\}$ and $L_2 = \{R_{m+1}, \ldots, R_n\}$.
2. Generate the packing for L_1 as follows.
   - 2.1. Find the boxes R_i in L_1 for which the area of $br(R_i)$ is greater than $1/4$ which are $R_{p+1}, \ldots, R_m$ w.l.o.g. Stack these on top of one another in direction z, each on its own level.
   - 2.2. Sort the remaining boxes $R_1, \ldots, R_p$ in non-increasing order of z_i, resulting in a list L_1'.
   - 2.3. Partition L_1' into consecutive sublists $L_1'', \ldots, L_v''$ where the total base area of each sublist is as close to $1/2$ as possible but not greater. Pack each of these sublists on a level by itself using Theorem 1. Stack all of these levels on top of one another in direction z.
3. Generate the packing for L_2 in a similar way as for L_1 by Theorem 1. The resulting steps are called Steps 3.1 – 3.3.
4. Concatenate the packings of L_1 and L_2 to obtain a packing of L.

Theorem 2. *For each list L of n boxes Algorithm A generates a packing of height at most $4V(L) + Z_1 + Z_2$ where Z_1 and Z_2 are the heights of the first levels generated in Steps 2.3 and 3.3. The construction can be carried out in time $\mathcal{O}(n \log^2 n / \log \log n)$.*

Proof. Consider Step 2.3 which generates levels $L_1'', \ldots, L_v''$. Let h_i denote the height of level L_i'' for each $i \in \{1, \ldots, v\}$. Each box R_i for $i \in \{p+1, \ldots, m\}$ processed in Step 2.1 forms a level itself, so the total height for the packing of L_1 is

$$H(L_1) := \sum_{i=p+1}^{m} z_i + \sum_{i=1}^{v} h_i \tag{1}$$

and we have $V(L) = V(L_1) + V(L_2)$. Furthermore,

$$V(L_1) = \sum_{i=p+1}^{m} x_i y_i z_i + \sum_{i=1}^{v} V(L_i'')$$

holds. In Step 2.1 we assert that $x_i y_i > 1/4$ for $i \in \{p+1, \ldots, m\}$, so we get

$$\sum_{i=p+1}^{m} x_i y_i z_i > \frac{1}{4} \sum_{i=p+1}^{m} z_i. \tag{2}$$

For each $i \in \{1, \ldots, v-1\}$ the total base area of the boxes in L_i'' is larger than 1/4 since otherwise the next box would have also been put in L_i' in Step 2.3. As the boxes are sorted in non-increasing order of height, and thus $z_j \geq h_{i+1}$ for $R_j \in L_i''$, we get that

$$\sum_{R_j \in L_i''} x_j y_j > \frac{1}{4} \quad \text{holds, hence} \quad V(L_i'') \geq \frac{1}{4} h_{i+1} \tag{3}$$

for each $i \in \{1, \ldots, v-1\}$. Combining (2) and (3) results in

$$V(L_1) > \frac{1}{4} \sum_{i=p+1}^{m} z_i + \frac{1}{4} \sum_{i=2}^{v} h_i$$

which together with (1) implies that

$$H(L_1) < 4V(L_1) + h_1 \leq 4V(L_1) + Z_1 \tag{4}$$

holds. For Step 3 we get

$$H(L_2) < 4V(L_2) + Z_2 \tag{5}$$

with a very similar analysis. From (4) and (5) we conclude that the height of our packing of L is bounded from above by $H(L_1) + H(L_2) < 4V(L) + Z_1 + Z_2$. The runtime is dominated by the application of Theorem 1 and thus bounded by $\mathcal{O}(n \log^2 n / \log \log n)$. □

The second part of the overall algorithm is a partition into at most 9 bins applied to the output of Algorithm A–see Figure 1.

Algorithm B.

1. Set $\delta := \epsilon/(9 + \epsilon)$. Use an FPTAS for KP from [17, 19] to select $L' \subseteq L$ such that $V(L') \leq 1$ and $P(L') \geq (1 - \delta)\mathrm{OPT}$ holds, where OPT denotes the optimum of the generated KP instance.
2. Use Algorithm A to generate a packing of L' into the strip but separate the first levels generated in Steps 2.3 and 3.3. Pack these into a bin each.

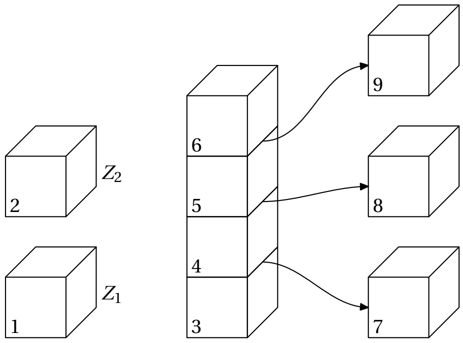

Figure 1: At most 9 bins are generated by Algorithm B.

3. By Theorem 2 the remaining strip has a height of at most $4V(L') \leq 4$. Consider the three cutting unit squares $[0,1] \times [0,1] \times \{i\}$ for $i \in \{1,2,3\}$. Generate a partition of the region $[0,1] \times [0,1] \times [0,4]$ into 7 subsets, namely 4 subsets which are each positioned in the regions $[0,1] \times [0,1] \times [i-1,i]$ for $i \in \{1,\dots,4\}$ but not intersecting any of the unit squares and 3 subsets of boxes which each intersect with one of the three cutting unit squares.

4. Out of the sets generated in Steps 2 and 3 return one with maximum profit.

Each set generated in Steps 2 and 3 permits a feasible packing into the unit cube which is available as a byproduct of Algorithm A. L' is partitioned into at most 9 subsets by Algorithm B, as illustrated in Figure 1.

Theorem 3. *Algorithm B is a* $(9+\epsilon)$*-approximation algorithm for OKP-3 with running time* $\mathcal{O}(\mathrm{T}_{\mathrm{KP}}(n,\epsilon) + n\log^2 n/\log\log n)$*, where* $\mathrm{T}_{\mathrm{KP}}(n,\epsilon)$ *is the running time of the FPTAS used for KP from [17, 19]* [1]*. Furthermore, this bound is tight.*

Proof. Clearly $9+\epsilon$ is an upper bound for the ratio and the running time is dominated by solving the knapsack instance and by Algorithm A since

$$\frac{(1-\delta)}{9} = \frac{1}{9+\epsilon}$$

holds. For the following instance this bound can be attained. We have 10 boxes $R_1 := (1/2, 1/2, 2/15)$, $R_2 := (1, 1/4, 2/15)$, $R_3 := (1, 2/7, 3/4)$, $R_4 := \ldots := R_7 := (1, 2/7, 1/2)$, $R_8 := (1, 2/7, 1/4 + 2/15)$, $R_9 := (1, 2/7, 2/15)$ and $R_{10} := (1,1,1)$. Furthermore, $p_1 := \ldots := p_9 := 1/(9+\epsilon)$ and $p_{10} := 1$. Let $S_1 := \{R_1, \ldots, R_9\}$ and $S_2 := \{R_{10}\}$. It is clear that S_2 is an optimal solution; elementary calculation shows $V(S_1) = 1$ and $P(S_1) = 1 - \delta$, hence S_1 may be selected in Step 1 of Algorithm B. Application of Algorithm B and assuming that the boxes are stacked in increasing order of index in Step 2.1 of Algorithm A yields 9 bins each containing an item with profit $1/(9+\epsilon)$–see Figure 2. □

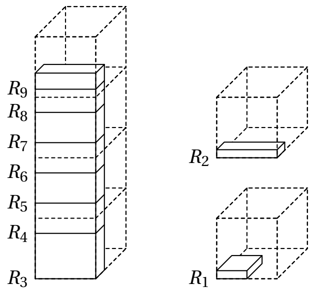

Figure 2: Packing of S_1 with Algorithm B.

Note that only the subset that is returned needs to be packed level-wise using the algorithm from Theorem 1 while the discarded subsets need not be arranged. Algorithm B can be used to solve the special cases where we wish to maximize the number of selected boxes or the volume by setting $p_i := 1$ or $p_i := x_i y_i z_i$ for each $i \in \{1, \dots, n\}$. This also holds for the other algorithms which we present.

In [17, 19] approximation algorithms for various knapsack problems are found. Using these, Algorithm B can be generalized by replacing the KP solver in Step 1, yielding algorithms for *unbounded* OKP-3 and *multiple-choice* OKP-3; see [17, 19] for notions and details. Algorithm B can be modified to yield a ratio of 18 with a much better running time by using a 2-approximation algorithm for classical KP from [17, 19], thus replacing $T_{\mathrm{KP}}(n, \epsilon)$ by $\mathcal{O}(n)$ in Theorem 3.

3 A Refined Construction

In this section we show how to refine Algorithm B to yield an approximation ratio of $(8+\epsilon)$. We identify two possible improvements on Algorithm A. First, by increasing the area guarantee and second, by decreasing the heights Z_1 and Z_2 of the additional strip packing levels.

In Algorithm A and the proof of Theorem 2, the area bound $1/2$ from Theorem 1 was used. We separated boxes with base area greater than $1/4$, resulting in the area guarantee of $1/2 - 1/4 = 1/4$ for each level generated in Steps 2.2 and 2.3 except the last ones. By improving the area guarantee we will improve the height bound of the strip.

So far we arbitrarily chose direction z to be the axis for level generation, but any direction $d \in \{x, y, z\}$ will do. This has the advantage that packing in a direction where all boxes are short, i.e. at most $1/2$, implies that the heights of the additional levels are bound by $1/2$.

Let us introduce the notion of big boxes in certain directions; for any direction $d \in \{x, y, z\}$ a box R_i is called *d-big* $:\Leftrightarrow d_i \in (1/2, 1]$ and we use X, Y and Z to denote the set of boxes that are d-big for the corresponding direction. Any box that is d-big

[1] In [17] the best running time for an FPTAS for the knapsack problem is $\mathcal{O}(n \min\{\log n, \log(1/\epsilon)\} + 1/\epsilon^2 \log(1/\epsilon) \min\{n, 1/\epsilon \log(1/\epsilon)\})$.

for every direction $d \in \{x, y, z\}$ will be called a *big* box. Finally, a box R_i is called *small* $:\Leftrightarrow x_i \in (0, 1/2]$ and $y_i \in (0, 1/2]$ and $z_i \in (0, 1/2]$.

The remainder of the section is organized as follows. First, we give a refined version of Algorithm A that is restricted to packing small boxes. Second, we show how to partition the boxes which are not small into three sets according to the big direction. Third, we give the overall algorithm which is based on the partition into d-big and small boxes.

The following algorithm is applied only on *small* items.

Algorithm C.

1. Find the boxes R_i in L for which the area of $br(R_i)$ is greater than $1/10$ which are $R_1, \ldots, R_m$ w.l.o.g. Sort these in non-increasing order of z_i, resulting in a list L_1. Arrange these in groups of 4 boxes each, except for the last group. Each group can be put on a seperate level by placing the boxes into the corners of the level. Stack these levels on top of one another in direction z.

2. Sort the remaining boxes $R_{m+1}, \ldots, R_n$ in non-increasing order of z_i, resulting in a list L_2.

3. Partition L_2 into consecutive sublists $L''_1, \ldots, L''_v$ where the total base area of each sublist is as close to $1/2$ as possible but not greater. Pack each of these sublists on a level by itself using Theorem 1. Stack all of these levels on top of one another in direction z.

4. Concatenate the packings of L_1 and L_2 to obtain a packing of L.

Note that we formed two groups and obtain an area guarantee of $2/5 = 1/2 - 1/10$ for each layer except the last ones generated in Steps 1 and 3. To avoid confusion we point out that the area guarantee does not hold for the *last* generated layers, while the summands Z_1 and Z_2 in Theorem 2 are the heights of the respective *first* layers. Similar to the proof of Theorem 2, we obtain the following results using the area guarantee of $2/5$.

Theorem 4. *For each list L of n small boxes Algorithm C generates a feasible packing of height at most $5/2V(L) + Z_1 + Z_2$ where $Z_1 \leq 1/2$ and $Z_2 \leq 1/2$ are the heights of the first levels generated in Steps 1 and 3. The construction can be carried out in time $\mathcal{O}(n \log^2 n / \log \log n)$.*

Lemma 5. *Each list L of n small boxes with $V(L) \leq 1$ permits a feasible packing into at most 5 bins. The construction can be carried out in time $\mathcal{O}(n \log^2 n / \log \log n)$; the bound of 5 is tight for the used construction.*

Proof. Use Algorithm C to arrange L in a strip, but separate the first levels generated in Steps 1 and 3. Since L contains only small boxes, these two levels can be packed together into a bin. By Theorem 4, the remaining strip has a height of at most $5/2$. Consider the two cutting unit squares $[0, 1] \times [0, 1] \times \{i\}$ for $i \in \{1, 2\}$. Generate a

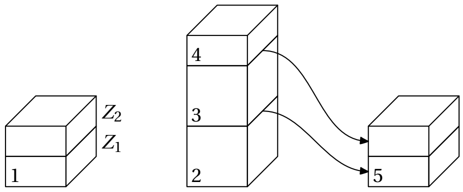

Figure 3: The small boxes can be packed into at most 5 bins.

partition of the region $[0,1] \times [0,1] \times [0,5/2]$ into 5 subsets, namely first 3 subsets which are each positioned in the regions $[0,1] \times [0,1] \times [i-1,i]$ for $i \in \{1,2\}$ as well as the region $[0,1] \times [0,1] \times [2,5/2]$ but not intersecting any of the unit squares, and furthermore 2 subsets of boxes which each intersect with one of the two cutting unit squares. The first three sets can be packed into one bin each. Since L contains only small boxes, the last two sets can be arranged together into one additional bin by aligning them at the top and bottom of the bin, respectively. We have at most 5 bins–see Figure 3. The running time is dominated by Algorithm C and thus bounded by $\mathcal{O}(n \log^2 n/ \log \log n)$.

To show the tightness of the bound let $\gamma := 1/500$ and consider the instance L consisting of $R_1 := \cdots := R_{29} := (1/2, 1/5 + \gamma, 1/3 + \gamma)$, $R_{30} := (\gamma, \gamma, 1/2)$, $R_{31} := \cdots := R_{33} := (1/2, 1/5, \gamma)$ and $R_{34} := (1/2, 1/5 - 2\gamma^2, \gamma)$. Note that $V(L) = 29/30 + 122/15\gamma + 15\gamma^2 - \gamma^3 < 29/30 + 1/60$. Application of Algorithm C packs $R_1, \ldots, R_{29}$ in Step 1, resulting in 8 layers. All remaining boxes are packed into one more layer in Step 3. The height of each layer is greater than 1/3, which means that the layers cannot be arranged in less than 5 bins. □

A partition of the boxes which are not small is given by the following lemma. In the sequel we use the notion of the *projection* of a box; for each box $R = (x, y, z)$ we call the rectangle (y, z) the x-projection, the rectangle (x, z) the y-projection and the rectangle (x, y) the z-projection of R.

Lemma 6. *Let L be a list of n boxes in which no small boxes and at most 3 big boxes occur. Then L can be partitioned into sets X', Y' and Z' in time $\mathcal{O}(n)$, such that each of these contains at most one big box and the x-projections of boxes in X', the y-projections of boxes in Y' and the z-projections of boxes in Z' contain no long or no wide rectangles.*

Proof. Remove the at most 3 big boxes from L and distribute them in X', Y' and Z' such that in each of these sets at most one big box occurs. Set $X' := X' \cup \{R_i \in L | x_i > 1/2, z_i \leq 1/2\}$, $Y' := Y' \cup \{R_i \in L | y_i > 1/2, x_i \leq 1/2\}$ and finally $Z' := Z' \cup \{R_i \in L | z_i > 1/2, y_i \leq 1/2\}$ to obtain the claim. To see that X', Y' and Z' form a partition consider w.l.o.g. a box R that is x-big. If this box is y-big and z-big it is big and therefore included in one of the sets. Otherwise it is either z-big and in Z' or small in direction z and thus in X'. On the other hand no box can be in more than one of the sets. □

We are now ready to give the overall algorithm. To avoid repetition, we enumerate the cases in the analysis only.

Algorithm D.

1. Set $\delta := \epsilon/(8+\epsilon)$. Use a PTAS for non-geometric 4D KP from [17, 19] to select $L' \subseteq L$ such that $P(L') \geq (1-\delta)\mathrm{OPT}$ where OPT denotes the optimum of the integral linear program

$$\text{maximize} \sum_{i=1}^{n} p_i R_i \text{ subject to } R \in P$$

   where R_i is an indicator variable for the box of the same name. Furthermore we define the set P by the constraints

$$\sum_{i=1}^{n} x_i y_i z_i R_i \leq 1, \quad \sum_{R_i \in X} y_i z_i R_i \leq 1, \quad \sum_{R_i \in Y} x_i z_i R_i \leq 1, \quad \sum_{R_i \in Z} x_i y_i R_i \leq 1.$$

   In total, P is a polytope of nonnegative integers.

2. Partition L' into at most 8 subsets which permit a feasible packing as described below. Out of these, return one with maximum profit.

Theorem 7. *Algorithm D is an $(8+\epsilon)$-approximation algorithm for OKP-3 with running time $\mathcal{O}(\mathrm{T}_{4\mathrm{DKP}}(n,\epsilon) + n\log^2 n/\log\log n)$, where $\mathrm{T}_{4\mathrm{DKP}}(n,\epsilon)$ is the running time of the PTAS used for 4D KP from [17, 19] [2]; furthermore this bound is tight.*

Proof. The first constraint of the integral linear program models the volume bound of the box. The other constraints are area bounds for d-big boxes for $d \in \{x,y,z\}$, motivated by the observation that the d- projections of d-big boxes do not overlap. Thus the given program is a relaxation of our problem.

We have not imposed a bound on the number of big boxes in the relaxation, but due to the area conditions there are at most 3 big boxes in the selected set. We consider two cases according to the total projection area of the d-big boxes.

Case 1: There is a direction $d \in \{x,y,z\}$ such that the total d-projection area of all d-big boxes in L' is larger than or equal to $1/2$. In this case all d-big boxes can be packed into at most 3 bins with a construction from [16], which can be done in time $\mathcal{O}(n\log^2 n/\log\log n)$, resulting in a volume of at least $1/4$ being packed. The total volume of the remaining boxes is bounded by $3/4$ and each remaining box has a d-height of at most $1/2$. We apply Algorithm A in direction d which results in a strip of d-height at most 3 and two additional levels of d-height at most $1/2$ each. Similar to the proof of Lemma 5 all these sets can be packed into at most 5 bins–see Figure 3, generating at most 8 bins in total.

[2] In [17] the best running time for a PTAS for the multidimensional knapsack problem is $\mathcal{O}(n^{\lceil d/\epsilon \rceil - d})$.

Case 2: For each $d \in \{x, y, z\}$ the total projection area of all d-big boxes is smaller than $1/2$. By Lemma 6 we partition the set $\{R_i \in L' | R_i \text{ is not small}\}$ into sets X', Y' and Z' such that the total projection area of X', Y' and Z' for the corresponding direction is not greater than $1/2$ and the x-projections of boxes in X', the y-projections of boxes in Y' and the z-projection of boxes in Z' contain no long or no wide rectangles, respectively. Furthermore, each of these sets contains at most one big box. By Theorem 1 the sets X', Y' and Z' can be packed into at most one bin each, resulting in at most 3 bins in total. Let S denote the set of small boxes; these are not yet packed. Clearly $V(S) \leq 1$ holds, so by Lemma 5 the set S can be packed into at most 5 bins, which results in at most 8 bins in total. The runtime bound follows from the fact that we can distinguish between the two cases in time $\mathcal{O}(n)$.

For the tightness of the bound, consider the instance L in which $R_1, \ldots, R_{34}$ are as in the proof of Lemma 5, $R_{35} := (1, 1, 1/180)$, $R_{36} := (1, 1/180, 1)$, $R_{37} := (1/180, 1, 1)$, and $R_{38} := (1, 1, 1)$. The profits are defined by $p_i := 1/[9(8+\epsilon)]$ for $i \in \{1, \ldots, 4, 30, \ldots, 34\}$, $p_i := 1/[8(8+\epsilon)]$ for $i \in \{5, \ldots, 28\}$, $p_i := 1/(8+\epsilon)$ for $i \in \{29, 35, 36, 37\}$ and $p_{38} := 1$. Let $S_1 := L \setminus \{R_{38}\}$ and $S_2 := \{R_{38}\}$. Since $P(S_1) = 8/(8+\epsilon) = (1-\delta) < 1 = P(S_2)$, S_2 is an optimal solution. Elementary calculation verifies that S_1 may be chosen in Step 1 of Algorithm D. Application of Algorithm D leads to Case 2 in the analysis above, where $X' = \{R_{35}\}$, $Y' = \{R_{37}\}$ and $Z' = \{R_{36}\}$. Each of these sets is packed into a separate bin. The remaining boxes are small and are packed into 5 bins as in the proof of Lemma 5. In total, 8 bins are generated; the profits are chosen such that each bin yields a profit of exactly $1/(8+\epsilon)$. □

4 A Packability Criterion

Theorem 1 has a number of applications besides the use in this paper–see [16, 24]. In this section we give a generalization to Theorem 1 for the three-dimensional case.

Lemma 8. *Let L be a list of n boxes such that $V(L) \leq 1/8$ holds and no d_1-big and no d_2-big boxes occur in L for $d_1, d_2 \in \{x, y, z\}$ and $d_1 \neq d_2$. Then L permits a feasible packing into the unit bin which can be generated in time $\mathcal{O}(n \log^2 n / \log\log n)$.*

Observe that similar to Theorem 1 we are allowed to have items that are d-big in a certain direction d but not in any other direction. A difference between the original two-dimensional version and our generalization is that in the original version a single big item–which is neither long nor wide–was allowed which is forbidden here.

Proof. Assume w.l.o.g. that there are no x-big and no z-big boxes in L. Obtain the set L_{scaled} by scaling the given instance by 2 in direction x and z. We use Algorithm A in direction z to pack the boxes in L_{scaled} into one stack S_1 of height at most $4V(L_{\text{scaled}}) \leq 4 \cdot 1/2 = 2$ and two additional stacks S_2 and S_3 of height limited by 1 which correspond to the summands Z_1 and Z_2 in Theorem 4. Rescaling everything yields a stack S_1' of height at most 1 and two additional stacks S_2' and S_3' of heights $1/2$. All stacks have a width of at most $1/2$ in direction x. Thus we can arrange all stacks in a unit cube as shown in Figure 4. □

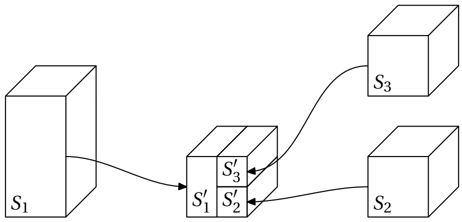

Figure 4: Arrangement of the stacks into a cube.

5 An Improved Strip Packing Algorithm

In Theorem 7 in [21], an approximation algorithm for 3D strip packing with asymptotic approximation ratio 13/4 is presented, more precisely the bound is $13/4\mathrm{OPT} + 8Z$, where Z is the height of the highest item. In this section we show that the additive constant of this bound can be easily improved upon by using Theorem 1; more precisely we obtain the following result by using a more suitable subdivision which results in fewer groups.

Theorem 9. *For each list L of n boxes a packing into the strip $[0,1] \times [0,1] \times [0,\infty)$ of height at most $13/4\mathrm{OPT}(L) + 4Z$ can be generated, where* OPT *denotes the minimum attainable packing height and Z is the height of the highest box. The running time is polynomial in n.*

As we already mentioned in the introduction, the currently best known *asymptotic* approximation ratio for this problem is 1.691 by Bansal et al. [4]. Our result improves the best known *absolute* approximation ratio of 45/4 which follows from [21] to 29/4.

Proof. The construction and the proof are similar as in [21] and included for completeness. We partition L in five groups by letting

$$
\begin{aligned}
L_1 &:= \{R_i | br(R_i) \text{ is big}\}, \\
L_2 &:= \{R_i | br(R_i) \text{ is long and } A(br(R_i)) \leq 1/6\}, \\
L_3 &:= \{R_i | br(R_i) \text{ is long and } A(br(R_i)) > 1/6\}, \\
L_4 &:= \{R_i | br(R_i) \text{ is wide or small, and } A(br(R_i)) \leq 1/6\}, \text{ and} \\
L_5 &:= \{R_i | br(R_i) \text{ is wide or small, and } A(br(R_i)) > 1/6\}
\end{aligned}
$$

and discuss how to obtain good corresponding area guarantees. Clearly, an area guarantee of 1/4 can be obtained for L_1 by simply stacking the boxes on top of one another. Furthermore, the groups $L_2, \ldots, L_5$ can be sorted in non-increasing order of z_i. Now for each of these, we proceed similar as in Step 2.3 of Algorithm A. For L_2 we generate layers using Theorem 1 and obtain an area guarantee of $1/2 - 1/6 = 1/3$ for each layer except the last one. In an even simpler way, the group L_3 can be packed by putting at least two boxes on each layer except the for the last one; we obtain an area guarantee of 1/3 for each but the last layer. By using Theorem 1, we generate layers for the items in

L_4 and again obtain an area guarantee of 1/3 for each but the last layer. Finally, the boxes of L_5 can be packed by placing two of them on each layer except for the last one; again we obtain an area guarantee of 1/3 for each but the last layer.

Now let $H(L_1)$ denote the height of the packing generated for L_1 and, for each $i \in \{2, \ldots, 5\}$, let $H(L_i)$ denote the height of the packing of L_i minus the height of the corresponding *first* layer generated. Let $H(L)$ denote the total height of the packing. For $H := \sum_{i=2}^{5} H(L_i)$ clearly

$$H(L) \leq H(L_1) + H + 4Z \tag{6}$$

holds; furthermore, we have

$$V(L_1) \geq \frac{H(L_1)}{4}.$$

Now let $i \in \{2, \ldots, 5\}$ and let $L'_1, \ldots, L'_j$ denote the levels generated for L_i; finally let $V(L'_k)$ denote the total volume of boxes in L'_k and let h_k denote the height of L'_k for each $k \in \{1, \ldots, j\}$. Since for $L'_1, \ldots, L'_{j-1}$ we have an area guarantee of at least 1/3, we obtain

$$V(L_i) > \sum_{k=1}^{j-1} V(L'_k) \geq \frac{1}{3} \sum_{k=2}^{j} h_k = \frac{H(L_i)}{3}.$$

In total, we obtain

$$\mathrm{OPT}(L) \geq V(L) = \sum_{i=1}^{5} V(L_i) > \frac{H(L_1)}{4} + \frac{H}{3}.$$

Obviously we also have $\mathrm{OPT}(L) \geq H(L_1)$, and thus

$$\mathrm{OPT}(L) \geq \max\left\{H(L_1), \frac{H(L_1)}{4} + \frac{H}{3}\right\} \tag{7}$$

holds. Now we study the ratio

$$r := \frac{H(L_1) + H}{\max\left\{H(L_1), \frac{H(L_1)}{4} + \frac{H}{3}\right\}}. \tag{8}$$

If $H(L_1) \geq H(L_1)/4 + H/3$, which means that $H(L_1) \geq 4/9H$, then we have

$$r = \frac{H(L_1) + H}{H(L_1)} = 1 + \frac{H}{H(L_1)} \leq 1 + \frac{9}{4} = \frac{13}{4}.$$

If $H(L_1) \leq H(L_1)/4 + H/3$, which means that $H(L_1) \leq 4/9H$, then

$$r = \frac{H(L_1) + H}{\frac{H(L_1)}{4} + \frac{H}{3}}$$

holds. In this case, r is a strictly monotonically increasing function of $H(L_1)$, thus the maximum is attained for $H(L_1) = 4/9H$ and 13/4 is the corresponding maximum value. In total, we have $r \leq 13/4$. Rearrangement of the inequalities above yields

$$H(L) \overset{(6)}{\leq} H(L_1) + H + 4Z \overset{(8)}{=} r \max\left\{H(L_1), \frac{H(L_1)}{4} + \frac{H}{3}\right\} + 4Z \overset{(7)}{\leq} \frac{13}{4}\mathrm{OPT}(L) + 4Z$$

which proves the claim. □

Since OPT $\geq Z$ we easily derive the following corollary:

Corollary 10. *The absolute approximation ratio of the given algorithm is at most* $29/4$*.*

6 Enumerations and a Shifting Technique

Algorithms B and D generate cutting areas in the strip, resulting in subsets that have to be repacked. We permit further loss of profit by removing more boxes to discard inconvenient layers; the loss will be suitably bounded. The improvement will be at the cost of a considerably larger running time due to a large enumeration; we thus omit a run time analysis. First we introduce a shifting technique to remove sets intersecting the cutting areas and the additional layers. In the sequel we use the notion of a *gap* which is a rectangular region in the strip that does not intersect the interiors of packed boxes.

Lemma 11. *Let* $L = \{R_1, \dots, R_n\}$ *be a list of boxes with* $z_i \leq \epsilon$ *for each* $R_i \in L$*. Suppose* L *admits a packing into a strip of height at most* h *and let* m *be a positive integer. Then we can create* m *gaps of shape* $[0,1] \times [0,1] \times [0,\epsilon]$ *in a packing of height* h *by deleting boxes such that for the remaining list* $L' \subseteq L$ *the inequality* $P(L') \geq (1 - 2(m+1)\epsilon/h)P(L)$ *holds. The construction can be done in time polynomial in* n*.*

Proof. Consider the original packing in a strip of height exactly h. We partition the strip into regions of height ϵ and eventually one region of smaller height. More precisely we define $p := \lceil h/\epsilon \rceil$ and partition the strip of height h into p regions $S_1, \dots, S_p$ of shape $[0,1] \times [0,1] \times [0,\epsilon]$ where the uppermost region is possibly of smaller height. Then for each $i \in \{1, \dots, p\}$ let $T_i = \{R_j \in L | R_j \cap S_i \neq \emptyset\}$ and let $U_1, \dots, U_{m+1}$ be the $m+1$ sets out of $T_1, \dots, T_p$ which have the smallest profit. It is easy to see that removing these from the packing causes a loss of profit which is at most $2(m+1)/pP(L)$ since each box is included in at most two of the sets $T_1, \dots, T_p$. We remove $m+1$ sets, since the uppermost region can be among the sets with lowest profit, but this region might have height smaller than ϵ; by removing $m+1$ sets we assert that we create at least m gaps of height ϵ. Let L' be the set of remaining boxes; then

$$P(L') \geq P(L) - \frac{2(m+1)}{p}P(L) = (1 - \frac{2(m+1)}{p})P(L) \geq (1 - 2(m+1)\frac{\epsilon}{h})P(L)$$

holds. □

Note that the construction above can be carried out in any direction.

Theorem 12. *Let* $R_1, \dots, R_n$ *be a list of boxes with* $z_i \leq \epsilon$ *for each* $R_i \in L$ *and* $V(L) \leq \alpha \leq 1$*. Then it is possible to select* $L'' \subseteq L$ *such that* $P(L'') \geq (1 - 12\epsilon)P(L)$ *holds and* L'' *admits a feasible packing into at most* $\lceil 4\alpha \rceil$ *bins. The construction can be carried out in time polynomial in* n*.*

Proof. W.l.o.g. $\epsilon \leq 1/12$ holds, since otherwise $L'' = \emptyset$ will do. Use Algorithm A to pack L into a strip which is of height $h \leq 4\alpha$ and two additional layers L_1 and L_2 by

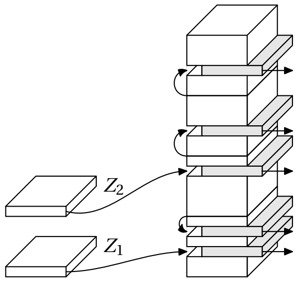

Figure 5: The shifting technique described in Theorem 12.

Theorem 2. If $h \leq 1-2\epsilon$ we can clearly pack L into 1 bin without losing any profit. For $1-2\epsilon < h < 1$ use Lemma 11 to generate 2 gaps in the strip, which causes a loss of profit of at most $6\epsilon/hP(L) \leq 12\epsilon P(L)$, since $\epsilon \leq 1/12$. Moreover, placing L_1 and L_2 into the gaps gives a feasible packing into 1 bin. For the remainder of the proof we thus assume $h \geq 1$. The following construction is illustrated in Figure 5. Use Lemma 11 to generate at most 5 suitable gaps in the strip, resulting in a loss of profit of at most $12\epsilon/hP(L)$; since $h \geq 1$, this loss is bounded by $12\epsilon P(L)$. The remaining set of boxes in the strip and L_1 and L_2 is denoted as L''. Consider the 3 cutting unit squares $[0,1] \times [0,1] \times \{i\}$ for $i \in \{1,2,3\}$ and let L_3, L_4 and L_5 be the sets of boxes in the strip that intersect with these unit squares, respectively. W.l.o.g. none of the sets $L_1, \ldots, L_5$ is empty; otherwise it is removed from consideration. Note that each of the sets $L_1, \ldots, L_5$ can be arranged on a layer of height at most ϵ, so we generate a feasible packing by arranging them into the 5 gaps. In the resulting packing, the 3 cutting unit squares $[0,1] \times [0,1] \times \{i\}$ for $i \in \{1,2,3\}$ do not intersect with any box. Furthermore, all layers $L_1, \ldots, L_5$ are merged in the strip; the packing can be rearranged into $\lceil 4\alpha \rceil$ bins. □

Like before for any $d \in \{x,y,z\}$ we call a box R_i *d-ϵ-big* $:\Leftrightarrow d_i \in (\epsilon, 1]$ and *d-ϵ-small* $:\Leftrightarrow d_i \in (0, \epsilon]$.

We now give the overall algorithm which is based on a separation into boxes that are d-δ-big in all directions and boxes that are d- δ-small in at least one direction. We explain the details in the proof only.

Algorithm E.

1. Set $\delta := \epsilon/[37(7+\epsilon)]$, let $L_1 := \{R_i | R_i$ is d-δ-big for each $d \in \{x,y,z\}\}$ and $L_2 := L \setminus L_1$.

2. For each $L_3 \subseteq L_1$ such that $|L_3| \leq \lfloor 1/\delta^3 \rfloor$ use an exact algorithm to test the packability of L_3. Store a feasible L_3 of maximum total profit.

3. Use an FPTAS for classical KP from [17, 19] to select $L_4 \subseteq L_2$ such that $V(L_4) \leq 1$ and $P(L_4) \geq (1-\delta)\text{OPT}$ holds.

4. Use the construction described below to select $L_5 \subseteq L_4$ which can be packed into at most 6 bins under a small loss of profit.

5. Out of the at most 7 sets generated in Step 2 and Step 4 return one with maximum profit.

Theorem 13. *Algorithm E is a $(7+\epsilon)$-approximation algorithm for OKP-3.*

Proof. Note that $\lfloor 1/\delta^3 \rfloor$ is an upper bound for the number of boxes from L_1 in a feasible solution since δ^3 is a lower bound for the volume of each $R_i \in L_1$. Step 2 can be carried out in time polynomial in δ and thus polynomial in $1/\epsilon$ using an exact optimization algorithm as in [3]. We show that in Step 4 at most 6 sets are generated, resulting in at most 7 bins in total. Partition L_4 into 3 subsets X', Y' and Z' such that in each of these all boxes R_i are d-δ-small for the correspondig direction; note that $V(X') + V(Y') + V(Z') \leq 1$ holds. We apply the construction from Theorem 12 in each of the three directions. Study the following cases, where $V(X') \geq V(Y') \geq V(Z')$ holds w.l.o.g.

Case 1: $V(X') \in (3/4, 1]$. The boxes in X' can be packed into at most 4 bins. We have $V(Y') + V(Z') < 1/4$. This means $V(Y') < 1/4$ and $V(Z') < 1/4$ hold. Consequently Y' and Z' can be packed into at most one bin each, resulting in at most 7 bins.

Case 2: $V(X') \in (1/2, 3/4]$. The boxes in X' can be packed into at most three bins. Furthermore, $V(Y') + V(Z') < 1/2$, which means that $V(Y') < 1/2$ holds. Consequently the boxes in Y' can be packed into at most 2 bins. Furthermore, $V(Z') < 1/4$ holds and finally the boxes in Z' can be packed into at most 1 bin; this generates at most 7 bins in total.

Case 3: We have $V(X') \in [0, 1/2]$. The boxes in X' can be packed into at most two additional bins. Furthermore, $V(Y') \leq 1/2$ and $V(Z') \leq 1/2$ hold. This means that the boxes in Y' and Z' can be packed into at most two bins each. In total at most 7 bins are generated.

To prove the approximation ratio fix an optimal solution S and let P_1^* be the profit of boxes in $S \cap L_1$ and let P_2^* be the profit of boxes in $S \cap L_2$. Consequently $P_1^* + P_2^* = \text{OPT}$ holds. Let P_1 be the profit of the set that is stored in Step 2 and let P_2 be the profit of the set that is selected in Step 3. By construction we have $P_1 \geq P_1^*$ and $P_2 \geq (1-\delta)P_2^*$. Furthermore, by threefold application of the construction from Theorem 12 the loss of profit in P_2 is bounded by $36\delta P_2$; let P_2' denote the remaining list. The profit of the set returned in Step 5 is at least

$$
\begin{aligned}
\frac{1}{7}(P_1 + P_2') &\geq \frac{1}{7}(P_1^* + (1-\delta)(1-36\delta)P_2^*) \\
&\geq \frac{1}{7}(P_1^* + P_2^*)(1-\delta)(1-36\delta) \\
&= \frac{1}{7}\text{OPT}(1-\delta)(1-36\delta) \geq \frac{1}{(7+\epsilon)}\text{OPT}
\end{aligned}
$$

which proves the claimed approximation ratio. □

Theorem 14. *The bound of $(7+\epsilon)$ for the approximation ratio of Algorithm E is asymptotically tight in the sense that it cannot be improved for ϵ arbitrary small.*

Proof. Let $\epsilon \in (0, 1/4]$ and δ be defined as in Step 1 of Algorithm E. Note that $\delta \leq 1/259$. Set $\gamma := \max\{(12i)^{-1} | i \in \mathbb{N}, (12i)^{-1} \leq \delta\}$. Let $\alpha \in \mathbb{R}_+$ such that $(3/4 - \alpha)(1/3 + \alpha) > 1/4$, $(1/3 - 2\alpha)(3/4 + 6\alpha) > 1/4$ and $2\alpha \leq \gamma$ hold. It is easy to see that such an α exists. Note that $\{1/(2\gamma), 1/(3\gamma), 1/(4\gamma), 1/(6\gamma)\} \subseteq \mathbb{N}$. We use the boxes

$$
\begin{array}{lll}
A_i := (\frac{1}{2}, \frac{1}{2} + \alpha, \gamma) & \text{for } i \in \{1, \dots, \frac{1}{\gamma}\}, & A := \{A_1, \dots, A_{\frac{1}{\gamma}}\}, \\
B_i := (\frac{1}{2}, \frac{1}{2} + \alpha, \gamma) & \text{for } i \in \{1, \dots, \frac{1}{4\gamma} - 1\}, & B := \{B_1, \dots, B_{\frac{1}{4\gamma}-1}\}, \\
C_i := (\gamma, \frac{1}{2} + \alpha, \frac{1}{2}) & \text{for } i \in \{1, \dots, \frac{1}{2\gamma}\}, & C := \{C_1, \dots, C_{\frac{1}{2\gamma}}\}, \\
D_i := (\gamma, 1, \frac{1}{4} + \alpha) & \text{for } i \in \{1, \dots, \frac{1}{2\gamma}\}, & D := \{D_1, \dots, D_{\frac{1}{2\gamma}}\}, \\
E_i := (\gamma, \frac{1}{3} + \alpha, \frac{3}{4} - \alpha) & \text{for } i \in \{1, \dots, \frac{1}{3\gamma}\}, & E := \{E_1, \dots, E_{\frac{1}{3\gamma}}\}, \\
F_i := (\frac{1}{3} + \alpha, \gamma, \frac{3}{4} - \alpha) & \text{for } i \in \{1, \dots, \frac{1}{6\gamma} - 1\}, & F := \{F_1, \dots, F_{\frac{1}{6\gamma}-1}\}, \\
G_i := (\frac{1}{3} + \alpha, \gamma, \frac{3}{4} - \alpha) & \text{for } i \in \{1, \dots, \frac{1}{2\gamma} - 1\}, & G := \{G_1, \dots, G_{\frac{1}{2\gamma}-1}\}, \\
H_i := (\frac{1}{3} - 2\alpha, \gamma, \frac{3}{4} + 6\alpha) & \text{for } i \in \{1, \dots, \frac{1}{2\gamma} - 1\}, & H := \{H_1, \dots, H_{\frac{1}{2\gamma}-1}\}, \\
R_1 := (\delta + \alpha, \delta + \alpha, \delta + \alpha). & &
\end{array}
$$

to define the list L.

In the following we will show that L admits a feasible packing into the unit cube. Note that each of the sets (without the box R_1) defined above can be arranged to a rectangular block by stacking them on top of each other in the direction d in which they are d-δ-small–see Figure 6. Furthermore, we note that

$$
\begin{array}{lll}
|A| = \frac{1}{\gamma}, & br_z(A_i) > \frac{1}{4} \text{ for } A_i \in A, & \sum_{A_i \in A} z(A_i) = 1, \\
|B| = \frac{1}{4\gamma} - 1, & br_z(B_i) > \frac{1}{4} \text{ for } B_i \in B, & \sum_{B_i \in B} z(B_i) = \frac{1}{4} - \gamma, \\
|C| = \frac{1}{2\gamma}, & br_x(C_i) > \frac{1}{4} \text{ for } C_i \in C, & \sum_{C_i \in C} x(C_i) = \frac{1}{2}, \\
|D| = \frac{1}{2\gamma}, & br_x(D_i) > \frac{1}{4} \text{ for } D_i \in D, & \sum_{D_i \in D} x(D_i) = \frac{1}{2}, \\
|E| = \frac{1}{3\gamma}, & br_x(E_i) > \frac{1}{4} \text{ for } E_i \in E, & \sum_{E_i \in E} x(E_i) = \frac{1}{3}, \\
|F| = \frac{1}{6\gamma} - 1, & br_y(F_i) > \frac{1}{4} \text{ for } F_i \in F, & \sum_{F_i \in F} y(F_i) = \frac{1}{6} - \gamma, \\
|G| = \frac{1}{2\gamma} - 1, & br_y(G_i) > \frac{1}{4} \text{ for } G_i \in G, & \sum_{G_i \in G} y(G_i) = \frac{1}{2} - \gamma, \\
|H| = \frac{1}{2\gamma} - 1, & br_y(H_i) > \frac{1}{4} \text{ for } H_i \in H, & \sum_{H_i \in H} y(H_i) = \frac{1}{2} - \gamma
\end{array}
$$

holds, where $br_d(R)$ denotes the base rectangle in direction d and $d(R)$ denotes component d for a box R and $d \in \{x, y, z\}$. As indicated above, these sets together with the box R_1 permit a feasible packing since

- F, E, C and F, E, B can be placed next to one another in direction y,
- C, B, D can be placed next to one another in direction z,
- A, H and A, G and B, G and C, G can be placed next to one another in direction y,

which can be verified by elementary calculation–see Figure 7.

In the following we discuss the execution of Algorithm E. The profits necessary for our construction will be described together with the presentation; first we require $P(L) = 1$;

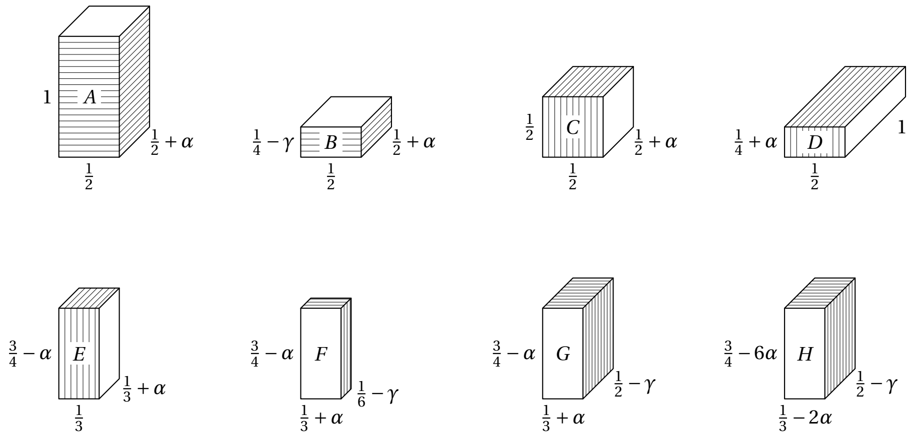

Figure 6: Boxes arranged in blocks.

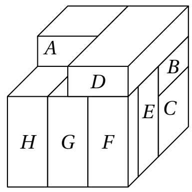

Figure 7: A feasible packing of the boxes into the unit cube.

this also equals OPT since L permits a feasible packing. In Step 1 we obtain $L_1 = \{R_1\}$ and $L_2 = L \setminus \{R_1\}$. Step 2 stores $\{R_1\}$ as a feasible candidate for selection later; here we require $P(R_1) = 1/(7+\epsilon)$. In Step 3 we assume that $L_4 = L_2$ which is possible since L_2 is an optimal solution for the generated knapsack instance; note that here $P(L_4) = 1 - 1/(7+\epsilon) = (6+\epsilon)/(7+\epsilon)$. Now we discuss how L_4 is packed in Step 4. First L_4 is partitioned into

$$X' = C \cup D \cup E, \quad Y' = F \cup G \cup H, \quad Z' = A \cup B.$$

Then each of these three sets is packed into a strip in the corresponding direction; note that

$$br_x(R) > \frac{1}{4} \text{ for } R \in X', \quad br_y(R) > \frac{1}{4} \text{ for } R \in Y', \quad br_z(R) > \frac{1}{4} \text{ for } R \in Z'$$

holds. Hence Algorithm E packs all boxes on top of one another in the corresponding direction; we assume that the boxes in each of the sets are packed in "lexicographical" order. Note that the first layer is separated. We denote the heights of the generated strips for X', Y' and Z' by $h_{X'}$, $h_{Y'}$ and $h_{Z'}$, respectively, and obtain

$$h_{X'} = \frac{4}{3} - \gamma, \quad h_{Y'} = \frac{7}{6} - 4\gamma, \quad h_{Z'} = \frac{5}{4} - 2\gamma.$$

Since $\gamma \le \delta \le 1/259$, all of these strips have height more than 1 and will later result in 2 bins. Let X'_1 denote the set of boxes that are in or intersect the region $[0,1] \times [0,1] \times [0,1]$ in the strip and set $X'_2 := X' \setminus X'_1$. We define Y'_1, Y'_2 and Z'_1, Z'_2 in a similar way. Now we require that

$$P(X'_1) = P(X'_2) = P(Y'_1) = P(Y'_2) = P(Z'_1) = P(Z'_2) = \frac{1}{6}\frac{6+\epsilon}{7+\epsilon}.$$

The profit is evenly partitioned among the items in the sets. Algorithm E uses the shifting technique from Lemma 11 to generate feasible packings for X', Y' and Z', respectively, into 2 bins. Hence three gaps of height δ are generated into each of the strips causing a small loss of profit; the separated layer and the boxes intersecting the cutting square are merged into each strip. It is easy to see that the shifting technique acts only on X'_1, Y'_1, and Z'_1, respectively, since the profit on each items here is lower than in the corresponding second sets. Thus no item is swapped between sets by the shifting technique. We obtain 6 bins each containing subsets of X'_1, X'_2, Y'_1, Y'_2, Z'_1, and Z'_2.

Since each of the sets holds a profit of $(6+\epsilon)/[6(7+\epsilon)]$, each bin holds a profit less than or equal to $(1+\epsilon/6)/(7+\epsilon)$ which finishes the proof. □

7 The Rotational Case

Finally, we discuss the application of Algorithm E on two different rotational scenarios. In both scenarios rotations of the boxes are only permitted by 90° around certain axes. In the first case, which we denote by *z-oriented* OKP-3, rotations are only permitted around the z-axis. This setting is motivated by packing fragile goods and has been considered in the strip packing variant in [9, 10, 27]. In the second case, which we denote by *rotational* OKP-3, rotations are permitted around all three axes. Note that for both scenarios $Q = (1,1,1)$ does not hold without loss of generality, but is an explicit assumption. Surprisingly, a better approximation ratio can be obtained easily although implicitly the search space is dramatically enlarged.

We show how Algorithm E can be revised to yield a better approximation ratio for both scenarios. Step 2 is modified in such a way that the exact packing algorithm takes rotations into account; as before, here at most 1 bin is generated. The most important part is the modification of Step 4. We separate the description of the two scenarios and start with the z-oriented setting. Let Z be the set of boxes $R_i \in L_4$ that are z-δ-small and $X = L_4 \setminus Z$. We introduce a preprocessing step in which each $R_i \in X$ is rotated in such a way that the side length x_i is minimal. Consequently, $x_i \le \delta$ holds for each $R_i \in X$. Hence, the generation of a strip for direction y can be removed; we build only two strips in directions x and z to which the shifting technique from Theorem 12 is applied. This again causes an additional loss of profit which is bounded, however. Let $i \in \{1, \ldots, 4\}$ such that $(i-1)/4 \le V(Z) \le i/4$. Then by Theorem 12, exactly i bins are needed to pack the strip in direction z. Furthermore, we get $V(X) \le V(L_4) - V(Z) \le 1 - (i-1)/4 = (5-i)/4$ which yields a number of $(5-i)$

bins for the corresponding strip. In total a number of $1 + i + (5 - i) = 6$ bins is needed to pack all items. We call the resulting approach Algorithm F and obtain the following result.

Theorem 15. *Algorithm F is a $(6+\epsilon)$-approximation algorithm for z-oriented OKP-3.*

For the *rotational* OKP-3 we have to continue the approach above. Since rotations around all axes are permitted, all items that are d-δ-small for any direction $d \in \{x, y, z\}$ can be packed into one strip in direction z by orienting them such that they are z-δ-small. Since $V(L_4) \leq 1$, application of the construction of Theorem 12 yields at most 4 additional bins. We call the resulting approach Algorithm G and obtain the following result.

Theorem 16. *Algorithm G is a $(5+\epsilon)$-approximation algorithm for rotational OKP-3.*

8 Conclusion

We contributed approximation algorithms for an NP-hard combinatorial optimization problem, where the runtimes of the simpler algorithms are practical. It is of interest whether here an algorithm with ratio $6+\epsilon$ or less exists. We are interested in a reduction of the running time, especially for Algorithm E. In [20] it was proved that it is NP-complete to decide whether a set of squares can be packed into the unit square. However, it is an open problem whether checking the packability of cubes into the unit cube is NP-complete.

Acknowledgements. The authors thank the anonymous referees for valuable comments. Florian Diedrich thanks Denis Naddef for hospitality during the preparation of this article.

References

1. B. S. Baker, D. J. Brown, and H. P. Katseff. A 5/4 algorithm for two-dimensional packing. *Journal of Algorithms*, 2(4):348–368, 1981.

2. N. Bansal, A. Caprara, and M. Sviridenko. Improved approximation algorithms for multidimensional bin packing problems. In *FOCS: Proc. 47th Symposium on Foundations of Computer Science*, pages 697–708, 2006.

3. N. Bansal, J. R. Correa, C. Kenyon, and M. Sviridenko. Bin Packing in Multiple Dimensions: Inapproximability Results and Approximation Schemes. *Mathematics of Operations Research*, 31(1):31–49, 2006.

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

5. A. Caprara. Packing two-dimensional bins in harmony. In *FOCS: Proc. 43rd Symposium on Foundations of Computer Science*, pages 490–499, 2005.

6. M. Chlebík and J. Chlebíková. Inapproximability results for orthogonal rectangle packing problems with rotations. In *CIAC: Proc. 6th Conference on Algorithms and Complexity*, pages 199–210, 2006.

7. F. Diedrich. Approximative Algorithmen für Rucksackprobleme. Diploma thesis, Institut für Informatik und Praktische Mathematik der Christian-Albrechts-Universität zu Kiel, 2004.

8. G. Dósa. The tight bound of first fit decreasing bin-packing algorithm is $\mathrm{FFD}(I) \leq 11/9\mathrm{OPT}(I) + 6/9$. In Bo Chen, Mike Paterson, and Guochuan Zhang, editors, *ESCAPE*, volume 4614 of *Lecture Notes in Computer Science*, pages 1–11. Springer, 2007.

9. L. Epstein and R. van Stee. This side up! In Giuseppe Persiano and Roberto Solis-Oba, editors, *WAOA*, volume 3351 of *Lecture Notes in Computer Science*, pages 48–60. Springer, 2004.

10. L. Epstein and R. van Stee. This side up! *ACM Transactions on Algorithms*, 2(2):228–243, 2006.

11. A. Feldmann, J. Sgall, and S.-H. Teng. Dynamic scheduling on parallel machines. *Theoretical Computer Science (Special Issue on Dynamic and On-line Algorithms)*, 130(1):49–72, 1994.

12. W. Fernandez de la Vega and G. Lueker. Bin packing can be solved within $1 + \epsilon$ in linear time. *Combinatorica*, 1(4):349–355, 1981.

13. Rolf Harren. Approximating the orthogonal knapsack problem for hypercubes. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *ICALP (1)*, volume 4051 of *Lecture Notes in Computer Science*, pages 238–249. Springer, 2006.

14. K. Jansen and R. Solis-Oba. An asymptotic approximation algorithm for 3d-strip packing. In *SODA: Proc. 17th ACM-SIAM Symposium on Discrete Algorithms*, pages 143–152, 2006.

15. K. Jansen and R. Solis-Oba. New approximability results for 2-dimensional packing problems. In Ludek Kucera and Antonín Kucera, editors, *MFCS*, volume 4708 of *Lecture Notes in Computer Science*, pages 103–114. Springer, 2007.

16. K. Jansen and G. Zhang. Maximizing the total profit of rectangles packed into a rectangle. *Algorithmica*, 47(3):323–342, 2007.

17. H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer, 2004.

18. C. Kenyon and E. Rémila. A near-optimal solution to a two dimensional cutting stock problem. *Mathematics of Operations Research*, 25(4):645–656, 2000.

19. E. Lawler. Fast approximation algorithms for knapsack problems. *Mathematics of Operations Research*, 4(4):339–356, 1979.

20. J. Y.-T. Leung, T. W. Tam, C. S. Wong, G. H. Young, and F. Y. L. Chin. Packing squares into a square. *Journal of Parallel and Distributed Computing*, 10(3):271–275, 1990.

21. K. Li and K.-H. Cheng. On three-dimensional packing. *SIAM Journal of Computing*, 19(5):847–867, 1990.

22. K. Li and K.-H. Cheng. Heuristic algorithms for on-line packing in three dimensions. *Journal of Algorithms*, 13(4):589–605, 1992.

23. R. H. Li and M. Y. Yue. The proof of $\mathrm{FFD(L)} \leq (11/9)\mathrm{OPT(L)} + (7/9)$. *Chinese Science Bulletin*, 42:1262–1265, 1997.

24. W. Ludwig and P. Tiwari. Scheduling malleable and nonmalleable parallel tasks. In *SODA: Proc. 5th ACM-SIAM Symposium on Discrete Algorithms*, pages 167–176, 1994.

25. S. Martello and P. Toth. *Knapsack Problems: Algorithms and Computer Implementations*. Wiley, 1990.

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

27. F. K. Miyazawa and Y. Wakabayashi. Approximation algorithms for the orthogonal z-oriented three-dimensional packing problem. *SIAM Journal of Computing*, 29(3):1008–1029, 1999.

28. I. Schiermeyer. Reverse-fit: A 2-optimal algorithm for packing rectangles. In Jan van Leeuwen, editor, *ESA*, volume 855 of *Lecture Notes in Computer Science*, pages 290–299. Springer, 1994.

29. D. D. K. Sleator. A 2.5 times optimal algorithm for packing in two dimensions. *Information Processing Letters*, 10(1):37–40, 1980.

30. A. Steinberg. A strip-packing algorithm with absolute performance bound 2. *SIAM Journal of Computing*, 26(2):401–409, 1997.

31. V. V. Vazirani. *Approximation Algorithms*. Springer, 2001.

32. D. Ye and G. Zhang. Online scheduling of parallel jobs with dependencies on 2-dimensional meshes. In Toshihide Ibaraki, Naoki Katoh, and Hirotaka Ono, editors, *ISAAC*, volume 2906 of *Lecture Notes in Computer Science*, pages 329–338. Springer, 2003.