A New Asymptotic Approximation Algorithm for 3-Dimensional Strip Packing*

Klaus Jansen and Lars Prädel

Universität Kiel, Institut für Informatik, Christian-Albrechts-Platz 4, 24118 Kiel, Germany
{kj,lap}@informatik.uni-kiel.de

Abstract. We study the 3-dimensional Strip Packing problem: Given a list of n boxes $b_1, \ldots, b_n$ of the width $w_i \leq 1$, depth $d_i \leq 1$ and an arbitrary length ℓ_i. The objective is to pack all boxes into a strip of the width and depth 1 and infinite length, so that the packing length is minimized. The boxes may not overlap or be rotated. We present an improvement of the current best asymptotic approximation ratio of 1.692 by Bansal et al. [2] with an asymptotic $3/2 + \varepsilon$-approximation for any $\varepsilon > 0$.

Keywords: Strip Packing, Packing Boxes, Approximation Algorithms.

1 Introduction

We study the 3-dimensional Strip Packing problem: Given a list of n boxes $b_1, \ldots, b_n$ of the width $w_i \leq 1$, depth $d_i \leq 1$ and an arbitrary length ℓ_i. The objective is to pack all boxes into a strip of the width and depth 1 and infinite length, so that the packing length is minimized. The boxes may not overlap or be rotated.

3-dimensional Strip Packing is known to be $\mathcal{NP}$-hard as it is the 2-dimensional counterpart. Thus, unless $\mathcal{P} = \mathcal{NP}$, there will be no polynomial time approximation algorithm that computes a packing with the optimal packing length. Therefore, we study approximation algorithms that have polynomial running time. An asymptotic approximation algorithm A for a minimization problem X with approximation ratio α and additive constant β is a polynomial-time algorithm, that computes for any instance I of the problem X a solution with $A(I) \leq \alpha \cdot \mathrm{OPT}(I) + \beta$, where $\mathrm{OPT}(I)$ is the optimal value of the instance and $A(I)$ is the value of the output. If $\beta = 0$, we call α also absolute approximation ratio. A family of asymptotic approximation algorithms with ratio $1 + \varepsilon$, for any $\varepsilon > 0$ is called an $\mathcal{APTAS}$.

Known results 3-dimensional Strip Packing is a generalization of the 2-dimensional Bin Packing Problem: Given is a list of rectangles $r_1, \ldots, r_n$ of the widths w_i and the heights h_i and an infinite set of 2-dimensional unit-squares, called bins. The objective is to pack all rectangles axis-parallel and non-overlapping into the bins in order to minimize the bins used. Rotations of the rectangles are not allowed. This problem

* Research supported by German Research Foundation (DFG) project JA612/12-2, "Approximation algorithms for two- and three-dimensional packing problems and related scheduling problems".

V. Geffert et al. (Eds.): SOFSEM 2014, LNCS 8327, pp. 327–338, 2014.

is a special case of the 3-dimensional Strip Packing problem, where the lengths of all boxes are 1. Thus, the lower bounds for 2-dimensional Bin Packing hold also for our problem. In the non-asymptotic setting, there is no approximation algorithm strictly better than 2, otherwise the problem Partition can be solved in polynomial time. In the asymptotic setting it is proven that there is no $\mathcal{APTAS}$ for this problem, unless $P = NP$ by Bansal et al. [1]. This lower bound was further improved by Chlebík & Chlebíková [3] to the value $1 + 1/2196$. On the positive side there is an asymptotic 3.25 [9], 2.89 [10], and 2.67 [11] approximation for our problem. More recently an asymptotic 2-approximation was given by Jansen and Solis-Oba [6] that was improved by Bansal et al. [2] to an asymptotic 1.692-approximation.

New results We present a significant improvement of the current best asymptotic approximation ratio:

Theorem 1. *For any $\varepsilon > 0$ and any instance I of the 3-dimensional Strip Packing problem that fits into a strip of length $\mathrm{OPT}_{3D}(I)$, we produce a packing of the length $A(I)$ such that*

$$A(I) \leq (3/2 + \varepsilon) \cdot \mathrm{OPT}_{3D}(I) + \varepsilon + f(\varepsilon, \ell_{\max}),$$

where $\ell_{\max}$ is the length of the largest box in I and $f(\varepsilon, \ell_{\max})$ is a function in ε and $\ell_{\max}$. The running time is polynomial in the input length.

Techniques In our work, we use a new result of the 2-dimensional Bin Packing Problem. In [5,12], there is the following result given:

Theorem 2. *For any $\varepsilon > 0$, there is an approximation algorithm A which produces a packing of a list I of n rectangles in $A(I)$ bins such that*

$$A(I) \leq (3/2 + \varepsilon) \cdot \mathrm{OPT}_{2D}(I) + 69,$$

where $\mathrm{OPT}_{2D}(I)$ is the optimal number of bins. The running time of A is polynomial in n.

There it is proven, that it is possible to round/enlarge some rectangles, so that there are only a constant number of different types and an optimal packing of the enlarged rectangles fit into roughly $(3/2 + \varepsilon) \cdot \mathrm{OPT}_{2D}(I)$ bins. Since there are only a constant number of different types of rectangles, a solution of them can be computed by solving an (Integer) Linear Program. In our work we present a non-trivial method to use these results for the 3-dimensional Strip Packing problem. Therefore, we adopt also some techniques from [6] to transform an instance of the 3-dimensional Strip Packing problem to an instance of the 2-dimensional Bin Packing problem. The main difficulty is to obtain a solution of our problem from the solution of the 2-dimensional Bin Packing problem.

2 2-Dimensional Bin Packing

As mentioned above, we use the results of the work [5,12] of the 2-dimensional Bin Packing problem. Thus, we give here a brief overview over the results obtained in this work.

2.1 Modifying Packings

We assume that we have an optimal solution in OPT_{2D} bins of an arbitrary instance of the 2-dimensional Bin Packing problem given. We denote by $a(X), w(X)$ and $h(X)$ the total area, width and height of a set X of rectangles. In the first step we find a value δ and divide the instance into big, wide, long, small and medium rectangles. We use therefore the following result given in [5,12], where ε' is in dependency of the precision of the algorithm specified later. A formal proof is given in the full version.

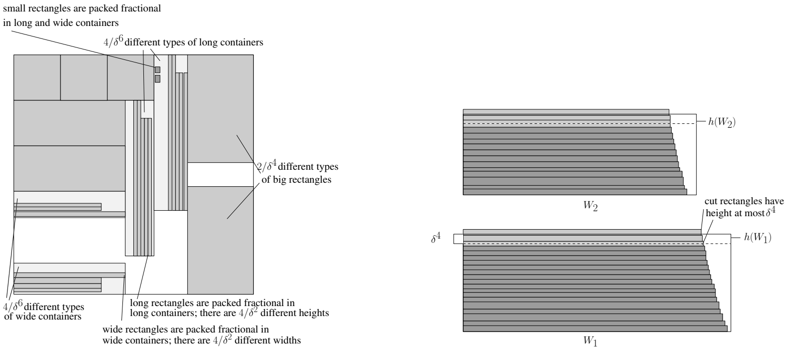
Lemma 1. *We find a value δ, so that ${\varepsilon'}^{2^{2/\varepsilon'}} < \delta \leq \varepsilon'$ and $1/\delta$ is a multiple of 24 holds and all rectangles r_i of the width $w_i \in [\delta^4, \delta)$ or the height $h_i \in [\delta^4, \delta]$ have a total area of at most $\varepsilon' \cdot \mathrm{OPT}_{2D}$.*

The value $1/\delta$ has to be a multiple of 24 for technical reasons, which we will not discuss further. A rectangle is *big* when the width $w_i \geq \delta$ and height $h_i \geq \delta$ holds, it is *wide* when the width $w_i \geq \delta$ and the height $h_i < \delta^4$ holds, when the width $w_i < \delta^4$ and the height $h_i \geq \delta$ holds it is *long*, when the width $w_i < \delta^4$ and height $h_i < \delta^4$ holds it is *small.* If none of these conditions holds, i.e. at least one side is within $[\delta^4, \delta)$ it is a *medium* rectangle. These medium rectangles are packed separately.

Our optimal solution can be transformed so that the widths and the heights of the big rectangles are rounded up to at most $2/\delta^4$ different types of rectangles (cf. Figure 1(a)). We denote the types by $B_1, \ldots, B_{2/\delta^4}$. The wide rectangles are cut in the height. The widths of the resulting slices of the wide rectangles are rounded up to at most $4/\delta^2$ values. The set of the slices of the wide rectangles of the different widths are denoted by $W_1, \ldots, W_{4/\delta^2}$. Vice versa, the long rectangles are cut in the width and the heights of the resulting slices are rounded up to at most $4/\delta^2$ different values. The sets of these slices of different heights are denoted by $L_1, \ldots, L_{4/\delta^2}$. The slices are packed into wide and long containers. There are at most $6/\delta^3$ wide and long containers in each bin. The wide containers have at most $4/\delta^2$ different widths and the containers of one certain width have at most $1/\delta^4$ different heights. Thus, there are at most $4/\delta^6$ different types. The same holds vice versa for the long containers. We denote the sets of wide containers of the different widths by $CW_1, \ldots, CW_{4/\delta^2}$, each set CW_i is separated in sets $CW_{i,1}, \ldots, CW_{i,1/\delta^4}$ of containers of different heights. The sets for the long containers of the different heights are denoted by $CL_1, \ldots, CL_{4/\delta^2}$ and each set CL_i is also separated into sets $CL_{i,1}, \ldots, CL_{i,1/\delta^4}$ of different widths. The small rectangles are cut in the width and height and are packed into the wide and long containers. The total number of bins without the medium rectangles is increased with these steps to at most $(3/2 + 22\delta)\mathrm{OPT}_{2D} + 53$ bins.

2.2 2-Dimensional Bin Packing Algorithm

After showing the above described modification steps of any optimal solution, it is possible to compute a solution with only a small increased number of bins. Therefore, all necessary values are guessed via an enumeration step. It is possible to find the value OPT_{2D}, the widths and the heights of the different types of big rectangles and long and wide containers, the different widths of the wide rectangles and the different heights



(a) Packing in one bin

(b) Greedy assignment of wide rectangles. Grey rectangles are packed separately.

Fig. 1. Structure of a packing in one bin and assignment of wide rectangles

of the long rectangles in polynomial time in the input length. Note that it is not beforehand clear to which values a rectangle is rounded, i.e. to which set it belongs. We only know that it is possible to enlarge/round it to one of the types. Thus, we have to assign each big, wide and long rectangle to one of the sets. The cardinality of the sets of big rectangles and containers are also guessed via an enumeration step. Whereas we guess approximately with an error of δ^4 the total heights of each set of wide rectangles $h(W_i)$ and the total width of each set of long rectangles $w(L_i)$. Since there are only fractions of these rectangles in these sets, it is not possible to enumerate the cardinality of them. In the following we assume that we are in the iteration of the right guess of all above described values. The big rectangles are assigned via a network flow algorithm to the sets. The wide rectangles are sorted by their widths and greedily assigned to one of these groups, beginning with the widest group, so that the total heights $h(W_i)$ of each set is strictly exceeded (cf. Figure 1(b)). With the right approximate guess of the total heights of each group, we can ensure that all rectangles are assigned to one group. While removing the rectangles of the total height at most $3\delta^4$, we secure that the total height of the wide rectangles is below $h(W_i)$ and thus they fit fractionally into the containers. We denote the set of rectangles that are rounded to the ith width by $\overline{W_i}$. We need 1 additional bin for the removed rectangles of all sets. We do the analogous steps for assigning the long rectangles to the groups. At this moment, the big, long and wide rectangles are assigned to one group and are rounded. The wide rectangles are packed into the wide containers via a linear program, that is similar to the linear program in [7]. We pack the wide rectangles fractionally into the containers of a certain width. Here $\mathfrak{C}_j^{(\ell)}$ represents a configuration of wide rectangles that fit into a wide container of the set CW_ℓ. The configurations are all possible multi-sets of wide rectangles that have a total width of at most the width of the container. There is only a bounded number q of possible configurations. $a(i, \mathfrak{C}_j^{(\ell)})$ represents the number of wide rectangles in the set W_i that are in configuration $\mathfrak{C}_j^{(\ell)}$. The variable $x_j^{(\ell)}$ gives the total height of one configuration in the container. $LP(1)$:

$$\sum_{j=1}^{q^{(\ell)}} x_j^{(\ell)} = h(CW_\ell) \qquad \ell \in \{1, \ldots, 4/\delta^2\}$$

$$\sum_{\ell=1}^{t} \sum_{j=1}^{q^{(\ell)}} a(i, \mathfrak{C}_j^{(\ell)}) \cdot x_j^{(\ell)} \geq h(\overline{W_i}) \qquad i \in \{1, \ldots, 4/\delta^2\}$$

$$x_j^{(\ell)} \geq 0 \qquad j \in \{1, \ldots, q^{(\ell)}\}, \ell \in \{1, \ldots, 4/\delta^2\}$$

The first line secures, that the total height of the containers of a certain width is not exceeded by the configurations, while the second line secures, that there is enough area to occupy all wide rectangles. Since we have at most $2 \cdot 4/\delta^2$ conditions, a basic solution has also at most $8/\delta^2$ non-zero variables, i.e. there are only $8/\delta^2$ different configurations in the solution. This and the fact that there is a bounded number of containers allows us to generate a non-fractional packing of the wide rectangles into the containers. The same is done with the long rectangles that are packed in the long containers. The small rectangles are packed with Next-Fit-Decreasing-Height by Coffman et al. [4] in the remaining gaps, by losing only a small amount of additional bins.

Finally, the big rectangles and wide and long containers are packed with an Integer Linear Program $ILP(1)$:

$$\min \sum_{k=1}^{q} x_k$$

$$s.t. \sum_{k=1}^{q} b(i, C_k) \cdot x_k \geq n_i^b \qquad i \in \{1, \ldots, 2/\delta^4\}$$

$$\sum_{k=1}^{q} w(i, j, C_k) \cdot x_k \geq n_{i,j}^w \qquad i \in \{1, \ldots, 4/\delta^2\}, j \in \{1, \ldots, 4/\delta^6\}$$

$$\sum_{k=1}^{q} \ell(i, j, C_k) \cdot x_k \geq n_{i,j}^\ell \qquad i \in \{1, \ldots, 4/\delta^2\}, j \in \{1, \ldots, 4/\delta^6\}$$

$$x_k \in \mathbb{N} \qquad k \in \{1, \ldots, q\}$$

Therefore, we build also configurations C_k of rectangles that fit into one sole bin. Since we have a constant number of rectangles/containers in one bin and only a constant number of different types, the number of configurations q is also only a constant. n_i^b represents the total number of big rectangles in the set B_i, and $b(i, C_k)$ gives the number of big rectangles in the set B_i that are in configuration C_k. The analogous values for the containers are represented by $n_{i,j}^w$ and $w(i, j, C_k)$ for the wide containers of type $CW_{i,j}$ and $n_{i,j}^\ell$ and $\ell(i, j, C_k)$ for the long containers of the type $CL_{i,j}$. This Integer Linear Program computes a value of at most $(3/2 + 22\delta)\mathrm{OPT}_{2D} + 53$ bins.

3 3-Dimensional Strip Packing

After giving the overview of the 2-dimensional Bin Packing algorithm we start with the presentation of our 3-dimensional Strip Packing algorithm. We start also with an optimal solution of an arbitrary given instance and show how to modify this. Some of the techniques used here are also used in [6]. Afterwards, we present our algorithm. We denote by $vol(X)$ the total volume of a set X of boxes. Furthermore, we call the rectangle of the width w_i and height d_i of a box b_i by the base of b_i.

3.1 Modifying Packings

We first modify an optimal solution of an instance for our problem that fits into a strip of the length OPT_{3D}. We scale the lengths of the whole instance by the value OPT_{3D}, the total length of the optimal packing is thus 1. Afterwards, we extend the lengths of the boxes to the next multiple of ε'/n for a given $\varepsilon' > 0$. This enlarges the strip by at most ε', since each box is enlarged by at most ε'/n and there are at most n boxes on top of each other. The length of the strip is $1 + \varepsilon'$. Furthermore, this allows us to place the boxes on z-coordinates that are multiples of ε'/n. A formal proof of this fact is already given in [6]. In the next step, we cut the strip horizontally on each z-coordinate that is a multiple of ε'/n. Each slice of length ε'/n of the packing is treated as one 2-dimensional bin. Note that each box intersects a slice of the solution completely, or not at all. Each slice of a box b_i is a copy of its base, i.e. a rectangle of the width w_i and height d_i. It follows that we obtain from the optimal solution of the 3-dimensional Strip Packing problem a solution of a 2-dimensional Bin Packing instance in $(1 + \varepsilon') \cdot n/\varepsilon' = n/\varepsilon' + n$ bins. We denote by $\mathrm{OPT}_{2D} \leq n/\varepsilon' + n$ the minimal number of bins used in an optimal packing of this 2-dimensional Bin Packing instance.

We use the modification steps of the 2-dimensional Bin Packing as described above. The medium rectangles/boxes are discarded. Thus we have a packing into one strip of the total length:

$$\begin{aligned}((3/2 + 22\delta)\mathrm{OPT}_{2D} + 53) \cdot \varepsilon'/n &\leq (3/2 + 22\delta) \cdot (n/\varepsilon' + n) + 53) \cdot \varepsilon'/n \\ &\leq (3/2 + 22\delta) \cdot (1 + \varepsilon') + 53\varepsilon'/n \\ &\leq 3/2 + 22\delta + 3/2\varepsilon' + 22\delta\varepsilon' + 53\varepsilon'\end{aligned}$$

After rescaling the lengths of the boxes by OPT_{3D}, we obtain a packing length of $(3/2 + 22\delta + 3/2\varepsilon' + 22\delta\varepsilon' + 53\varepsilon')\mathrm{OPT}_{3D}$.

3.2 Algorithm

In the first step we set ε' as the largest value so that $1/\varepsilon'$ is a multiple of 24 and $\varepsilon' \leq \min\{\varepsilon/236, 1/48\}$ holds. Thus, $\varepsilon' \geq \varepsilon/260$.

For dual approximation we approximately guess the optimal length $L_{\mathrm{OPT}_{3D}}$ so that $\mathrm{OPT}_{3D} \leq L_{\mathrm{OPT}_{3D}} < \mathrm{OPT}_{3D} + \varepsilon'$ holds. To do this we use a naïve approach. It holds $\mathrm{OPT}_{3D} \in [\ell_{\max}, n \cdot \ell_{\max}]$. Thus, we test less than $n \cdot \ell_{\max}/\varepsilon'$ values with binary search. This takes time at most $\mathcal{O}(\log(n \cdot \ell_{\max}/\varepsilon'))$ and is thus polynomial in the encoding length of the input. In the following we assume that we are in the iteration where we

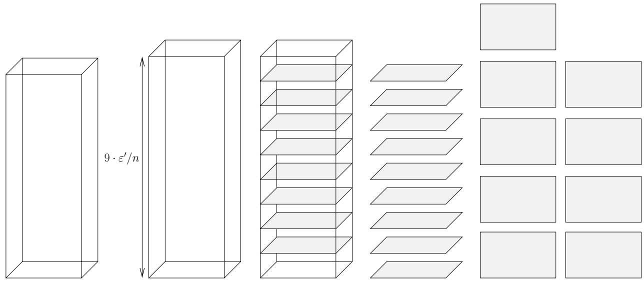


Fig. 2. Transform a box into rectangles

found the correct value $L_{\mathrm{OPT}_{3D}}$. We scale the lengths of the boxes in the input by the value $L_{\mathrm{OPT}_{3D}}$. An optimal packing fits now into a strip of length 1.

Our algorithm rounds afterwards the lengths of the boxes to the next multiple of ε'/n and we cut each box at each multiple of ε'/n. Each slice of a box b_i is treated as a 2-dimensional rectangle of the width w_i and the height d_i. There are now at most $n \cdot n/\varepsilon' = n^2/\varepsilon'$ rectangles that fit in an optimal packing into at most $\mathrm{OPT}_{2D} \leq (1+\varepsilon') \cdot n/\varepsilon'$ bins.

Gap-Creation and Medium Boxes. We find a value δ with Lemma 1, and partition the instance into big, wide, long, small and medium rectangles. We also guess all necessary values that are needed to run the 2-dimensional Bin Packing algorithm. We assume that we are in the iteration, where all values are guessed correctly (cf. also Algorithm 1).

The medium rectangles are divided into two sets $M_{w\delta}$ and $M_{h\delta}$ of rectangles of the width within $[\delta^4, \delta)$ and the remaining rectangles of the height within $[\delta^4, \delta)$. The total area $a(M_{w\delta} \cup M_{h\delta})$ is bounded by $\varepsilon'\mathrm{OPT}_{2D} \leq \varepsilon'(n/\varepsilon' + n) = n + \varepsilon' n$. Thus, the total volume of the corresponding 3-dimensional (medium, scaled) boxes is bounded by $\varepsilon'/n \cdot (n+\varepsilon' n) = \varepsilon'(1+\varepsilon')$. After rescaling by $L_{\mathrm{OPT}_{3D}}$, the total volume is increased to $vol(M_{w\delta} \cup M_{h\delta}) \leq L_{\mathrm{OPT}_{3D}}\varepsilon'(1+\varepsilon') \leq (\mathrm{OPT}_{3D}+\varepsilon')\varepsilon'(1+\varepsilon') \leq 2\varepsilon'(\mathrm{OPT}_{3D}+\varepsilon')$. We pack the medium boxes into a strip S_0 with the following Lemma. Furthermore, we assign the wide and long rectangles non-fractional to the groups $W_1, \ldots, W_{4/\delta^2}$ and $L_1, \ldots, L_{4/\delta^2}$, so that all slices of one box belong to one group. This is done similarly as assigning the wide and long rectangles in the 2-dimensional Bin Packing algorithm (cf. Figure 1(b)). Therefore, we have to pack some wide and long boxes that cannot be assigned into S_0. The proof of the following Lemma is given in the full version.

Lemma 2. *We need a strip S_0 of the total length $6\varepsilon'\mathrm{OPT}_{3D} + \varepsilon' + 6\ell_{\max}$ to*

1. *pack the medium boxes and*
2. *assign the wide and long rectangles into the groups $W_1, \ldots, W_{4/\delta^2}$ and $L_1, \ldots, L_{4/\delta^2}$ so that all slices of one box belong to one group.*

Packing the Containers and Big-Slices In the end of the 2-dimensional Bin Packing algorithm, an Integer Linear Programs ($ILP(1)$) is solved to pack the 2-dimensional containers and the slices of the big boxes into the bins. In our case it is an advantage to use the relaxation of the Integer Linear Program, since the basic solution consists of at most $m \leq 2/\delta^4 + 4/\delta^6 + 4/\delta^6 \leq 9/\delta^6$ configurations. W.l.o.g. we denote these non-zero configurations by $C_1, \ldots, C_m$. We treat each 2-dimensional object in the non-zero configurations as 3-dimensional object of length ε'/n and pack the objects of each configuration C_k on top of each other. Thus, we obtain at most m 3-dimensional strips. The length of the strip S_k, for $k \in \{1, \ldots, m\}$ is the value of the configuration $\overline{x}_k$ multiplied with ε'/n. The total length of these strips is at most $\varepsilon'/n \cdot ((3/2 + 24\delta) \cdot \mathrm{OPT}_{2D} + 53) \leq \varepsilon'/n \cdot ((3/2+24\delta) \cdot (1+\varepsilon') \cdot n/\varepsilon' + 53) \leq (3/2+24\delta) \cdot (1+\varepsilon') + 53\varepsilon'$.

After rescaling the boxes by the length $L_{\mathrm{OPT}_{3D}}$ we obtain the following total packing length:

$$\begin{aligned}
L &\leq L_{\mathrm{OPT}_{3D}}((3/2 + 24\delta) \cdot (1 + \varepsilon') + 53\varepsilon') \\
&\leq (\mathrm{OPT}_{3D} + \varepsilon')((3/2 + 24\delta) \cdot (1 + \varepsilon') + 53\varepsilon') \\
&\leq (3/2 + 24\delta) \cdot (\mathrm{OPT}_{3D} + \varepsilon' + \varepsilon'\mathrm{OPT}_{3D} + \varepsilon'^2) + 53\varepsilon'\mathrm{OPT}_{3D} + 53\varepsilon'^2 \\
&= (3/2 + 24\delta + 3/2\varepsilon' + 24\delta\varepsilon' + 53\varepsilon')\mathrm{OPT}_{3D} + (3/2 + 24\delta) \cdot (\varepsilon' + \varepsilon'^2) + 53\varepsilon'^2 \\
&\leq (3/2 + 80\varepsilon')\mathrm{OPT}_{3D} + 6\varepsilon',
\end{aligned}$$

since $\delta \leq \varepsilon' \leq 1/48$. It is left to pack the big boxes into the strip at the places of their placeholders and to pack the wide, long and small boxes into the containers. Remember that we assume that we have guessed all values correctly, so there is a fractional packing of the big boxes into the strips. We show in the full version how to use a result by Lenstra et al. [8] for scheduling jobs on unrelated machines to pack the big boxes into the strips.

3.3 3-Dimensional Containers

We describe in this section how to pack the wide and long boxes into the corresponding containers. We will focus on the wide boxes, since the steps for the long boxes are analogous.

Packing the Wide and Long Boxes into the Containers We remain in the 2-dimensional representation to pack the slices into the 2-dimensional containers. We use the linear program $LP(1)$ to select at most $8/\delta^2$ configurations in the containers so that all wide slices are fractionally covered.

Afterwards, we transform the slices of the containers to 3-dimensional objects by adding lengths of the value $\varepsilon'/n \cdot L_{\mathrm{OPT}_{3D}}$. We keep the configurations of the linear program, that forms slots in the 3-dimensional containers (cf. Figure 3). Each 3-dimensional container is divided into slots and possibly an empty space on the right side for the small boxes. If a 3-dimensional container consists of different configurations, we split the entire strip at this length. This increases the number of strips by less than $8/\delta^2$. By doing the analogous steps for the long slices, the number of strips grows to at most $\overline{m} \leq m + 16/\delta^2 = 9/\delta^6 + 16/\delta^2$ strips. At this moment there is only one configuration in each strip and each container. Furthermore, all wide boxes fit fractionally (cut in the depth and length) into the 3-dimensional slots inside the wide containers. We increase all strips by the length $\ell_{\max}$. The total length of all $\overline{m}$ strips is

$$\overline{L} \leq L + \overline{m} \cdot \ell_{\max} \leq (3/2 + 80\varepsilon')\mathrm{OPT}_{3D} + 6\varepsilon' + (9/\delta^6 + 16/\delta^2) \cdot \ell_{\max}.$$

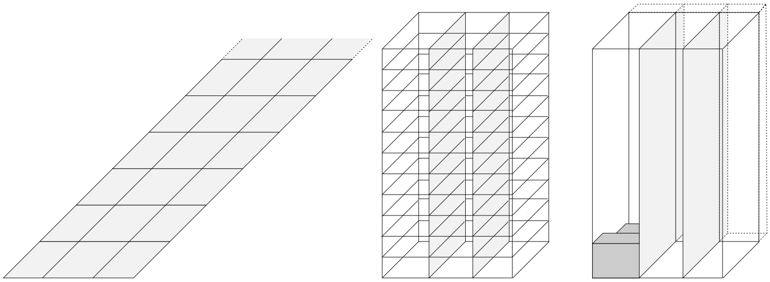


Fig. 3. Configurations of the linear program build slots in the wide container

We pack the wide boxes into the 3-dimensional wide containers, therefore we focus one 3-dimensional wide container C of the width w_C, depth d_C and length ℓ_C. We increased the length by $\ell_{\max}$, thus the length is $\ell_C + \ell_{\max}$. Since all wide boxes fit fractionally into the slots inside the wide containers (when all values are guessed correctly) it holds that the total volume of the wide boxes is at most the total volume of these slots. The same holds with the long boxes and the slots in the long containers. All small boxes fit also fractionally into the wide and long containers in the remaining gaps. Thus, if we find a packing of the boxes that occupies the total volume of the containers, then we know that all boxes are packed. We prove that we either occupy the total volume of one container, or we are running out of boxes. Therefore, we have to extend the side-lengths of each container. We already increased the length of each container by $\ell_{\max}$. Now we extend also the depth by δ^4 (cf. Figure 4) and the width by δ^4. The side-lengths of C is now $w_C + \delta^4$, $d_C + \delta^4$ and $\ell_C + \ell_{\max}$.

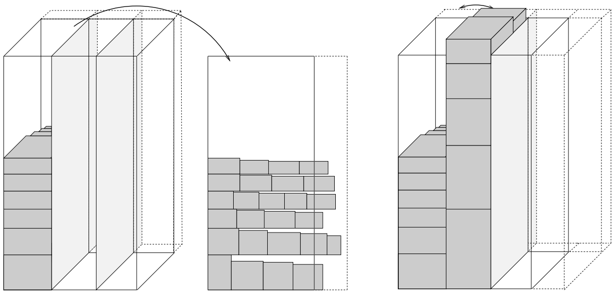


Fig. 4. Next-Fit heuristic

The left side of the container is parted by the slots of the linear program $LP(1)$. We focus one of these slots S of the width w_S, depth $d_S = d_C + \delta^4$ and length $\ell_C + \ell_{\max}$. In this slot we pack only boxes b_i of the (rounded) width $w_i = w_S$. We sort the boxes by their lengths and pack them with a Next-Fit heuristic into the slots. When the next box does not fit into the slot we form a new level on top of the first box and continue to pack the boxes until the length of the container is exceeded. Each box has a depth of at most δ^4, thus we exceed the depth d_C in each level (cf. Figure 4). It follows that the area covered by the bases of the boxes is at least $a := w_S \cdot d_C$ in each level. With the following Lemma 3 we cover a total volume of $w_S \cdot d_C \cdot (\ell_C + \ell_{\max} - \ell_{\max}) = w_S \cdot d_C \cdot \ell_C$.

Lemma 3. *Given a target region X of the width w_X, depth d_X and length ℓ_X and k levels $\mathcal{L}_1, \ldots, \mathcal{L}_k$, where the bases of the boxes in each level covers a total area of at least a. Let w.l.o.g. b_i be the largest box in level $\mathcal{L}_i$ and $b_{i'}$ be the smallest box. If $\ell_{i'} \geq \ell_{i+1}$ for all $i \in \{1, \ldots, k-1\}$, then we are able to pack boxes into X with a total volume of at least $a \cdot (\ell_X - \ell_{\max})$.*

Proof. We pack the levels on top of each other until the next level does not fit into the target region. Let $\ell_{k+1} := \ell_X - \sum_{i=1}^{k} \ell_i$ be the length on top of the uppermost target region. It holds $\ell_{i'} \geq \ell_{i+1}$ for all $i \in \{1, \ldots, k\}$. Furthermore, we have $vol(\mathcal{L}_i) \geq a \cdot \ell_{i'}$. Thus, $\sum_{i=1}^{k} vol(\mathcal{L}_i) \geq \sum_{i=1}^{k} a \cdot \ell_{i'} \geq \sum_{i=1}^{k} a \cdot \ell_{i+1} = a \cdot (\sum_{i=2}^{k+1} \ell_i) \geq a \cdot (\ell_X - \ell_{\max})$ □

We do this with all slots in all configurations. Since the total volume of the boxes of one width w is at most the total volume of all slots of the same width w, we are able to pack all boxes. It happens for each slot of one specified width exactly once that the boxes running out and that there is some free space in the slot that we have to use for the small boxes. In this case, we change the order of the slots and exchange this slot with the rightmost slot. If there are several slots in one strip where this happens, then we sort the slots by non-increasing packing lengths (cf. Figure 4). Each time when this happens, we split the entire strip into two strips. The length of the lower part is extended by the length $\ell_{\max}$, so that the cut boxes still fit in the lower strip. There are $2\delta^2$ different widths of wide and $2/\delta^2$ different depths of long boxes. Therefore, the number of strips increases to $\overline{\overline{m}} \leq \overline{m} + 4\delta^2 \leq 9/\delta^6 + 20\delta^2$. The total length of all strips is increased to

$$\overline{\overline{L}} \leq \overline{L} + 4/\delta^2 \cdot \ell_{\max} \leq (3/2 + 80\varepsilon')\mathrm{OPT}_{3D} + 6\varepsilon' + (9/\delta^6 + 20/\delta^2) \cdot \ell_{\max}.$$

The advantage is, that we have containers with some slots and some cubic free space at the right side. In this free space we pack the small boxes with the 2-dimensional Next-Fit-Decreasing-Height algorithm [4]. We describe in the full version how to pack the small boxes into the remaining free space and how to remove the extensions of the boxes with the following Lemma:

Lemma 4. *We are able to remove the extensions of the containers and to pack the intersecting boxes into a strip $S_{\overline{\overline{m}}+1}$ of the length $48\delta\overline{\overline{L}} \leq 152\varepsilon'\mathrm{OPT}_{3D} + 6\varepsilon' + (9/\delta^6 + 20/\delta^2) \cdot \ell_{\max}$.*

3.4 Summary

To summarize our results, we state the entire algorithm in Algorithm 1.

Algorithm 1. Algorithm for 3-dimensional Strip Packing

Set $\varepsilon' := \min\{\varepsilon/236, 1/48\}$, so that $1/\varepsilon'$ is a multiple of 24
Find $L_{\mathrm{OPT}_{3D}}$, so that $\mathrm{OPT}_{3D} \leq L_{\mathrm{OPT}_{3D}} < \mathrm{OPT}_{3D} + \varepsilon'$ holds with binary search **for each guess do**
  Scale length of boxes by $1/L_{\mathrm{OPT}_{3D}}$ and round them to next multiple of ε'/n
  Split boxes at multiple of ε'/n and obtain instance for 2-dim. Bin Packing
  \\begin 2-dim. Bin Packing algorithm
  Find OPT_{2D} **for each guess do**
    Compute δ and partition the rectangles
    Find widths and heights and number of $2/\delta^4$ different types of big rectangles;
    widths and approx. total height of $4/\delta^2$ different types of wide rectangles;
    heights and approx. total width of $4/\delta^2$ different types of long rectangles;
    widths and heights and number of $4/\delta^6$ different types of long and wide containers **for each guess do**
      Greedy assignment of wide and long rectangles to different types
      Solve $LP(1)$ to find fractional packing of wide rectangles in wide containers and long rectangles in long containers
      Solve relaxation of $ILP(1)$ to find fractional packing of big rectangles and wide and long containers
      \\end 2-dim. Bin Packing algorithm
      Add lengths of value $\varepsilon' L_{\mathrm{OPT}_{3D}}/n$ to 2-dim. bins and build strip for each configuration in the basic solution of relaxation of $ILP(1)$
      Extend each strip by $\ell_{\max}$
      Pack big boxes into strip with result by Lenstra et al. [8]
      Pack wide, long and small boxes into extended 3-dimensional containers and remove the extensions
      Pack medium boxes with use of Steinbergs algorithm [13]

We have packed the boxes into the strips $S_0, \ldots, S_{\overline{\overline{m}}+1}$. We simply stack them on top of each other and obtain one strip of the total length:

$$\begin{aligned}
&4\varepsilon'\mathrm{OPT}_{3D} + \varepsilon' + 6\ell_{\max} + \overline{\overline{L}} + 48\delta\overline{\overline{L}} \\
&\leq (3/2 + 84\varepsilon')\mathrm{OPT}_{3D} + 7\varepsilon' + (6/\delta^6 + 20/\delta^2 + 6)\cdot \ell_{\max} + 48\delta\overline{\overline{L}} \\
&\leq (3/2 + 236\varepsilon')\mathrm{OPT}_{3D} + 13\varepsilon' + (15/\delta^6 + 40/\delta^2 + 6)\cdot \ell_{\max} \\
&\leq (3/2 + 236\varepsilon')\mathrm{OPT}_{3D} + 13\varepsilon' + (16/\delta^6)\cdot \ell_{\max} \\
&\leq (3/2 + \varepsilon)\mathrm{OPT}_{3D} + \varepsilon + 16/\varepsilon'^{12/\varepsilon'}\ell_{\max} \\
&\leq (3/2 + \varepsilon)\mathrm{OPT}_{3D} + \varepsilon + 4160/\varepsilon^{3120/\varepsilon}\ell_{\max}.
\end{aligned}$$

By stacking the strips on top of each other it follows Theorem 1 with $f(\varepsilon, \ell_{\max}) = 4160/\varepsilon^{3120/\varepsilon}\ell_{\max}$.

4 Conclusion

We presented an asymptotic $3/2+\varepsilon$-approximation for the 3-dimensional Strip Packing problem. This is a significant improvement over the previous best known asymptotic approximation ratio of 1.692 by Bansal et al. [2]. It is of interest, if it is possible to improve our new upper bound or to show that the lower bound of $1+1/2196$ by Chlebík & Chlebíková [3] can be lifted.

References

1. Bansal, N., Correa, J.R., Kenyon, C., Sviridenko, M.: Bin packing in multiple dimensions: Inapproximability results and approximation schemes. Mathematics of Operations Research 31(1), 31–49 (2006)
2. Bansal, N., Han, X., Iwama, K., Sviridenko, M., Zhang, G.: A harmonic algorithm for the 3d strip packing problem. SIAM Journal on Computing 42(2), 579–592 (2013)
3. Chlebík, M., Chlebíková, J.: Inapproximability results for orthogonal rectangle packing problems with rotations. In: Calamoneri, T., Finocchi, I., Italiano, G.F. (eds.) CIAC 2006. LNCS, vol. 3998, pp. 199–210. Springer, Heidelberg (2006)
4. Coffman Jr., E.G., Garey, M.R., Johnson, D.S., Tarjan, R.E.: Performance bounds for level-oriented two-dimensional packing algorithms. SIAM Journal on Computing 9(4), 808–826 (1980)
5. Jansen, K., Prädel, L.: New approximability results for two-dimensional bin packing. In: Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2013), pp. 919–936 (2013)
6. Jansen, K., Solis-Oba, R.: An asymptotic approximation algorithm for 3d-strip packing. In: Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm (SODA 2006), pp. 143–152. ACM Press (2006)
7. Kenyon, C., Rémila, E.: A near-optimal solution to a two-dimensional cutting stock problem. Mathematics of Operations Research 25(4), 645–656 (2000)
8. Lenstra, J.K., Shmoys, D.B., Tardos, É.: Approximation algorithms for scheduling unrelated parallel machines. Mathematical Programming 46, 259–271 (1990)
9. Li, K., Cheng, K.-H.: On three-dimensional packing. SIAM Journal on Computing 19(5), 847–867 (1990)
10. Li, K., Cheng, K.-H.: Heuristic algorithms for on-line packing in three dimensions. Journal of Algorithms 13(4), 589–605 (1992)
11. Miyazawa, F.K., Wakabayashi, Y.: An algorithm for the three-dimensional packing problem with asymptotic performance analysis. Algorithmica 18(1), 122–144 (1997)
12. Prädel, L.: Approximation Algorithms for Geometric Packing Problems. PhD thesis, University of Kiel (2012), http://eldiss.uni-kiel.de/macau/receive/dissertation_diss_00010802
13. Steinberg, A.: A strip-packing algorithm with absolute performance bound 2. SIAM Journal on Computing 26(2), 401–409 (1997)