Approximating Smallest Containers for Packing Three-Dimensional Convex Objects*

Helmut Alt[1] and Nadja Scharf[2]

1 **Institute of Computer Science, Freie Universität Berlin, Berlin, Germany**
alt@mi.fu-berlin.de

2 **Institute of Computer Science, Freie Universität Berlin, Berlin, Germany**
nadja.scharf@fu-berlin.de

Abstract

We investigate the problem of computing a minimum-volume container for the non-overlapping packing of a given set of three-dimensional convex objects. Already the simplest versions of the problem are $\mathcal{NP}$-hard so that we cannot expect to find exact polynomial time algorithms. We give constant ratio approximation algorithms for packing axis-parallel (rectangular) cuboids under translation into an axis-parallel (rectangular) cuboid as container, for packing cuboids under rigid motions into an axis-parallel cuboid or into an arbitrary convex container, and for packing convex polyhedra under rigid motions into an axis-parallel cuboid or arbitrary convex container. This work gives the first approximability results for the computation of minimum volume containers for the objects described.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases computational geometry, packing, approximation algorithm

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2016.11

1 Introduction

The problem of efficiently packing objects without overlap arises in a large variety of contexts. Apart from the obvious ones, where concrete objects need to be packed for transportation or storage, there are more abstract ones, for example cutting stock or scheduling. Given a set of objects that have to be cut out from the same material the objective is to minimize the waste, i.e., place the pieces to be cut out as close as possible. In the case of scheduling, a list of jobs is given. Each job needs a certain amount of given resources and the aim is to minimize under certain constraints this need of resources such as time, space, or number of machines. Altogether, this situation can be described as a problem of packing high-dimensional cuboids into a strip with bounded side lengths. So, both problems can be viewed as a given list of objects for which a container of minimum size is wanted.

In this work, we consider the more general and abstract problem of packing three-dimensional convex polyhedra into a minimum volume container. All variants of this problem are $\mathcal{NP}$-hard and we will develop constant factor approximation algorithms for some of them. The worst case constant factors are still very high, but probably they will be much lower for realistic inputs. The major aim of this paper, however, is to show the existence of constant factors at all, i.e., that the problems belong to the complexity class APX.

* This work was partially supported by DFG (Deutsche Forschungsgemeinschaft) under grant no. AL253/7-2.

Related Work

So far, there are only few results about finding containers of minimum volume. Related problems include strip packing and bin packing. In two-dimensional strip packing the width of a strip is given and the objects should be packed in order to minimize the length of the strip used. In three dimensions, the rectangular cross section of the strip is fixed. Bin-packing is the problem where the complete container is fixed and the objective is to minimize the number of containers to pack all objects. For both problems usually only translations are allowed to pack the objects.

For two-dimensional bin packing there exists an algorithm with an asymptotic approximation ratio of 1.405 [3] and Bansal et al. proved that there cannot be an APTAS unless $\mathcal{P} = \mathcal{NP}$ [2]. For two-dimensional strip packing there exists an AFPTAS [7]. In three dimensions there are algorithms with an asymptotic approximation ratio of 4.89 for bin packing [9] and an asymptotic approximation ratio of $\frac{3}{2} + \varepsilon$ for strip packing [6]. The best known worst case (non-asymptotic) approximation ratio for three-dimensional strip packing is $\frac{29}{4}$ [5].

For two dimensions, von Niederhäusern [11] gave algorithms for packing rectangles or convex polygons in a minimum-area rectangular container with approximation ratios 3 and 5 respectively. A recent result shows that packing convex polygons under translation into a minimum-area rectangular or convex container can be approximated with ratios 17.45 and 27 respectively [1].

PARTITION can be reduced to one-dimensional bin packing and one-dimensional bin packing is a special case of higher dimensional bin or strip packing. If one-dimensional bin packing could be approximated with a ratio smaller than $\frac{3}{2}$, we could solve PARTITION. Therefore, none of the mentioned problems can be approximated better than with ratio $\frac{3}{2}$ unless $\mathcal{P} = \mathcal{NP}$. PARTITION can also be reduced to our problem showing $\mathcal{NP}$-hardness.

Our Results

In this work we give the first approximation results for packing three-dimensional convex objects in a minimum-volume container. For packing axis-parallel rectangular cuboids under translation into an axis-parallel rectangular cuboid as a container, we achieve a $7.25 + \varepsilon$ approximation. If we allow the cuboids to be packed under rigid motions (translation and rotation) then we achieve an approximation ratio of 17.737 for an axis-parallel cuboid as container and an approximation ratio of 29.135 for an arbitrary convex container. For packing convex polyhedra under rigid motions we achieve an approximation ratio of 277.59 for computing an axis-parallel cuboid as container and 511.37 for a convex container.

2 Preliminaries and Reduction to Strip Packing

For most algorithms considered here, the input is a set of rectangular boxes $\mathcal{B} = \{b_1, b_2, \dots b_n\}$. We denote a box b_i in axis-parallel orientation by a tuple of its height, width and depth (h_i, w_i, d_i). We denote by $h_{\max} = \max\{h_i \mid b_i \in \mathcal{B}\}$, $w_{\max} = \max\{w_i \mid b_i \in \mathcal{B}\}$ and $d_{\max} = \max\{d_i \mid b_i \in \mathcal{B}\}$.

For points P and Q we denote by $\overline{PQ}$ the line segment between P and Q of length $|PQ|$. $\overrightarrow{PQ}$ denotes the vector from P to Q. When we write "axis-parallel container" we mean "axis-parallel rectangular cuboid as a container". We use the term box as a synonym for rectangular cuboid.

Packing under translation means that a separate translation is applied to each object moving it inside the container. The translated objects are not allowed to overlap. Packing under rigid motion means that a (separate) rotation may be applied to each object before it is translated into the container.

▶ **Definition 1** (strip packing). An instance for the *strip packing problem* consists of an axis parallel strip and a set of axis parallel boxes, i.e. in two dimensions the width and in three dimensions the width and the depth are fixed and the objective is to pack the boxes under translation such that the height is minimized.

▶ **Definition 2** (orthogonal minimal container packing – OMCOP). An instance of this problem is a set of convex polyhedra. The aim is to pack these polyhedra non-overlapping such that the minimal axis-parallel container has minimal volume. Variants include the kind of motions allowed or that more specialized objects are to be packed.

This work only considers algorithms in two or three dimensions. For ease of notation we always assume the lower left (front) corner of the container to lie in the origin. V_{opt} denotes the minimal possible volume for a container.

The following algorithm was given by von Niederhäusern [11]. It will be used later as a subroutine. For an example see Figure 1.

Algorithm 1:

Input: A list $\mathcal{S}$ of rectangles r_i, denoted by their width w_i and height h_i, a width for the strip w

1. Order the rectangles in $\mathcal{S}$ by decreasing width, such that if $i < j$ then $w_i \geq w_j$.
2. Split $\mathcal{S}$ in sublists $\mathcal{S}_j = \left\{r_i \in \mathcal{S} \mid \frac{w}{2^{j-1}} \geq w_i > \frac{w}{2^j}\right\}$ for $j \geq 1$.
3. Start with packing the rectangles in $\mathcal{S}_1$ on top of each other in the strip $[0, w] \times [0, \infty)$.
4. Split the remaining strip in two substrips with width $\frac{w}{2}$ and pack the rectangles in $\mathcal{S}_2$ one after another into these substrips. Each rectangle r_i is packed in the substrip with current minimal height.
5. Again split the substrips into two and pack S_3. Iterate that process until everything is packed.

▶ Remark. Note that the strip is half filled with rectangles up to the lower boundary of the highest rectangle that touches the upper end of the packing. Otherwise,this rectangle could have been placed lower. That means that the strip is half filled with rectangles except for a part with area at most $w \cdot h_{\max}$.

▶ Remark. Steps 1 and 2 can be done in $\mathcal{O}(n \log n)$ time where n is the size of $\mathcal{S}$. Steps 4 and 5 are presented in a simplified way in order to convey the idea of the algorithm in a more understandable manner. In reality it may happen that sublists $\mathcal{S}_j$ are empty and therefore splitting all substrips until they have the suitable width takes too much time. Hence, we split off a new substrip of suitable width from an existing one only when needed. To maintain all substrips with their currently occupied height, a heap-like data structure is used. Then, we can perform steps 3 to 5 in $\mathcal{O}(n \log n)$ time.

In this section we consider the version of OMCOP where the given objects are axis-parallel boxes that are to be packed under translation. The idea behind the reduction of OMCOP to strip packing is to test different base areas for the strip and to return the result with minimal volume. Assuming that the lower left corner of the base area is located at the origin, we test each point in a set $\mathcal{S}$ as a possible upper right corner for the base area. Testing means that we call a strip packing algorithm with the given boxes and the base area

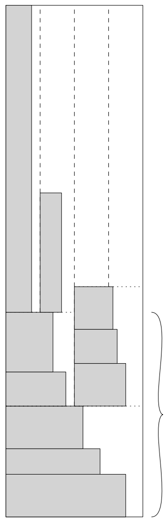


Figure 1 Result of Algorithm 1.

implied by the point of $\mathcal{S}$. $\mathcal{S}$ will be determined by a parameter ε: the smaller ε, the more elements $\mathcal{S}$ contains, the better the approximation ratio gets.

Note that for the width W_{opt} of an optimal container, the following inequalities hold:

1. $W_{\text{opt}} \leq W_\Sigma$, where W_Σ denotes the sum of all widths of the boxes to be packed. It is an upper bound because the width of an optimal container has to be the sum of the widths of some of the objects. Otherwise they can be pushed together reducing the width of the container and thereby its volume.
2. $W_{\text{opt}} \geq w_{\max}$, where $w_{\max}$ denotes the width of the widest box. Since this box needs to be packed, this is a lower bound for the width of the container.

The analogous bounds for the depth of an optimal container hold for the same reasons. In the following H_{opt}, W_{opt}, D_{opt}, and V_{opt} denote the height, width, depth, and volume of the same optimal container. Let $\varepsilon' = \frac{\varepsilon}{2(\varepsilon+\alpha)}$ for a constant α defined later.

The set $\mathcal{S}$ is obtained by dividing the intervals of possible width and depth logarithmically:

$$\begin{aligned}\mathcal{S} =& \{W_\Sigma (1-\varepsilon')^i \mid i \in \mathbb{N}, W_\Sigma (1-\varepsilon')^i > w_{\max}\} \cup \{w_{\max}\} \times \\ & \{D_\Sigma (1-\varepsilon')^j \mid j \in \mathbb{N}, D_\Sigma (1-\varepsilon')^j > d_{\max}\} \cup \{d_{\max}\}.\end{aligned}$$

For an example for $\mathcal{S}$ see Figure 2.

▶ **Theorem 3.** *If we use an α-approximation algorithm of runtime $T(n)$ to pack n boxes under translation into the strips and the set $\mathcal{S}$ defined above, we obtain an $(\alpha+\varepsilon)$-approximation algorithm for the* OMCOP *variant where n axis aligned boxes are to be packed under translation. Its runtime is $\mathcal{O}\left(T(n)\frac{\log^2 n}{\varepsilon^2}\right)$.*

Proof. There exist $a, b \in \mathbb{N}$ with $W_\Sigma (1-\varepsilon')^{a+1} < W_{\text{opt}} \leq W_\Sigma (1-\varepsilon')^a$ and $D_\Sigma (1-\varepsilon')^{b+1} < D_{\text{opt}} \leq D_\Sigma (1-\varepsilon')^b$. Eventually the boxes will be packed in a strip with base area $W \times D$ with $W = W_\Sigma (1-\varepsilon')^a$ and $D = W_\Sigma (1-\varepsilon')^b$. Since $W \geq W_{\text{opt}}$ and $D \geq D_{\text{opt}}$, the minimal height for a strip packing with base area $W \times D$ is at most H_{opt}. Therefore, we obtain a packing with height $H \leq \alpha H_{\text{opt}}$. The associated container has volume

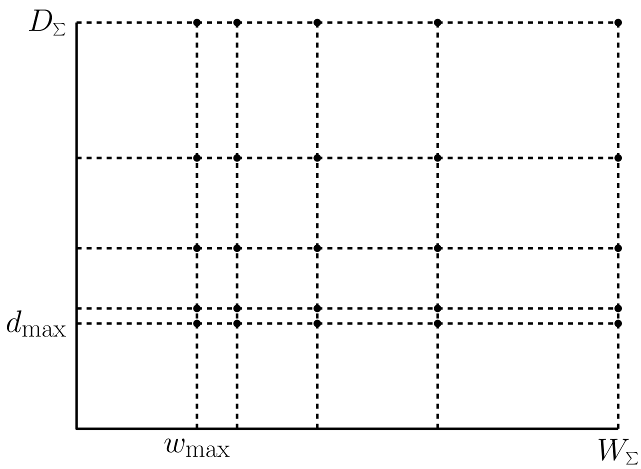


Figure 2 Example for Set $\mathcal{S}$ with $\varepsilon = \frac{3}{4}$ and $\alpha = 1.5$.

V with

$$\begin{aligned}
V &= HWD \\
&\leq (\alpha H_{\text{opt}}) \left(W_\Sigma (1-\varepsilon')^a\right) \left(D_\Sigma (1-\varepsilon')^b\right) \\
&\leq (\alpha H_{\text{opt}}) \left(\frac{W_{\text{opt}}}{1-\varepsilon'}\right) \left(\frac{D_{\text{opt}}}{1-\varepsilon'}\right) \\
&\leq \frac{\alpha}{(1-\varepsilon')^2} V_{\text{opt}} \\
&\leq \frac{\alpha}{1-2\varepsilon'} V_{\text{opt}} = (\alpha+\varepsilon) V_{\text{opt}} && \text{, since } \varepsilon' = \frac{\varepsilon}{2(\varepsilon+\alpha)}.
\end{aligned}$$

The size of $\mathcal{S}$ is

$$\begin{aligned}
|\mathcal{S}| &= \left(\left\lceil \log_{\frac{1}{1-\varepsilon'}} W_\Sigma \right\rceil - \left\lfloor \log_{\frac{1}{1-\varepsilon'}} w_{\max} \right\rfloor + 1\right) \left(\left\lceil \log_{\frac{1}{1-\varepsilon'}} D_\Sigma \right\rceil - \left\lfloor \log_{\frac{1}{1-\varepsilon'}} d_{\max} \right\rfloor + 1\right) \\
&= \mathcal{O}\left(\frac{\log^2 n}{(-\log(1-\varepsilon'))^2}\right) \text{, since } \frac{W_\Sigma}{w_{\max}} \leq n \text{, where } n \text{ is the number of boxes} \\
&= \mathcal{O}\left(\frac{\log^2 n}{\varepsilon^2}\right),
\end{aligned}$$

since $-\log(1-x) \geq x$ for $x \in [0,1]$ and $\varepsilon' \geq c\varepsilon$ for some constant $c > 0$.

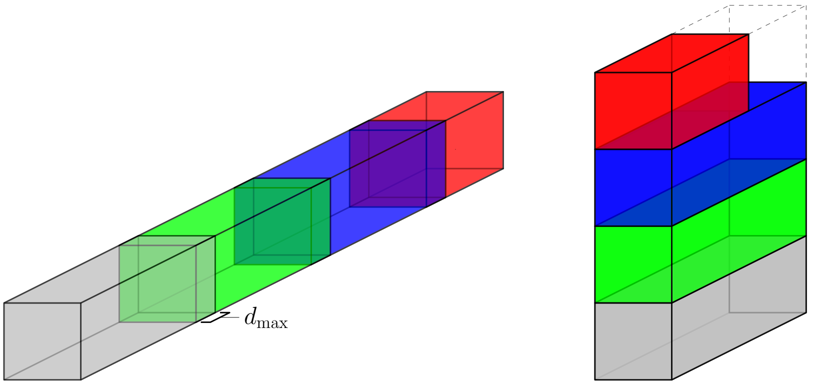
Therefore we get the desired running time. ◀

If we use the algorithm given by Diedrich et al. [5] to pack the boxes into the strips, we obtain the following corollary.

▶ **Corollary 4.** *There exists a $(7.25+\varepsilon)$-approximation algorithm for packing axis-parallel boxes under translation into a minimum volume axis-parallel box with running time polynomial in both the input size and $\frac{1}{\varepsilon}$.*

3 Algorithms for Variants of OMCOP

In this section, we will give algorithms for variants of OMCOP. The basic idea is to get rid of the third dimension by dividing the set of objects into sets of objects with similar height



(a) Cut strip

(b) Pieces obtained from one strip stacked on top of each other

Figure 3

and then packing those using an algorithm for two-dimensional boxes. These containers then get cut into pieces with equal base area and the pieces will be stacked on top of each other.

3.1 Packing Cuboids under Translation

Even though this algorithm gets outperformed by the construction in the previous section, we state it here as base for the algorithms for the other variants. Let $\alpha \in (0,1)$ and $c > 1$ be two parameters that we will choose later.

Algorithm 2:

Input: A set of axis parallel boxes $\mathcal{B} = \{b_1, \dots, b_n\}$

1. Partition $\mathcal{B}$ into subsets of boxes that have almost the same height: $\mathcal{B}_j = \left\{b_i \in \mathcal{B} \mid h_{\max} \cdot \alpha^j < h_i \leq h_{\max} \cdot \alpha^{j-1}\right\}$.
2. Pack the boxes of every $\mathcal{B}_j$ into a strip with width $w_{\max}$ and height $h_{\max} \cdot \alpha^{j-1}$ considering the depth of the boxes instead of the height, i.e., the strip grows into the depth. This is done by applying Algorithm 1 to pack the lower facets of the boxes (rectangles) into the lower facet of the strip (2d-strip).
3. Divide the strips into pieces with depth $(c-1) \cdot d_{\max}$, ignoring the last part of the strip of depth $d_{\max}$. (Parts of boxes contained in this part of the strip will be covered in step 5 anyway.)
4. Assign each box to the piece its front lies in.
5. Extend each piece to depth $c \cdot d_{\max}$ such that every assigned box lies entirely in the piece.
6. Stack the pieces on top of each other.

For an illustration of steps 3 to 6 see Figure 3. The first step can be done in $\mathcal{O}(n)$ time. The second step needs time $\mathcal{O}(n \log n)$ (see Remarks on Algorithm 1). The rest can be done in linear time. Therefore, Algorithm 2 runs in $\mathcal{O}(n \log n)$ time. We obtain

▶ **Theorem 5.** *For suitable values of c and α Algorithm 2 computes a $\left(\frac{3}{\sqrt[3]{2}-1} \approx 11.542\right)$-approximation for the variant of three-dimensional* OMCOP *where n axis parallel cuboids are packed under translation in $\mathcal{O}(n \log n)$ time.*

Proof. Let D_j denote the depth of the strip obtained in step 2 for the boxes in $\mathcal{B}_j$. Then we get by step 3 $\left\lceil \frac{D_j - d_{\max}}{(c-1)d_{\max}} \right\rceil$ pieces. After step 5 each piece has volume $c \cdot d_{\max} w_{\max} h_{\max} \alpha^{j-1}$.

Consider the total volume V_j of the pieces obtained for the subset $\mathcal{B}_j$:

$$V_j = c \cdot d_{\max} \left\lceil \frac{D_j - d_{\max}}{(c-1)\, d_{\max}} \right\rceil w_{\max} h_{\max} \alpha^{j-1}$$
$$< \frac{c}{c-1} \left(D_j - d_{\max}\right) w_{\max} h_{\max} \alpha^{j-1} + c \cdot d_{\max} w_{\max} h_{\max} \alpha^{j-1}.$$

We know from the two-dimensional packing algorithm that the base area of the strip is half filled with boxes except for the last part of depth at most $d_{\max}$ (see Remarks on Algorithm 1), so $(D_j - d_{\max})\, w_{\max} \leq 2 \sum_{b \in \mathcal{B}_j} A_B(b)$ where $A_B(b)$ denotes the base area of box b. We also know that for every $b_i \in \mathcal{B}_j$ the inequality $h_{\max} \alpha^{j-1} < \frac{h_i}{\alpha}$ holds. Therefore, we get for the total volume of the packing V that

$$V \leq \sum_j \left(\frac{c}{c-1} \left(D_j - d_{\max}\right) w_{\max} h_{\max} \alpha^{j-1} + c \cdot d_{\max} w_{\max} h_{\max} \alpha^{j-1} \right)$$
$$\leq \sum_j \left(\frac{2c}{\alpha(c-1)} \sum_{b \in \mathcal{B}_j} V(b) + c \cdot w_{\max} \cdot d_{\max} \cdot h_{\max} \alpha^{j-1} \right)$$
$$\leq \frac{2c}{\alpha(c-1)} \underbrace{\sum_{b \in \mathcal{B}} V(b)}_{\leq V_{\text{opt}}} + c \cdot \underbrace{w_{\max} \cdot d_{\max} \cdot h_{\max}}_{\leq V_{\text{opt}}} \cdot \sum_{l=0}^{\infty} \alpha^l \tag{1}$$
$$\leq \left(\frac{2c}{\alpha(c-1)} + \frac{c}{1-\alpha} \right) V_{\text{opt}}. \tag{2}$$

The factor before V_{opt} in term (2) is minimized if the partial derivatives with respect to c and α are 0. Solving the resulting system of equations we get $c = \sqrt[3]{2} + 1 \approx 2.2599$ and $\alpha = \frac{1}{3}\left(2 - \sqrt[3]{4} + \sqrt[3]{2}\right) \approx 0.5575$. This gives an approximation ratio of $\frac{3}{\sqrt[3]{2}-1} \approx 11.542$. ◀

3.2 Packing Cuboids under Rigid Motions

3.2.1 Cuboid as Container

Now we consider the variant of OMCOP where the objects to be packed are boxes and rigid motions are allowed. Let V_{opt} denote the volume of an optimal container for the given setting. We basically use the algorithm stated above but with an extra preprocessing step, namely rotating every box $b_i \in \mathcal{B}$ such that it becomes axis parallel and $h_i \geq w_i \geq d_i$. This can be done in $\mathcal{O}(n)$ time. To prove the performance bound of this algorithm we need the following lemma.

▶ **Lemma 6.** *If every $b_i = (h_i, w_i, d_i) \in \mathcal{B}$ is oriented such that $h_i \geq w_i \geq d_i$, then $h_{\max} \cdot w_{\max} \cdot d_{\max} \leq \sqrt{6} \cdot V_{opt}$.*

Proof. Since an optimal container has to contain the box determining $h_{\max}$, it contains a line segment of length $h_{\max}$. The projection of that line segment on at least one of the axes has to have length at least $\frac{1}{\sqrt{3}} h_{\max}$. W.l.o.g. let this axis be the x-axis. Therefore, the optimal container has an extent of at least $\frac{1}{\sqrt{3}} h_{\max}$ in x-direction.

Since every box is at least as high as wide, a box with width $w_{\max}$ contains a disk D with diameter $w_{\max}$ and so the optimal container does. Observe that D contains a diametric line segment l which is parallel to the y-z-plane. Consequently, the projection of l and therefore the one of the whole box on the y-axis or on the z-axis has a length of at least $\frac{1}{\sqrt{2}} w_{\max}$. W.l.o.g. let this be the y-axis.

A box with depth $d_{\max}$ contains a sphere with diameter $d_{\max}$. The projection of this sphere on any axis has length at least $d_{\max}$.

Summarizing, each optimal box has volume at least $\frac{1}{\sqrt{6}} h_{\max} \cdot w_{\max} \cdot d_{\max}$ ◀

Observe that every argument leading to inequality (1) still holds for this variant of the algorithm. Using Lemma 6 to estimate $h_{\max} \cdot w_{\max} \cdot d_{\max}$ we get an approximation factor of $\frac{2c}{\alpha(c-1)} + \frac{c \cdot \sqrt{6}}{1-\alpha}$. Minimizing this expression as before yields the following theorem.

▶ **Theorem 7.** *The given algorithm computes a* 17.738*-approximation for the variant of three-dimensional* OMCOP *where* n *axis parallel cuboids are packed under rigid motions in* $\mathcal{O}(n \log n)$ *time.*

3.2.2 Convex Container

If we allow a convex container instead of an orthogonal container, we can use the same algorithm but adapt the analysis. The arguments leading to inequality (1) still hold since they only use the total volume of the boxes as estimate for the volume of an optimal container. To estimate $h_{\max} \cdot w_{\max} \cdot d_{\max}$, we use the following lemma. Note that V_{opt} here denotes the volume of a minimal convex container instead of an axis parallel container.

▶ **Lemma 8.** *If every* $b_i = (h_i, w_i, d_i) \in \mathcal{B}$ *is oriented such that* $h_i \geq w_i \geq d_i$*, then* $h_{\max} \cdot w_{\max} \cdot d_{\max} \leq 6 \cdot V_{opt}$.

Proof. Consider the line segment, disk and sphere from the proof of Lemma 6. The line segment has length $h_{\max}$. The disk with diameter $w_{\max}$ contains a line segment of length $w_{\max}$ that is perpendicular to the first line segment. The sphere with diameter $d_{\max}$ contains a line segment of length $d_{\max}$ that is perpendicular to the first two line segments. It is well known (see, e.g., Lemma 6 from [8]) that the convex hull of these three line segments has a volume of at least $\frac{1}{6} h_{\max} w_{\max} d_{\max}$. ◀

This leads with inequality (1) to the approximation ratio $\frac{2c}{\alpha(c-1)} + \frac{c \cdot 6}{1-\alpha}$. Minimizing this term as before yields the following theorem.

▶ **Theorem 9.** *Using the algorithm described in section 3.2 we get a* 29.135*-approximation for packing* n *axis parallel boxes under rigid motions into a smallest-volume convex container in time* $\mathcal{O}(n \log n)$.

3.3 Packing Convex Polyhedra under Rigid Motions

3.3.1 Cuboid as Container

We use the algorithm from the previous sections to pack convex polyhedra under rigid motions into an axis-parallel box of minimal volume. To do so, we add another preprocessing step where we compute a bounding box for every polyhedron according to the following lemma. We then pack these boxes with the algorithm discussed in the previous section.

▶ **Lemma 10.** *For every* m*-vertex convex polyhedron* K *in* $\mathbb{R}^d$*, there is a box* B *that contains* K *with* $V(B) \leq d! V(K)$ *that can be computed in* $O(d^2 m^2)$ *time, or* $O(m \log m)$ *time if* $d = 3$.

Proof by induction on the dimension d. In one dimension, the Lemma holds obviously.

In higher dimensions d, let P, Q be two points of K with maximum distance and $|PQ| = l$. Let π_P be the hyperplane normal to $\overline{PQ}$ in the point P. Let K' be the orthogonal projection of K onto π_P. By the inductive hypothesis there is a $(d-1)$-dimensional box B' containing K'

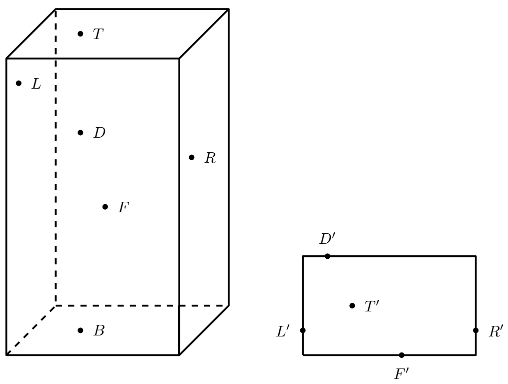


Figure 4 Box with a point of the enclosed polyhedron in every facet and the projection of the box on a plane perpendicular to $\overline{TB}$. By construction, the images of T and B under the projection are the same.

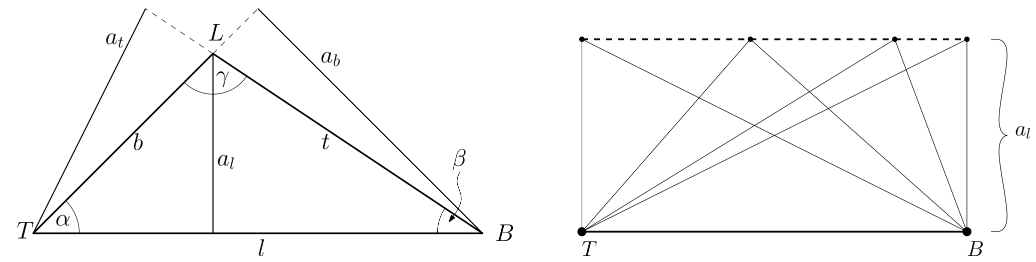
for which $V'(B') \leq (d-1)!V'(K')$ where V' denotes the $(d-1)$-dimensional volume. Then K is contained in the box B with base B' and height l and $V(B) = lV'(B') \leq l(d-1)!V'(K')$. It is well known (see e.g. [8]) that for any convex body K, its projection K' on some hyperplane π_P, and a line segment l perpendicular to π_P, it holds: $V(K) \geq \frac{1}{d} \cdot l \cdot V'(K')$. Hence, we get for the volume of B: $V(B) \leq d!V(K)$. B can be computed by testing every pair of vertices to find P and Q that have maximal distance. This takes $\mathcal{O}(dm^2)$ time. Then K gets projected on a hyperplane perpendicular to $\overline{PQ}$. This is possible in $\mathcal{O}(dm)$ time. Then we proceed recursively with the projection of K. In total we need $\mathcal{O}(d^2m^2)$. The asymptotically fastest algorithm for dimension three however has runtime $\mathcal{O}(m \log m)$, see [10]. ◀

The construction in the proof of Lemma 10 is the same as in Lemma 7 from [8]. We get a total running time of $\mathcal{O}(m \log m)$ for computing the bounding boxes of three-dimensional polyhedra with m vertices in total.

For the analysis of the algorithm presented in this section we need several notations and lemmata that follow. Consider the box $b = (h, w, d)$ obtained from the polyhedron p by Lemma 10 after the algorithm rotated it in axis-parallel position such that $h \geq w \geq d$. Notice that in every facet of b lies at least one point of p. We call the top and bottom one T and B, which are unique by construction. In the left and right facet of b, we choose such a point from each and call them L and R. By construction, the distance from them to the front facet has to be the same. We do the same for the front and rear facet and call them F and D respectively. We know from the construction that $|TB| = h$ and $\overline{TB}$ is parallel to the longest edge of b. If we project the polyhedron onto a plane perpendicular to $\overline{TB}$, we call the images of T, L, R, F and D under the projection T', L', R', F' and D', respectively. See Figure 4 for illustration. Due to the construction of b, $|L'R'| = w$ holds.

▶ **Lemma 11.** *Let $b = (h, w, d)$ with $h \geq w \geq d$ be the enclosing box obtained for polyhedron p by the algorithm from Lemma 10. Then, parallel to any given plane, p contains a line segment of length at least $w \cdot \frac{1}{\sqrt{5}}$.*

Proof. Consider the points T, B, L and R as described above. The distance between line segment $\overline{TB}$ and L or the distance between line segment $\overline{TB}$ and R is at least $\frac{w}{2}$. Let w.l.o.g. L be the point with larger distance to $\overline{TB}$. Consider the triangle $\triangle(T, B, L)$ with edges and angles labeled according to Figure 5a. Notice that $\alpha \leq 90°$ and $\beta \leq 90°$. Let a_t be the height of the triangle on edge t, a_b on edge b, and a_l on edge l.



(a) Labelled triangle $\triangle(T, B, L)$ **(b)** Possible triangles $\triangle(T, B, L)$

Figure 5

Due to the construction of $\triangle(T, B, L)$, we know that $a_l \geq \frac{w}{2}$. We will later show that $a_b \geq \frac{w}{\sqrt{5}}$ and $a_t \geq \frac{w}{\sqrt{5}}$. If we choose a plane parallel to the given one, such that the intersection between the plane and $\triangle(T, B, L)$ contains T, B or L but is not only one point, then we know that the intersection is at least a line segment with length at least $\min(a_t, a_b, a_l) \geq \frac{w}{\sqrt{5}}$ which completes the proof. It remains to show that $a_t, a_b \geq \frac{w}{\sqrt{5}}$.

We only show that $a_b \geq \frac{w}{\sqrt{5}}$ since the proof for a_t is analogous. Figure 5b depicts possible triangles with given distance $|TB|$ and height a_l. a_b is the distance between B and the line defined by T and L. Since $\beta \leq 90°$ this distance is minimal for $\beta = 90°$.

Let A be the area of $\triangle(T, B, L)$ with $\beta = 90°$.

It holds

$$\frac{a_l \cdot |TB|}{2} = A = \frac{a_b \cdot |TL|}{2}.$$

Hence

$$a_l \cdot h = a_b \cdot \sqrt{h^2 + a_l^2},$$

since $|TB| = h$ and using Pythagoras' theorem for replacing $|TL|$. That gives

$$\begin{aligned} a_b &= \frac{a_l \cdot h}{\sqrt{h^2 + a_l^2}} \\ &= \frac{1}{\sqrt{\frac{1}{a_l^2} + \frac{1}{h^2}}} \\ &\geq \frac{1}{\sqrt{\frac{4}{w^2} + \frac{1}{w^2}}} \\ &= \frac{w}{\sqrt{5}} \end{aligned}$$ ◀

▶ **Lemma 12.** *Let $b = (h, w, d)$ with $h \geq w \geq d$ be the enclosing box obtained for a convex polyhedron p by the algorithm from Lemma 10. Then the projection of p onto any arbitrary line g has length at least $\frac{1}{8\sqrt{3}}d$.*

This Lemma is shown by an elaborate construction, where we find four line segments inside p such that the projection of at least one of them onto g has length at least $\frac{1}{8\sqrt{3}}d$. See Appendix A for the complete proof.

Summarized, the algorithm for packing convex polyhedra works as follows: First, we compute a bounding box for every polyhedron with the algorithm from Lemma 10, then we rotate each box b_i together with its contained polyhedron p_i in an axis-prallel orientation such that $h_i \geq w_i \geq d_i$. Finally, we run Algorithm 2 with the rotated boxes.

Now consider the polyhedra p_1, p_2, p_3 that determine $h_{\max}$, $w_{\max}$ and $d_{\max}$ in the placement of the enclosing boxes the described algorithm computes. p_1 contains a line segment of length $h_{\max}$ and so its projection to at least one of the axes is at least $\frac{1}{\sqrt{3}}h_{\max}$. W.l.o.g. let this axis be the x-axis. Furthermore, by Lemma 11 the projection of p_2 onto the y-z-plane contains a line of length at least $\frac{1}{\sqrt{5}}w_{\max}$. Therefore, the projection of p_2 onto the y-axis or the one onto the z-axis has length at least $\frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{5}}w_{\max} = \frac{1}{\sqrt{10}}w_{\max}$. The projection of p_3 to the remaining axis has length at least $\frac{1}{8\sqrt{3}}d_{\max}$ by Lemma 12. An axis parallel box with minimal volume containing p_1, p_2, p_3 has at least the described side lengths and so we get the following lemma:

▶ **Lemma 13.** *For packing convex polyhedra under rigid motions into a minimum-volume axis parallel container, the following inequality holds:* $h_{\max} \cdot w_{\max} \cdot d_{\max} \leq 24\sqrt{10}V_{opt}$.

From Lemma 10 we know that the volume of the smallest enclosing box for a polyhedron is at most 6 times the volume of the polyhedron. With the previous lemma and this knowledge we derive the following approximation ratio from inequality (1):

$$\frac{12c}{\alpha\,(c-1)} + \frac{c \cdot 24\sqrt{10}}{1-\alpha}. \tag{3}$$

The running time of this algorithm is determined by the computation of the bounding boxes and the packing of these boxes: $\mathcal{O}\,(m \log m + n \log n) = \mathcal{O}\,(m \log m)$ where m is the total number of vertices of the polyhedra. Hence, by minimizing term (3) as before we get the following theorem.

▶ **Theorem 14.** *The given algorithm computes an orthogonal container with volume at most* 277.59 *times the volume of an orthogonal minimal container for the variant of three-dimensional* OMCOP *where a set of convex polyhedra having m vertices in total are to be packed under rigid motions. The runtime of the algorithm is* $\mathcal{O}\,(m \log m)$.

3.3.2 Convex Container

Next, we show that the algorithm from the previous section is not only a constant factor approximation for the smallest axis parallel cuboid under rigid motions but even for the smallest convex container. Of course, the approximation factor is higher and, first, we get the following lemma instead of Lemma 13:

▶ **Lemma 15.** *For packing convex polyhedra under rigid motions into a minimum-volume convex container, the following inequality holds:* $h_{\max} \cdot w_{\max} \cdot d_{\max} \leq 48\sqrt{15}V_{opt}$.

Proof. As before let p_1, p_2, p_3 be the polytopes that determine $h_{\max}$, $w_{\max}$ and $d_{\max}$. p_1 contains a line segment of length $h_{\max}$. By Lemma 11, p_2 contains a line segment of length $\frac{w_{\max}}{\sqrt{5}}$ that is perpendicular to the first line segment. By Lemma 12, p_3 contains a line segment with length $\frac{d_{\max}}{8\sqrt{3}}$ that is perpendicular to the first two lines. Since any convex body containing three pairwise perpendicular line segments of length a, b, c has volume at least $\frac{1}{6}abc$ (cf. Lemma 6 in [8]), we get a lower bound on the volume of the convex hull which is also a lower bound for the volume of an optimal container. ◀

As before we use Lemma 10 and the previous lemma to estimate inequality (1) and obtain the following approximation ratio: $\frac{12c}{\alpha(c-1)} + \frac{c \cdot 48\sqrt{15}}{1-\alpha}$. Minimizing this term as before yields the following result.

▶ **Theorem 16.** *The algorithm given in Section 3.3 computes a convex container with volume at most* 511.37 *times the volume of a minimal convex container for packing a set of convex polyhedra having* m *vertices in total under rigid motions in time* $\mathcal{O}(m \log m)$.

References

1 Helmut Alt, Mark de Berg, and Christian Knauer. Approximating minimum-area rectangular and convex containers for packing convex polygons. In *Proc. 23th Annual European Symp. Algorithms (ESA)*, pages 25–34, 2015.

2 Nikhil Bansal, José R. Correa, Claire Kenyon, and Maxim Sviridenko. Bin packing in multiple dimensions: inapproximability results and approximation schemes. *Math. Oper. Res.*, 31(1):31–49, 2006.

3 Nikhil Bansal and Arindam Khan. Improved approximation algorithm for two-dimensional bin packing. In *Proc. 25th Annual ACM-SIAM Symp. Discrete Algorithms (SODA)*, pages 13–25, 2014.

4 H. S. M. Coxeter. *Introduction to geometry*, page 12. John Wiley & Sons, Inc., New York-London-Sydney, second edition, 1969.

5 Florian Diedrich, Rolf Harren, Klaus Jansen, Ralf Thöle, and Henning Thomas. Approximation algorithms for 3D orthogonal knapsack. *J. Comput. Sci. Tech.*, 23(5):749–762, 2008.

6 Klaus Jansen and Lars Prädel. A new asymptotic approximation algorithm for 3-dimensional strip packing. In *Proc. 40th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM)*, pages 327–338, 2014.

7 Claire Kenyon and Eric Rémila. A near-optimal solution to a two-dimensional cutting stock problem. *Math. Oper. Res.*, 25(4):645–656, 2000.

8 A. M. Macbeath. A compactness theorem for affine equivalence-classes of convex regions. *Canadian J. Math.*, 3:54–61, 1951.

9 F. K. Miyazawa and Y. Wakabayashi. Three-dimensional packings with rotations. *Comput. Oper. Res.*, 36(10):2801–2815, 2009.

10 Edgar A. Ramos. An optimal deterministic algorithm for computing the diameter of a three-dimensional point set. *Discrete & Computational Geometry*, 26(2):233–244, 2001.

11 Léonard von Niederhäusern. Packing polygons: Research of approximation algorithms. Master's thesis, Freie Universität Berlin and École Polytechnique Fédérale de Lausanne, 2014.

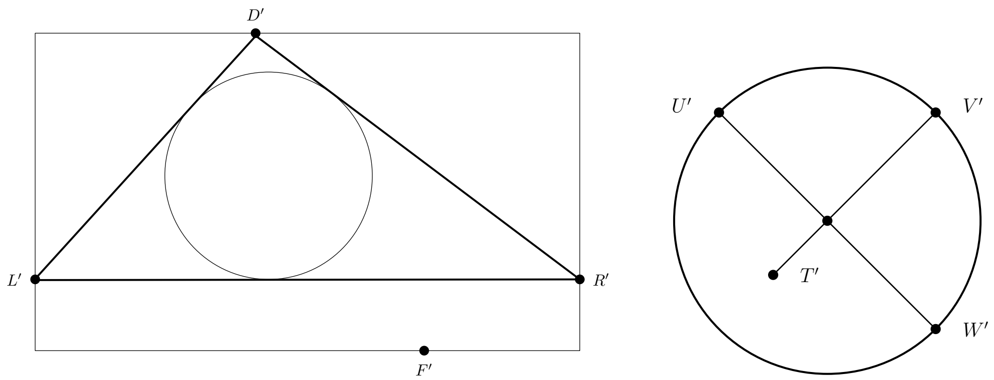
A Proof of Lemma 12

We construct four line segments inside of p such that the projection of at least one of them onto the line has the desired length.

Consider the projection of p onto a plane perpendicular to $\overline{TB}$ as described above (Figure 4). Then $\triangle(L', R', F')$ or $\triangle(L', R', D')$ has an area $A \geq \frac{dw}{4}$. The perimeter of the projection of the box, namely $2(w+d)$, gives an upper bound for the perimeter u of the triangles. It is well known (see, e.g.,[4]) that the radius of the incircle of a triangle with area A and perimeter u is $r = \frac{2A}{u}$. Hence, we know that the projection of p contains a circle with radius r where

$$r = \frac{2A}{u} \geq \frac{dw}{4(d+w)} \geq \frac{1}{8}d \text{ , since } d \leq w. \tag{4}$$

See Figure 6a for an example.



(a) Circle in the projection of p that has radius at least $\frac{1}{8}d$

(b) Construction of U', V' and W'

Figure 6

Now we can find points U', V', W' in the projection, such that U', V', W' lie on the circle with radius r and $|T'V'| = k \geq r$, $|U'W'| = l = 2r$ and $\overline{T'V'} \perp \overline{U'W'}$. To obtain V', we shoot a ray from T' through the center of the circle until we hit the circle and call this point V'. $\overline{U'W'}$ is the diameter of the circle perpendicular to $\overline{T'V'}$. See Figure 6b for an example.

Let U, V, W be preimages of U', V', W' under the projection. Hence, they lie inside p. The line segments whose projections on the given line g we consider are $\overline{BT}$, $\overline{BV}$, $\overline{VT}$ and $\overline{WU}$.

The length of the projection of a line segment onto g is the scalar product of the vector between the endpoints of the line segment and a unit vector with same direction as g. To simplify the computation of the scalar product, we define the coordinate system as follows: B is equal to the origin. T lies on the z-axis. The y-coordinate of V is 0. Then U and W have the same x-coordinate. Now we have

$$\overrightarrow{BT} = \begin{pmatrix} 0 \\ 0 \\ h \end{pmatrix} \quad \overrightarrow{BV} = \begin{pmatrix} k \\ 0 \\ h_V \end{pmatrix} \quad \overrightarrow{VT} = \begin{pmatrix} -k \\ 0 \\ h - h_V \end{pmatrix} \quad \overrightarrow{WU} = \begin{pmatrix} 0 \\ l \\ h_{WU} \end{pmatrix},$$

for values k, l with properties described above, and h_V, h_{WU} where $0 \leq h_V \leq h$ and $|h_{WU}| \leq h$. Let $\overrightarrow{g} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ be the direction of g in the defined coordinate system, with $|\overrightarrow{g}| = 1$. We now look at the lengths of the projections of the line segments onto the given line and distinguish four cases.

Case 1: $|x| \geq \frac{1}{\sqrt{3}}$. Then, using inequality (4), if $\operatorname{sgn}(z) = \operatorname{sgn}(x)$

$$|\overrightarrow{BV} \cdot \overrightarrow{g}| \geq k|x| \geq r|x| \geq \frac{1}{\sqrt{3} \cdot 8} d$$

or, if $\operatorname{sgn}(z) \neq \operatorname{sgn}(x)$

$$|\overrightarrow{VT} \cdot \overrightarrow{g}| \geq k|x| \geq \frac{1}{\sqrt{3} \cdot 8} d.$$

Case 2: $|z| \cdot h \geq \frac{1}{\sqrt{3}\cdot 8} d$. Note, that this inequality is satisfied if $|z| \geq \frac{1}{\sqrt{3}}$. Then

$$|\overrightarrow{BT} \cdot \overrightarrow{g}| = h \cdot |z| \geq \frac{1}{\sqrt{3} \cdot 8} d.$$

Case 3: $|y| \geq \frac{1}{\sqrt{3}}$ and $\mathrm{sgn}(y) = \mathrm{sgn}(h_{WU} z)$. Then

$$|\overrightarrow{WU} \cdot \overrightarrow{g}| \geq l|y| \geq \frac{1}{\sqrt{3} \cdot 8} d.$$

Case 4: $|y| \geq \frac{1}{\sqrt{3}}$ and $\mathrm{sgn}(y) \neq \mathrm{sgn}(h_{WU} z)$ and $|z| \cdot h < \frac{1}{\sqrt{3}\cdot 8} d$. Note: $|h_{WU} z| \leq h|z| < \frac{1}{\sqrt{3}\cdot 8} d$ and $l|y| = 2r|y| \geq \frac{2}{\sqrt{3}\cdot 8} d$, hence

$$|\overrightarrow{WU} \cdot \overrightarrow{g}| = l|y| - |h_{WU} z| \geq \frac{1}{\sqrt{3} \cdot 8} d.$$

Since $|\overrightarrow{g}| = 1$, $|x| \geq \frac{1}{\sqrt{3}}$ or $|y| \geq \frac{1}{\sqrt{3}}$ or $|z| \geq \frac{1}{\sqrt{3}}$ holds. Hence, at least one of the 4 cases occurs because $h \geq d$.