Constant Ratio Approximation Algorithms for Weighted Single Container Packing

Rolf Harren *

Graduate School of Informatics, Kyoto University, Yoshida-Honmachi, Sakyo-ku, Kyoto 606-8501, Japan

and

Fachbereich Informatik, Universität Dortmund, August-Schmidt-Str. 12, 44227 Dortmund, Germany

Rolf.Harren@uni-dortmund.de

Abstract

Given a set of rectangular three-dimensional items, all of them associated with a profit, and a single bigger rectangular three-dimensional bin, we can ask to find a non-rotational, non-overlapping packing of a selection of these items into the bin to maximize the profit. This problem differs from three-dimensional strip- and bin-packing as we are to pack into a single bounded bin. We derive a $(16 + \epsilon)$-approximation algorithm and improve the algorithm to an approximation ratio of $(9 + \epsilon)$.

`It turned out, that there is a mistake in Lemma 3.2 that affects the correctness of the` $(9+\epsilon)$ `algorithm. Instead of correcting the mistake here, we refer to [4]. Based on the ideas developed in this work, we derived a` $(9+\epsilon)$ `and a` $(8+\epsilon)$ `algorithm in [4].`

1 Introduction

Different packing problems have been in the focus of study for a long time. This paper concentrates on weighted three-dimensional packing problem with the objective to pack a selection of the rectangular items without rotations into a single bin. In contrast to this, bin packing problems require to pack all items into a minimal number of bins. Strip packing is another well-studied packing problem where items have to be packed into a strip of bounded basis and unlimited height. The aim is to minimize the height that is needed to pack all items. For all these problems different variants are examined. Rotation can be allowed or forbidden. A speciality for three-dimensional packing is the z-orientated packing where rotations are allowed only along the z-axis. Furthermore, for single container packing we can maximize the number of items or maximize the covered space. These are special cases of the weighted packing where the profits are equal to 1 and equal to the space, respectively. Although these problems have some similarities, they differ considerably and the results cannot be adopted directly.

The two-dimensional packing problems have been studied more broadly. We present some of the results that are connected to this work. For two-dimensional strip-packing the best-known result with absolute approximation ratio is given by Steinberg [10] with an approximation ratio of 2. Bansal and Sviridenko [1] proved that two-dimensional bin packing does not admit an asymptotic PTAS. The best-known positive result is an asymptotic approximation ratio of 1, 69... given by Caprara [2]. For the special case of packing squares into a minimum number of bins an asymptotic PTAS is given by Bansal and Sviridenko [1]. Finally, Jansen and Stee presented a $(2 + \epsilon)$-approximation for rotational bin packing [6]. Packing rectangles into a single bin has been slightly neglected in the past. Therefore only a limited number of results are presently available. The best result is a $(2 + \epsilon)$-algorithm given by Jansen and Zhang in 2004 [7]. For weighted

*This work was partly supported by DAAD (German Academic Exchange Service)

packing of a *large* number of items into a bin (which means, that the size of the items is much smaller than the size of the bin) Fishkin, Gerber and Jansen [5] gave a PTAS. The special case of maximizing the number of packed squares into a rectangular bin admits an asymptotic FPTAS and therefore a PTAS (Jansen and Zhang [8]).

There are much less results for three-dimensional packing and all corresponding papers focus on strip- or bin-packing rather than on single container packing or give heuristics without an analysis of their approximation ratio. Miyazawa and Wakabayashi gave asymptotic 2, 67-approximation algorithms for strip packing [11] and z-oriented packing [12].

Formally the single container packing problem (**SCPP**) is stated as follows. We are given a list I of n rectangular items $r_1, \dots r_n$ of sizes (a_i, b_i, c_i) and profit p_i. The objective is to find a non-overlapping packing of a selection of these items into a bigger container $C = (a, b, c)$ such that the total profit of the packed items is maximized.

Obviously three-dimensional packing finds many applications, especially for the transportation and distribution of goods. Rectangle packing also finds an application in the advertisement placement which is to maximize the profit for rectangular placards that can be sticked to a limited billboard. Not so obviously we can find analogies to non-preemptive scheduling problems. Container packing (especially in the z-orientated setting) is related to job scheduling in partitionable mesh connected systems.

The paper is organized as follows. In the preliminaries in Section 2 we make some preparations as well as introducing an important result from two-dimensional strip packing, which will be employed by our algorithms. In Section 3 we develop a $(16 + \epsilon)$-approximation algorithm for SCPP, based on the ideas of a $(3 + \epsilon)$-algorithm for rectangle packing in [7]. Later we improve this algorithm to a $(9 + \epsilon)$-algorithm.

2 Preliminaries

At first we introduce some notations for two-dimensional packing and describe the measurement of approximation algorithms. Two-dimensional packing (or rectangle packing) is defined similarly as follows. We are given a list I of n rectangular items $r_1, \dots r_n$ of sizes (a_i, b_i) and profit p_i. The objective is to find a non-overlapping packing of a selection of these items into a bigger bin $C = (a, b)$ such that the total profit of the packed items is maximized.

We distinguish rectangles as *wide*, *high*, *big* and *small*. An item $r_i = (a_i, b_i)$ is called wide if $a_i > a/2$ and $b_i \le b/2$, high if $a_i \le a/2$ and $b_i > b/2$, big if $a_i > a/2$ and $b_i > b/2$ and finally small if $a_i \le a/2$ and $b_i \le b/2$. See Figure 1 for an illustration.

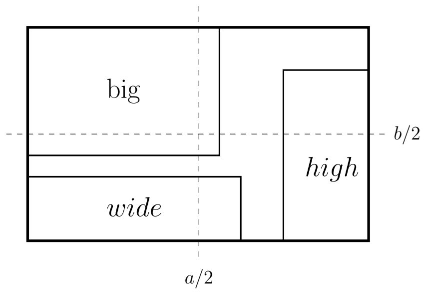

Figure 1: Wide, high and big items

In 1990, Leung et al. [9] proved the NP-completeness in the strong sense for the problem of determining whether a set of squares can be packed into a bigger square. As this is a very special case of weighted rectangle packing (and also single container packing) we directly obtain the NP-hardness in the strong sense for our problem. Therefore we concentrate on approximation algorithms. To evaluate approximation algorithms, we use the standard measure *absolute performance ratio*, which is defined for maximization problems as $R_A = sup_I OPT(I)/A(I)$ where $OPT(I)$ and $A(I)$ are the optimal value and the objective value given by an approximation algorithm A for any instance I, respectively. An algorithm A is called a δ-approximation algortihm if $OPT(I)/A(I) \le \delta$ for any instance I.

Although our task is to find a concrete packing of a selection of items, we will usually not really describe the packing, but rather show that a certain selection of items can be packed. We can show this by using a result from two-dimensional strip packing, known as Steinberg's Theorem [10]. We give a variant of this theorem, which is proved in [7].

Theorem 2.1 (Variant of Steinberg's Theorem). *If the total area of a set T of items is at most $\frac{ab}{2}$ and there are no high items (or wide items), then* *a packing for the items in T into a bin of size (a, b) can be found in polynomial time.*

3 Single container packing

3.1 A constant ratio approximation algorithm for single container packing

At first we show a relatively simple $(16+\epsilon)$-approximation algorithm which we improve with some minor tweaks to a $(9+\epsilon)$-approximation. The main idea for our algorithm is to use a PTAS for the m-dimensional knapsack problem to get a selection of items with almost optimal profit and pack this selection in up to 16 bins.

The m-dimensional knapsack problem is to find for given matrix $A \in N^{m \cdot n}$ and vectors $b \in N^m$ and $p \in N^n$ a solution $x \in N^n$ which maximizes the profit $\sum_{i=1}^{n} p_i x_i$ under the side condition $Ax \leq b$. All numbers are nonnegative integers. Chandra et al. [3] proved the existence of a PTAS for the m-dimensional knapsack problem.

For every direction a, b and c we create a side condition which bounds the sum of the side planes of the long items. An item is called *long in one direction* if the length in this direction is more than half of the bins size. The conditions are:

$$\sum_{i \in I,\, a_i > a/2} b_i c_i \cdot x_i \leq bc$$

$$\sum_{i \in I,\, b_i > b/2} a_i c_i \cdot x_i \leq ac$$

$$\sum_{i \in I,\, c_i > c/2} a_i b_i \cdot x_i \leq ab$$

A fourth and fifth condition is added in order to limit the total space used by all items and the number of big items, respectively.

$$\sum_{i \in I} a_i b_i c_i \cdot x_i \leq abc$$

$$\sum_{i \in I\ a_i > a/2\ b_i > b/2\ c_i > c/2} x_i \leq 1$$

The objective remains untouched:

$$\sum_{i \in I} p_i x_i$$

Lemma 3.1. *Every packable selection I' of items can be canonically interpreted as a vector that satisfies the m-dimensional knapsack instance.*

Proof. We show that all the constraints are satisfied. Obviously the *total space*- and the *big item*-constraint are satisfied by every feasible packing. Now observe that two items that are long in direction a can not be packed behind each other in this direction (they would exceed the boundary of the bin) - see Figure 2. Therefore the sum of the projected area in bc-direction of items that are long in direction a is bounded by bc - see Figure 3. Similarly we can derive the side conditions for the other directions. □

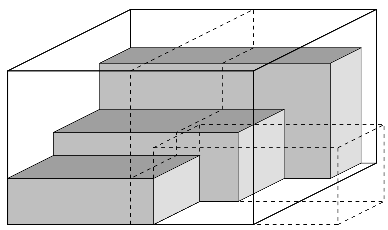

Figure 2: Packing of long items in direction a

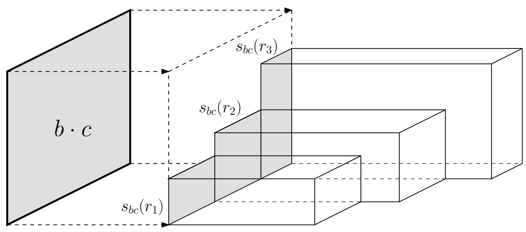

Figure 3: Side condition for long items in direction a

We conclude that the optimal solution of the knapsack instance has at least the same profit as the optimal solution of the SCPP instance. Now we use the PTAS for m-dimensional knapsack to find a list I_k of items with almost optimal profit $p(I_k) \geq (1-\epsilon)OPT_{knapsack}(I) \geq (1-\epsilon)OPT(I)$.

Suppose we have a set I_k of items with almost optimal profit $p(I_k)$ from the knapsack problem. Now we separate I_k into three sets S_a, S_b, S_c of long items (see Figure 4), a set S_{big} of at most one big item and a set S_{small} of small items such that

$$
\begin{aligned}
S_a &= \{i \in I_k \quad | \quad a_i > a/2, c_i \leq c/2\} \\
S_b &= \{i \in I_k \quad | \quad b_i > b/2, a_i \leq a/2\} \\
S_c &= \{i \in I_k \quad | \quad c_i > c/2, b_i \leq b/2\} \\
S_{big} &= \{i \in I_k \quad | \quad a_i > a/2, b_i > b/2, c_i > c/2\} \\
S_{small} &= \{i \in I_k \quad | \quad a_i \leq a/2, b_i \leq b/2, c_i \leq c/2\}
\end{aligned}
$$

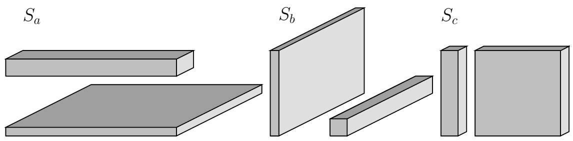

Figure 4: Illustration of shapes of items in S_a, S_b and S_c

We pack each set (except of S_{big} which we include to one of the others later) separately into up to 3 (as for S_a, S_b and S_c) and up to 7 bins (as for S_{small}). In total we get along with 16 bins and declare the bin with highest profit P^* as the solution, achieving an approximation ratio of $(16+\epsilon)$.

Packing of the long items: Exemplarily we describe the packing of S_a. As we saw before the projections on the (b, c)-side plane of the bin do not overlap in a regular packing. We use this observation to reduce the problem to rectangle packing. The algorithm we use equals the $(3+\epsilon)$-approximation algorithm in Jansen and Zhang [7]. Pack the bc-projections of the items into a rectangle of size $(2b, c)$ using the variant of Steinberg's Theorem 2.1. This is possible since the total area of the projections is $\leq bc$ and there are no wide items (if we regard the b direction as the horizontal direction). Draw a vertical line to divide the $(2b, c)$ rectangle into two rectangles of size (b, c) and pack the items, that are cut by this line, into a third rectangle of the same size. To do this, retain the order of the items and pack them left aligned into the rectangle - see Figure 5 for an illustration.

Finally we fit these three (b, c)-rectangles into three (a, b, c)-bins and erect the associated items in a direction on their projections - see Figure 6. The same algorithm can be applied to S_b and S_c, packing each in 3 bins of size (a, b, c). S_{big} contains at most one item it can be packed together with any of the sets S_a, S_b or S_c (since the *surface*-conditions already consider the big item).

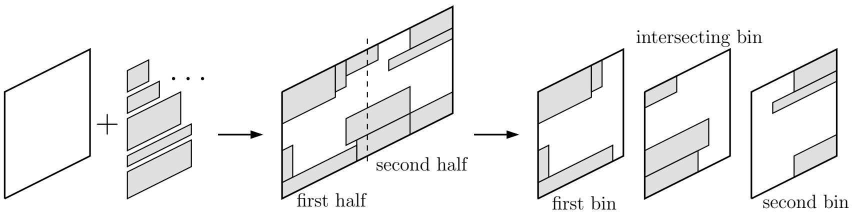

Figure 5: Packing the projections of S_a into three bins

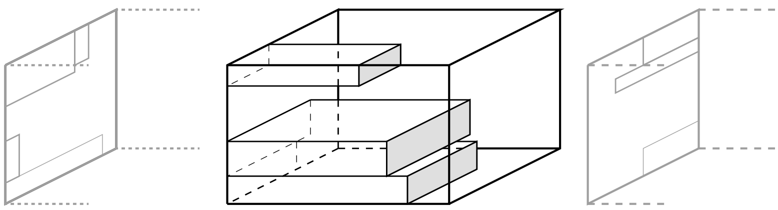

Figure 6: Erecting the items of S_a on the pattern of their projections

Packing of the small items: To pack the set of small items S_{small} we use a combination of rectangle packing with Steinberg and a layer packing into an unlimited three-dimensional strip similar to the simple two-dimensional algorithm *Next-Fit-Decreasing-Height (NFDH)* which orders the items in non-increasing height and fills the strip level by level. First, we order the items in non-increasing height $c_1 \geq c_2 \geq \ldots \geq c_k$. Second, we group the items such that the sum of the *ab*-

size of each group is as close to $ab/2$ as possible. Formally, we define variables $g_1, \dots g_l$ such that $g_1 = 1$ and

$$\begin{aligned} S_i = \sum_{j=g_i}^{g_{i+1}-1} a_j b_j &\leq ab/2 \quad \text{for all} \quad 1 \leq i \leq l-1 \quad \text{and} \\ \sum_{j=g_i}^{g_{i+1}} a_j b_j &> ab/2 \quad \text{for all} \quad 1 \leq i \leq l-1 \quad \text{and} \\ S_i = \sum_{j=g_l}^{k} a_j b_j &\leq ab/2 \end{aligned}$$

As all the items are small, the ab-area of each item is $\leq ab/4$. Therefore $S_i \geq ab/4$ for all $1 \leq i \leq l-1$. Furthermore the ab-projection of each group $G_i = (g_i, g_i+1, \dots, g_{i+1}-1)$ of items can be packed into a rectangle of size (a,b) with Steinberg's algorithm, as their total area is bounded by $ab/2$ and there are neither wide nor high items. As for the large items we can erect the small items on their projection and therefore pack each group G_i into one layer of size (a, b, c_{g_i}) with the highest height in that group. Insert the layers in non-increasing order into a strip with the basis (a,b) and unlimited height - see Figure 7.

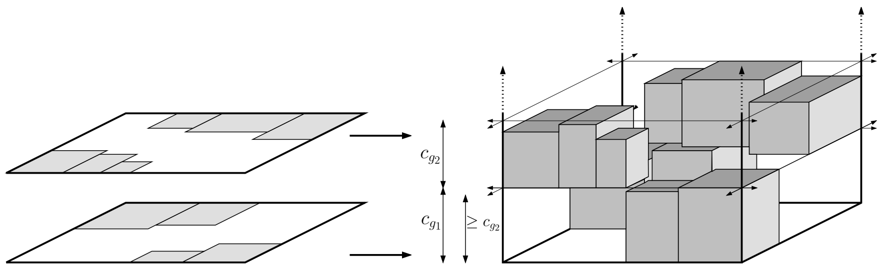

Figure 7: Layer packing of S_{small}

We analyse the height of this strip packing. Suppose the total volume of small items is bounded by $V = \sum_{i \in S_{small}} a_i b_i c_i \leq \alpha\, abc$. In every layer $1 \leq i \leq l-1$ all items have a c-length of at least $c_{g_{i+1}}$. Hence, the filling of every layer can be estimated by the height of the succeeding layer. Therefore the total volume can be estimated by

$$V \geq \sum_{i=1}^{l-1} c_{g_{i+1}} S_i \geq \sum_{i=1}^{l-1} c_{g_{i+1}} ab/4$$

omitting the thinnest layer. Thus we get

$$\sum_{i=1}^{l-1} c_{g_{i+1}} \leq 4\alpha\, c.$$

With this estimation we can bound the total height of the strip to:

$$H = \sum_{i=1}^{l} c_{g_i} = c_{g_1} + \sum_{i=2}^{l} c_{g_i} \leq \left(\frac{1}{2} + 4\alpha\right) c$$

In this basic analysis we use $\alpha = 1$ as an upper bound (we need a more precise volume estimation for the improvement in the next step) and derive an absolute height limit of $\frac{9}{2}c$. Using the same idea as for the long items - but this time the three-dimensional equivalent - we cut the strip with vertical planes into 4 bins of size (a,b,c) and one bin of size $(a,b,c/2)$ - see Figure 8.

The items which are cut by the 4 planes can be put together into two more bins of size (a,b,c). To do this, retain the order of the intersections on two of the cutting planes and insert them bottom aligned and top aligned respectively into one bin. As the c-length is bounded by $c/2$ the items do not overlap. Thus we can pack S_{small} into at most 7 bins which proves

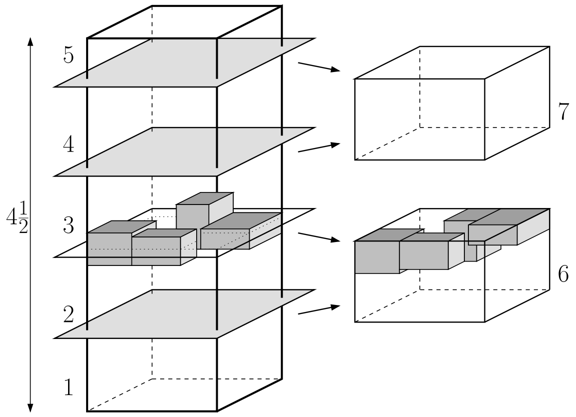

Figure 8: Layer packing of S_{small}

Theorem 3.1 (Single container packing)**.** *There is a polynomial time approximation algorithm with performance ratio of at most $16+\epsilon$ for single container packing, for any $\epsilon > 0$.*

3.2 Improvement

`As already mentioned in the modified abstract, Lemma 3.2, which is basic for the improvement of the algorithm, is not correct. Nevertheless the idea of separating the items into` S_a, S_b, S_c, S_{big} `and` S_{small} `and the Observation 1 are still correct and are used in [4] to derive a` $(9+\epsilon)$`- and a` $(8+\epsilon)$`-approximation. Furthermore the idea of using placeholder items might still be useful for improving on these results.`

The basic $(16+\epsilon)$-approximation algorithm finds a selection of items with the m-dimensional knapsack and packs this selection into up to 16 bins. This method is adopted from the $(3+\epsilon)$-algorithm for rectangle packing in [7]. It is not surprising, that the algorithm can be improved by using the best-known result, which is the $(2+\epsilon)$-algorithm of Jansen and Zhang [7]. To do this we have to change our perspective. We can not longer pack all items into a certain number of bins, but separate the selection into several sets of items and apply 'as good as possible'-approximation algorithms on them. Therefore we need the following lemma

Lemma 3.2. *Given a set of items S which fulfills the m-dimensional knapsack instance of the previous section, an algorithm P that produces a partition $S_1 \cup S_2 \cup \ldots \cup S_l = S$ and a list of approximation algorithms $A_1, A_2, \ldots, A_l$ which have an approximation ratio of $\delta_1, \delta_2, \ldots, \delta_l$ on $S_1, S_2, \ldots, S_l$, respectively, there is an approximation algorithm A for SCPP with approximation ratio at most $\delta_1 + \delta_2 + \ldots + \delta_l + \epsilon$ for every $\epsilon > 0$.*

Proof. Let A be the algorithm which first finds a selection of items S with the m-dimensional knapsack of the previous section, using $\epsilon' = \epsilon/(\delta_1 + \delta_2 + \ldots + \delta_l)$. Then it applies P on S and A_i on S_i for $1 \leq i \leq l$. Let $\overline{S_i}$ be the output of the algorithm A_i. A outputs $\max(p(\overline{S_1}), p(\overline{S_2}), \ldots, p(\overline{S_l}))$ together with the corresponding packing. We prove that $OPT(I) \leq (\delta_1 + \delta_2 + \ldots + \delta_l + \epsilon)A(I)$.

$$
\begin{aligned}
OPT(I) &\leq (1+\epsilon')p(S) \\
&= (1+\epsilon')(p(S_1) + p(S_2) + \ldots + p(S_l)) \\
&\leq (1+\epsilon')(\delta_1 p(\overline{S_1}) + \delta_2 p(\overline{S_2}) + \ldots + \delta_l p(\overline{S_l})) \\
&\leq (1+\epsilon')(\delta_1 + \delta_2 + \ldots + \delta_l) \max(p(\overline{S_1}), p(\overline{S_2}), \ldots, p(\overline{S_l})) \\
&= (\delta_1 + \delta_2 + \ldots + \delta_l + \epsilon)A(I)
\end{aligned}
$$

□

`The mistake in this prove is that` $p(S_i) \leq \delta_i p(\overline{S}_i)$ `doesn't hold. Instead we have` $OPT(S_i) \leq \delta_i p(\overline{S}_i)$. `The former equation would only be true, if algorithm` A_i `packed all items` S_i `into the bins, which is not true for the` $(2+\epsilon)$ `algorithm.`

Note, that the final ϵ can also absorb other arbitrarily small number in the approximation ratios of $A_1, \dots, A_l$. The algorithm P is only needed to ensure later, that the sets S_i have certain properties from which the algorithm A_i can benefit.

The main idea of the improved algorithm is to consider the volumes of the different sets. Obviously the strip height of S_{small} decreases if the volume of these items is smaller. We make two observations before we give the algorithm.

Observation 1 (Packing little volume of S_a)**.** *S_a can be packed into one bin if $Vol(S_a) \leq \frac{1}{4}\,abc$.*

Proof. If $Vol(S_a) \leq \frac{1}{4}\,abc$ then $S_{bc}(S_a) \leq \frac{1}{2}\,bc$ where $S_{bc}(S_a)$ is the sum of the bc-side planes of the items in S_a. As there are no high item in S_a (which means long in direction c), the bc- projections of the items can be packed into one bin of size (b, c) with Steinbergs algorithm. Finally the items can be erected in a- direction as usual. Obviously the same applies to the sets S_b and S_c and we can include S_{big} to any of these sets. □

Observation 2 (Packing little volume of S_{small})**.** *S_{small} can be packed into $f(\alpha)$ bins if $Vol(S_{small}) \leq \alpha\,abc$. Where $f(\alpha)$ is defined as*

$$f(\alpha) = \begin{cases} 7 & \text{if} \quad 7/8 < \alpha \leq 1 \\ 6 & \text{if} \quad 6/8 < \alpha \leq 7/8 \\ 5 & \text{if} \quad 5/8 < \alpha \leq 6/8 \\ 4 & \text{if} \quad 3/8 < \alpha \leq 5/8 \\ 3 & \text{if} \quad 2/8 < \alpha \leq 3/8 \\ 2 & \text{if} \quad 1/8 < \alpha \leq 2/8 \\ 1 & \text{if} \quad 0 \leq \alpha \leq 1/8 \end{cases}$$

Unfortunately the funktion $f(\alpha)$ can not be defined directly because of an odd gap between the α-values of 3/8 and 5/8. Nevertheless the proof is quite straightforward.

Proof. Recall, that we analyzed the height of the strip packing for the small items already bearing in mind their total volume. Therefore we can use the resulting strip height of $\leq (\frac{1}{2} + 4\alpha)c$. Now it is easy to see, that for certain strip heights a certain number of bins is sufficient. Exemplarily, a strip height of up to $3c$ can be packed into 4 bins (we have two cutting planes whose intersection can be packed together into one bin) - and this height is sufficient for $\alpha \leq 5/8$. Note, that for odd numbers of cutting planes (for example the height of $3\frac{1}{2}c$ for $\alpha \leq 6/8$ has 3 cutting planes) the remaining last regular bin might be only half-filled and can therefore hold the surplus intersection. □

Now we give a first improved algorithm with approximation ratio $(10 + \epsilon)$ which we improve with a last tweak to $(9 + \epsilon)$ afterwards.

1. Find a selection S of items with the m-dimensional knapsack problem as in the basic algorithm,

2. separate the items in S into the sets S_a, S_b, S_c and S_{small} as defined above (but consider a potential big item belonging to S_a),

3. if $Vol(S_a) \leq 1/4\,abc$ than pack S_a into one bin, otherwise use the $(2+\epsilon)$-algorithm for rectangle packing for the bc-projection and erect the items on the pattern (deal similarly with S_b and S_c),

4. pack S_{small} into a strip and transfer the strip into a minimal number of bins according to $f(\alpha)$,

5. choose the bin with the highest profit.

Analysis: We consider four different cases:

1. $Vol(S_a), Vol(S_b), Vol(S_c) \leq 1/4\,abc$
   In this case we need 3 bins for S_a, S_b and S_c and (like in the basis algorithm) 7 bins for S_{small} - summing up to 10 bins in total.

2. $Vol(S_a) > 1/4\,abc$ and $Vol(S_b), Vol(S_c) \leq 1/4\,abc$
   Using Lemma 3.2 we get $\delta_a = 2 + \epsilon$ for packing S_a and $\delta_b = \delta_c = 1$ for packing S_b and S_c (because they are packed completely in one bin each). Finally $\alpha \leq 6/8$ and therefore $\delta_{small} = 5$ (as S_{small} can be packed into 5 bins). In total we get $\delta_a + \delta_b + \delta_c + \delta_{small} = 9 + \epsilon$.

3. $Vol(S_a), Vol(S_b) > 1/4\,abc$ and $Vol(S_c) \leq 1/4\,abc$
   Analogue we get $\delta_a = \delta_b = 2+\epsilon$, $\delta_c = 1$ and $\delta_{small} = 4$ (as $\alpha \leq 4/8$) - summing up to $\delta_a + \delta_b + \delta_c + \delta_{small} = 9 + 2\epsilon$.

4. $Vol(S_a), Vol(S_b), Vol(S_c) > 1/4\,abc$
   Analogue we get $\delta_a = \delta_b = \delta_c = 2 + \epsilon$ and $\delta_{small} = 2$ (as $\alpha \leq 2/8$) - summing up to $\delta_a + \delta_b + \delta_c + \delta_{small} = 8 + 3\epsilon$.

As we can interchange S_a, S_b and S_c we proved the approximation ratio of $(10 + \epsilon)$ for our improved algorithm. Obviously only the first case causes difficulties for the improvement to $(9 + \epsilon)$. In this case S_a, S_b and S_c contain only very limited volume - this leads to the idea to pack some of the small items together with the long items.

If the total volume of items packed together with S_a, S_b and S_c into 3 bins is $\geq \frac{1}{8}abc$ then the remaining small items can be packed into 6 bins (see Observation 2). Hence, assume that $Vol(S_a \cup S_b \cup S_c) < \frac{1}{8}abc$. W.l.o.g $Vol(S_a) \leq Vol(S_b) \leq Vol(S_c)$ and therefore $Vol(S_a), Vol(S_b) < \frac{1}{16}abc$. Obviously the sums of the side planes of the items in S_a and S_b are therefore bounded by $S_{bc}(S_a) < \frac{1}{8}bc$ and $S_{ac}(S_b) < \frac{1}{8}ac$. We can add a placeholder item P_a of size $(a, b/2, c/2)$ in S_a and a placeholder item P_b of size $(a/2, b, c/2)$ in S_b and pack S_a and S_b together with their placeholder items into one bin each. Now we add some items of S_{small} into the space of the placeholder item. Therefore we define

$$S_a^* = \{r \in S_{small} \quad | \quad b_r c_r \geq \frac{1}{8}bc\}$$
$$S_b^* = \{r \in S_{small} \quad | \quad a_r c_r \geq \frac{1}{8}ac\}$$

and pile up items of S_a^* in a-direction into placeholder item P_a until we either exceed a height of $a/2$ or run out of items. Similarly we pile up items of S_b^* in b-direction into P_b. As items can be included in both of the sets S_a^* and S_b^* we pay attention that each item is only included in one placeholder. If the item supply from S_a^* and S_b^* is sufficient, P_a and P_b have a volume of $\geq \frac{1}{16}abc$ each (ground $\geq \frac{1}{8}$ and height $\geq \frac{1}{2}$ in the fitting dimensions), summing up to $\geq \frac{1}{8}abc$. This permits to pack the remaining small items in S_{small} into 6 bins.

Now assume w.l.o.g a lack of items in S_a^*. Then it is possible to pack all items of S_a^* into one of the two placeholders (items that are also contained in S_b^* can be put into P_b too) and the bc-side plane of each remaining small item in S_{small} is $< \frac{1}{8}bc$. Change the direction of the layer packing of the remaining items into direction a. Considering the analysis of the layer packing we can improve the layer filling from $S_i \geq \frac{1}{4}bc$ to $S_i \geq \frac{3}{8}bc$. Taking this into account we derive a strip height of $H \leq (\frac{1}{2} + \frac{8}{3}\alpha)a$ which is $\leq 3\frac{1}{2}a$ for $\alpha = 1$. Therefore we can pack the remaining small items into 5 bins. We proved

Theorem 3.2 (Improved single container packing)**.** *There is a polynomial time approximation algorithm with performance ratio of at most* $9 + \epsilon$ *for single container packing, for any* $\epsilon > 0$.

Remark: The running time of this improved algorithm is dominated by several applications of the $(2 + \epsilon)$-algorithm for rectangle packing and therefore of extremely high order. It is also possible to derive a $(10 + \epsilon)$-algorithm without using the $(2 + \epsilon)$-algorithm for rectangle packing and thus getting a much better running time.

References

[1] Bansal, N. and Sviridenko, M. (2004). New approximability and inapproximability results for 2-dimensional bin packing. Proc. 15th ACM-SIAM Symposium on Discrete Algorithms, 196-203.

[2] Caprara, A. (2002). Packing 2-dimensional bins in harmony. Proc. 43rd Foundations of Computer Science, 490-499.

[3] Chandra, A.K., Hirschberg D. S. and Wong, C. K. (1976). Approximate algorithms for some generalized knapsack problems. *Theoretical Computer Science 3*, 293-304.

[4] Diedrich, F., Harren, R., Jansen, K., Thöle, R. and Thomas, H (2007). Approximation Algorithms for 3D Orthogonal Knapsack. Proc. 4th International Conference on Theory and Applications of Models of Computation, 34-45.

[5] Fishkin, A., Gerber, O. and Jansen, K. (2004). On weighted rectangle packing with large resources. Proc. 3rd IFIP International Conference on Theoretical Computer Science, 237-250.

[6] Jansen, K. and Stee, R. (2005). On strip packing with rotations. Proc. 37th ACM Symposium on Theory of Computing, 755-761.

[7] Jansen, K. and Zhang, G. (2004). On rectangle packing: maximizing benefits. Proc. 15th ACM-SIAM Symposium on Discrete Algorithms, 204-213.

[8] Jansen, K. and Zhang, G. (2004). Maximizing the number of packed rectangles. Proc. 9th Scandinavian Workshop on Algorithm Theory, 362-371.

[9] Leung, J.Y.-T, Tam, T.W., Wong, C.S., Young, G.H. and Chin, F.Y.L. (1990). Packing squares into a square. *Parallel and Distributed Computation 10*, 271-275

[10] Steinberg, A. (1997). A strip-packing algorithm with absolute performance bound 2. SIAM Journal on Computing 26, 401-409.

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

[12] Miyazawa, F.K. and Wakabayashi, Y. (1999). Approximation algorithms for the orthogonal z-oriented three-dimensional packing problem. SIAM Journal on Computing, 1008-1029