Journal of Combinatorial Optimization (2019) 37:1011–1044
https://doi.org/10.1007/s10878-018-0338-y

Drawer algorithms for 1-space bounded multidimensional hyperbox packing

Paulina Grzegorek[1] · **Janusz Januszewski**[1]

Published online: 17 August 2018

Abstract
We study a multidimensional hyperbox packing with one active bin. The items (d-dimensional hyperboxes of edge length not greater than 1) arrive one by one. Each item must be packed online into a hypercube bin of edge 1 and 90°-rotations are allowed. If it is impossible to pack an item into an active bin, we close the bin and open a new active bin to pack that item. In this paper, we present a 3.5^d-competitive as well as a $12 \cdot 3^d$-competitive online d-dimensional hyperbox packing algorithm with one active bin.

Keywords Online algorithms · Bin packing · Multidimensional · One-space bounded

Mathematics Subject Classification 68W27

1 Introduction

A finite sequence S of items is given. When all the items of S are accessible, the packing method is called *offline*. When items arrive one by one and each item that has arrived must be packed into a bin and cannot be moved thereafter, the packing method is called *online*. In the online version of packing a crucial parameter is the number of bins available for packing, i.e., *active bins*. It is natural to expect a packing method to be less efficient with fewer number of active bins. Online packing methods are further divided into two classes: *unbounded space* when no restriction on the number of active bins occurs and t-*space bounded* with the maximum of t active bins at the same time.

✉ Paulina Grzegorek
Paulina.Grzegorek@utp.edu.pl

Janusz Januszewski
januszew@utp.edu.pl

1 Institute of Mathematics and Physics, UTP University of Science and Technology, Al. Prof. S. Kaliskiego 7, 85-789 Bydgoszcz, Poland

In this paper we consider an online version of packing with only one active bin, i.e., a 1-space bounded model. Each item of the sequence S is packed into the active bin. When the packing is not possible, the bin is closed and is never used again. A new active bin is opened.

Let $A(S)$ be the number of bins used by the algorithm A to pack a sequence S. Let $OPT(S)$ be the number of bins used to pack items from S with the most efficient offline method, i.e., the minimum number of bins containing whole sequence S. The *asymptotic competitive ratio* for algorithm A is defined as:

$$R_A^\infty = \lim_{n\to\infty} \sup_S \left\{ \frac{A(S)}{OPT(S)} \mid OPT(S) = n \right\}.$$

The online bin packing is a classical problem studied for more than 40 years. One-dimensional bin packing was first investigated in Ullman (1971) (see also Johnson et al. 1974), where the performance ratio of the First Fit algorithm was proved to be 17/10. The analysis of the Next Fit algorithm can be found in Johnson (1974), where the author shows that the performance ratio is not greater than 2. Revised First Fit presented in Yao (1980) has performance ratio 5/3. The article also gives the lower bound 3/2 of one dimensional online bin packing. The improvement of this result can be found in Brown (1979) and Liang (1980), who proved that the lower bound is not smaller than 1.53635. Currently the best know lower bound is 1.54014, proved by van Vliet (1992).

Concerning the two-dimensional online bin packing algorithms Coppersmith and Raghavan (1989) presented the algorithm with competitive ratio 3.25. The result was later improved by Csirik et al. (1993) to 3.0625 and by Han et al. (2001) to 2.7834. Further improvements can be found in Seiden and van Stee (2003), where the authors show the upper bound 2.66013 of the asymptotic competitive ratio. The upper bound currently stands at 2.5545 (see Han et al. 2011).

One can also consider general bounded space packing methods, where the number of active bins is finite, but not specified. Lee and Lee (1985) presented the Harmonic algorithm with competitive ratio not greater than 1.63597. Ramanan et al. (1989) showed that the upper bound can be improved to 1.61217 and gave the lower bound 1.58333. Seiden (2002) further improved the upper bound to 1.58889. The best know upper bound 1.5813 is proved in Heydrich and van Stee (2016).

In optimal algorithms (Harmonic algorithm and its improvements) when the asymptotic competitive ratio approaches the optimal value, the number of active bins diverges to infinity. It is hard to expect to use this result in practical applications. Thus a question arises: What asymptotic competitive ratio can we achieve when the number of active bins is bounded above by a given (small) natural number? This question was addressed by Woeginger (1993) whose Simplified Harmonic 6-space bounded online algorithm has competitive ratio beneath 17/10.

Different types of items in the sequence S can be considered. In d-dimensional bin packing problem a bin is a unit hypercube and each item of S is a hyperbox of edge lengths not greater than 1. Items can be rotated by 90° in any plane defined by arbitrary two of the item's edges.

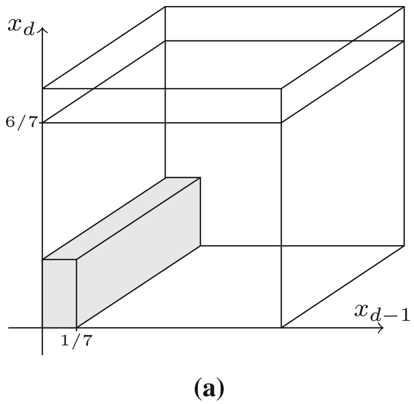
For d-dimensional hyperbox packing Epstein and van Stee (2005) gave a $(\Pi_\infty)^d$-competitive space bounded algorithm, where $\Pi_\infty \approx 1.69103\ldots$ is the competitive ratio of the one-dimensional harmonic algorithm, see Lee and Lee (1985). Algorithms with only one active bin and 2-dimensional items were presented for the first time in Zhang et al. (2010b), where a method with competitive ratio 8.84 was given. A paper by Zhang et al. (2010a) gives an online packing algorithm with competitive ratio 5.155 for rectangles and a 4.5-competitive algorithm for squares. Another result was obtained by Zhang et al. (2014) with competitive ratios 5.06 and 4.3 achieved for squares. Also a paper by Grzegorek and Januszewski (2015) presents a 3.883-competitive online square packing algorithm. In an article by Januszewski and Zielonka (2018) the authors describe a 4.84-competitive 1-space bounded 2-dimensional bin packing algorithm and present the lower bound of 3.246 for the competitive ratio. In a paper by Januszewski and Zielonka (2016) the reader can find a 3.8165-competitive 2-space bounded algorithm for rectangles and a 3.6-competitive model for squares. A 3-space bounded 3.577-competitive square packing method is given in Grzegorek and Januszewski (2014). The d-dimensional case of one-space bounded hyperbox packing is considered in Zhang et al. (2013). The authors give an online algorithm with competitive ratio equal to 4^d. Two-space bounded hypercube packing with competitive ratio $32/21 \cdot 2^d$ is discussed in Zhao and Shen (2015). Online packing of d-dimensional hypercubes with total volume not greater than $(n+1)2^{-d}$ into n unit d-dimensional hypercubes is considered in Zielonka (2016).

We focus on the problem of online packing of d-dimensional hyperboxes into one active bin. The paper contains two algorithms $D_1(d)$ and $D_2(d)$: the first method is a 3.5^d-competitive algorithm and for $d < 17$ works better than the second algorithm having the $12 \cdot 3^d$ competitive ratio, which is a significant improvement of the ratio 4^d from Zhang et al. (2013). Both algorithms are defined inductively from lower dimensions to higher. The inductive step goes two dimensions back and thus the core of the algorithm is the method of packing rectangles on the front wall of the unit hypercube. Since in three-dimensions this looks a lot like drawers we decided to name it: *the drawer method*. As a base, for $d = 1$ both algorithms take the Next Fit algorithm and for $d = 2$ the 1-space bounded algorithm from Zhang et al. (2014).

2 Intuitions on how algorithms work

We give several examples of packing some 'easy' items to introduce the reader to general rules of packing used by both algorithms. Since it is difficult to handle items of completely arbitrary size, we decided to assign two-dimensional items into categories, called λ-rectangles (and their analogue r-rectangles for the second algorithm) depending on lengths of their edges. λ-rectangles also differ in size, however there are only countably many of them.

Each drawer is a hyperbox with $d-2$ edges of length 1 and two edges forming a λ-rectangle. Thus to define a drawer we only need to provide the size of the front wall, i.e., lengths of the two smallest edges of the hyperbox. An algorithm for choosing the right place for a drawer is described in Sect. 3: Λ_1-method for packing λ-rectangles.



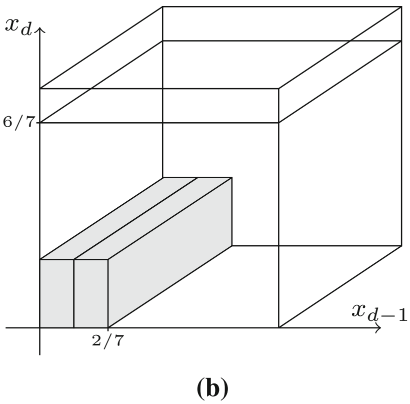


Fig. 1 The packing of the first two items. **a** A bin with the first packed item. **b** A bin with two packed items

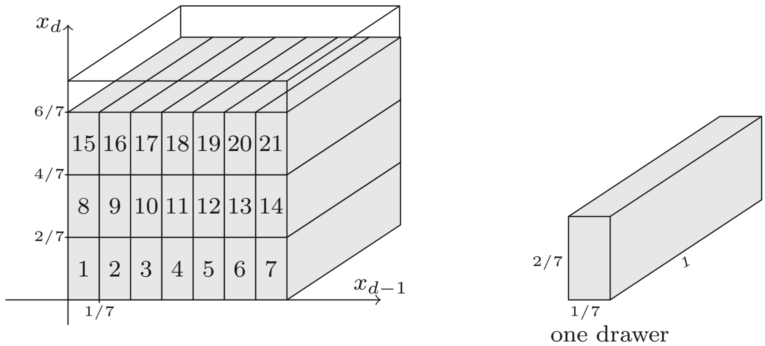


Fig. 2 21 hyperboxes $1^{d-2} \times 1/7 \times 2/7$

During packing, there are no empty drawers in the bin (a hypercube of edge 1). A new drawer is opened only when an item arrives and there is no place to pack it in matching drawers opened earlier. Two smallest edges of the item are taken into account and a drawer with appropriate front wall is created. The item is packed into the new drawer immediately. The rest of the bin is not divided into drawers until a need for a specific drawer occurs.

In the following examples $\lambda = 2/7$. We chose such λ to balance the average packing ratio of big and small (see Sect. 4 for the definitions) hyperboxes for the $D_1(d)$-algorithm.

In Examples 1–3 each drawer is entirely packed because the items are as big as drawers. Examples 4–6 could picture an actual situation where incoming items are of arbitrary size and do not fill entire drawers.

Example 1 The packing of 21 congruent hyperboxes $1^{d-2} \times 1/7 \times 2/7$, see Figs. 1 and 2. Front walls of the hyperboxes are λ-rectangles $1/7 \times 2/7$. Upon arrival of each hyperbox a new drawer of opened. Each item is packed into an individual drawer.

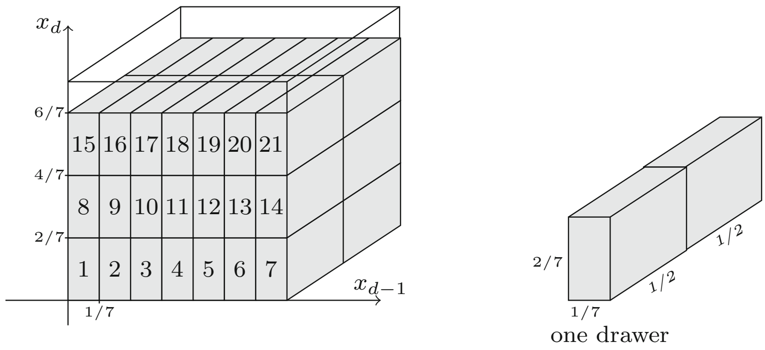


Fig. 3 42 hyperboxes $1^{d-3} \times 1/2 \times 1/7 \times 2/7$

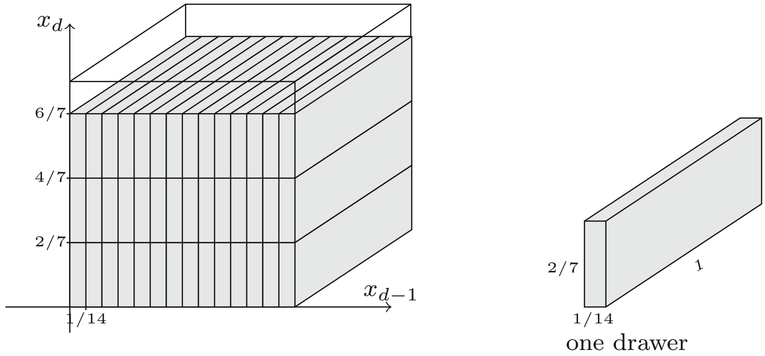


Fig. 4 42 hyperboxes $1^{d-2} \times 1/14 \times 2/7$

Example 2 The packing of 42 congruent hyperboxes $1^{d-3} \times 1/2 \times 1/7 \times 2/7$, see Fig. 3. Each drawer is packed with two items.

Example 3 The packing of 42 congruent hyperboxes $1^{d-2} \times 1/14 \times 2/7$, see Fig. 4. Front walls of the hyperboxes are λ-rectangles. For each item an appropriate drawer is created and the item is packed into it.

The following examples are for $d = 3$.

Example 4 The packing of 21 congruent boxes $0.6 \times 0.1 \times 0.2$, see Fig. 5. In this example lengths of edges of the front wall of each box satisfy: $1/14 < 0.1 < 1/7$ and $1/7 < 0.2 < 2/7$. If we let $H = 0.6 \times 0.1 \times 0.2$, then the smallest λ-rectangle $P(H)$ containing the front wall of H (see Sect. 4 for precise definitions) is $1/7 \times 2/7$. For each item a new drawer with front wall $P(H)$ is opened. Each item is packed into an individual drawer.

Example 5 The packing of 42 congruent boxes $0.4 \times 0.1 \times 0.2$, see Fig. 6. This example is a mix of Examples 2 and 4. If we let $H = 0.4 \times 0.1 \times 0.2$, then the smallest λ-rectangle $P(H)$ containing the front wall of H (see Sect. 4 for precise definitions) is

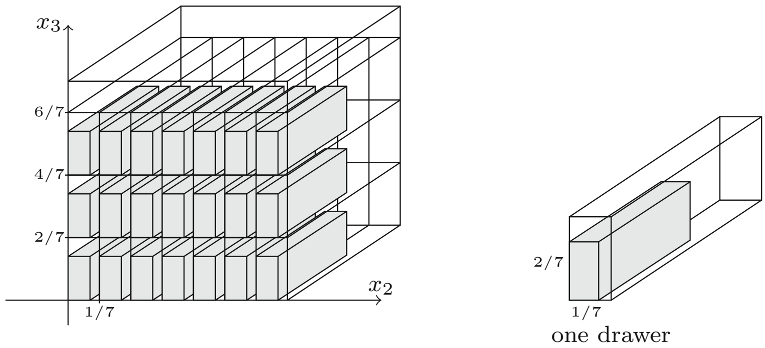


Fig. 5 21 boxes $0.6 \times 0.1 \times 0.2$

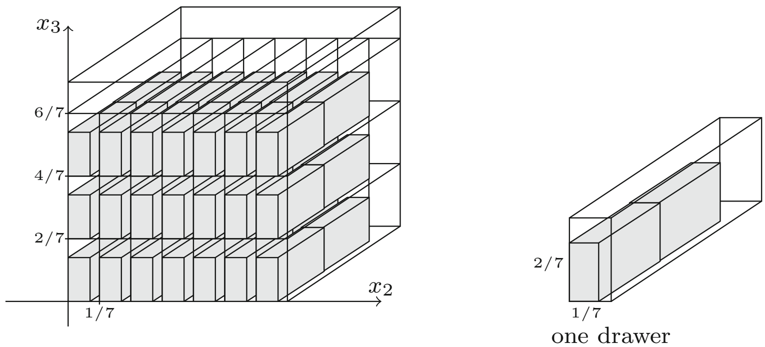


Fig. 6 42 boxes $0.4 \times 0.1 \times 0.2$

$1/7 \times 2/7$. Items are packed into drawers $1 \times 1/7 \times 2/7$. It is possible to fit two items in one drawer, thus each drawer is packed with two boxes $0.4 \times 0.1 \times 0.2$.

Example 6 The packing of 56 boxes: 26 boxes $0.45 \times 0.1 \times 0.25$ (we name them *thick*) and 30 boxes $0.35 \times 0.06 \times 0.2$ (we name them *slim*), see Fig. 7. The order of arrival of items is the following: 8 thick boxes, 24 slim boxes, 14 thick boxes, 6 slim boxes and 4 thick boxes.

Two types of drawers were created. Since $1/7 < 0.25 < 2/7$ and $1/14 < 0.1 < 1/7$ each thick box is packed into a drawer with the front wall $1/7 \times 2/7$, two thick boxes per drawer. Since $1/7 < 0.2 < 2/7$ and $1/28 < 0.06 < 1/14$ each slim box is packed into a drawer with the front wall $1/14 \times 2/7$, again two slim boxes per drawer.

3 Λ_1-method for packing λ-rectangles

Let $\lambda > 0$ and let k be a non-negative integer. A λ_k-*unit* is a rectangle with height $\lambda/2^k$ and width $\lambda/2^{k+1}$. A *basic unit* is λ_0-unit. See Fig. 8.

For basic units we consider rectangles with side ratio 1:2. It makes possible to pack a square in a union of two such rectangles. During packing basic units are divided into

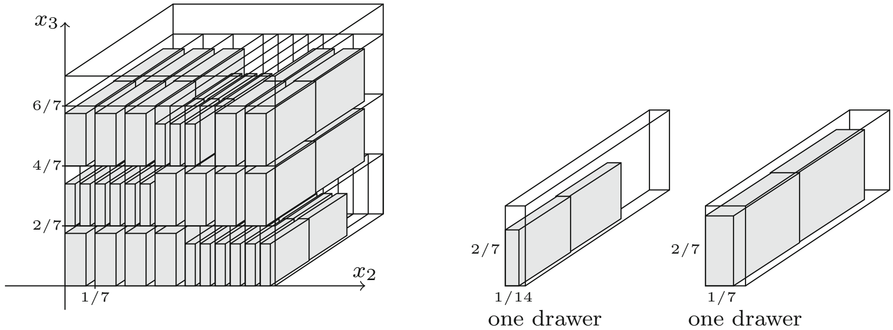


Fig. 7 26 boxes 0.45 × 0.1 × 0.25 and 30 boxes 0.35 × 0.06 × 0.2

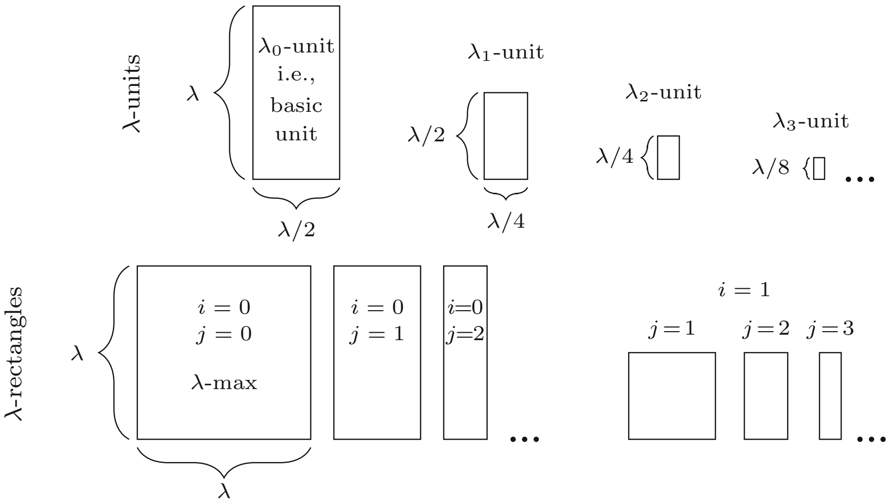


Fig. 8 λ-units and λ-rectangles

smaller rectangles. Every division creates four congruent rectangles, each one being similar to a basic unit.

Consider a square $I = 1 \times 1$. For $\epsilon > 0$ the square I is divided into two rectangles $I_\epsilon = 1 \times (1 - \epsilon)$ and $T_\epsilon = 1 \times \epsilon$. The rectangle T_ϵ will be used for packing big items, while I_ϵ—for small items (definitions can be found in Sect. 4).

Small items are packed into basic units $\lambda/2 \times \lambda$ and to obtain high packing ratio, we wish to fit in I_ϵ as many basic units as possible. Of course, we also want to divide the whole I_ϵ, therefore the widths of basic units in one row must sum up to 1. For $D_1(d)$-algorithm the following values are sufficient: $\epsilon = 1/7$ and $\lambda = 2/7$. The rectangle I_ϵ (called B_1 in this case) with edges $1 \times (1 - 1/7) = 3.5\lambda \times 3\lambda$ is divided into 21 basic units $\lambda/2 \times \lambda$.

Suppose we pack rectangles $R_1, R_2, \ldots$ with side lengths smaller than or equal to 2/7 into B_1. For each R_i we find the smallest λ-rectangle containing R_i (the area of

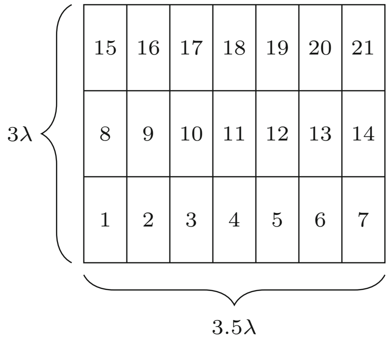


Fig. 9 The rectangle B_1

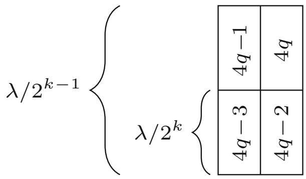


Fig. 10 The division of a λ_{k-1}-unit into four λ_k-units

this λ-rectangle is smaller than the area of four R_i's). Then this λ-rectangle (along with R_i inside) is packed into B_1.

In the next section we will pack d-dimensional hyperboxes into d-dimensional drawers. The first stage is finding the proper drawer: since we consider only front walls (of the hyperbox and of the drawer) it is exactly the same as packing rectangles with side lengths smaller or equal to 2/7 into B_1. That is why the method of packing λ-rectangles into B_1 is crucial in $D_1(d)$-algorithm.

We decided to describe the general case with an arbitrary λ, since some of the reasoning is used again in Sect. 5 for $\lambda = 1/3$.

Denote by B_1 a rectangle $3.5\lambda \times 3\lambda$ divided into twenty one basic units numbered with natural numbers in the order showed on Fig. 9. During the packing process basic units will be divided into smaller units. When a λ_{k-1}-unit (for $k \geq 1$) numbered with q is partitioned into four λ_k-units, these λ_k-units are numbered from $4q-3$ to $4q$ as on Fig. 10.

Let *λ-rectangle* be a rectangle of width $\lambda_j = \lambda/2^j$ and height $\lambda_i = \lambda/2^i$ for some $0 \leq i \leq j$. *λ-max* is a square of side length λ (see Fig. 8). A unit is called *empty*, if its interior has an empty intersection with any packed λ-rectangle.

Λ_1-method of packing λ-*rectangles into* $\mathbf{B_1}$

1. A λ-rectangle of height λ and width less then λ is packed as much to the left as possible into the lowest indexed basic unit in B_1 that has enough empty space.
2. λ-max is packed into the union of two consecutive, empty, lowest indexed basic units. Clearly, λ-max cannot be packed into units 7 and 8 or into units 14 and 15.
3. A λ-rectangle of height $\lambda_i = \lambda/2^i$, $i \geq 1$ that is not a square is packed as much to the left as possible into the lowest indexed λ_i-unit and, obviously, with enough empty space.

   If there is no such unit, find the greatest $k \leq i$ such that there is an empty λ_k-unit.

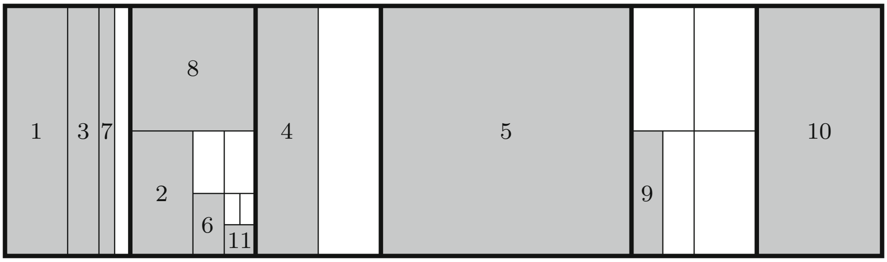


Fig. 11 Packing items into basic units 1–7 by Λ_1-method. Numbers indicate the order in which the items arrive

From among empty λ_k-units choose the lowest indexed one and then divide it into four λ_{k+1}-units. If $\lambda_{k+1} > \lambda_i$, then the lowest indexed λ_{k+1}-unit is divided again into four smaller units. The division is repeated until a λ_i-unit is created. Now the λ-rectangle is packed into the lowest indexed λ_i-unit as much to the left as possible.

4. A λ-rectangle that is a square of side length λ_i, $i \geq 1$, is packed, if possible, into two empty λ_i-units, that were created through the division of one λ_{i-1}-unit. If there are no such units the division of a bigger unit is conducted as described in the previous case. Finally a λ-rectangle is packed into two lowest indexed λ_i-units (such that the union of these units is a square).

Example 7 Figure 11 illustrates Λ_1-method. The first λ-rectangle, by Rule 1, is packed into the first basic unit as much to the left as possible. The second λ-rectangle is packed by Rule 3: it must be packed into a λ_1-unit. These units are created through the division, see Fig. 10, solely from empty units, thus we cannot take the first basic unit. The second basic unit is divided and the second item is packed into it. The third item is packed by Rule 1 and so is the fourth: we look for a basic unit with enough empty space. The fifth item is λ-max, therefore by Rule 2 it is packed into the union of two consecutive, empty, lowest indexed basic units. To pack the sixth item we use Rule 3: a division of an empty, lowest indexed (which would be 6 in this example) λ_1-unit is conducted and the item is packed. The seventh item can be fitted into the first basic unit (Rule 1). The eighth λ-rectangle is packed by Rule 4 into two empty λ_1-units contained in the second basic unit. Since after this packing there is no empty λ_1-unit left to pack the ninth item, we perform a new division (Rule 3). Enough empty space for the tenth item is only in a new, empty basic unit. The last, eleventh item is packed by Rule 4 into freshly created λ_3-units from the lowest indexed λ_2-unit.

λ-rectangles shown on Fig. 11 can be front walls of drawers with $(d-2)$ edges of length 1 (see Sect. 4).

Lemma 1 *Let $k \geq 3$ and let a λ-rectangle of width smaller then λ be packed into the unit number k in B_1. The empty space in units numbered from 1 to k is smaller than $7\lambda^2/6$.*

Proof First, we will show that the empty space in all units that are partially packed is smaller than $\frac{2}{3}\lambda^2$.

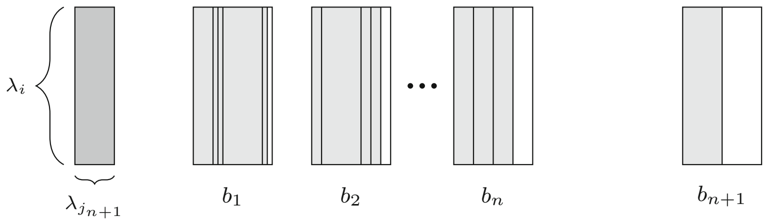


Fig. 12 Units of height λ_i

Let $i \geq 0$. Consider all units of height λ_i (and width $\frac{1}{2}\lambda_i$) into which an item of height λ_i was packed. Let b_{n+1} be the last such unit. From among remaining units (different from b_{n+1}) we choose units $b_1, b_2, \ldots, b_n$ that are not entirely packed (see Fig. 12). Let R_{j_i} be an item with the smallest width λ_{j_i} (and of height λ_i) packed into b_i for $i = 1, \ldots, n+1$. The width of empty space in b_n is smaller than $\lambda_{j_{n+1}}$, otherwise $R_{j_{n+1}}$ can be packed into this unit. This implies that $\lambda_{j_n} < \lambda_{j_{n+1}}$ (a rectangle of width smaller than $\lambda_{j_{n+1}}$ was packed into b_n). Since the width of empty space in b_n must be a multiple of λ_{j_n} (the width of b_n as well as the width of each item packed into b_n is a multiple of λ_{j_n}), it follows that the width of empty space in b_n is not greater than $\lambda_{j_{n+1}} - \lambda_{j_n}$. For the same reason the width of empty space in b_{n-1} is not greater than $\lambda_{j_n} - \lambda_{j_{n-1}}$. Repeating this argument, we get that the sum of widths of empty space in all $b_1, b_2, \ldots, b_n$ is not greater than

$$\lambda_{j_{n+1}} - \lambda_{j_n} + \lambda_{j_n} - \lambda_{j_{n-1}} + \cdots + \lambda_{j_2} - \lambda_{j_1} < \lambda_{j_{n+1}}.$$

The empty space in b_{n+1} is not greater than the area of this unit ($\lambda_i^2/2$) minus the area of $R_{j_{n+1}}$. Consequently, the empty space in all units $b_1, \ldots, b_{n+1}$ (of height λ_i) is less than

$$\lambda_{j_{n+1}} \cdot \lambda_i + \frac{1}{2}\lambda_i^2 - \lambda_{j_{n+1}} \cdot \lambda_i = \frac{1}{2}\lambda_i^2.$$

Finally, the empty space in partially packed units of all heights is not greater than

$$\sum_{i \geq 0} \frac{1}{2}\lambda_i^2 = \frac{1}{2} \cdot \lambda^2 \sum_{i \geq 0} \frac{1}{4^i} = \frac{1}{2}\lambda^2 \cdot \frac{4}{3} = \frac{2}{3}\lambda^2.$$

Entirely packed units do not add anything to the empty space, therefore they are omitted.

Now, we will calculate the number of empty units of all sizes. There is no empty basic unit, otherwise it would be used for packing. Smaller units are created during the division to pack smaller items. At each step four units are created, but at least one unit is immediately used either for the next division or for packing. Thus at most 3 units of height smaller than λ can be empty. This gives

$$3 \cdot \sum_{i \geq 1} \frac{1}{2}\lambda_i^2 = \frac{3}{2}\lambda^2 \cdot \sum_{i \geq 1} \frac{1}{4^i} = \frac{1}{2}\lambda^2.$$

The empty space in all basic units numbered from 1 to k is less than

$$\frac{2}{3}\lambda^2 + \frac{1}{2}\lambda^2 = \frac{7}{6}\lambda^2.$$

□

Lemma 2 *Let R be λ-max. If R is packed into B_1 into the union of two consecutive units numbered $k-1$ and k, where $2 \leq k \leq 21$ and, obviously, $k \neq 8$ as well as $k \neq 15$, then the empty space in units 1 to k is not greater than*

- $13\lambda^2/6$, *for* $k \in \{16, 18, 19, 20, 21\}$
- $5\lambda^2/3$, *for* $k \in \{9, 11, 13\}$,
- $7\lambda^2/6$, *for* $k \in \{2, 3, 4, 5, 6, 7, 10, 12, 14, 17\}$.

Moreover, if there is not enough space to pack λ-max into B_1, then the empty space in B_1 is smaller than $8\lambda^2/3$.

Proof **Bottom row,** $2 \leq k \leq 7$
If R is packed in the bottom row, there are no empty basic units preceding R, thus by Lemma 1 we get that the empty space is not greater than $7\lambda^2/6$.

Middle row, $9 \leq k \leq 14$

Case M1 R is packed into units 8–9. Use Lemma 1 or the case above to show that when unit 7 is

- occupied by a λ-rectangle, then the empty space is at most $7\lambda^2/6$,
- empty, then unit 6 is not empty (otherwise R can be packed into the union of two consecutive units 6 and 7). Empty space is at most

$$\frac{7}{6}\lambda^2 + \frac{1}{2}\lambda^2 = \frac{5}{3}\lambda^2.$$

Case M2 R is packed into units 9–10. Unit 8 contains a smaller λ-rectangle, thus by Lemma 1 the empty space in units 1 to 10 is not greater than $7\lambda^2/6$.

Case M3 R is packed into units 10–11 *or* 12–13. If R is packed into units 12–13 and unit 11 does not contain λ-max, then the empty space is smaller then $7\lambda^2/6$. If units 10–11 contain λ-max , then consider the contents of unit 9. If there is a smaller λ-rectangle use Lemma 1 ($7\lambda^2/6$ of the empty space), otherwise use Case M1. The empty space does not exceed $5\lambda^2/3$.

Case M4 R is packed into units 11–12 *or* 13–14. Use the similar reasoning as in the Case M3. By Lemma 1 and Case M2 the empty space does not exceed $7\lambda^2/6$.

Upper row, $k \geq 16$

Case U1 λ-max is packed into units 15–16.

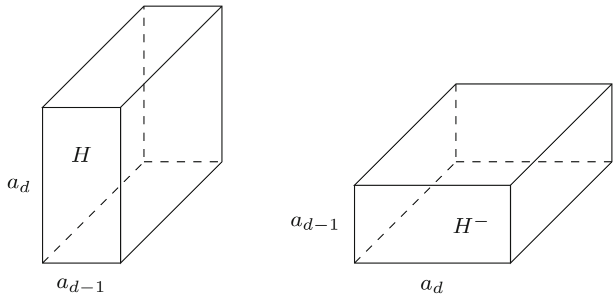


Fig. 13 H and H^-

- If unit 14 contains a λ-rectangle, then by Lemma 1 or Case M4 there is at most $7\lambda^2/6$ empty space.
- If unit 14 is empty, then unit 13 is not empty. It contains either λ-max (Case M3) or a smaller λ-rectangle. Consequently, the empty space is at most $5\lambda^2/3 + \lambda^2/2 = 13\lambda^2/6$.

Case U2 λ-max is packed into units 16–17. Unit 15 must be packed with a smaller λ-rectangle. By Lemma 1 the empty space is less than $7\lambda^2/6$.

Case U3 λ-max is packed into two consecutive units from 17–21. Use Lemma 1 or Case U1 or Case U2. The empty space is less than $13\lambda^2/6$.

Case E: There is not enough space to pack λ-max into B_1. The worst case is when unit 21 is empty. Empty space in B_1 does not exceed $13\lambda^2/6 + \lambda^2/2 = 8\lambda^2/3$. □

4 First drawer algorithm

Let $\lambda = 2/7$. In this section a drawer algorithm $D_1(d)$ with the asymptotic competitive ratio not greater than 3.5^d is presented.

The value of λ was chosen to ensure that the average packing ratios of big and small items were similar to each other and were both not smaller than $(2/7)^d$.

The ratio of packing big items is of the form $c \cdot \lambda^d$, thus smaller λ gives smaller packing ratio. On the other hand, if λ is greater than 2/7, then I_ϵ contains less basic units. Moreover the volume of open drawers (in the proof of Theorem 1 it is showed to be bounded from above by $8\lambda^2/3$) and the empty space (see Lemmas 1 and 2) increases. The average occupation in closed drawers would be smaller and consequently the packing ratio of small items is less than $(2/7)^d$.

Each hyperbox H is rotated to satisfy

$$a_1 \geq \cdots \geq a_{d-2} \geq a_d \geq a_{d-1},$$

where a_j is the length of the jth edge of H (see Fig. 13, left).

If W is a hyperbox $[v_1, w_1] \times \cdots \times [v_d, w_d]$, i.e., $W = \{(x_1, \ldots, x_d) : v_1 \leq x_1 \leq w_1, \ldots, v_d \leq x_d \leq w_d\}$, then by the *front wall* of W we mean the set of its points

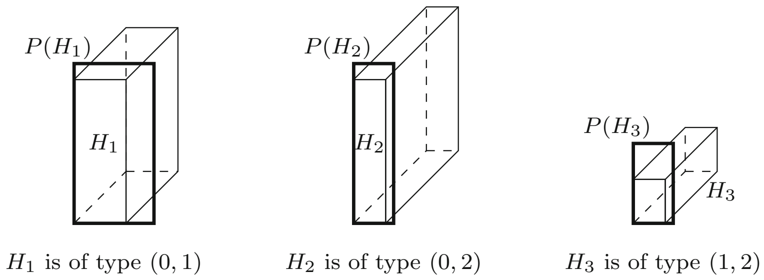


Fig. 14 Hyperboxes H_i and their λ-rectangles $P(H_i)$

with $x_1 = v_1, \ldots, x_{d-2} = v_{d-2}$. Without loss of generality we can assume that the active bin is $[0, 1]^d$. Let $B_1(d) = [0, 1]^{d-1} \times [0, 6/7]$. Obviously, the *front wall* of $B_1(d)$ is the set of its points with $x_1 = \cdots = x_{d-2} = 0$.

A hyperbox with $a_d > \lambda$ is called *big*, otherwise it is called *small*. Note that big hyperboxes fulfill $a_1 \geq \cdots \geq a_{d-2} \geq a_d > \lambda$, while small hyperboxes satisfy $a_{d-1} \leq a_d \leq \lambda$.

For each small hyperbox H, let $P(H)$ be the smallest λ-rectangle containing the front wall of H. There are integers i and j such that the height of $P(H)$ equals $\lambda/2^i$, the width of $P(H)$ equals $\lambda/2^j$ and moreover $\lambda/2^{i+1} < a_d \leq \lambda/2^i$ and $\lambda/2^{j+1} < a_{d-1} \leq \lambda/2^j$, see Fig. 14. We say then that H is *of type* (i, j).

Using the division of B_1, the hyperbox $B_1(d)$ can be divided into 21 hyperboxes $1 \times \cdots \times 1 \times 1/7 \times 2/7$ that will be called *basic drawers* or *drawers* (see Fig. 17). During the packing process each drawer can be divided into smaller *drawers*, i.e., hyperboxes of edges $1 \times \cdots \times 1 \times \lambda/2^j \times \lambda/2^i$ called (i, j)-drawers. As described above, the front wall of any (i, j)-drawer is a rectangle $\lambda/2^j \times \lambda/2^i$ and B_1 is the front wall of $B_1(d)$.

Drawers are created for packing items: there can be drawers of different sizes and many drawers of the same size, however at most one drawer of a fixed size is open at each stage of the packing process. Moreover, any two (open or closed) drawers have disjoint interiors. For example, if we treat the numbered rectangles on Fig. 11 as the front walls of all drawers used for the packing, then drawers with numbers $2, \ldots, 10$ are open while the drawer with number 1 must be closed (its size is equal to the size of the drawer with number 4). Let us add that the proper drawer for a hyperbox with $1/7 < a_{d-1} \leq a_d \leq 2/7$ is the union of two adjacent basic drawers (see 5 on Fig. 11).

Since the $D_1(d)$-algorithm is defined inductively two dimensions back, for $d = 1$ and $d = 2$ we use some already known algorithms. The exact method of packing is not crucial, we only need to achieve a certain average occupation. For $d = 1$, we chose the Next Fit algorithm as the $D_1(1)$-algorithm, i.e., we pack any item into the active bin as much to the left as it is possible. Clearly, the average occupation in each bin is greater than 1/2, however any other method with the average occupation greater than 1/3.5 is also suitable. For $d = 2$, as $D_1(2)$-algorithm we use the method described in Zhang et al. (2014) (average occupation greater than 0.197). Again, the reader can take any other 1-space bounded packing algorithm with the average occupation greater than $1/(3.5)^2 \approx 0.0816$.

Assume that $d \geq 3$.

If $H = a_1 \times \cdots \times a_{d-2} \times a_{d-1} \times a_d$ is a hyperbox such that $a_1 \geq \cdots \geq a_{d-2} \geq a_d \geq a_{d-1}$, then $H^- = a_1 \times \cdots \times a_{d-2} \times a_d \times a_{d-1}$ is the image of H in rotation of 90° on the plane $x_{d-1}x_d$ (see Fig. 13). By the height h of H^- we mean a_{d-1}. Clearly, $h = \min(a_1, \ldots, a_d)$ and h is the length of the edge of H^- parallel to the x_d-axis.

If W is a hyperbox $[v_1, w_1] \times \cdots \times [v_d, w_d]$, then by the *the top* of W we mean the set of its points with $x_d = w_d$. The $(d-2)$*-dimensional bottom* of W is the set of its points with $x_d = v_d$ and $x_{d-1} = v_{d-1}$. We say that W is packed *along the right edge* of the bin $[0, 1]^d$ provided W is contained in the bin and there is $p_d \in [0, 1 - w_d + v_d]$ such that W contains the segment with endpoints $(0, \ldots, 0, 1, p_d)$ and $(0, \ldots, 0, 1, p_d + w_d - v_d)$.

$\mathbf{D_1(d)}$-algorithm of packing a hyperbox H

- If H is big, it is rotated to obtain H^-. Then H^- is packed into the active bin from top to bottom along the right edge of the bin (see Fig. 17) provided the interior of the packed hyperbox is disjoint with those basic drawers that already contain small hypercubes. If that is not possible, we close the active bin and open a new active bin to pack H.
- If H is a small hyperbox of type (i, j), then
  - it is packed into the open (i, j)-drawer in such a way that the $(d-2)$-dimensional bottom of H is packed into the $(d-2)$-dimensional bottom of the drawer (i.e., is packed into the $(d-2)$-dimensional unit hypercube) using the $D_1(d-2)$-algorithm.
  - If there is not enough empty space in the open (i, j)-drawer to pack H, close this drawer. A new drawer is opened in the following way. First, determine the proper place $Q \subset B_1$ to pack the rectangle $P(H)$ into B_1 by Λ_1-method.
    - If the drawer with the front wall equal to Q is disjoint with the interior of any packed big hypercube, then the new open (i, j)-drawer is the one with the front wall equal to Q. The rectangle $Q \subset B_1$ is treated as a λ-rectangle packed into B_1 in Λ_1-method. The hyperbox H is packed into this open (i, j)-drawer in such a way that the $(d-2)$-dimensional bottom of H is packed into the $(d-2)$-dimensional bottom of the drawer by using the $D_1(d-2)$-algorithm.
    - Otherwise we close the active bin and open a new active bin to pack H.

Three examples presented below illustrate the packing method.

Example 8 A list of 3-dimensional boxes $H_1 = (1/7 + \epsilon, 1/14 + \epsilon, 1/7 + \epsilon)$, $H_2 = (1, \epsilon, 1/7 + \epsilon)$, $H_3 = (1/7 + 2\epsilon, 1/14 + 2\epsilon, 1/7 + 2\epsilon)$, $H_4 = (1, 2\epsilon, 1/7 + 2\epsilon), \ldots, H_{240}$, for sufficiently small $\epsilon > 0$ is packed as shown on Figs. 15 and 16 (on these figures ϵ equals 1/70).

The front wall of H_1 is a rectangle $(1/14 + \epsilon) \times (1/7 + \epsilon)$ and $F_1 = P(H_1) = 1/7 \times 2/7$ is the smallest λ-rectangle containing this rectangle. We open a drawer with the front wall F_1, i.e., the first basic drawer. H_1 is packed into this drawer in the place $[0, 1/7 + \epsilon] \times [0, 1/14 + \epsilon] \times [0, 1/7 + \epsilon]$.

Let j be the greatest integer such that $\epsilon \leq \lambda/2^j$ and let $\zeta = \lambda/2^j$. The front wall of H_2 is a rectangle $\epsilon \times (1/7 + \epsilon)$ and $F_2 = P(H_2) = \zeta \times 2/7$ is the smallest

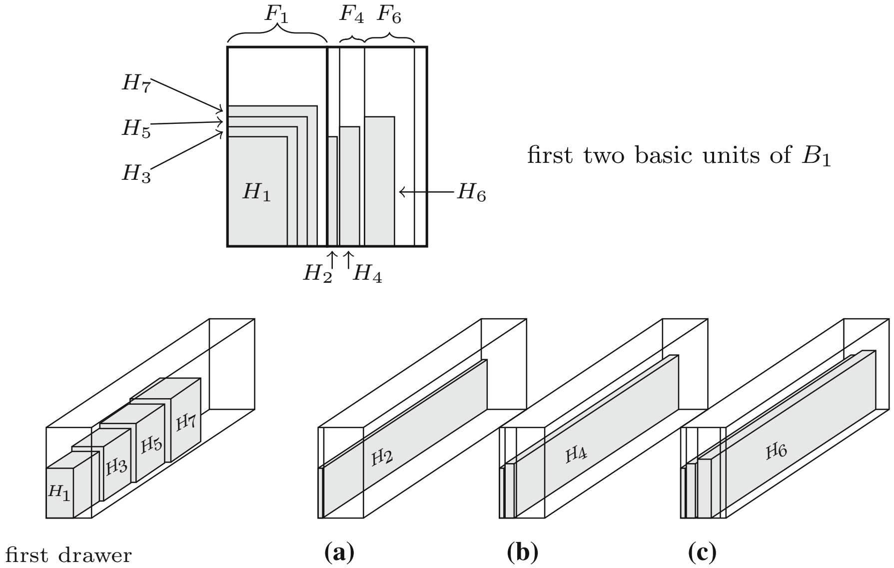


Fig. 15 Example 8, first drawer after packing H_1, H_3, H_5 and H_7 and next three drawers created in the second basic drawer for packing: **a** H_2, **b** H_4, **c** H_6

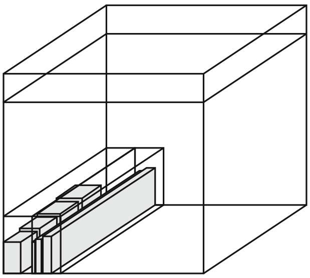


Fig. 16 Example 8, the bin

λ-rectangle containing this rectangle. Since there is no open drawer with front wall F_2, we open a new drawer $[0, 1] \times [1/7, 1/7 + \zeta] \times [0, 2/7]$ and pack H_2 in the place $[0, 1] \times [1/7, 1/7 + \epsilon] \times [0, 1/7 + \epsilon]$.

H_3 is packed similarly to H_1. The front wall of H_3 is a rectangle $(1/14 + 2\epsilon) \times (1/7 + 2\epsilon)$ and $F_3 = P(H_3) = 1/7 \times 2/7$. There is an open drawer with the front wall $F_3 = F_1$ (the first basic unit) with enough empty space, so we pack H_3 in the place $[1/7 + \epsilon, 2/7 + 3\epsilon] \times [0, 1/14 + 2\epsilon] \times [0, 1/7 + 2\epsilon]$.

The front wall of H_4 is a rectangle $2\epsilon \times (1/7+2\epsilon)$ and $F_4 = P(H_4) = 2\zeta \times 2/7$ is the smallest λ-rectangle containing this rectangle. There is no open drawer with front wall F_4, therefore we open such drawer $[0, 1] \times [1/7 + \zeta, 1/7 + 3\zeta] \times [0, 2/7]$ and pack H_4 in the place $[0, 1] \times [1/7 + \zeta, 1/7 + \zeta + 2\epsilon] \times [0, 1/7 + \epsilon]$. We continue to pack odd items into the first drawer as long as it is possible. Then a new drawer will be opened for them. Each even item is packed into an individual drawer of height λ.

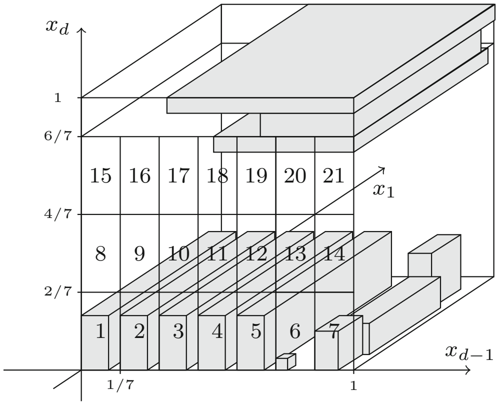


Fig. 17 Drawers in $B_1(d)$, $d = 3$

Note that ϵ is relatively large on Fig. 15. However, if ϵ is sufficiently small, for example smaller than 10^{-6}, then $H_2, H_4, \ldots, H_{240}$ are packed into individual drawers of height λ created in the second basic drawer. Six items $H_1, H_3, H_5, H_7, H_9, H_{11}$ are packed into the first basic drawer. Moreover, we pack six items $H_{12k-23}, H_{12k-21}, \ldots, H_{12k-13}$ into the kth basic drawer, for $k = 3, 4, \ldots, 21$. Clearly, $H_{241} = H_{12\cdot 22-23}$ is the first item that cannot be packed into the active bin by $D_1(d)$-algorithm.

Example 9 The following example is a precise description of the packed items showed on Fig. 17. Let $d = 3$ and let $H_1 = \cdots = H_5 = 0.9 \times 0.1 \times 0.2$, $H_6 = 0.05 \times 0.05 \times 0.05$, $H_7 = 0.17 \times 0.1 \times 0.17$, $H_8 = 0.49 \times 0.13 \times 0.13$, $H_9 = 0.2 \times 0.1 \times 0.23$, $H_{10} = 1 \times 0.6 \times 0.05$, $H_{11} = 0.9 \times 0.34 \times 0.08$ and $H_{12} = 0.95 \times 0.51 \times 0.05$.

For $i = 1, \ldots, 5$, each item H_i is packed into the $(0, 1)$-drawer (i.e., the basic drawer) number i in the place $[0, 0.9] \times [(i-1)/7, (i-1)/7+0.1] \times [0, 0.2]$. The item H_6 is packed into the drawer $1 \times 1/14 \times 1/14$ with the front wall contained in the unit number 6 in the place: $[0, 0.05] \times [5/7, 5/7+0.05] \times [0, 0.05]$, H_7 is packed into the drawer with the front wall contained in the unit number 7 so that its $(d-2)$-dimensional bottom (the segment of length 0.17) is packed into the $(d-2)$-dimensional bottom of the drawer (the segment of length 1) by the $D_1(1)$ method, i.e., we pack H_7 in the place $[0, 0.17] \times [6/7, 6/7+0.1] \times [0, 0.17]$. Since $P(H_8) \neq P(H_6)$, the item H_8 cannot be packed in one drawer with H_6. Notice that $P(H_8) = P(H_7)$ as well as $P(H_9) = P(H_7)$ and since there is enough space in drawer with the front wall contained in the unit number 7 we do not open a new drawer for H_8 or H_9. According to the Next Fit algorithm H_8 is packed in $[0.17, 0.17 + 0.49] \times [6/7, 6/7 + 0.13] \times [0, 0.13]$ and H_9 in $[0.66, 0.66 + 0.2] \times [6/7, 6/7 + 0.1] \times [0, 0.23]$. The big item H_{10} is packed in $[0, 1] \times [1-0.6, 1] \times [1-0.05, 1]$ (along the right edge of the bin, i.e., the segment with endpoints $(0, 1, 0.95)$ and $(0, 1, 1)$ is contained in this box), H_{11} in $[0, 0.9] \times [1-0.34, 1] \times [0.95-0.08, 0.95]$ and H_{12} in $[0, 0.95] \times [1-0.51, 1] \times [0.87-0.05, 0.87]$.

Example 10 Assume that $d = 5$ and that the $(d-2)$-dimensional bottoms of hyperboxes U_i (for $i = 1, \dots 12$) are boxes H_i described in Example 9. Let the length of the 4th and the 5th edge of each U_i be equal to 0.01, i.e., $U_1 = 0.9 \times 0.1 \times 0.2 \times 0.01 \times 0.01, \dots, U_{12} = 0.95 \times 0.51 \times 0.05 \times 0.01 \times 0.01$.

Since $0.01 < 2/7$ all these hyperboxes are small items. Let U_{13} be a big item of the size $0.4 \times 0.4 \times 0.4 \times 0.4 \times 0.1$.

All small items $U_1, \dots, U_{12}$ are packed into the first drawer $1 \times 1 \times 1 \times 1/56 \times 1/56$ with the front wall contained in the first unit so that the $(d-2)$-dimensional bottoms of U_i (the boxes H_i) are packed into the $(d-2)$-dimensional bottom of the drawer (the unit cube) as in Example 9. For instance, U_8 is packed in the place $[0.17, 0.66] \times [6/7, 6/7 + 0.13] \times [0, 0.13] \times [0, 0.01] \times [0, 0.01]$, U_{11} is packed in $[0, 0.9] \times [0.66, 1] \times [0.87, 0.95] \times [0, 0.01] \times [0, 0.01]$. The only big item U_{13} is packed in $[0, 0.4] \times [0, 0.4] \times [0, 0.4] \times [0.6, 1] \times [0.9, 1]$ along the right edge of the bin, i.e., the segment with endpoints $(0, 0, 0, 1, 0.9)$ and $(0, 0, 0, 1, 1)$ is contained in U_{13}.

Theorem 1 *The asymptotic competitive ratio for the $D_1(d)$-algorithm is not greater than 3.5^d.*

Proof We show that the average occupation in each bin is greater than

$$\sigma_d = (2/7)^d.$$

The proof is inductive.

For $d = 1$ the average occupation in each bin is greater than $1/2 > 2/7$, for $d = 2$ (see Zhang et al. 2014) it is not smaller than $0.197 > (2/7)^2$.

Let $d \geq 3$. Assume that the statement holds in each dimension $n \in \{1, 2, \dots, d-1\}$.

Denote by $\sigma(l)$ the total volume of items packed into the lth bin $\mathcal{B}_l$. To prove that the average occupation in each bin is greater than σ_d it suffices to show that either $\sigma(l) > \sigma_d$ or $\sigma(l)$ plus the volume of the first item that cannot be packed into $\mathcal{B}_l$ is greater than $2\sigma_d$ (consequently, $\sigma(l) + \sigma(l+1) > 2\sigma_d$).

Consider a few cases depending on the size of the first hyperbox item

$$H_u = u_1 \times \cdots \times u_{d-1} \times u_d,$$

where $u_1 \geq \cdots \geq u_{d-2} \geq u_d \geq u_{d-1}$, that cannot be packed into $\mathcal{B}_l$. Denote by h the sum of heights of big items packed into $\mathcal{B}_l$ and let $h_u = u_{d-1}$.

By the inductive assumption (for $n = d-2$) and by the fact that the smallest λ-rectangle containing a rectangle $R = a_{d-1} \times a_d$ is of area smaller than 4 times the area of R, the average occupation in any closed (i, j)-drawer $\Lambda_{i,j}$ is greater than

$$\frac{1}{4} \cdot \sigma_{d-2} \cdot \mathrm{vol}(\Lambda_{i,j}),$$

where $\mathrm{vol}(D)$ denotes the d-dimensional volume of a drawer D. The total volume of items packed in any open drawer can be close to 0. There is at most one open

(i, j)-drawer for any integers i and j. The total volume of open drawers of height λ is smaller than $\lambda \cdot \lambda + \frac{1}{2}\lambda \cdot \lambda + \frac{1}{4}\lambda \cdot \lambda + \cdots = 2\lambda^2$. The sum of volumes of open drawers of height $\lambda/2$ is smaller than $\frac{1}{2}\lambda \cdot \frac{1}{2}\lambda + \frac{1}{4}\lambda \cdot \frac{1}{2}\lambda + \cdots = \frac{1}{2}\lambda^2$. The total volume of open drawers of height $\lambda/4$ is smaller than $\frac{1}{4}\lambda \cdot \frac{1}{4}\lambda + \frac{1}{8}\lambda \cdot \frac{1}{4}\lambda + \cdots = \frac{1}{8}\lambda^2$ and so on. Consequently, the sum of volumes of open drawers is smaller than

$$2\lambda^2 + \frac{1}{2}\lambda^2 + \frac{1}{8}\lambda^2 + \cdots = \frac{8}{3}\lambda^2.$$

Denote by $k \in \{1, \ldots, 21\}$ the greatest integer such that the interior of a drawer with the front wall contained in the kth basic unit has a non-empty intersection with a packed small item.

Case 1 H_u is small and $h \leq 1/7$.

For a moment we forget about drawers and consider only two-dimensional packing. We count the sum of areas of λ-rectangles being the front walls of all open and closed drawers. Let $F_u = P(H_u)$ be the smallest λ-rectangle containing the front wall of H_u (i.e, containing the rectangle $u_{d-1} \times u_d$). Clearly, F_u cannot be packed into B_1 by Λ_1-method. If F_u is λ-max, then by Lemma 2 the empty space in B_1 is smaller than $8\lambda^2/3$. Otherwise, $k = 21$. If a λ-rectangle of width smaller than λ was packed in the unit number 21, then by Lemma 1 the empty space in B_1 is smaller than $7\lambda^2/6$. If a rectangle λ-max was packed in the union of units with number 20 and 21, then by Lemma 2 the empty space in B_1 is smaller than $13\lambda^2/6$. This means that the total area of λ-rectangles packed into B_1, i.e., the sum of areas of front walls of all open and closed drawers, is greater than $21 \cdot \frac{1}{2}\lambda^2 - \frac{8}{3}\lambda^2 = \frac{47}{6}\lambda^2$.

Consequently, the total volume of all closed and open drawers is greater than $\frac{47}{6}\lambda^2$. Since the sum of volumes of open drawers is smaller than $8\lambda^2/3$, it follows that the total volume of closed drawers is greater than $\frac{47}{6}\lambda^2 - \frac{8}{3}\lambda^2 = \frac{31}{6}\lambda^2$. Hence, total volume of small items packed into $\mathcal{B}_l$ is greater than

$$\frac{1}{4} \cdot \sigma_{d-2} \cdot \frac{31}{6}\lambda^2 = \frac{1}{4} \cdot \left(\frac{2}{7}\right)^{d-2} \cdot \frac{31}{6} \cdot \left(\frac{2}{7}\right)^2 > \left(\frac{2}{7}\right)^d = \sigma_d.$$

Case 2 H_u is small and $1/7 < h < 2/7$. Clearly, $k \geq 13$.

The total volume of big items packed into $\mathcal{B}_l$ is greater than $h \cdot (2/7)^{d-1}$. The total volume of open and closed drawers, by Lemma 2, is greater than $13 \cdot \frac{1}{2}\lambda^2 - \frac{5}{3}\lambda^2 = \frac{29}{6}\lambda^2$. The total volume of closed drawers is greater than $\frac{29}{6}\lambda^2 - \frac{8}{3}\lambda^2 = \frac{13}{6}\lambda^2$. Consequently,

$$\sigma(l) \geq \frac{1}{4} \cdot \sigma_{d-2} \cdot \frac{13}{6}\lambda^2 + h \cdot \left(\frac{2}{7}\right)^{d-1} > \left(\frac{13}{24} + \frac{1}{7} \cdot \frac{7}{2}\right) \cdot \left(\frac{2}{7}\right)^d > \sigma_d.$$

Case 3 $h \geq 2/7$. The total volume of big items packed into $\mathcal{B}_l$ is greater than

$$h \cdot (2/7)^{d-1} \geq (2/7)^d.$$

In the next cases we will assume that $h < 2/7$ and that H_u is big.

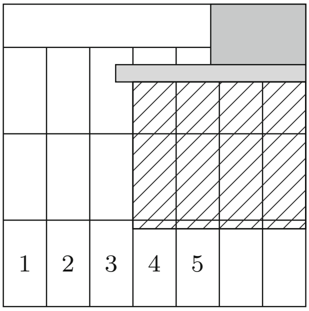


Fig. 18 Case 4

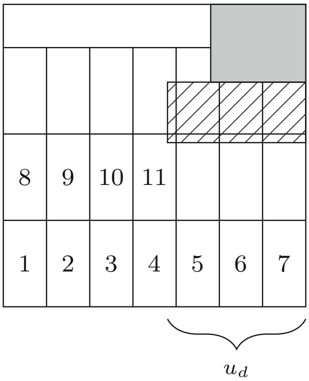


Fig. 19 Case 5, $k = 11$

Case 4 *H_u is big and $k \in \{1, \ldots, 7\}$.* Since $h < 2/7$ and $h + h_u > 5/7$ (see Fig. 18, where the front wall of the active bin and front walls of big items are shown), it follows that $h_u > 3/7$ and

$$\sigma(l) + \sigma(l+1) \geq \text{vol}(H_u) > \left(\frac{3}{7}\right)^d > 2 \cdot \left(\frac{2}{7}\right)^d = 2\sigma_d.$$

Case 5 *H_u is big and $k \in \{8, \ldots, 11\}$.* If $h + h_u > 5/7$, then we proceed as in Case 4. Otherwise, $u_d > 3/7$ and $h + h_u > 3/7$ (see Fig. 19). Since $h < 2/7$, it follows that

$$\begin{aligned}
\sigma(l) + \text{vol}(H_u) &> h\left(\frac{2}{7}\right)^{d-1} + h_u \cdot (u_d)^{d-1} > h\left(\frac{2}{7}\right)^{d-1} + \left(\frac{3}{7} - h\right) \cdot \left(\frac{3}{7}\right)^{d-1} \\
&= \left[h + \left(\frac{3}{7} - h\right)\left(\frac{3}{2}\right)^{d-1}\right] \cdot \left(\frac{2}{7}\right)^{d-1} \\
&\geq \left[h + \left(\frac{3}{7} - h\right)\left(\frac{3}{2}\right)^{2}\right] \cdot \left(\frac{2}{7}\right)^{d-1} \\
&> \left[\frac{2}{7} + \left(\frac{3}{7} - \frac{2}{7}\right) \cdot \frac{9}{4}\right] \cdot \frac{7}{2} \cdot \left(\frac{2}{7}\right)^d = \frac{17}{8} \cdot \left(\frac{2}{7}\right)^d > 2\sigma_d.
\end{aligned}$$

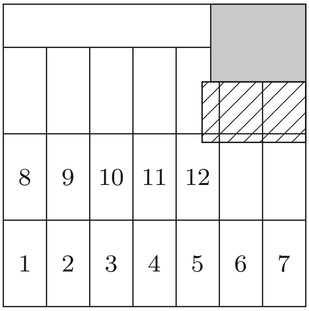


Fig. 20 Case 6, $k = 12$

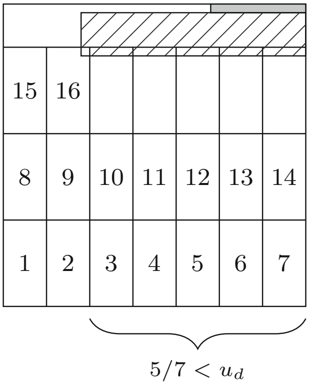


Fig. 21 Case 7, $k = 16$

Case 6 H_u is big and $k \in \{12, 13, 14\}$. If $k = 12$, then the total volume of small items in $\mathcal{B}_l$ is, by Lemma 2, greater than

$$\left(12 \cdot \frac{1}{2}\lambda^2 - \frac{7}{6}\lambda^2 - \frac{8}{3}\lambda^2\right) \cdot \frac{1}{4}\sigma_{d-2} = \frac{13}{24}\left(\frac{2}{7}\right)^d.$$

If $k \in \{13, 14\}$, then the total volume of small items in $\mathcal{B}_l$ is greater than

$$\left(13 \cdot \frac{1}{2}\lambda^2 - \frac{5}{3}\lambda^2 - \frac{8}{3}\lambda^2\right) \cdot \frac{1}{4}\sigma_{d-2} = \frac{13}{24}\left(\frac{2}{7}\right)^d.$$

Since $h + h_u > 3/7$ (see Fig. 20),

$$\sigma(l) + \text{vol}(H_u) \geq \frac{13}{24} \cdot \left(\frac{2}{7}\right)^d + \frac{3}{7} \cdot \left(\frac{2}{7}\right)^{d-1} > 2 \cdot \left(\frac{2}{7}\right)^d.$$

Case 7 H_u is big and $k \in \{15, 16\}$. In the case when $k = 15$, the total volume of small items in $\mathcal{B}_l$ is by Lemma 1 greater than

$$\left(15 \cdot \frac{1}{2}\lambda^2 - \frac{7}{6}\lambda^2 - \frac{8}{3}\lambda^2\right) \cdot \frac{1}{4}\sigma_{d-2} = \frac{11}{12}\left(\frac{2}{7}\right)^d > \frac{19}{24}\left(\frac{2}{7}\right)^d.$$

In the case when $k = 16$, the total volume of small items in $\mathcal{B}_l$ is, by Lemma 2, greater than

$$\left(16 \cdot \frac{1}{2}\lambda^2 - \frac{13}{6}\lambda^2 - \frac{8}{3}\lambda^2\right) \cdot \frac{1}{4}\sigma_{d-2} = \frac{19}{24}\left(\frac{2}{7}\right)^d.$$

If $h \geq 5/84$, then

$$\sigma(l) > \frac{19}{24}\left(\frac{2}{7}\right)^d + \frac{5}{84} \cdot \left(\frac{2}{7}\right)^{d-1} = \sigma_d.$$

If $h + h_u > 3/7$, we proceed as in Case 6. If $h + h_u \leq 3/7$ and $h < 5/84$, then

$$h_u > 1/7 - 5/84 = 1/12$$

and $u_d > 5/7$ (see Fig. 21). This implies that

$$\sigma(l) + \mathrm{vol}(H_u) > \frac{19}{24}\left(\frac{2}{7}\right)^d + \frac{1}{12} \cdot \left(\frac{5}{7}\right)^{d-1}.$$

It is easy to check that

$$\left(\frac{5}{7}\right)^{d-1} > \frac{29}{2}\left(\frac{2}{7}\right)^d$$

for $d \geq 3$. Consequently,

$$\sigma(l) + \mathrm{vol}(H_u) > \frac{19}{24}\left(\frac{2}{7}\right)^d + \frac{1}{12} \cdot \frac{29}{2}\left(\frac{2}{7}\right)^d = 2\left(\frac{2}{7}\right)^d.$$

Case 8 H_u is big and $k \geq 17$. According to Lemma 2, if $k = 17$, then the total volume of small items in $\mathcal{B}_l$ is greater than

$$\left(17 \cdot \frac{1}{2}\lambda^2 - \frac{7}{6}\lambda^2 - \frac{8}{3}\lambda^2\right) \cdot \frac{1}{4}\sigma_{d-2} = \frac{7}{6}\left(\frac{2}{7}\right)^d > \sigma_d.$$

If $k = 18$, then the total volume of small items in $\mathcal{B}_l$ is greater than

$$\left(18 \cdot \frac{1}{2}\lambda^2 - \frac{13}{6}\lambda^2 - \frac{8}{3}\lambda^2\right) \cdot \frac{1}{4}\sigma_{d-2} = \frac{25}{24}\left(\frac{2}{7}\right)^d > \sigma_d.$$

If $k \geq 19$, then the total volume of small items in $\mathcal{B}_l$ is greater than

$$\left(19 \cdot \frac{1}{2}\lambda^2 - \frac{8}{3}\lambda^2 - \frac{8}{3}\lambda^2\right) \cdot \frac{1}{4}\sigma_{d-2} = \frac{25}{24}\left(\frac{2}{7}\right)^d > \sigma_d.$$

The average occupation in each bin is greater than $(2/7)^d$. Consequently, the asymptotic competitive ratio of this packing strategy is not greater than

$$\left[(2/7)^d\right]^{-1} = 3.5^d.$$

□

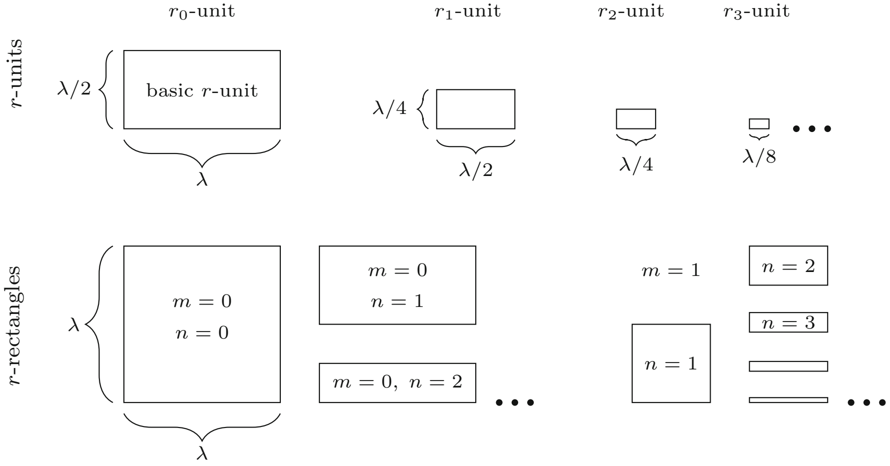


Fig. 22 r-units and r-rectangles

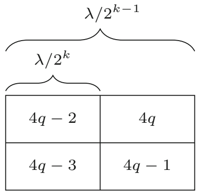


Fig. 23 The division of an r_{k-1}-unit into four r_k-units

5 Λ_2-method for packing *r*-rectangles

Let $\lambda > 0$ and let k be a non-negative integer. An r_k*-unit* is a rectangle of width $\lambda/2^k$ and height $\lambda/2^{k+1}$ (see Fig. 22). A *basic r-unit* is r_0-unit. Of course, an r_k-unit is the image of a λ_k-unit in rotation of 90°. In Sect. 3, basic units were divided into smaller units, and here basic r-units will be divided into smaller r-units in the same way (see Fig. 23).

Let r-*rectangle* be a rectangle of width $\lambda/2^m$ and height $\lambda/2^n$ for some $0 \le m \le n$. Clearly, each r-rectangle is the image of a λ-rectangle in rotation of 90°.

Let U be the union of k basic r-units with pairwise disjoint interiors numbered with natural numbers from 1 to n. When an r_{k-1}-unit ($k \ge 1$) numbered with q is divided into four r_k-units, these r_k-units are numbered from $4q-3$ to $4q$ as on Fig. 23.

Consider a sequence of r-rectangles $R_1, R_2, \ldots$ of widths not greater than $\lambda/2$. Note than neither $\lambda \times \lambda$ nor $\lambda \times \frac{1}{2}\lambda$ occurs in the sequence.

We will use the analogue of the Λ_1-method of packing (see Fig. 24).

Λ_2-method of packing an r-rectangle of width $l_i = \lambda/2^i$ into U

1. If an r-rectangle is not a square, then it is packed as low as possible into the lowest indexed r_i-unit and, obviously, with enough empty space.

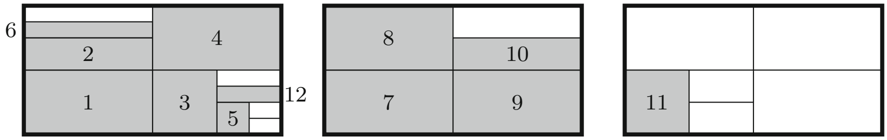


Fig. 24 Packing items into three basic r-units by Λ_2-method. Numbers indicate the order in which the items arrive

If there is no such unit, find the greatest $k \leq i$ such that there is an empty r_k-unit. From among empty r_k-units choose the lowest indexed one and then divide it into four r_{k+1}-units. If $r_{k+1} > r_i$, then the lowest indexed r_{k+1}-unit is again divided into four smaller units. The division is repeated until an r_i-unit is created. Now the r-rectangle is packed into the lowest indexed r_i-unit as low as possible.

2. An r-rectangle that is a square of side length $l_i = \lambda/2^i$, $i \geq 1$, is packed, if possible, into two empty r_i-units, that were created through the division of one r_{i-1}-unit. If there are no such units the division of a bigger unit is conducted as described in the previous case. Finally the r-rectangle is packed into two lowest indexed r-units.

Example 11 Figure 24 illustrates Λ_2-method. The method is very similar to Λ_1-method shown in Example 7. The main difference between the two methods is that Λ_1 packs as much to the left as possible, while Λ_2 puts items as low as it is possible.

The first r-rectangle, by Rule 1, should be packed into an r_1-unit. Since there is no such unit, the first r_0-unit is divided and the r-rectangle is then packed into the lowest indexed r_1-unit. The second item is packed, by Rule 1, into the lowest indexed (the second) r_1-unit as low as possible. The third item is a square thus we use Rule 2 and divide the lowest indexed r_1-unit. The third r-rectangle is packed into two lowest indexed r_2-units. The fourth item is packed, by Rule 1, into an r_1-unit with enough empty space. To pack the fifth r-rectangle, we need to perform another division and the item is packed, by Rule 2, into two r_3-units. Items from 6 to 10 are all packed by Rule 1 into lowest indexed r_1-units with enough empty space as low as possible. The eleventh item is a square and thus is packed by using Rule 2: the third basic r-unit is partitioned and then the lowest indexed r_1-unit is divided again. The eleventh item is packed into two lowest indexed r_2-units. We use Rule 1 for the twelfth r-rectangle which can be fitted into an r_2-unit in the first basic r-unit.

Lemma 3 *Assume that a sequence of r-rectangles of width not greater than $\lambda/2$ was packed into U by Λ_2-method. Let $W \subset U$ be the union of basic r-units into which an r-rectangle was packed. The empty space in W is smaller than $\eta = \frac{2}{3}\lambda^2$.*

Proof We follow the same steps as in the proof of Lemma 1. However, now there are no r-rectangles of width λ in the sequence.

First, we calculate the empty space in r-units that are partially packed. Using the same arguments, we get that the empty space in all r-units of width $\lambda_i = \lambda/2^i$ is not greater than $\frac{1}{2}\lambda_i^2$. Since the sequence of items to pack consists of r-rectangles of width smaller than λ, there are no partially packed basic r-units (of area $\lambda^2/2$). Consequently,

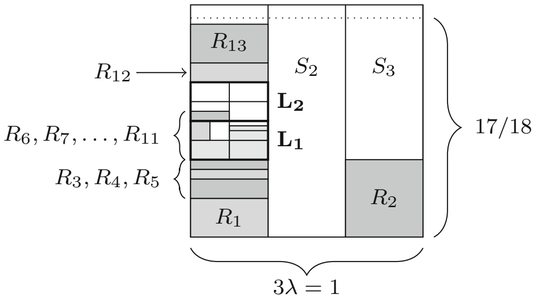


Fig. 25 Λ_2^+-method

the empty space in partially packed r-units is not greater than

$$\sum_{i\geq 1} \frac{1}{2}\lambda_i^2 = \sum_{i\geq 1} \frac{1}{2}\Big(\frac{1}{2^i}\lambda\Big)^2 = \frac{1}{2}\lambda^2 \sum_{i\geq 1} \frac{1}{4^i} = \frac{1}{6}\lambda^2.$$

Note that in the proof of Lemma 1 the empty space in partially packed λ-units was not greater than $2\lambda^2/3$; now in r-units it is not greater than $2\lambda^2/3 - \lambda^2/2 = \lambda^2/6$.

The number of empty r-units in Λ_2-method is the same as in Λ_1-method, thus the empty space in empty r-units is not greater than $\frac{1}{2}\lambda^2$.

Finally, the empty space in W is less than

$$\frac{1}{6}\lambda^2 + \frac{1}{2}\lambda^2 = \frac{2}{3}\lambda^2.$$

□

Let

$$\lambda = 1/3,$$
$$B_2 = [0, 1] \times [0, 1]$$

and let $R_1, R_2, \ldots$ be a sequence of r-rectangles. B_2 is partitioned into three rectangles $S_w = [(w-1)/3, w/3] \times [0, 1]$ for $w = 1, 2, 3$.

As in Sect. 4 the value of λ is chosen to balance the average ratios of packing of big and small items. Exactly the same reasoning is behind the choice of the height 17/18 of packing in S_1, S_2, S_3.

We give an outline of the Λ_2^+-packing method presented below. All r-rectangles are packed from bottom to top.

- Squares $1/3 \times 1/3$ are packed first into S_3 up to the height 17/18, then into S_2 up to the height 17/18 (see R_2 on Fig. 25).
- r-rectangles of width smaller than 1/3 are packed into basic r-units $1/3 \times 1/6$, called *containers* (see L_1 and L_2 on Fig. 25), by using the Λ_2-method (see $R_6, R_7, \ldots, R_{11}$ on Fig. 25).

- Containers (see L_1 and L_2 on Fig. 25) are created and other items of width 1/3 that are not squares (as $R_1, R_3, R_4, R_5, R_{12}, R_{13}$ on Fig. 25) are packed first into S_1 up to the height 17/18, then into S_2 up to the height 17/18, then into S_3.
- If an r-rectangle cannot be packed or a container cannot be created under the height 17/18, it is packed or created in B_2 as low as possible.

Let P be the topmost r-rectangle preceding R_i packed in S_w. Denote by $b_w(i)$ the distance between the bottom of S_w and

- the top of P, provided the width of P is 1/3;
- the top of the container into which P was packed, provided the width of P is smaller than 1/3.

Moreover, $b_{\max}(i) = \max[b_1(i), b_2(i), b_3(i)]$, $b_{\min}(i) = \min[b_1(i), b_2(i), b_3(i)]$. For example, on Fig. 25, $b_1(1) = b_2(1) = b_3(1) = 0$, $b_1(2) = b_1(3) = 1/6$, $b_3(3) = 1/3$, $b_1(4) = 1/4$, $b_1(5) = 7/24$, $b_1(6) = 1/3$, $b_1(7) = \cdots = b_1(11) = 1/2$, $b_1(12) = 2/3$.

Λ_2^+-method of packing of an r-rectangle R_i into B_2

1. Any item that is packed or any container that is created is placed in a bin in such a way that its interior is disjoint with any r-rectangle packed so far and with any container created earlier.
2. If R_i is a square $1/3 \times 1/3$, then
   (a) if $b_3(i) \leq 17/18 - 1/3 = 11/18$, then R_i is packed into S_3 as low as it is possible, (see the two squares on the right on Fig. 31);
   (b) if $b_3(i) > 11/18$ and $b_2(i) \leq 11/18$, then R_i is packed into S_2 as low as it is possible, (see the middle square $1/3 \times 1/3$ on Fig. 31; now $b_3(i) = 2/3 > 11/18$ and $b_2(i) = 11/18$);
   (c) otherwise R_i is packed into S_q as low as it is possible, where $q \in \{1, 2, 3\}$ is an integer such that $b_q(i) = b_{\min}(i)$. If $b_{\min}(i)$ is not unique then R_i is packed into the lowest indexed $b_q(i) = b_{\min}(i)$.
3. If R_i is a rectangle of width $w_i = 1/3$ and height $t_i \leq 1/6$, then
   (a) if $b_1(i) \leq 17/18 - t_i$, then R_i is packed into S_1 as low as it is possible, (see $R_1, R_3, R_4, R_5, R_{12}, R_{13}$ on Fig. 25);
   (b) if $b_1(i) > 17/18 - t_i$ and $b_2(i) \leq 17/18 - t_i$, then R_i is packed into S_2 as low as it is possible, (see the first r-rectangle packed in S_2 on Fig. 29);
   (c) if $b_1(i) > 17/18 - t_i$, $b_2(i) > 17/18 - t_i$ and $b_3(i) \leq 17/18 - t_i$, then R_i is packed into S_3 as low as it is possible (see the third, i.e., the topmost r-rectangle packed in S_3 on Fig. 29);
   (d) otherwise, R_i is packed into S_q as low as it is possible, where $q \in \{1, 2, 3\}$ is an integer such that $b_q(i) = b_{\min}(i)$ (see the patterned r-rectangle that we try to pack in S_2 on Fig. 29). If $b_{\min}(i)$ is not unique then R_i is packed into the lowest indexed $b_q(i) = b_{\min}(i)$.
4. If $w_i \leq 1/6$, then let U be the union of created containers contained in B_2. If $U = \emptyset$, then $k = 0$. Otherwise denote by k the number of created containers. These containers are numbered from 1 to k.

(a) if R_i can be packed into U by Λ_2-method, then we do it ($R_7, \ldots, R_{10}$ on Fig. 25 are packed into the existing container L_1);
(b) otherwise we open a new container of number $k+1$ as follows:
  – if $b_1(i) \leq 17/18 - 1/6 = 7/9$, then L is a basic r-unit packed into S_1 as low as it is possible,
    For example, on Fig. 25, if $i = 6$, then there is no created container; the new container L_1 is opened (R_6 is the first item packed into a freshly created container) in such a place into which a basic r-unit should be packed, i.e., in S_1 as low as it is possible; if $i = 11$, then R_{11} cannot be packed in L_1; therefore we create a new container L_2 (to pack this rectangle) in such a place into which a basic r-unit should be packed, i.e., in S_1 as low as it is possible;
  – if $b_1(i) > 7/9$ and $b_2(i) \leq 7/9$, then L is a basic r-unit packed into S_2 as low as it is possible;
  – if $b_1(i) > 7/9$, $b_2(i) > 7/9$ and $b_3(i) \leq 7/9$, then L is a basic r-unit packed into S_3 as low as it is possible;
  – otherwise, L is a basic r-unit packed into S_q as low as it is possible, where $b_q(i) = b_{\min}(i)$;

R_i is packed into $U \cup L$ by Λ_2-method.

6 Second drawer algorithm

In this section a drawer algorithm $D_2(d)$ with the asymptotic competitive ratio not greater than $12 \cdot 3^d$ is presented.

Let $\lambda = 1/3$. Each hyperbox H is rotated to satisfy $a_1 \geq \cdots \geq a_{d-1} \geq a_d$, where a_j is the length of the jth edge of H. By the *height* of H we mean a_d. By the *width* of H we mean a_{d-1}. A hyperbox with $a_{d-1} > \lambda = 1/3$ is called *big*, otherwise it is called *small*. Note that small hyperboxes satisfy $a_d \leq a_{d-1} \leq 1/3$. For each small hyperbox H, let $Q_{m,n}$ be an r-rectangle $\frac{1}{3 \cdot 2^m} \times \frac{1}{3 \cdot 2^n}$ such that $\frac{1}{3 \cdot 2^{m+1}} < a_{d-1} \leq \frac{1}{3 \cdot 2^m}$ and $\frac{1}{3 \cdot 2^{n+1}} < a_d \leq \frac{1}{3 \cdot 2^n}$. We say then that H is *of type* (m, n).

By an (m, n)-*drawer* we mean a hyperbox

$$1 \times \cdots \times 1 \times \frac{1}{3 \cdot 2^m} \times \frac{1}{3 \cdot 2^n}.$$

Items of type (m, n) will be packed into (m, n)-drawers. For example, item H_1 of type $(0, 0)$ is packed into $(0, 0)$-drawer, see Fig. 26. The *front wall* of any (m, n)-drawer is the rectangle $\frac{1}{3 \cdot 2^m} \times \frac{1}{3 \cdot 2^n}$. We can also treat B_2 as the front wall of $[0, 1]^d$. Drawers are created for packing items: there can be drawers of different sizes and many drawers of the same size, however at most one drawer of a fixed size is open at each stage of the packing process. Moreover, any two (open or closed) drawers have disjoint interiors.

For $d = 1$, $D_2(1)$-algorithm is the Next Fit algorithm, i.e., we pack any item into the active bin as much to the left as it is possible; the average occupation in each bin is greater than 1/2. For $d = 2$, as $D_2(2)$-algorithm we use the method described in Zhang

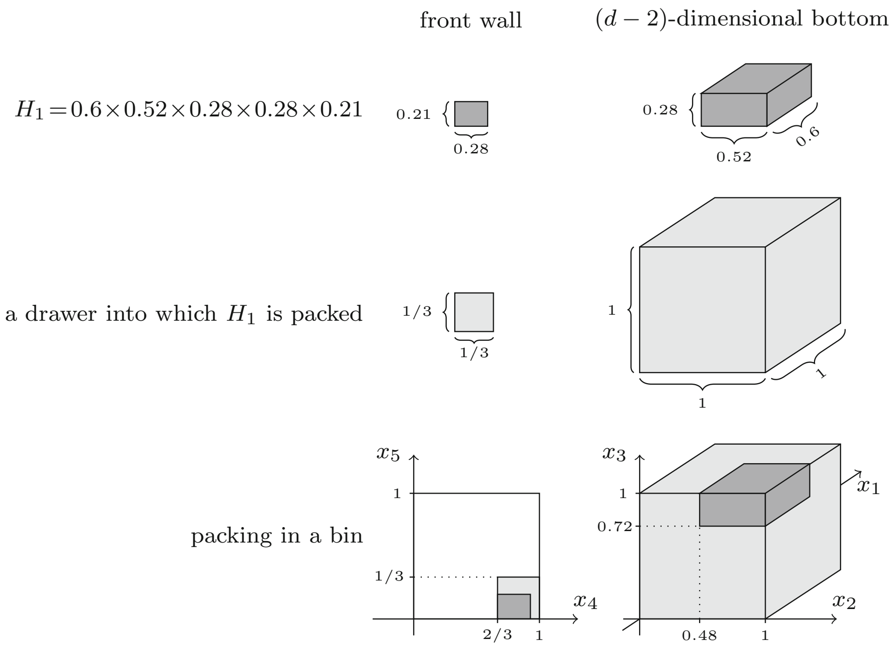


Fig. 26 $d = 5$: a drawer $1 \times 1 \times 1 \times 1/3 \times 1/3$ is created in the active bin and an item $H_1 = 0.6 \times 0.52 \times 0.28 \times 0.28 \times 0.21$ is packed into it

et al. (2014); the average occupation in each bin is greater than 0.197. Similarly as for $D_1(d)$-algorithm, the reader can take any other 1-space bounded packing algorithm with the average occupation greater than 1/36 for $d = 1$ or greater than 1/108 for $d = 2$.

Assume that $d \geq 3$.

$D_2(d)$-**algorithm**

- If H is big, then it is packed into the active bin from top to bottom along the right edge of the bin (see Sect. 4 for the definition) provided the interior of the packed item is disjoint with the union of open and closed drawers. If either H should be packed so that its interior intersects a drawer that already contains a small hypercube or there is not enough empty space to pack H, then we close the active bin and open a new active bin to pack H.
- If H is a small hyperbox of type (m, n), then
  - it is packed into the open (m, n)-drawer in such a way that the $(d-2)$-dimensional bottom of H is packed into the $(d-2)$-dimensional bottom of the drawer (i.e., into the $(d-2)$-dimensional unit cube) using the $D_2(d-2)$-algorithm.
  - If there is not enough empty space in the open (m, n)-drawer to pack H, we close this drawer. A new drawer in opened in the following way. First, use the Λ_2^+-method to determine the proper rectangle to pack $Q_{m,n}$ into B_2, let it be called Q. The rectangle Q is treated as an r-rectangle packed into B_2.

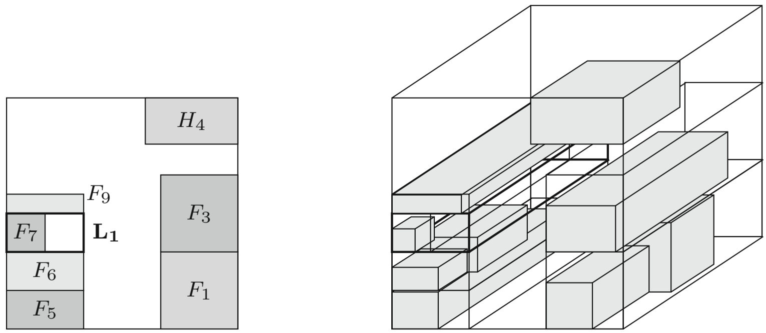


Fig. 27 $D_2(d)$-algorithm

- If the drawer with the front wall equal to Q is disjoint with the interior of any packed big item, then the new open (m, n)- drawer is the one with the front wall equal to Q. The hyperbox H is packed into this open drawer in such a way that the $(d-2)$-dimensional bottom of H is packed into the $(d-2)$-dimensional bottom of the drawer using the $D_2(d-2)$-algorithm.
- If either there is no empty space in B_2 to pack $Q_{m,n}$ by Λ_2^+-method or the drawer with the front wall equal to Q intersects the interior of a big packed item, then we close the active bin and open a new active bin to pack H.

Example 12 Figure 27 illustrates $D_2(d)$-algorithm. The first three 3-dimensional boxes $H_1 = 0.4 \times 0.2 \times 0.2$, $H_2 = 0.3 \times 0.3 \times 0.3$, $H_3 = 0.8 \times 0.3 \times 0.2$ are of type $(0, 0)$ and they are packed into $(0, 0)$-drawers: items H_1 and H_2 into the drawer with the front wall F_1 while H_3 into the drawer with the front wall F_3 (there is not enough empty space to pack H_3 in the drawer with the front wall F_1). The next item is big; $H_4 = 0.4 \times 0.4 \times 0.2$ is packed at the top of the bin along its right edge. $H_5 = 0.9 \times 0.2 \times 0.16$ is of type $(0, 1)$ and it is packed into the drawer with the front wall F_5. $H_6 = 0.2 \times 0.2 \times 0.1$ and $H_8 = 0.35 \times 0.25 \times 0.15$ are also of type $(0, 1)$; they are packed into the drawer with the front wall F_6. $H_7 = 0.13 \times 0.1 \times 0.1$ is of type $(1, 1)$ and it is packed into the drawer with the front wall F_7. This front wall is contained in the container L_1. $H_9 = 1 \times 0.3 \times 0.08$ is of type (0,2) and it is packed into the drawer with the front wall F_9.

For $d \geq 3$ we will also use the notion of $b_1(i)$, $b_2(i)$, $b_3(i)$. B_2 is the front wall of the active bin, thus we can adopt the definitions of $b_w(i)$ for higher dimensions.

Theorem 2 *The asymptotic competitive ratio for the $D_2(d)$-algorithm is not greater than $12 \cdot 3^d$.*

Proof We show that the average occupation in each bin is greater than

$$\vartheta_d = \frac{1}{12} \cdot \left(\frac{1}{3}\right)^d.$$

The proof is inductive. For $d = 1$ the average occupation in each bin is greater than $1/2 > (1/12) \cdot (1/3)$, for $d = 2$, (see Zhang et al. 2014) it is greater than $0.197 > (1/12) \cdot (1/3)^2$. Assume that $d \geq 3$ and that the statement holds in each dimension $n \in \{1, 2, \ldots, d-1\}$.

Denote by $\vartheta(l)$ the total volume of items packed into the lth bin $\mathcal{B}_l$. To prove that the average occupation in each bin is greater than ϑ_d it suffices to show that either $\vartheta(l) > \vartheta_d$ or $\vartheta(l)$ plus the volume of the first item that cannot be packed into $\mathcal{B}_l$ is greater than $2\vartheta_d$.

Consider a few cases depending on the size of the first hyperbox item

$$H_u = u_1 \times \cdots \times u_{d-1} \times u_d$$

that cannot be packed into $\mathcal{B}_l$.

Denote by ω the sum of volumes of open drawers. There is at most one open drawer of each size. The total volume of open drawers of width 1/3 is smaller than $\frac{1}{3} \cdot \frac{1}{3} + \frac{1}{3} \cdot \frac{1}{6} + \frac{1}{3} \cdot \frac{1}{12} + \cdots = \frac{2}{9}$. The sum of volumes of open drawers of width 1/6 is smaller than $\frac{1}{6} \cdot \frac{1}{6} + \frac{1}{6} \cdot \frac{1}{12} + \frac{1}{6} \cdot \frac{1}{24} + \cdots = \frac{2}{36}$. The total volume of open drawers of width $\lambda/12$ is smaller than $\frac{1}{12} \cdot \frac{1}{12} + \frac{1}{12} \cdot \frac{1}{24} + \cdots = \frac{2}{144}$ and so on. Consequently,

$$\omega < \frac{2}{9} + \frac{2}{36} + \frac{2}{144} + \cdots = \frac{2}{9} \cdot \frac{4}{3} = \frac{8}{27}.$$

By the inductive assumption and by the fact that the smallest r-rectangle containing a rectangle $R = a_{d-1} \times a_d$ is of area not greater than 4 times the area of R, the average occupation in any closed (m, n)-drawer $L_{m,n}$ is greater than

$$\frac{1}{4} \cdot \vartheta_{d-2} \cdot \mathrm{vol}(L_{m,n}).$$

On Figs. 28, 29, 30 and 31 the front wall of an active bin is shown. We use grey or patterned rectangles to indicate one of the following things:

- front walls of big items;
- front walls of $(0, n)$-drawers;
- containers.

For example, on Fig. 28 the fifth and the sixth rectangle in S_1 (the left column) can be either front walls of (0, 1)-drawers or containers as L_1 or L_2 on Fig. 25. The patterned rectangle is the front wall of the big item that cannot be packed in the bin.

Denote by h the sum of heights of big items packed into the active bin.

If H_u is big and if $h + u_d > 1/18$ (see Fig. 28), then

$$\vartheta(l) + \mathrm{vol}(H_u) > \frac{1}{18} \cdot \left(\frac{1}{3}\right)^{d-1} = \frac{1}{6} \cdot \left(\frac{1}{3}\right)^{d} = 2\vartheta_d.$$

In the next cases we will assume that $h + u_d \leq 1/18$ provided H_u is big.

Case 1 $b_{max}(u) \leq 17/18$ *and* $u_d \leq 1/6$.

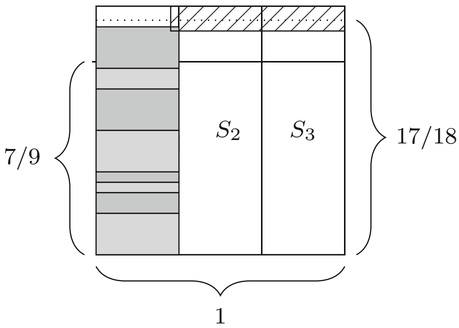


Fig. 28 Big items: $h + u_d > 1/18$

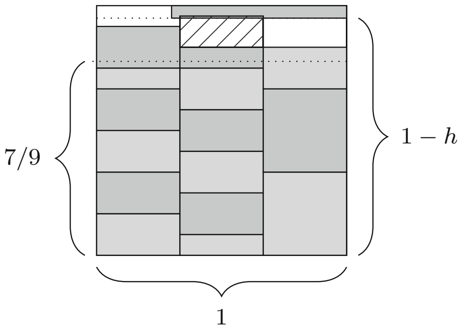


Fig. 29 Case 1

Since $b_{max}(u) \leq 17/18$ (see Fig. 29), we can assume that H_u is small; otherwise $h + u_d > 1/18$.

The total volume of items packed into an open drawer can be close to 0, thus we have to subtract the sum of volumes of open drawers $\omega < 8/27$. Moreover, by Lemma 3, the area of empty space in B_2 is not greater than $\eta = \frac{2}{3} \cdot \left(\frac{1}{3}\right)^2$. Therefore the area of the part of B_2 covered by the front walls of open and closed drawers and, consequently, the volume of all open and closed drawers is greater than

$$\frac{1}{3}\big[b_1(u) + b_2(u) + b_3(u)\big] - \eta.$$

Since $b_w(u) > 1 - h - \frac{1}{6}$ for $w = 1, 2, 3$, it follows that

$$\begin{aligned}\vartheta(l) &> \left(1 - h - \frac{1}{6} - \omega - \eta\right) \cdot \frac{1}{4} \cdot \vartheta_{d-2} + h \cdot \left(\frac{1}{3}\right)^{d-1} \\ &> \left(1 - \frac{1}{6} - \frac{8}{27} - \frac{2}{27}\right) \cdot \frac{1}{4} \cdot \frac{1}{12} \cdot \left(\frac{1}{3}\right)^{d-2} > \vartheta_d.\end{aligned}$$

Case 2 $b_{max}(u) \leq 17/18$ *and* $u_d > 1/6$. Similarly as in Case 1 we can assume that H_u is small. Since $u_{d-1} \geq u_d > 1/6$, by the description of Λ_2^+ method, H_u should be packed into an open (0, 0)-drawer (i.e., drawer whose front wall is a square

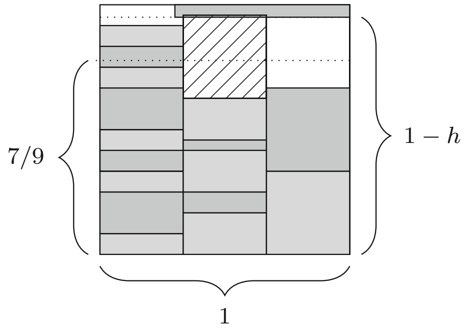


Fig. 30 Case 2

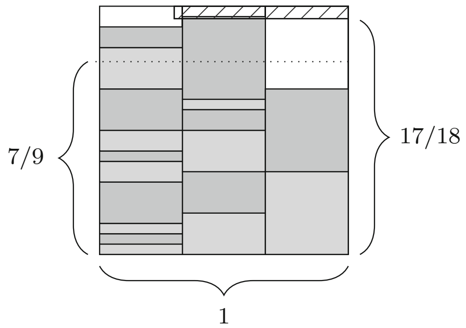


Fig. 31 Case 4

$1/3 \times 1/3$). Since it is impossible, there is no open (0, 0)-drawer. This implies that the total volume of open drawers is smaller than $\omega - \frac{1}{3} \cdot \frac{1}{3}$.

First assume that at least four (0, 0)-drawers are closed. The total volume of small items is greater than $4 \cdot \frac{1}{9} \cdot \frac{1}{4} \cdot \vartheta_{d-2} = \vartheta_d$.

Now assume that at most three (0, 0)-drawers are closed (see Fig. 30). By $b_1(u) > \frac{17}{18} - \frac{1}{6} = \frac{7}{9}$ (at least one item of height not greater than 1/6 was packed into $S_2 \cup S_3$), by $b_2(u) > 1 - h - \frac{1}{3}$ as well as by $b_3(u) > 1 - h - \frac{1}{3}$ we deduce that

$$\vartheta(l) > \left[\frac{7}{9} \cdot \frac{1}{3} + 2\left(1 - h - \frac{1}{3}\right) \cdot \frac{1}{3} - \left(\omega - \frac{1}{9}\right) - \eta\right] \cdot \frac{1}{4} \cdot \vartheta_{d-2} + h \cdot \left(\frac{1}{3}\right)^{d-1}$$
$$> \left(\frac{7}{27} + 2 \cdot \frac{2}{3} \cdot \frac{1}{3} - \frac{8}{27} + \frac{1}{9} - \frac{2}{27}\right) \cdot \frac{1}{4} \cdot \frac{1}{12} \cdot \left(\frac{1}{3}\right)^{d-2} = \vartheta_d.$$

Case 3 $b_{max}(u) > 17/18$ *and* $b_{\min}(u) \geq 7/9$. In this case

$$\vartheta(l) > \left(2 \cdot \frac{7}{9} \cdot \frac{1}{3} + \frac{17}{18} \cdot \frac{1}{3} - \omega - \eta\right) \cdot \frac{1}{4} \cdot \vartheta_{d-2}$$
$$> \left(2 \cdot \frac{7}{9} \cdot \frac{1}{3} + \frac{17}{18} \cdot \frac{1}{3} - \frac{8}{27} - \frac{2}{27}\right) \cdot \frac{9}{4} \cdot \frac{1}{12} \cdot \left(\frac{1}{3}\right)^{d} > \vartheta_d.$$

Case 4 $b_{max}(u) > 17/18$ *and* $b_{\min}(u) < 7/9$.

If at least four $(0,0)$-drawers are closed, then the total volume of small items is greater than $4 \cdot \frac{1}{9} \cdot \frac{1}{4} \cdot \vartheta_{d-2} = \vartheta_d$.

Now assume that at most three $(0,0)$-drawers are closed (see Fig. 31). Observe that at least one small item R_k of height not greater than 1/6 was packed into S_2 (otherwise $b_{\max}(u) < 17/18$). Thus $7/9 < b_1(k) \leq 17/18$. Moreover, all small items packed into S_3 have height greater than 1/6. The reason is that if a small item R_l of height not greater than 1/6 was packed into S_3, then $7/9 < b_1(l) \leq 17/18$ as well as $7/9 < b_2(l) \leq 17/18$, which is a contradiction with $b_{\min}(u) < 7/9$ and $b_{\max}(u) > 17/18$. Consequently, at least two drawers of height 1/3 were opened into S_3 and $b_3(u) \geq 2/3$. Moreover, $7/9 \leq b_1(u) < 17/18$.

Subcase 4a: A drawer of height smaller than 1/6 (i.e., *not greater that* $1/12$) *was opened in* S_2. This implies that $b_1(u) > \frac{17}{18} - \frac{1}{12} = \frac{31}{36}$ (otherwise this drawer should be created in S_1). Hence

$$\vartheta(l) > \left(\frac{31}{36} \cdot \frac{1}{3} + \frac{17}{18} \cdot \frac{1}{3} + \frac{2}{3} \cdot \frac{1}{3} - \omega - \eta\right) \cdot \frac{1}{4} \cdot \vartheta_{d-2}$$
$$> \left(\frac{31}{36} \cdot \frac{1}{3} + \frac{17}{18} \cdot \frac{1}{3} + \frac{2}{3} \cdot \frac{1}{3} - \frac{8}{27} - \frac{2}{27}\right) \cdot \frac{9}{4} \cdot \frac{1}{12} \cdot \left(\frac{1}{3}\right)^d > \vartheta_d.$$

Subcase 4b: No drawer of height smaller than 1/6 *was opened in* S_2. This implies that $b_2(u)$ is a multiple of 1/6 (also $b_3(u)$ is a multiple of 1/6).

If $b_{\max}(u) = b_2(u)$, then by $b_{\max}(u) > 17/18$ we deduce that $b_2(u) = 1$. As a consequence,

$$\vartheta(l) > \left(\frac{7}{9} \cdot \frac{1}{3} + 1 \cdot \frac{1}{3} + \frac{2}{3} \cdot \frac{1}{3} - \omega - \eta\right) \cdot \frac{1}{4} \cdot \vartheta_{d-2}$$
$$> \left(\frac{7}{9} \cdot \frac{1}{3} + 1 \cdot \frac{1}{3} + \frac{2}{3} \cdot \frac{1}{3} - \frac{8}{27} - \frac{2}{27}\right) \cdot \frac{9}{4} \cdot \frac{1}{12} \cdot \left(\frac{1}{3}\right)^d = \vartheta_d.$$

If $b_{\max}(u) = b_3(u)$, then $b_3(u) = 1$ and $b_2(u) \geq 2/3$. Hence

$$\vartheta(l) > \left(\frac{7}{9} \cdot \frac{1}{3} + \frac{2}{3} \cdot \frac{1}{3} + 1 \cdot \frac{1}{3} - \omega - \eta\right) \cdot \frac{1}{4} \cdot \vartheta_{d-2} = \vartheta_d.$$

The average occupation in each bin is greater than $(1/12) \cdot (1/3)^d$. Consequently, the asymptotic competitive ratio of this packing strategy is not greater than $12 \cdot 3^d$. □

References

Brown DJ (1979) A lower bound for on-line one-dimensional bin packing algorithms. Technical report R-864, Coordinated Sci. Lab, Urbana, Illinois

Coppersmith D, Raghavan P (1989) Multidimensional on-line bin packing: algorithms and worst-case analysis. Oper Res Lett 8(1):17–20. https://doi.org/10.1016/0167-6377(89)90027-8

Csirik J, Frenk J, Labb M (1993) Two-dimensional rectangle packing: on-line methods and results. Discrete Appl Math 45(3):197–204. https://doi.org/10.1016/0166-218X(93)90009-D

Epstein L, van Stee R (2005) Optimal online algorithms for multidimensional packing problems. SIAM J Comput 35(2):431–448. https://doi.org/10.1137/S0097539705446895

Grzegorek P, Januszewski J (2014) Online algorithms for 3-space bounded 2-dimensional bin packing and square packing. Rom J Inf Sci Technol 17(2):190–203

Grzegorek P, Januszewski J (2015) A note on one-space bounded square packing. Inf Process Lett 115(11):872–876. https://doi.org/10.1016/j.ipl.2015.06.001

Han X, Fujita S, Guo H (2001) A two-dimensional harmonic algorithm with performance ratio 2.7834. IPSJ SIG Notes 93:43–50

Han X, Chin FYL, Ting HF, Zhang G, Zhang Y (2011) A new upper bound 2.5545 on 2d online bin packing. ACM Trans Algorithms 7(4):50:1–50:18. https://doi.org/10.1145/2000807.2000818

Heydrich S, van Stee R (2016) Beating the harmonic lower bound for online bin packing. In: 43rd international colloquium on automata, languages, and programming, Schloss Dagstuhl—LZI, Wadern, LIPIcs Leibniz international proceedings in informatics, vol 55, pp 41:1–41:14

Januszewski J, Zielonka Ł (2016) Improved online algorithms for 2-space bounded 2-dimensional bin packing. Int J Found Comput Sci 27(04):407–429. https://doi.org/10.1142/S0129054116500076

Januszewski J, Zielonka Ł (2018) Online packing of rectangular items into square bins. In: Solis-Oba R, Fleischer R (eds) Approximation and online algorithms. Lecture notes in computer science. WAOA 2017, vol 10787. Springer, Cham, pp 147–163

Johnson DS (1974) Fast algorithms for bin packing. J Comput Syst Sci 8(3):272–314. https://doi.org/10.1016/S0022-0000(74)80026-7

Johnson DS, Demers A, Ullman JD, Garey MR, Graham RL (1974) Worst-case performance bounds for simple one-dimensional packing algorithms. SIAM J Comput 3(4):299–325. https://doi.org/10.1137/0203025

Lee CC, Lee DT (1985) A simple on-line bin-packing algorithm. J ACM 32(3):562–572. https://doi.org/10.1145/3828.3833

Liang FM (1980) A lower bound for on-line bin packing. Inf Process Lett 10(2):76–79. https://doi.org/10.1016/S0020-0190(80)90077-0

Ramanan P, Brown DJ, Lee C, Lee D (1989) On-line bin packing in linear time. J Algorithms 10(3):305–326. https://doi.org/10.1016/0196-6774(89)90031-X

Seiden SS (2002) On the online bin packing problem. J ACM 49(5):640–671. https://doi.org/10.1145/585265.585269

Seiden SS, van Stee R (2003) New bounds for multidimensional packing. Algorithmica 36(3):261–293. https://doi.org/10.1007/s00453-003-1016-7

Ullman JD (1971) The performance of a memory allocation algorithm. Technical report 100, Princeton University, Department of Electrical Engineering, Computer Sciences Laboratory

van Vliet A (1992) An improved lower bound for on-line bin packing algorithms. Inf Process Lett 43(5):277–284. https://doi.org/10.1016/0020-0190(92)90223-I

Woeginger G (1993) Improved space for bounded-space, on-line bin-packing. SIAM J Discrete Math 6(4):575–581. https://doi.org/10.1137/0406045

Yao ACC (1980) New algorithms for bin packing. J ACM 27(2):207–227. https://doi.org/10.1145/322186.322187

Zhang Y, Chen J, Chin FYL, Han X, Ting HF, Tsin YH (2010a) Improved online algorithms for 1-space bounded 2-dimensional bin packing. Springer, Berlin, pp 242–253

Zhang Y, Chin FYL, Ting HF (2010b) One-space bounded algorithms for two-dimensional bin packing. Int J Found Comput Sci 21(06):875–891. https://doi.org/10.1142/S0129054110007611

Zhang Y, Chin FYL, Ting HF, Han X (2013) Online algorithms for 1-space bounded multi dimensional bin packing and hypercube packing. J Comb Optim 26(2):223–236. https://doi.org/10.1007/s10878-012-9457-z

Zhang Y, Chin FYL, Ting HF, Han X, Poon CK, Tsin YH, Ye D (2014) Online algorithms for 1-space bounded 2-dimensional bin packing and square packing. Theor Comput Sci 554:135–149. https://doi.org/10.1016/j.tcs.2014.02.045

Zhao X, Shen H (2015) On-line algorithms for 2-space bounded cube and hypercube packing. Tsinghua Sci Technol 20(3):255–263

Zielonka Ł (2016) On-line packing cubes into n unit cubes. Bull Pol Acad Sci Math 64:185–198