Algorithmica (2020) 82:3216–3249
https://doi.org/10.1007/s00453-020-00723-5

Efficient 1-Space Bounded Hypercube Packing Algorithm

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

Received: 24 May 2017 / Accepted: 2 May 2020 / Published online: 5 June 2020

Abstract
A space bounded $O(d/\log d)$-competitive hypercube packing algorithm with one active bin only is presented. As a starting point we give a simple 1-space bounded hypercube packing algorithm with competitive ratio $(3/2)^d + O((21/16)^d)$, for $d \geq 3$.

Keywords Bin packing · Online algorithm · Asymptotic competitive ratio · Cube · Hypercube · One-space bounded

1 Introduction

In the bin packing problem, we receive a sequence S of items of different sizes that must be packed into a finite number of bins in a way that minimizes the number of bins used. When all the items of S are accessible, the packing method is called *offline*. The packing method is called *online*, when items arrive one by one and each item has to be packed irrevocably into a bin before the next item is presented.

One can consider an online method with t bins available for packing at each point in time. It is called *t-space bounded*. There are three types of bins: active, open and closed. At each point in time, exactly t bins are declared active. At the beginning, t bins are declared active and the remaining bins are open (there are no closed bins). Each incoming item is packed into one of the active bins; the remaining open bins are not available at this moment. We can decide to close an active bin. The most frequent reason for doing this is not enough space to pack an incoming item, however there may be other reasons based on the packing algorithm. When an active bin is

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

Janusz Januszewski
januszew@utp.edu.pl

Łukasz Zielonka
Lukasz.Zielonka@utp.edu.pl

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

closed, a new bin from among open bins is declared active. None of the closed bins is used again. It is natural to expect a packing method to be less efficient with fewer number of active bins. An unbounded space model does not impose any limits on the number of active bins.

Let S be a sequence of items, let $A(S)$ be the number of bins used by algorithm A and let $OPT(S)$ be the minimum possible number of bins used to pack items from S. The *asymptotic competitive ratio* for algorithm A is defined as:

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

Online bin packing is a classical problem studied for more than forty years. One-dimensional bin packing was first investigated in [27] (see also [20]), where the performance ratio of the First Fit algorithm was proved to be 17/10. The Next Fit algorithm with performance ratio not greater than 2 was discussed in [19]. Revised First Fit presented in [30] has performance ratio 5/3. The article also gives the lower bound 3/2 on the competitive ratio. The result was then improved in [3, 22] (the lower bound not smaller than 1.53635) and in [28], where the reader can find the lower bound 1.54014. First Fit and Best Fit algorithms can be found in [7]. The authors of the article [23] improve the upper bound to 1.61217 and give the lower bound 1.58333 for the class of Modified Harmonic algorithms. Seiden in [24] further improved the upper bound to 1.58889. Moreover, the upper bound 1.5813 was proved by Heydrich and van Stee (see [14]). Recently, the lower bound on the asymptotic competitive ratio of any online algorithm for bin packing was improved to 1.54278 (see [2]). Furthermore, an algorithm AH (Advanced Harmonic) whose asymptotic competitive ratio does not exceed 1.57829 was presented in [1].

Coppersmith and Raghavan in [5] presented the 2-dimensional online bin packing algorithm with competitive ratio 3.25. The result was later improved in [6] to 3.0625 and in [13] to 2.7834. Further improvements can be found in [25], where the authors show the upper bound of 2.66013 of the asymptotic competitive ratio. Currently the upper bound stands at 2.5545 (see [12]).

The classical 1-dimensional result in bin packing comes from Lee and Lee [21]. The authors presented an online bounded-space algorithm called Harmonic with the lower bound with the competitive ratio $\Pi_\infty \approx 1.69103$. The authors also showed that there is no bounded space algorithm with performance ratio below Π_∞.

In the Harmonic algorithm and its improvements when the asymptotic competitive ratio approaches the optimal value, the number of active bins diverges to infinity. A question arises: What asymptotic competitive ratio can be achieved when the number of active bins is bounded above by a small natural number? This question was addressed by Woeginger in [29] whose Simplified Harmonic 6-space bounded online algorithm has competitive ratio beneath 17/10.

Let $d \geq 3$. We focus on the problem of packing d-dimensional hypercubes of the edge lengths not greater than 1 into a bin (a hypercube of the edge length 1).

The problem of multidimensional bin packing is discussed in [4]. Although previous studies used space bounded models, the number of active bins was usually large (for example, greater than 9 in [8]). The paper [8] by Epstein and van Stee gives a space bounded multidimensional hypercube packing algorithm with competitive ratio $O(d/\log d)$, however the number of active bins is about $d/\log d$. In this paper we describe a hypercube packing algorithm with competitive ratio $O(d/\log d)$ and with one active bin only.

Articles [15, 17] provide optimal estimates for online packing of hypercubes to a single bin for $d \geq 4$: any sequence of d-dimensional hypercubes of total volume not greater than 2^{1-d} can be packed online into a unit hypercube. The first paper concerns the case $d \geq 5$, the second $d = 4$. Online packing of hyperboxes into a single bin is studied in [18], where the following upper bound is presented: any sequence of d-dimensional hyperboxes of the edge length smaller than or equal to 1 with total volume not greater than $(3 - 2\sqrt{2}) \cdot 3^{-d}$ can be packed online into the d-dimensional unit hypercube.

An algorithm with competitive 2^{d+1} that uses only one active bin is presented in the article [31]. In the same paper the authors provide 1-space bounded algorithm for hyperbox packing with competitive ratio 4^d. Algorithms with smaller competitive ratios can be found in [11], where the authors give hyperbox packing methods with ratios $(3.5)^d$ and $12 \cdot 3^d$. 1-space bounded 2-dimensional online packing algorithms were studied in [32] (a 4.3-competitive algorithm) and [10] (a 3.888-competitive algorithm). The 2-space bounded 3.8165-competitive algorithm can be found in [16]. The 3-space bounded 3.577-competitive algorithm is presented in [9].

In the first part of the paper we provide the online algorithm $tt(d)$ (with two types of small items) that uses one active bin only with competitive ratio not greater than $(3/2)^d + O((21/16)^d)$ which is a significant improvement of the previous result (2^{d+1}, see [31]). The algorithm distinguishes three types of hypercubes: big, 2-small and 3-small. Big hypercubes are packed alone, while the last two types are packed starting from the opposite corners of the bin. Counting the number of 2- and 3-small hypercubes that could be packed together led us to Gould's sequence and a sequence of its partial sums linked with odd entries in Pascal's triangle, which we consider to be a fact worth noting.

The second part of the paper contains the algorithm $har(d)$ in which about $d/\log d$ types of items are distinguished. The algorithm is a generalization of the $tt(d)$ algorithm and is based on the well-known Harmonic algorithm. Instead of using weights we give a direct proof that also in 1-space bounded model the competitive ratio $O(d/\log d)$ is achievable.

2 The *tt*(*d*) Algorithm for $d \geq 3$

2.1 Types of Items

Given an item (a d-dimensional hypercube) C_i, denote by a_i its edge length. Items are divided into types:

1. C_i is *big*, provided $a_i > 1/2$;
2. C_i is *small*, provided $a_i \leq 1/2$;

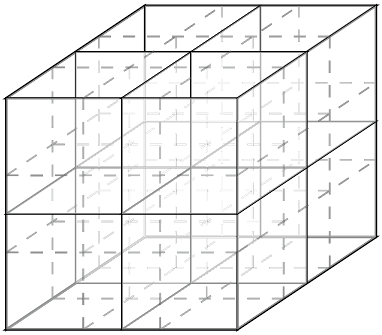


Fig. 1 Partition into 2-cubes and 3-cubes

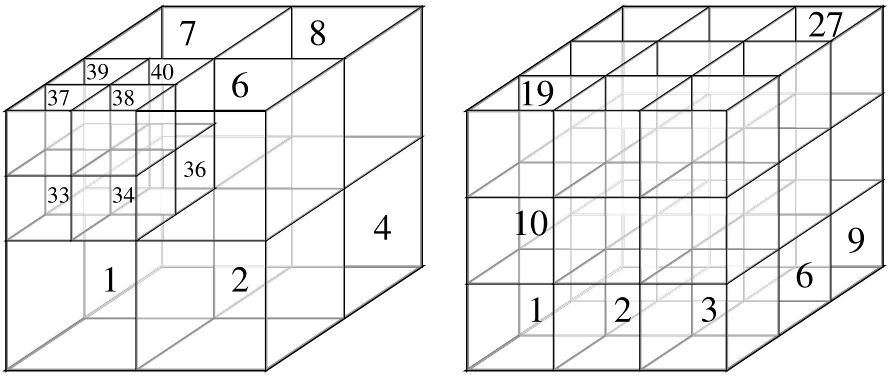


Fig. 2 Numbering of some 2-cubes and 2-subcubes (left) and 3-cubes (right)

2.1 a small C_i is 2-small if a_i belongs to $\bigcup_{n=0}^{\infty}(1/3 \cdot 2^{-n}, 1/2 \cdot 2^{-n}]$;
2.2 a small C_i is 3-small if a_i belongs to $\bigcup_{n=0}^{\infty}(1/4 \cdot 2^{-n}, 1/3 \cdot 2^{-n}]$.

Each small item is either 2-small or 3-small. Items from these classes will be packed starting from the opposite vertices of the hypercube.

Any bin $\mathscr{B}$ can be partitioned into 2^d congruent hypercubes called 2-*cubes* as well as into 3^d congruent hypercubes called 3-*cubes* (see Fig. 1). Moreover, in the course of packing every k-cube can be partitioned into 2^{dn} congruent hypercubes with the edge lengths $1/(k \cdot 2^n)$, called k-*subcubes*, for $k \in \{2, 3\}$ and $n \in \{0, 1, 2, \dots\}$. 2-cubes [3-cubes] are also called 2-subcubes [3-subcubes, respectively].

We lose no generality in assuming that $\mathscr{B} = [0, 1]^d$. We will define an ordering of 2-cubes in $\mathscr{B}$. For $d = 1$ the bin $\mathscr{B}$ is the interval [0, 1] and the 2-cubes are numbered from left to right: 1 and 2. When the order is defined for all dimensions up to $d - 1$, we define the order in $\mathscr{B} = [0, 1]^d$ in such way that all d-dimensional 2-cubes of $[0, 1]^{d-1} \times [0, 1/2]$ are numbered as in dimension $d - 1$: from 1 to 2^{d-1}. Then the d-dimensional 2-cubes of $[0, 1]^{d-1} \times [1/2, 1]$ are numbered from $2^{d-1} + 1$ to 2^d in the order borrowed from the dimension $d - 1$ (see Fig. 2, where $d = 3$).

The order of 3-cubes in $\mathscr{B}$ is defined in an analogous way.

Furthermore, each subcube has a number assigned to it.

Any bin $\mathscr{B}$ is partitioned into congruent 2-subcubes with the edge length 1/2, numbered from 1 to 2^d. If S is the subcube with the edge length 1/2 and with number λ_1 ($\lambda_1 \in \{1, \dots, 2^d\}$), then all 2-subcubes with the edge length $1/2^2$ (the partition of S) are numbered in an arbitrary order from $2^d(\lambda_1 - 1) + 1$ to $2^d\lambda_1$ (see Fig. 2). Moreover, if S is the subcube with the edge length $1/2^2$ and with number λ_2 ($\lambda_2 \in \{1, \dots, 4^d\}$), then all 2-subcubes with the edge length $1/2^3$ (the partition of S) are numbered in an arbitrary order from $2^d(\lambda_2 - 1) + 1$ to $2^d\lambda_2$. Generally, 2^d congruent 2-subcubes forming the partition of S are numbered in an arbitrary order from $2^d(\lambda - 1) + 1$ to $2^d\lambda$.

Similarly, any bin $\mathscr{B}$ is partitioned into congruent 3-subcubes with the edge length 1/3, numbered from 1 to 3^d. If S is the subcube with the edge length 1/3 and with number μ_1 ($\mu_1 \in \{1, \dots, 3^d\}$), then all 3-subcubes with the edge length $1/(2 \cdot 3)$ (the partition of S) are numbered in an arbitrary order from $2^d(\mu_1 - 1) + 1$ to $2^d\mu_1$ (see Fig. 2). Moreover, if S is the subcube with the edge length $1/(2 \cdot 3)$ and with number μ_2 ($\mu_2 \in \{1, \dots, 6^d\}$), then all 3-subcubes with the edge length $1/(2^2 \cdot 3)$ (the partition of S) are numbered in an arbitrary order from $2^d(\mu_2 - 1) + 1$ to $2^d\mu_2$. Generally, 2^d congruent 3-subcubes into which the 3-subcube with number μ is divided are numbered in an arbitrary order from $2^d(\mu - 1) + 1$ to $2^d\mu$.

2.2 Packing Algorithm

If C_i is a k-small item, then denote by K_i the smallest hypercube of the edge lengths from the set $\{\frac{1}{k}, \frac{1}{2k}, \frac{1}{4k}, \dots\}$ into which C_i can be packed, $k \in \{2, 3\}$. For example, if $a_i = 10/81$ (C_i is 2-small), then K_i is a hypercube of the edge length 1/8; if $a_i = 10/31$, then the edge length of the smallest hypercube K_i containing the 3-small item C_i equals 1/3.

A k-subcube of $\mathscr{B}$ is *empty* if its interior has an empty intersection with any item packed so far.

Algorithm $tt(d)$ **for packing of** C_i **into** $\mathscr{B}$.

1. If C_i is big, then we close the active bin, open a new bin, pack C_i and close the bin. Then we open a new active bin.
2. If C_i is a 2-small item, then we pack C_i into the empty 2-subcube congruent to K_i with the smallest number. If there is no empty subcube, we close the active bin and open a new bin to pack C_i.
3. If C_i is a 3-small item, then we pack C_i into the empty 3-subcube congruent to K_i with the greatest number. If there is no empty subcube, we close the active bin and open a new active bin to pack C_i.

Example We need to pack a sequence of four items C_1, C_2, C_3, C_4 according to the $tt(3)$ algorithm, see Fig. 3. The edge lengths of arriving items are 1/9, 1/2, 10/81 and 10/31.

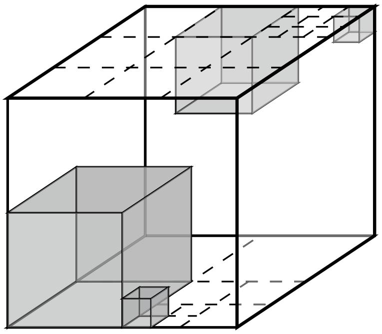


Fig. 3 Four items packed according to the *tt*(3) algorithm

The first hypercube is a 3-small item. Moreover $C_1 = K_1$ and thus it is packed into the empty 3-subcube with the greatest number (inside the 3-cube numbered 27, see Fig. 2). The hypercube $C_2 = K_2$ is packed into the first 2-cube. For C_3 the smallest 2-subcube into which it can be packed is congruent to a cube K_3 of the edge length 1/8. C_3 is packed into K_3 and then K_3 is packed into a 2-subcube of the second 2-cube. Finally, C_4 is packed into a cube K_4 of the edge length 1/3, which is then packed into the empty 3-cube with the greatest number, which would be the 3-cube with number 26.

Denote by $|A|$ the d-dimensional volume of an item A.

Lemma 1 *Suppose a* 2*-small item* C_i *was packed into a* 2*-subcube congruent to* K_i*. At least* $\left(2/3\right)^d$ *units of volume of the* 2*-subcube is occupied, i.e.,* $|C_i| \geq \left(2/3\right)^d |K_i|$*.*

Proof Recall that K_i is congruent to the smallest 2-subcube into which C_i can be packed. C_i is 2-small thus its edge length is greater than $3^{-1} \cdot 2^{-n}$ and not greater than 2^{-n-1} for some non-negative integer n. We use that lower bound in calculations.

$$\frac{|C_i|}{|K_i|} > \frac{(3^{-1} \cdot 2^{-n})^d}{(2^{-n-1})^d} = \left(\frac{2}{3}\right)^d.$$

□

We now give the analogous lemma for 3-small items.

Lemma 2 *Suppose a* 3*-small item* C_i *was packed into a* 3*-subcube congruent to* K_i*. At least* $\left(3/4\right)^d$ *of the* 3*-subcube is occupied, i.e.,* $|C_i| \geq \left(3/4\right)^d |K_i|$*.*

Proof

$$\frac{|C_i|}{|K_i|} > \frac{(2^{-n-2})^d}{(3^{-1} \cdot 2^{-n})^d} = \left(\frac{3}{4}\right)^d.$$

□

Let $\rho_2 = \left(2/3\right)^d$, $\rho_3 = \left(3/4\right)^d$ and let either $k = 2$ or $k = 3$. We say that a k-subcube is *used for packing*, provided a k-small item was packed into it.

Lemma 3 *Assume that C_i is the first k-small item that cannot be packed into a k-cube of a bin $\mathscr{B}$ by our algorithm. Denote by m_k the number of k-cubes (of the edge lengths k^{-1}) used for packing. Then the total volume of all k-small items preceding C_i packed into $\mathscr{B}$ is greater than $\rho_k m_k \cdot k^{-d} - |C_i|$.*

Proof Let C_i be a k-small item. By the assumption, there is no empty k-subcube greater than or congruent to K_i, i.e., all empty k-subcubes in $\mathscr{B}$ are smaller than K_i. Furthermore, for each $i = 1, 2, \ldots$ there are at most $2^d - 1$ empty k-subcubes congruent to $2^{-i}K_i$. Notice that if there were 2^d empty k-subcubes congruent to $2^{-i}K_i$, they would form a whole empty k-subcube congruent to $2^{-i+1}K_i$ (recall that the algorithm imposes packing to the subcube with the smallest number).

The total volume of empty k-subcubes in $\mathscr{B}$ is smaller than

$$(2^d - 1) \cdot (2^{-d} + 4^{-d} + \cdots) \cdot |K_i| = |K_i|.$$

By Lemmas 1 and 2 we know that the total volume of k-small items packed into $\mathscr{B}$ is greater than

$$\rho_k(m_k \cdot k^{-d} - |K_i|) = \rho_k m_k \cdot k^{-d} - \rho_k |K_i| > \rho_k m_k \cdot k^{-d} - |C_i|.$$

□

2.3 2-Cubes Versus 3-Cubes

Denote by n_2 the number of 2-cubes used for packing and by U_{n_2} the union of 2-cubes used for packing in a bin $\mathscr{B}$. Moreover, denote by n_3^- the number of 3-cubes contained in $\mathscr{B} \backslash U_{n_2}$.

For $d = 1$ we have $n_3^- = 3$ for $n_2 = 0$, $n_3^- = 1$ for $n_2 = 1$ and $n_3^- = 0$ for $n_2 = 2$ (see Fig. 4).

We want to have a relation between n_3^- and n_2. In Lemma 4 we will present a recursive formula. Clearly, n_3^- also depends on d, thus we will use the notation $n_3^- = y_d(n_2)$ (see Fig. 4).

The number of empty 3-cubes can be calculated directly for $d = 2$ (see Fig. 5):

- $y_2(0) = 3^2$ (all 3-cubes are empty),
- $y_2(1) = 5$ (exactly one 2-cube is packed),
- $y_2(2) = 3^1$ (only one row of 3-cubes is empty),
- $y_2(3) = 1$,
- $y_2(4) = 0$.

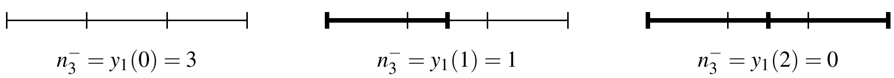


Fig. 4 2-cubes and 3-cubes for $d = 1$

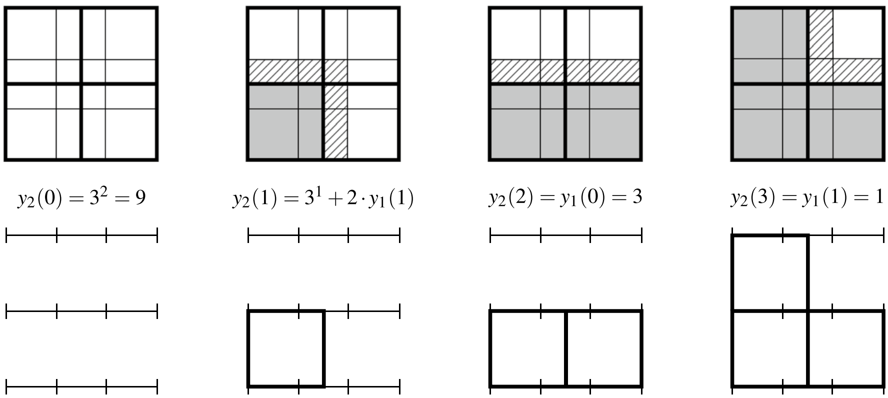


Fig. 5 2-cubes, 3-cubes for $d = 2$ visualised as layers

If $\mathscr{B} = [0, 1]^d$, then all 2-cubes of $\mathscr{B}$ are of the form

$$C_2(\alpha_1, \dots, \alpha_d) = [\alpha_1, \alpha_1 + 1/2] \times \cdots \times [\alpha_d, \alpha_d + 1/2],$$

where $\alpha_i \in \{0, 1/2\}$ for $i = 1, 2, \dots, d$. Moreover, all 3-cubes of $\mathscr{B}$ are of the form

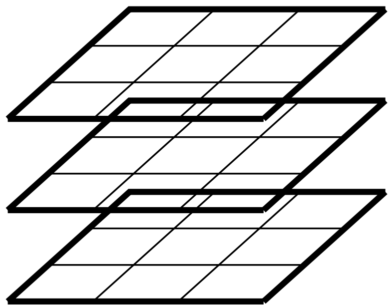
$$C_3(\beta_1, \dots, \beta_d) = [\beta_1, \beta_1 + 1/3] \times \cdots \times [\beta_d, \beta_d + 1/3],$$

where $\beta_i \in \{0, 1/3, 2/3\}$ for $i = 1, 2, \dots, d$. We will use the notation $C_2[\alpha_d] = C_2(\alpha_1, \dots, \alpha_d)$ and $C_3[\beta_d] = C_3(\beta_1, \dots, \beta_d)$.

For $d \geq 2$ the values of the sequence y_d can be calculated inductively (see the bottom row of Fig. 5). There is a 1-1 correspondence between 3-cubes in a d-dimensional unit hypercube and 3-cubes in three $(d-1)$-dimensional unit hypercubes. Assign all 3^{d-1} hypercubes $C_3[0]$ to 3-cubes in the first $(d-1)$-dimensional unit hypercube, all 3^{d-1} hypercubes $C_3[1/3]$ to 3-cubes in the second $(d-1)$-dimensional unit hypercube and finally all hypercubes $C_3[2/3]$ to 3-cubes in the third $(d-1)$-dimensional unit hypercube (see Fig. 6, where $d = 3$).

If $n_2 \leq 2^{d-1}$, i.e., if no hypercube $C_2[1/2]$ was used for packing, then all 3^{d-1} hypercubes $C_3[2/3]$ remain empty. The number of empty hypercubes $C_3[1/3]$ as

Fig. 6 3-cubes $C_3[0]$, $C_3[1/3]$ and $C_3[2/3]$ in a three-dimensional cube visualised as 3 layers of two-dimensional 3-cubes in a square



well as the number of empty hypercubes $C_3[0]$ equals $y_{d-1}(n_2)$. The number of all empty 3-cubes is equal to $3^{d-1} + 2y_{d-1}(n_2)$.

If $n_2 > 2^{d-1}$, then all 2^{d-1} hypercubes $C_2[0]$ were used for packing. There is no empty hypercube $C_3[0]$ as well as there is no empty hypercube $C_3[1/3]$. Let m be an integer such that $n_2 = 2^{d-1} + m$. Obviously, exactly m hypercubes $C_2[1/2]$ were used for packing. The number of empty 3-cubes is exactly the same as in the dimension $d - 1$ and equals $y_{d-1}(m)$.

Consequently, we get the following inductive formula

$$\begin{aligned} y_d(0) &= 3^d, \\ y_d(n_2) &= 3^{d-1} + 2 \cdot y_{d-1}(n_2), \quad \text{for } n_2 < 2^{d-1} \end{aligned} \tag{1}$$

$$y_d(n_2) = y_{d-1}(m), \quad \text{for } n_2 = 2^{d-1} + m, \quad m \le 2^{d-1}. \tag{2}$$

Note that the sequence $y_d(n_2)$ is directly related to Gould's sequence, what is proved in "Appendix 1".

Now we give the relation between the number n_2 of 2-cubes used for packing and the number $n_3^- = y_d(n_2)$.

Lemma 4 *Let n_2 be the number of 2-cubes already used for packing, let U_{n_2} be the union of all 2-cubes used for packing and let n_3^- be the number of 3-cubes contained in $\mathscr{B} \setminus U_{n_2}$. For $d \ge 1$ and $n_2 \le 2^d$ we get*

$$n_2 4^d + 3^d n_3^- \ge 8^d - \frac{5}{8} \cdot 7^d.$$

Proof Let

$$R = \frac{1}{3^d}\left(8^d - \frac{5}{8} \cdot 7^d - n_2 4^d\right)$$

and let $n_3^- = y_d(n_2)$. We will show that

$$y_d(n_2) \ge R.$$

The inequality is obvious for $n_2 = 0,\ 2^{d-1},\ 2^d$ and arbitrary d.

Notice that for $d = 1$ the inequality holds true.

- For $n_2 = 0,\ y_1(0) = 3$ we have $R = \frac{1}{3}(8 - \frac{5}{8} \cdot 7 - 0 \cdot 4) = \frac{29}{24} < 3 = y_1(0)$.
- For $n_2 = 1,\ y_1(1) = 1$ we have $R = \frac{1}{3}(8 - \frac{5}{8} \cdot 7 - 1 \cdot 4) < 0 < 1 = y_1(1)$.
- For $n_2 = 2,\ y_1(2) = 0$ we have $R = \frac{1}{3}(8 - \frac{5}{8} \cdot 7 - 2 \cdot 4) < 0 = y_1(2)$.

The rest of the proof goes by induction. Let $d \ge 2$.

Case 1 $n_2 < 2^{d-2}$. By the inductive assumption,

$$\begin{aligned} y_d(n_2) &= 3^{d-1} + 2y_{d-1}(n_2) \\ &\geq 3^{d-1} + 2 \cdot \frac{1}{3^{d-1}}\left(8^{d-1} - \frac{5}{8} \cdot 7^{d-1} - n_2 4^{d-1}\right) \\ &= R + \frac{1}{3^d}\left(3^{2d-1} - 2 \cdot 8^{d-1} - 2n_2 4^{d-1} + \frac{5}{8} \cdot 7^{d-1}\right) \\ &> R + \frac{1}{3^d}\left(3^{2d-1} - 2 \cdot 8^{d-1} - 2 \cdot 2^{d-2} \cdot 4^{d-1}\right) \\ &= R + \frac{1}{3^d}\left(3 \cdot 9^{d-1} - 3 \cdot 8^{d-1}\right) > R. \end{aligned}$$

Case 2 $n_2 = 2^{d-2}$.

$$\begin{aligned} y_d(n_2) = y_d(2^{d-2}) = 3^{d-1} + 2y_{d-1}(2^{d-2}) = 3^{d-1} + 2 \cdot 3^{d-2} = \frac{5}{9} \cdot 3^d \\ > \left(\frac{8}{3}\right)^d - \frac{5}{8} \cdot \left(\frac{7}{3}\right)^d = \frac{1}{3^d}\left(8^d - \frac{5}{8} \cdot 7^d\right) > R. \end{aligned}$$

Case 3 $2^{d-2} < n_2 < 2^{d-1}$. We can assume that $d \geq 3$ (if $d = 2$, then there is no n_2 such that $2^0 < n_2 < 2^1$).

Clearly, $n_2 = 2^{d-2} + l$ for some $0 < l < 2^{d-2}$. Consequently,

$$\begin{aligned} y_d(n_2) &= 3^{d-1} + 2y_{d-1}(n_2) \\ &= 3^{d-1} + 2y_{d-1}(2^{d-2} + l) \\ &= 3^{d-1} + 2y_{d-2}(l) \\ &\geq 3^{d-1} + 2 \cdot \frac{1}{3^{d-2}}\left(8^{d-2} - \frac{5}{8} \cdot 7^{d-2} - l4^{d-2}\right) \\ &= 3^{d-1} + 2 \cdot \frac{1}{3^{d-2}}\left(8^{d-2} - \frac{5}{8} \cdot 7^{d-2} - (n_2 - 2^{d-2})4^{d-2}\right) \\ &= R + \frac{1}{3^d}\left(3 \cdot 9^{d-1} - \frac{7}{2} \cdot 8^{d-1} + \frac{155}{8} \cdot 7^{d-2} - 2n_2 4^{d-2}\right) \\ &> R + \frac{1}{3^d}\left(3 \cdot 9^{d-1} - \frac{9}{2} \cdot 8^{d-1} + 2 \cdot 7^{d-1}\right) > R. \end{aligned}$$

Case 4 $2^d - 2 \leq n_2 \leq 2^d$.

If $n_2 = 2^d$, then $y_d(n_2) = 0 > R$.
If $n_2 \in \{2^d - 2, 2^d - 1\}$, then $y_d(n_2) \geq 1 > \frac{1}{3^d}\left(8^d - \frac{5}{8} \cdot 7^d - (2^d - 2) \cdot 4^d\right) \geq R$.

Case 5 $2^{d-1} \leq n_2 \leq 2^d - 3$.

Observe that there are integers j and k such that $1 \leq j \leq d - 2$, $0 \leq k < 2^{d-j-1}$ and that

$$n_2 = 2^{d-1} + 2^{d-2} + \cdots + 2^{d-j} + k = 2^d(1 - 2^{-j}) + k.$$

Using the inductive definition of the sequence (2), i.e., $y_d(2^{d-1} + m) = y_{d-1}(m)$, for $0 < m \leq 2^{d-1}$, we get

$$\begin{aligned} y_d(n_2) &= y_d(2^{d-1} + 2^{d-2} + \cdots + 2^{d-j} + k) \\ &= y_{d-1}(2^{d-2} + \cdots + 2^{d-j} + k) \\ &= y_{d-2}(2^{d-3} + \cdots + 2^{d-j} + k) \\ &= \cdots = y_{d-j}(k) \\ &\geq y_{d-j}(2^{d-j-1} - 1) \\ &= 3^{d-j-1} + 2y_{d-j-1}(2^{d-j-1} - 1) \\ &= 3^{d-j-1} + 2. \end{aligned}$$

Since $n_2 \geq 2^d(1 - 2^{-j})$, to prove $y_d(n_2) \geq R$, is suffices to show the following inequality

$$3^{d-j-1} + 2 \geq \frac{1}{3^d}\left(8^d - \frac{5}{8}7^d - 2^d(1 - 2^{-j})4^d\right),$$

i.e.,

$$\frac{9^d}{3^{j+1}} - \frac{8^d}{2^j} + \frac{5}{8} \cdot 7^d + 2 \cdot 3^d \geq 0.$$

This is equivalent to check that

$$f_d(j) \geq 0,$$

where

$$f_d(x) = \frac{9^d}{3^{x+1}} - \frac{8^d}{2^x} + \frac{5}{8} \cdot 7^d + 2 \cdot 3^d.$$

The function f_d has a local minimum at

$$x_{\min} = \log_{2/3}\left(\left(\frac{8}{9}\right)^d 3\log_3 2\right) = d\log_{2/3}\frac{8}{9} + \log_{2/3}(3\log_3 2),$$

and

$$0.29d - 1.58 < x_{\min} < 0.3d - 1.57.$$

Moreover, f_d is continuous, decreasing on the interval $(-\infty, x_{\min}]$ and increasing on $[x_{\min}, +\infty)$.

First assume that $d \in \{2, 3, \ldots, 11\}$. It is easy to verify that $f_d(1) = 9^{d-1} - \frac{8^d}{2} + \frac{5}{8} \cdot 7^d + 2 \cdot 3^d > 0$ as well as $f_d(2) = \frac{9^{d-1}}{3} - \frac{8^d}{4} + \frac{5}{8} \cdot 7^d + 2 \cdot 3^d > 0$. Since $x_{\min} < 0.3 \cdot 11 - 1.57 < 2$, it follows that

$$f_d(j) \geq \min\bigl(f_d(1), f_d(2)\bigr) > 0$$

for $j \in \{1, 2, \dots, d-2\}$.

Now assume that $d \geq 12$. For $j \in \{1, 2 \dots, d-2\}$ we get

$$\begin{aligned} f_d(j) \geq f_d(x_{min}) &= \frac{9^d}{3 \cdot 3^{x_{\min}}} - \frac{8^d}{2^{x_{\min}}} + \frac{5}{8} \cdot 7^d + 2 \cdot 3^d \\ &> \frac{9^d}{3 \cdot 3^{0.3d} \cdot 3^{-1.57}} - \frac{8^d}{2^{0.29d} \cdot 2^{-1.58}} + 0.625 \cdot 7^d \\ &> 1.87 \cdot (6.473)^d - 2.99 \cdot (6.544)^d + 0.625 \cdot 7^d \\ &> (6.473)^d \cdot \left(1.87 - 2.99 \cdot (1.011)^d + 0.625 \cdot (1.081)^d\right) \\ &\geq (6.473)^d \cdot \left(1.87 - 2.99 \cdot (1.011)^{12} + 0.625 \cdot (1.081)^{12}\right) > 0 \end{aligned}$$

□

2.4 Competitive Ratio

Lemma 5 *Assume that only small items are in the sequence S. The total volume of small items packed into any closed bin $\mathscr{B}$ is greater than*

$$\rho = \left(1 - 5/8 \cdot (7/8)^d\right) \cdot (2/3)^d - 2^{-d} - 3^{-d}$$

for $d \geq 3$.

Proof By Lemma 3 we know that the total volume of 2-small items packed into $\mathscr{B}$ is greater than $\rho_2 n_2 \cdot 2^{-d} - 2^{-d}$. Moreover, the total volume of 3-small items packed into $\mathscr{B}$ is greater than $\rho_3 n_3 \cdot 3^{-d} - 3^{-d}$. Consequently, the sum of the volumes of packed small items is greater than

$$\begin{aligned} &(2/3)^d n_2 (1/2)^d + (3/4)^d n_3 (1/3)^d - 1/2^d - 1/3^d \\ &= 12^{-d} \cdot (n_2 \cdot 4^d + n_3 \cdot 3^d) - 1/2^d - 1/3^d. \end{aligned}$$

This value, by Lemma 4, is greater than

$$12^{-d} \cdot \left(8^d - 5/8 \cdot 7^d\right) - 1/2^d - 1/3^d = \rho.$$

□

Theorem 1 *The asymptotic competitive ratio of the tt(d) algorithm is not greater than*

$$(3/2)^d + O((21/16)^d).$$

Proof Let S be a sequence of items of the total volume v, let λ_1 denote the number of big items in S and let β be the number of bins used to pack items from S according to the $tt(d)$ algorithm.

Obviously, $OPT(S) \geq v$ as well as $OPT(S) \geq \lambda_1$. By Lemma 5 and Rule (1) from the description of the $tt(d)$ algorithm we have

$$v > \frac{1}{2^d} \cdot \lambda_1 + \rho \cdot (\beta - 2\lambda_1 - 1),$$

i.e.,

$$\beta < \frac{1}{\rho} v + \left(2 - \frac{1}{2^d \rho}\right) \lambda_1 + 1.$$

Let $\mu = \max(v, \lambda_1)$. Since $OPT(S) \geq \mu$, we get

$$\frac{\beta}{OPT(S)} \leq \frac{\beta}{\mu} < \frac{\frac{1}{\rho}\mu + \left(2 - \frac{1}{2^d \rho}\right)\mu + 1}{\mu} = \frac{1}{\rho}\left(1 - \frac{1}{2^d}\right) + 2 + \frac{1}{\mu}.$$

Consequently, the asymptotic competitive ratio for $tt(d)$ algorithm is not greater than

$$\frac{1}{\rho}\left(1 - \frac{1}{2^d}\right) + 2 = \left(\frac{3}{2}\right)^d + \left(\frac{21}{16}\right)^d \cdot \left(\frac{\frac{5}{8} + (\frac{6}{7})^d}{1 - \frac{5}{8} \cdot (\frac{7}{8})^d - (\frac{3}{4})^d - (\frac{1}{2})^d} + 2\left(\frac{16}{21}\right)^d \right).$$

□

3 The *har*(*d*) Algorithm for $d \geq 5$

In this section we describe a one-space bounded algorithm for packing d-dimensional hypercubes with competitive ratio $O(d/\log d)$. The $har(d)$ algorithm is based on the well-known Harmonic algorithm as well as on the $tt(d)$ algorithm and thus some definitions, notations and even lemmas are similar to those related to the $tt(d)$ algorithm. For instance, if all the hypercubes have the edge lengths from the set $\{1/2, 1/4, 1/8, 1/16, \dots\}$, then the algorithm works exactly the same as the $tt(d)$ algorithm. Some of the presented bounds are not sharp and can be improved; we preferred non-optimal constants rather than complicated calculations.

By $a_1 \times a_2 \times \cdots \times a_d$ we mean a hyperbox such that its edges parallel to the k-th axis of the coordinate system are of the length a_k, for $k = 1, 2, \dots, d$. Furthermore, we will write $a^{d-q} \times b^q$ instead of $\underbrace{a \times \cdots \times a}_{(d-q)\ \text{times}} \times \underbrace{b \times \cdots \times b}_{q\ \text{times}}$.

3.1 Intuition About How the Algorithm Works

Different types of items are packed into different active bins by the Harmonic algorithm. Since any unit d-dimensional hypercube can be divided into

2^d hypercubes of the edge length 1/2 (we called such hypercubes 2-cubes, see Sect. 2.2), our first idea for packing with one active bin only was to place different items into different 2-cubes instead of into many active bins. Unfortunately, the losses were too large.

For example, any unit hypercube can be partitioned into hypercubes of the edge length 1/5, but a 2-cube cannot. Any 2-cube contains only 2^d hypercubes of the edge length equal to 1/5. Consequently, only $(4/5)^d$ of any 2-cube is occupied by such hypercubes and this ratio is close to zero for large dimensions d. If hypercubes were packed into $B_1 = 1^{d-1} \times (1/2)$, then 4/5 of B_1 would be occupied. Unfortunately, each unit hypercube contains only two hyperboxes congruent to B_1. Therefore items should be packed into hyperboxes smaller than B_1 but larger than 2-cubes.

In the online version of packing we do not know anything about the size of incoming items, so we should prepare an empty space to pack items of different sizes. For the items of the edge length $1/t$, where $t = 5, 7, 9, 11, \dots$ we will reserve an empty space in special hyperboxes $1^{d-q} \times (1/2)^{q-1} \times (1/t)$, called layers of the height $1/t$. Let us mention that $q = q(t)$ depends on t and that $q(t_1) \geq q(t_2)$, provided $t_1 \geq t_2$.

For example, the items of the edge length 1/5 will be packed into layers $1^{d-4} \times (1/2)^3 \times (1/5)$. Such a layer contains $5^{d-4} \cdot 2^3$ hypercubes of the edge length 1/5. This means that 64/125 of the layer is occupied by the hypercubes and this ratio does not depend on d. The items of the edge length 1/t, for $t \in \{7, 9\}$, will be packed into layers $1^{d-4} \times (1/2)^3 \times (1/t)$. Since $\frac{1}{5} + \frac{1}{7} + \frac{1}{9} < \frac{1}{2}$, it is possible to create three layers (of pairwise disjoint interiors) of the height 1/5, 1/7 and 1/9 in any hyperbox $1^{d-4} \times (1/2)^4$. At least 64/125 of any layer of the height $1/t$ is occupied by hypercubes of the edge length $1/t$. Similarly, it is possible to create eight layers $1^{d-5} \times (1/2)^4 \times (1/t)$, for $t = 11, 13, \dots, 25$, in any hyperbox $1^{d-5} \times (1/2)^5$ (we have $\frac{1}{11} + \frac{1}{13} + \cdots + \frac{1}{23} + \frac{1}{25} < \frac{1}{2}$) . In a similar way, layers $1^{d-6} \times (1/2)^5 \times (1/t)$, for $t = 27, 29, \dots, 69$, will be created in $1^{d-6} \times (1/2)^6$. Generally, layers of small height will be created in hyperboxes with comparatively large number of edges of the length 1/2. The number of such hyperboxes in the active bin is relatively large. In any case, despite many edges of the layer are of the length 1/2, at least 64/125 of any layer of the height $1/t$ is occupied by hypercubes of the edge length $1/t$.

Items (d-dimensional hypercubes) will first be packed into hypercubes, called m-cubes, of the edge lengths $1/2, 1/3, 1/4, 1/5, 1/6, 1/7, \dots$ in our main method. For each m-cube we describe a place in the bin into which this hypercube will be packed. First, we will assign an open B-box (the union of some 2-cubes). Then we will determine the right place in the open B-box to pack the hypercube. Some of m-cubes will be contained in layers created in open B-boxes. The hypercubes of the edge length 1/2 will be packed into open 2-cubes. The hypercubes of the edge length 1/3 will be placed into open B-boxes $1^{d-3} \times (1/2)^3$. Moreover, the items of the edge lengths $1/(2t)$ will be packed next to the items of the edge lengths $1/t$, for $t = 2, 3, \dots$.

All open hyperboxes have pairwise disjoint interiors. Moreover, all created layers have pairwise disjoint interiors.

Since hypercubes will be packed into layers $1^{d-q} \times (1/2)^{q-1} \times (1/t)$, where $q \geq 4$, we will assume that

$$d \geq 5$$

in this section. Let us add that it is possible to modify the algorithm and create layers also in smaller dimensions. However, such a modified algorithm would not be effective. Even for $d = 5$ the upper bound presented in Theorem 2 is distinctively worse than the one given in the proof Theorem 1. The aim of this paper is to present a one-space bounded algorithm efficient for large d.

Example Incoming items will be packed into open hyperboxes of a proper size. Let $d = 5$. At the beginning of the packing process three hyperboxes

$$H_1 = [0, 1/2]^5,\ H_2 = [0, 1] \times [1/2, 1] \times [0, 1/2]^3 \text{ and } H_3 = [0, 1]^2 \times [1/2, 1] \times [0, 1/2]^2$$

are open in the active bin; the place of the bin into which these hyperboxes are contained is determined by the *box*(d) algorithm described in Sect. 3.2. H_1 will be called BC_2-box in Sect. 3.6; hypercubes of the edge length $1/2, 1/4, 1/8, \ldots$ will be packed into it. H_3 will be called BC_3-box; hypercubes of the edge length $1/3, 1/6, 1/12, \ldots$ will be packed into it. H_2 will be called BL_4-box; three layers of the height 1/5, 1/7 and 1/9 are created in it.

The items of the edge lengths

$$1/4, 1/3, 1/8, 1/5, 1/7, 1/2, 10/51$$

are packed as on Fig. 7. The first item is placed into H_1. We will use the $aux^+(5)$ algorithm described in Sect. 3.4: the first item is packed in the first 2-subcube of the edge length 1/4 contained in H_1. The second item is placed into H_3. We will use the $aux^+(5)$ algorithm described in Sect. 3.4: the item is packed in the first subcube of the edge length 1/3 contained in H_3. The third item is packed into H_1 by the $aux^+(5)$ algorithm; it is packed into the first empty 2-subcube of H_1 of the edge length 1/4 (see Fig. 7, left). The fourth item is packed in H_2 into the layer $[0, 1] \times [1/2, 1] \times [0, 1/2]^2 \times [0, 1/5]$ of the height 1/5 (see Fig. 7, right). The

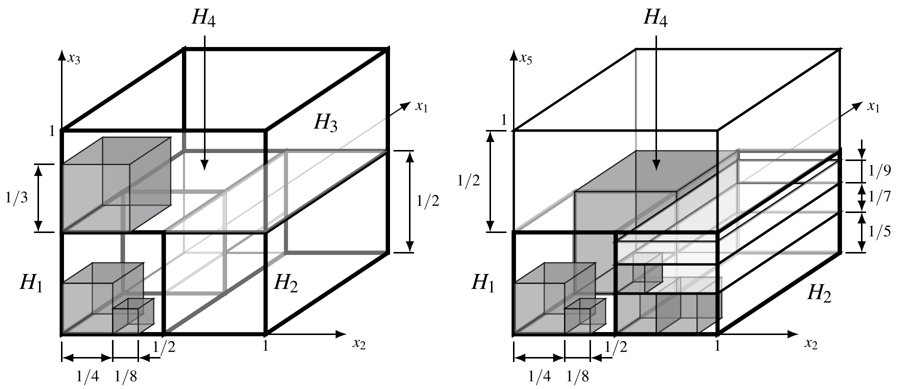


Fig. 7 Items in the bin

place in the layer into which this item is packed is determined by the $aux^{+}(5)$ algorithm described in Sect. 3.4, just the item is packed into the first hypercube of the edge length 1/5 contained in this layer. The next item is placed into the layer $[0,1] \times [1/2,1] \times [0,1/2]^2 \times [1/5, 1/5+1/7]$ contained in H_2. The place in the layer into which this item is packed is determined by the $aux^{+}(5)$ algorithm: just the item is packed into the first hypercube of the edge length 1/7 contained in this layer (see Fig. 7, right). There is no empty space in H_1 to pack the penultimate item of the edge length 1/2. We open a new BC_2-box $H_4 = [1/2,1] \times [0,1/2]^4$ (this place is determined by the *box*(d) algorithm) and pack the item into H_4. The last item is packed into the layer $[0,1]^{d-4} \times [1/2,1] \times [0,1/2]^2 \times [0,1/5]$ contained in H_2. The place in the layer into which this item is packed is determined by the *har*(5) algorithm; just the item is packed into the second hypercube of the edge length 1/5 contained in this layer.

Let us add that if all subsequent items in the sequence (eighth, ninth, ...) have the edge lengths 1/7, then $7 \cdot 3^3 - 1$ of them are packed in the layer $[0,1] \times [1/2,1] \times [0,1/2]^2 \times [1/5, 1/5+1/7]$ contained in H_2. Then a new BL_4-box is opened and three layers of the height 1/7 are created in it. Details of this method are described in the next subsections.

3.2 *B*-Boxes

As it was said in Sect. 3.1, items will be packed into open hyperboxes with pairwise disjoint interiors. Any hypercube of the edge length 1/2 was packed in the empty 2-cube with the smallest number by the *tt*(d) algorithm. We will present a similar method for hyperboxes that are the union of 2-cubes.

By a B_q-*box* (a *B-box*, for short), where $q \in \{0, \dots, d\}$, we mean $1^{d-q} \times (1/2)^q$ (see Fig. 8). Any B_q box is the union of 2^{d-q} many 2-cubes.

For each $q \in \{0, \dots, d\}$ the bin $\mathscr{B}$ is partitioned into 2^q congruent B_q-boxes that are numbered as follows. The entire bin $\mathscr{B} = [0,1]^d$ is a B_0-box and is numbered with 1. Two B_1-boxes contained in $\mathscr{B}$, i.e., hyperboxes $[0,1]^{d-1} \times [0,1/2]$ and $[0,1]^{d-1} \times [1/2,1]$, are numbered 1 and 2, respectively. Finally, let $k < d$ and let

$$X = [0,1]^k \times [\gamma_{k+1}, \gamma_{k+1} + 1/2] \times \cdots \times [\gamma_d, \gamma_d + 1/2],$$

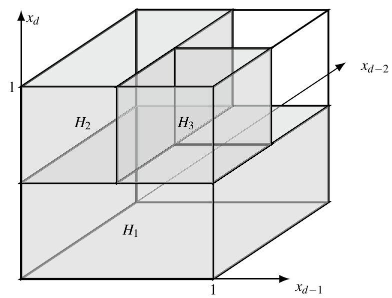


Fig. 8 B_q-boxes in the bin

where $\gamma_{k+1}, \dots \gamma_d \in \{0, 1/2\}$, be a B_k-box numbered λ. X is partitioned into two congruent B_{k+1}-boxes:

$$[0,1]^{k-1} \times [0,1/2] \times [\gamma_{k+1}, \gamma_{k+1} + 1/2] \times \cdots \times [\gamma_d, \gamma_d + 1/2],$$
$$[0,1]^{k-1} \times [1/2,1] \times [\gamma_{k+1}, \gamma_{k+1} + 1/2] \times \cdots \times [\gamma_d, \gamma_d + 1/2]$$

with numbers $2\lambda - 1$ and 2λ, respectively.

For example, $H_1 = [0,1]^{d-1} \times [0,1/2]$ on Fig. 8 is the B_1-box with number 1 and number 2 is assigned to the other B_1-box $\mathscr{B} \backslash H_1$. Moreover, H_1 contains: two B_2-boxes with numbers 1 and 2 as well as four B_3-boxes with numbers 1, 2, 3 and 4, eight B_4-boxes with numbers $1, 2, \dots, 8$ and so on. $H_2 = [0,1]^{d-2} \times [0,1/2] \times [1/2,1]$ is the B_2-box with number 3 and number 4 is assigned to the B_2-box $\mathscr{B} \backslash (H_1 \cup H_2)$. $H_3 = [0,1]^{d-3} \times [0,1/2] \times [1/2,1]^2$ is the B_3-box with number 7 and number 8 is assigned to the B_3-box $\mathscr{B} \backslash (H_1 \cup H_2 \cup H_3)$.

Incoming items will be packed into open B-boxes by the *har*(d) algorithm described in Sect. 3.8. If there is no empty space, then we will open a new B_q-box in the place described by the following algorithm.

Let $H_1, H_2, \dots$ be a sequence of B-boxes. A B-box of $\mathscr{B}$ is called *empty* if its interior is disjoint with any open B-box.

Algorithm *box*(d)

- the first open B-box is the first B-box of $\mathscr{B}$ that is congruent to H_1;
- if B-boxes of $\mathscr{B}$ congruent to $H_1, \dots, H_{j-1}$ are open, then the the next open B-box is the empty B-box of $\mathscr{B}$ with the smallest number that is congruent to H_j.

For example, B_d-box H_1, B_{d-1}-box H_2, B_{d-2}-box H_3 and B_d box H_4 are open in the active bin by *box*(d) in places shown on Fig. 7 ($d = 5$). H_1 is the first B_d-box, H_2 is the second B_{d-1}-box, H_3 is the second B_{d-2}-box and H_4 is the second B_d-box in the active bin.

Another example is illustrated on Fig. 8, where hyperboxes: B_1-box H_1, B_2-box H_2 and B_3-box H_3 are open in places described by *box*(d). H_1 is the first B_1-box, H_2 is the third B_2-box and H_3 is the seventh B_3-box in the active bin.

Lemma 6 *Let $H_1, H_2, \dots$ be a sequence of B -boxes. If B-boxes congruent to H_1, $\dots, H_{z-1}$ are open in $\mathscr{B}$ and if there is no empty space in $\mathscr{B}$ to open a new B-box congruent to H_z, then $\sum_{i=1}^{z-1} |H_i| > 1 - |H_z|$.*

Proof H_z is a B_q-box for some $q \in \{0, \dots, d\}$. Obviously, $|H_z| = 2^{-q}$. Since it is not possible to open a B-box congruent to H_z, there are no empty B_k-boxes in $\mathscr{B}$ for $k \le q$. There can be at most one empty B_{q+1}-box. Otherwise suppose there are two empty B_{q+1}-boxes X_1 and X_2. These B_{q+1}-boxes of $\mathscr{B}$ are not contained in one B_q-box, because that would make a whole empty B_q-box in $\mathscr{B}$. Therefore X_1 and X_2 lay in different B_q-boxes, say X_2 is in a B_q-box with a higher number than X_1 is. Let X_2' be the complementary half of X_2 in a B_q-box they lay in. Since there is no empty B_q-box, X_2' contains an open B-box. It means that a B_{q+1}-box X_2' contains an open B-box and has higher number than X_1 that is empty. This contradicts the *box*(d) algorithm. Therefore there is at most one empty B_{q+1}-box in $\mathscr{B}$.

Using the above argument we obtain at most one empty B_k-box of $\mathscr{B}$ for each $k \in \{q+1, \dots, d\}$ with the interior disjoint with other empty B-boxes of $\mathscr{B}$.

Finally, the empty space in $\mathscr{B}$ does not exceed

$$\sum_{k=q+1}^{d} 2^{-k} = 2^{-q} - 2^{-d} < 2^{-q} = |H_z|.$$

□

3.3 Layers and *m*-Cubes

To prepare an empty space for incoming items of different sizes, we will create layers of the height 1/t, for $t = 5, 7, 9, \dots$ in open B_q-boxes. Items of the edge lengths 1/t, 1/(2t), $1/(4t), \dots$ will be placed in layers of the height 1/t.

Let $10 \le m \le 2^{d-1}$ be an even number and let $q \in \{1, \dots, d\}$.

The integer m determines the number of layers of different heights. For fixed m only layers of the height $1/5, 1/7, 1/9, \dots, 1/(m-1)$ can be created. For example, $d = 5$ and $m = 10$ on Fig. 7 and only layers of the height 1/5, 1/7 and 1/9 were created. Ultimately, we will take m close to $4d/\log d$ in the proof of Theorem 2.

By an m-cube K^+ we mean a hypercube of the edge length $1/(t \cdot 2^p)$, where $p \ge 0$ is an integer and $t \in \{1, 2, \dots, m\}$. We say then that K^+ is a cube *of type* (t, p). When the value of p can be arbitrary we say that K^+ is a cube *of class* t.

For instance, 6-cubes have the edge lengths from the set

$$\left\{1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}, \frac{1}{6}, \frac{1}{8}, \frac{1}{10}, \frac{1}{12}, \frac{1}{16}, \frac{1}{20}, \frac{1}{24}, \dots\right\}$$

while cubes of class 6 have the edge lengths from the set $\left\{\frac{1}{6}, \frac{1}{12}, \frac{1}{24}, \dots\right\}$.

Let us note that any cube of type $(2^n \cdot t, p)$ is also of type $(t, p+n)$. This fact will be used in the description (Rule 5) of the *har*$^+$(d) algorithm. For instance, the 6-cube of the edge length $\frac{1}{32} = \frac{1}{2^1 \cdot 2 \cdot 2^3} = \frac{1}{2 \cdot 2^{3+1}}$ is of type $(2^1 \cdot 2, 3)$ as well as $(2, 3+1)$.

By a (t, q)*-layer* we mean $1^{d-q} \times (1/2)^{q-1} \times (1/t)$; the value 1/$t$ is called the *height* of the layer (see Fig. 9).

Clearly, each (2, q)-layer is a B_q-box. We emphasize the fact that in the process of packing we will only use layers for t being odd. In the main packing method q depends on t (for detailed description see Sect. 3.5): if $t \in \{5, 7, 9\}$, then $q = 4$; if $t = 11$, then $q = 5$; if $t = 111$, then $q = 6, \dots$.

For example, (5, 4)-layer, (7, 4)-layer and (9, 4)-layer were created on Fig. 7. The first layer contains $5 \cdot 2^3$ hypercubes of the edge length 1/5. The first and the second hypercube are used for packing. The second layer contains $7 \cdot 3^3$ hypercubes of the edge length 1/7. The first hypercube is used for packing.

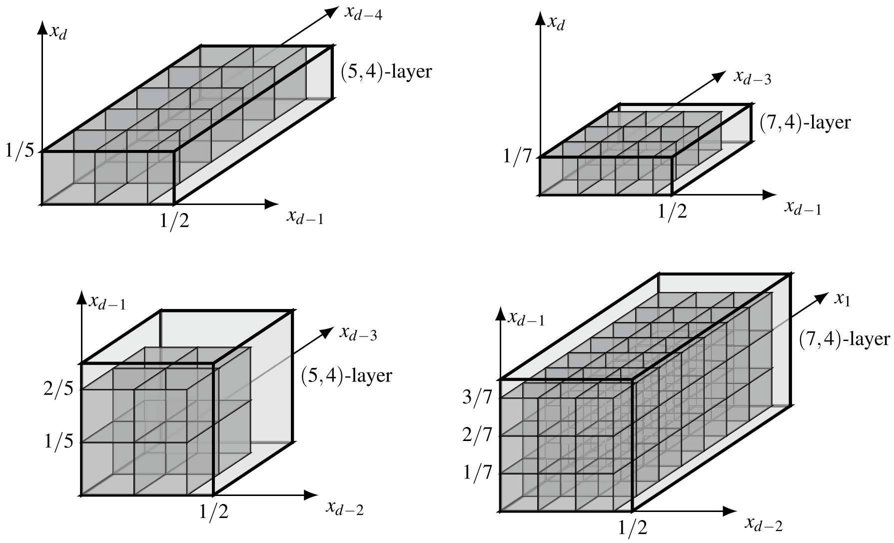


Fig. 9 Layers filled with m-cubes

It is easy to see that for an odd number t any (t, q)-layer contains $\left(\frac{1}{2}t - \frac{1}{2}\right)^{q-1} \cdot t^{d-q}$ cubes of type $(t, 0)$ (see Fig. 9, where (5, 4)-layer contains $2^3 \cdot 5^{d-4}$ hypercubes of the edge length 1/5, while (7, 4)-layer contains $3^3 \cdot 7^{d-4}$ hypercubes of the edge length 1/7).

Lemma 7 *If* $q \geq 4$ *and if* $t \geq 2^{q-2} + 1$, *then at least* 64/125 *of any* (t, q)*-layer is occupied by cubes of type* $(t, 0)$.

Proof $\left(\frac{1}{2}t - \frac{1}{2}\right)^{q-1} \cdot t^{d-q}$ cubes of type $(t, 0)$ are contained in each (t, q)-layer. The volume of any (t, q)-layer equals $2^{1-q} \cdot t^{-1}$. Observe that

$$\frac{\left(\frac{1}{2}t - \frac{1}{2}\right)^{q-1} \cdot t^{d-q} \cdot t^{-d}}{2^{1-q} \cdot t^{-1}} = \left(1 - \frac{1}{t}\right)^{q-1}.$$

Since $t \geq 2^{q-2} + 1$, we get

$$\left(1 - \frac{1}{t}\right)^{q-1} \geq \left(1 - \frac{1}{2^{q-2} + 1}\right)^{q-1}.$$

If $q = 4$, then $(1 - \frac{1}{2^2+1})^3 = \frac{64}{125}$. If $q \geq 5$, then

$$\left(1 - \frac{1}{2^{q-2} + 1}\right)^{q-1} = \frac{1}{\left(1 + \frac{1}{2^{q-2}}\right)^{q-1}} > e^{-\frac{q-1}{2^{q-2}}} \geq e^{-\frac{4}{8}} > \frac{64}{125}.$$

This implies that at least 64/125 of any (t, q)-layer is occupied by cubes of type $(t, 0)$. □

3.4 Auxiliary Algorithm $aux^+(d)$

If all items have the edge lengths from the set $\{1/2, 1/4, 1/8, \dots\}$, then we will use the $tt(d)$ algorithm described in Sect. 2. Each item will be packed into the proper subcube with the smallest number contained in the union of hypercubes of the edge length 1/2. Similarly, each item of the edge length slightly smaller than either 1/2 or 1/4 or 1/8, ...will be packed into the proper subcube with the smallest number contained in the union of hypercubes of the edge length 1/2. In the main packing method the items of the edge length slightly smaller than either $1/t$ or $1/(2t)$ or $1/(4t)$, ...will be placed into layers of the height $1/t$ (for odd $t \geq 5$). More precisely, the items will be packed into the union of hypercubes of the edge length $1/t$ contained in layers of the height $1/t$. Therefore we need an algorithm that packs items into hypercubes. Our algorithm is similar to the $tt(d)$ algorithm.

Let $10 \leq m \leq 2^{d-1}$ be an even number and let either $3 \leq t \leq m-1$ be an odd number or $t = 2$. Moreover let p be a non-negative integer.

Consider the union U_T of cubes of type $(t, 0)$ with pairwise disjoint interiors. All cubes are numbered with successive positive integers (compare Fig. 2, left, where $t = 2$). Furthermore, 2^d cubes of type (t, p) contained in the cube of type $(t, p-1)$ with number η are numbered in an arbitrary order from $2^d(\eta - 1) + 1$ to $2^d\eta$.

Consider a sequence $K_1^+, K_2^+, \dots$ of cubes of class t. We say that a cube of type (t, p) of U_T is *empty* if its interior has an empty intersection with any cube packed so far.

Algorithm $aux^+(d)$ **for packing of** K_i^+ **into** U_T.

- pack K_i^+ into the empty hypercube of U_T with the smallest number that is congruent to K_i^+.

If U_T is the unit hypercube partitioned into the union of 2^d many 2-cubes and if $t = 2$, then $aux^+(d)$ works exactly the same as the $tt(d)$ algorithm. For instance, the first item from the Example of Sect. 3.1 is packed by the $aux^+(d)$ algorithm into the first hypercube of the edge length 1/4; the third item is packed into the first empty hypercube of the edge length 1/8. The second item is packed by the $aux^+(d)$ algorithm into the first hypercube of the edge length 1/3 contained in H_3. The third item is packed by the $aux^+(d)$ algorithm into the first hypercube of the edge length 1/5 contained in the proper layer (see Fig. 7).

Lemma 8 *Let S_t be a sequence of cubes of class t. If K_z^+ is the first hypercube from S_t that cannot be packed in U_T by $aux^+(d)$ algorithm, then the total volume of hypercubes packed in U_T plus the volume of K_z^+ is greater than $|U_T|$.*

Proof We proceed as in the proof of Lemma 3. Since K_z^+ is a cube of class t, there is $p \geq 0$ such that K_z^+ is a cube of type (t, p). Clearly, there is no empty cube of type (t, q) in U_T for any $q \leq p$. Furthermore, for each $q > p$ there are at most $2^d - 1$ empty cubes of type (t, q). The total volume of empty cubes in U_T is smaller than

$$(2^d - 1) \cdot (2^{-d} + 4^{-d} + \cdots) \cdot \left(t \cdot 2^p\right)^{-d} = \left(t \cdot 2^p\right)^{-d}.$$

This implies that the sum of the volumes of all cubes packed into U_T is greater than $|U_T| - \left(t \cdot 2^p\right)^{-d}$. By $|K_z^+| = (t \cdot 2^p)^{-d}$, it follows that the total volume of items packed in U_T is greater than $|U_T| - |K_z^+|$. □

3.5 Layers in *B*-Boxes

Since the series $\sum_{j=2}^{\infty} \frac{1}{2j-1}$ is divergent, there is no space in the bin to create too many layers of different heights.

Clearly, the sum of the heights of layers created in any B_q-box (of the height 1/2) is not greater than 1/2.

Since $\frac{1}{5} + \frac{1}{7} + \frac{1}{9} \leq \frac{1}{2}$, it follows that three layers: $(5, q)$, $(7, q)$ and $(9, q)$ can be created in one B_q-box. Similarly, $\frac{1}{11} + \frac{1}{13} + \cdots + \frac{1}{25} \leq \frac{1}{2}$ implies that eight layers: $(11, q), (13, q), \ldots, (25, q)$ can be created in one B_q-box.

Now we will estimate the number of layers of different heights that can be created in B_q-boxes.

Let $n_3 = 3$ and, for $k = 4, 5 \ldots$, let n_k be the greatest odd number such that

$$\frac{1}{n_{k-1} + 2} + \frac{1}{n_{k-1} + 4} + \cdots + \frac{1}{n_k} \leq \frac{1}{2}.$$

This means that the layers of the height $n_{k-1} + 2, n_{k-1} + 4, \ldots, n_k$ can be created in one B_q-box for any $k = 3, 4, \ldots$. It is easy to check that $n_4 = 9$, $n_5 = 25$, $n_6 = 69$, $n_7 = 189$, $n_8 = 515$, $n_9 = 1401$ and $n_{10} = 3809$. By the definition of n_k we know that the layers: $(5, q)$, $(7, q)$, $\ldots, (n_k, q)$ can be accommodated into the union of $k - 3$ many B_q-boxes. It is enough now to give a lower bound for n_k (see Lemma 9).

During the packing process $(t,\ q(t))$-layers (for odd $t \geq 5$) will be created in $B_{q(t)}$-boxes (see Sect. 3.6). Let us add that for large t the integer $q = q(t)$ will be comparatively large, i.e., hypercubes of the edge length $1/t$ will be packed into layers with comparatively large number of edges of the length 1/2. The connection between t and $q(t)$ is as follows. Given an odd number $t \geq 5$, let $q(t)$ be an integer such that

$$n_{q(t)-1} < t \leq n_{q(t)}.$$

For example,

$q(5) = q(7) = q(9) = 4$ $(n_{4-1} = 3 < 5 < 7 < 9 = n_4)$;
$q(11) = q(13) = \cdots = q(25) = 5$ $(n_{5-1} = 9 < 11 < \cdots < 25 = n_5)$,
$q(27) = \cdots = q(69) = 6$ $(n_6 = 69)$.

Lemma 9 *For any integer* $l \geq 4$ *we get* $n_l \geq 2^{l-1} - 1$.

$$\sum_{j=3}^{\infty} \frac{1}{2j-1} = \left(\frac{1}{5}+\frac{1}{7}\right)+\left(\frac{1}{9}+\frac{1}{11}+\frac{1}{13}+\frac{1}{15}\right)+\left(\frac{1}{17}+\cdots+\frac{1}{31}\right)+\cdots$$
$$+\left(\frac{1}{2^{l-1}+1}+\cdots+\frac{1}{2^{l}-1}\right)+\cdots$$
$$< \underbrace{\left(\frac{1}{4}+\frac{1}{4}\right)}_{2^1 \text{ times}}+\underbrace{\left(\frac{1}{8}+\frac{1}{8}+\frac{1}{8}+\frac{1}{8}\right)}_{2^2 \text{ times}}+\underbrace{\left(\frac{1}{16}+\cdots+\frac{1}{16}\right)}_{2^3 \text{ times}}+\cdots$$
$$+\underbrace{\left(\frac{1}{2^{l-1}}+\cdots+\frac{1}{2^{l-1}}\right)}_{2^{l-2} \text{ times}}+\cdots = \frac{1}{2}+\frac{1}{2}+\frac{1}{2}+\cdots$$

Proof

This implies that: $n_4 \geq 7 = 2^3 - 1,\ n_5 \geq 15 = 2^4 - 1, \ldots,\ n_{l+1} \geq 2^l - 1.$ □

3.6 Description of the *har*$^+$(*d*) Algorithm

The main algorithm is presented in two steps. First, we give the algorithm $har^{+}(d)$ that packs only m-cubes into the unit cube $\mathscr{B}$. Then we provide the final algorithm $har(d)$: a method of packing items of arbitrary sizes into $\mathscr{B}$. Simply we will pack any item into the smallest possible m-cube and then this m-cube will be packed into $\mathscr{B}$.

Let $d \geq 5$. Moreover, let $10 \leq m \leq 2^{d-1}$ be an even number.

Since some m-cubes will be packed into $B_{q(m-1)}$-boxes, i.e., hypercubes $1^{d-q(m-1)} \times (1/2)^{q(m-1)}$, first we show that

$$q(m-1) \leq d.$$

By the choice of $q(t)$ (see Sect. 3.5) we know that $n_{q(m-1)-1} < m - 1 \leq n_{q(m-1)}$. By Lemma 9 we get

$$m - 1 > n_{q(m-1)-1} > 2^{q(m-1)-2} - 1.$$

Since $2^{d-1} \geq m$, we have $2^{d-1} > 2^{q(m-1)-2}$, i.e., $q(m-1) < d + 1$. Consequently, $q(m-1) \leq d$. This implies that it is possible to create a $B_{q(m-1)}$-box in $\mathscr{B}$.

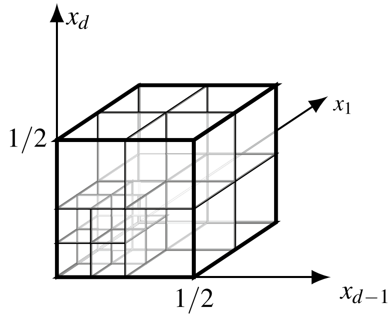
We distinguish two classes of B-boxes based on the size of items packed into them:

- *BC-boxes* (see Figs. 10 and 11);
- *BL-boxes* in which layers will be created (see Fig. 12).

Each time a B-box is opened it is assigned to one of those classes and we mention whether it is a BC- or a BL-box.

At the beginning of the process of packing the following B-boxes are open in $\mathscr{B}$:

Fig. 10 B_d-box (BC_2-box)



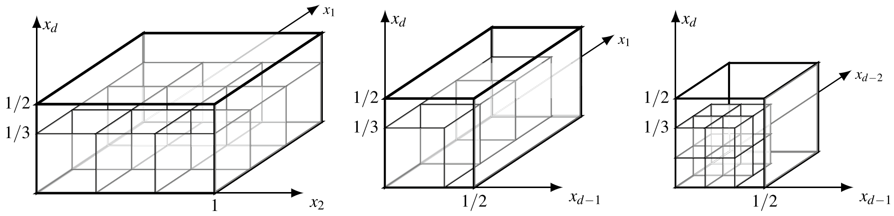


Fig. 11 B_3-box (BC_3-box)

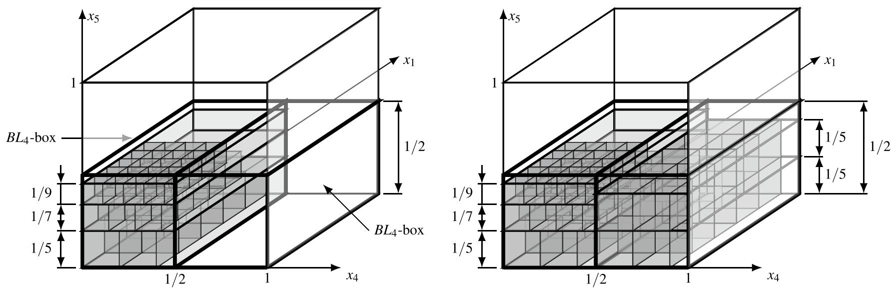


Fig. 12 Layers in BL_4-boxes for $d = 5$

- two BC-boxes: one B_d-box called BC_2-box into which hypercubes of the edge lengths $1/2, 1/4, 1/8, \ldots$ will be packed and one B_3-box called BC_3-box into which hypercubes of the edge lengths $1/3, 1/6, 1/12, \ldots$ will be packed;
- $q(m-1)-3$ many BL-boxes: one B_k-box called BL_k-box, for each $k = 4, 5, \ldots, q(m-1)$.

The place in the bin into which these $q(m-1)-1$ many B-boxes are contained is described by the *box*(d) algorithm. Just we open: one B_d-box, one B_3-box, one B_4 -box, ...and one $B_{q(m-1)}$-box in $\mathscr{B}$ in places determined by the *box*(d) algorithm (see Fig. 7, where $m = 10$ and $q(m-1) = q(9) = 4$).

Into BL_4-boxes the hypercubes of the edge lengths $1/5, 1/7, 1/9, 1/10, 1/14, 1/18, 1/20, 1/28, 1/36, \dots$ will be packed. The hypercubes of the edge lengths $1/11, 1/13, \dots, 1/25, 1/22, 1/26, \dots$ will be placed into BL_5-boxes. Into BL_s-boxes hypercubes of the edge lengths

$$1/(n_{s-1}+2), 1/(n_{s-1}+4), \dots, 1/n_s, 1/(2n_{s-1}+4), 1/(2n_{s-1}+8), \dots, 1/(2n_s), \dots$$

will be packed, for $s = 4, 5, \dots, q(m-1)$. The number of open B-boxes increases during the packing process.

Note that it is possible to pack m-cubes of types (2, 0) and (3, 0) into one B-box, as in the $tt(d)$ algorithm, however we will pack these cubes separately to simplify the calculations.

Three layers are created in the first open BL_4-box: one (5, 4)-layer, one (7, 4)-layer and one (9, 4)-layer as on Fig. 12. Let us note that $5 = n_3 + 2$ and $9 = n_4$. Then, depending on the size of incoming items it is possible that in the second open BL_4-box only two (5, 4)-layers were created as on Fig. 12, right. It is also possible that in the second open BL_4-box only one (5, 4)-layer and one (9, 4)-layers were created (see Fig. 13).

Similarly, in the first BL_k-box ($k \in \{4, 5, \dots, q(m-1)\}$) the following layers are created: one $(n_{k-1}+2, k)$-layer, one $(n_{k-1}+4, k)$-layer, ⋯, one (n_k, k)-layer.

We will use the algorithm $aux^+(d)$, therefore we enumerate m-cubes contained in layers. All cubes of type (t,0) contained in the created (t, q)-layer with number ξ are numbered in an arbitrary order from $\left(\frac{1}{2}t - \frac{1}{2}\right)^{q-1} \cdot t^{d-q} \cdot (\xi - 1) + 1$ to $\left(\frac{1}{2}t - \frac{1}{2}\right)^{q-1} \cdot t^{d-q} \cdot \xi$. Furthermore, 2^d congruent hypercubes into which the cube of type (t, p) with number λ can be partitioned are numbered in an arbitrary order from $2^d(\lambda - 1) + 1$ to $2^d\lambda$.

Algorithm $har^+(d)$ **for packing of** K_i^+ **into** $\mathscr{B}$.

Let K_i^+ be a cube of type (t, p). All B_q-boxes of $\mathscr{B}$ are numbered as in Sect. 3.2.

1. If $(t, p) = (1, 0)$, i.e., K_i^+ is a hypercube of the edge length 1, then we close the active bin, open a new bin, pack K_i^+ and close the bin. Then we open a new active bin.

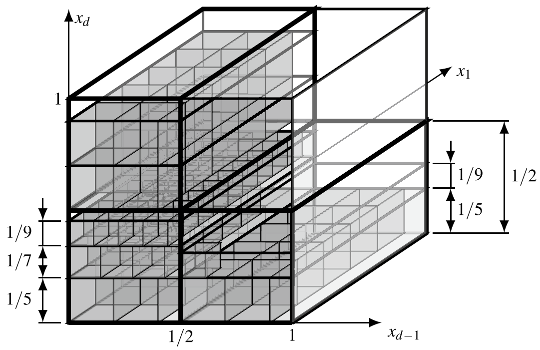


Fig. 13 Layers in three BL_4 -boxes

2. If $(t, p) = (2^j, n)$, for some $j \geq 1$ and $n \geq 0$, then K_i^+ is packed into the union of cubes of type (2, 0) contained in the union of open BC_2-boxes according to the $aux^+(d)$ algorithm.

   If it is impossible, we open in $\mathscr{B}$ a new BC_2-box: an empty B_d-box with the smallest number (compare the $box(d)$ algorithm). If there were v open BC_2-boxes so far, then the number assigned to the cube of type (2, 0) that is the open BC_2 -box is $v + 1$. We pack K_i^+ into this cube by the $aux^+(d)$ algorithm.
3. If $(t, p) = (2^j \cdot 3, n) = (3, j + n)$, for some $j,\ n \geq 0$, then K_i^+ is packed into the union of cubes of type (3, 0) contained in open BC_3-boxes according to the $aux^+(d)$ algorithm.

   If it is impossible, we open in $\mathscr{B}$ a new BC_3-box: an empty B_3-box with the smallest number (compare the $box(d)$ algorithm). If there were w open BC_3-boxes so far, then the cubes of type (3, 0) contained in the open BC_3-box are numbered from $w \cdot 3^{d-3} + 1$ to $(w + 1) \cdot 3^{d-3}$. We pack K_i^+ by the $aux^+(d)$ algorithm.
4. If t is an odd number greater then 3 and $p \geq 0$ is arbitrary, then K_i^+ is packed into the union of cubes of type $(t, 0)$ contained in the union of $(t, q(t))$-layers created in open $BL_{q(t)}$-boxes by $aux^+(d)$ algorithm.

   If it is impossible, we create a new $(t, q(t))$-layer in an open $BL_{q(t)}$-box (as low with respect to x_d-axis as possible). If there were u created layers of the height $1/t$ so far, then the cubes of type $(t, 0)$ contained in the created layer are numbered from $u \cdot \left(\frac{1}{2}t - \frac{1}{2}\right)^{q(t)-1} \cdot t^{d-q(t)} + 1$ to $(u + 1) \cdot \left(\frac{1}{2}t - \frac{1}{2}\right)^{q(t)-1} \cdot t^{d-q(t)}$. We pack K_i^+ by the $aux^+(d)$ algorithm. If it is impossible, we open a new $BL_{q(t)}$-box: an empty $B_{q(t)}$-box with the smallest number (compare the $box(d)$ algorithm) and create at the bottom of this box (with respect to x_d-axis) a new $(t, q(t))$-layer to pack K_i^+.
5. If t is an even number but not a power of 2 and $p \geq 0$ is arbitrary, then there is an odd number t_1 and an integer s such that K_i^+ is also of type $(t_1, s + p)$. If $t_1 = 3$, then K_i^+ is packed as described in Rule (3) into BC_3-boxes. If $t_1 \geq 5$, then K_i^+ is packed as described in Rule (4) into $BL_{q(t_1)}$-boxes.
6. If there is no empty m-cube in open B-boxes to pack K_i^+ and if there is no empty space in $\mathscr{B}$ to open a new B-box, we close the active bin and open a new bin to pack K_i^+.

3.7 Packing Density

In this subsection we will show that the total volume of m-cubes packed into each closed bin is greater than 1/9.

Assume that m-cubes were packed into $\mathscr{B}$ by the algorithm $aux^+(d)$ and that $\mathscr{B}$ is closed.

Lemma 10 *Denote by v_{bc} the total volume of BC-boxes (contained in a closed bin $\mathscr{B}$) into which at least one m-cube was packed. The total volume of m -cubes packed in BC-boxes is greater than* $\frac{8}{27}\left(v_{bc} - 2^{-3} - 2^{-d}\right)$.

Proof Each BC_2-box is a cube of type (2, 0) (see Fig. 10). Each BC_3-box (of the volume 2^{-3}) contains 3^{d-3} cubes of type (3, 0) of the total volume $3^{d-3} \cdot 3^{-d} = 3^{-3} = \left(\frac{2}{3}\right)^3 \cdot 2^{-3}$ (see Fig. 11).

Denote by b_2 the number of open BC_2-boxes and by b_3 the number of open BC_3-boxes. From among open BC_2-boxes the first $b_2 - 1$ boxes are called *basic*, provided $b_2 \geq 2$. Moreover, from among open BC_3-boxes the first $b_3 - 1$ boxes are called *basic*, provided $b_3 \geq 2$. Clearly, $v_{bc} = b_2 \cdot 2^{-d} + b_3 \cdot 2^{-3}$. The total volume of basic boxes is greater than or equal to $(b_2 - 1) \cdot 2^{-d} + (b_3 - 1) \cdot 2^{-3}$. The total volume of cubes of type (2, 0) and (3, 0) contained in basic boxes is greater than

$$(b_2 - 1) \cdot 2^{-d} + \left(\frac{2}{3}\right)^3 \cdot (b_3 - 1) \cdot 2^{-3} \geq \left(\frac{2}{3}\right)^3 \cdot \left((b_2 - 1) \cdot 2^{-d} + (b_3 - 1) \cdot 2^{-3}\right)$$
$$= \left(\frac{2}{3}\right)^3 \cdot \left(v_{bc} - 2^{-d} - 2^{-3}\right).$$

By Lemma 8, the sum of the volumes of m-cubes packed in basic boxes plus the volumes of the first m-cubes packed in the last BC_2- and BC_3-boxes is greater than the total volume of basic boxes. Consequently, the total volume of all m-cubes packed in BC-boxes is greater than $\left(\frac{2}{3}\right)^3 \cdot \left(v_{bc} - 2^{-d} - 2^{-3}\right)$. □

Lemma 11 *Denote by v_{bl} the total volume of BL -boxes (contained in a closed bin $\mathscr{B}$) into which at least one m-cube was packed. Then the total volume of m-cubes packed in BL-boxes is greater than* $\frac{192}{625} v_{bl} - \frac{64}{625}$.

Proof Each BL-box is a B_k-box for some $k \in \{4, 5 \ldots, q(m-1)\}$.

Since $n_{q(t)-1} < t \leq n_{q(t)}$ and both t and $n_{q(t)-1}$ are odd numbers, it follows that $t \geq n_{q(t)-1} + 2$. By Lemma 9 we get

$$t \geq n_{q(t)-1} + 2 > 2^{q(t)-2} - 1 + 2 = 2^{q(t)-2} + 1.$$

This means, by Lemma 7, that at least 64/125 of any $(t, q(t))$-layer is occupied by cubes of type $(t, 0)$.

For $k \in \{4, 5, \ldots, q(m-1)\}$ denote by b_k the number of open BL_k-boxes. If $b_k \geq 2$, then the first $b_k - 1$ boxes are called *basic*. Observe that at least 3/5 of any basic BL_k-box is occupied by layers. The reason is that the height of the highest layer that can be created in any B_q-box is not greater than $1/5$. Since in any basic BL_k-box there is no space to create a new layer, the sum of the heights of all layers contained in any basic BL_k-box is greater than $\frac{1}{2} - \frac{1}{5} = \frac{3}{10}$ (see Fig. 13). This implies that the total volume of all layers created in a BL_k-box is greater than $\frac{3}{10} \cdot (\frac{1}{2})^{k-1}$. This volume divided by the volume 2^{-k} of a BL_k-box gives the desired ratio.

Finally we estimate the total volume of packed m-cubes.

Let $t \in \{4, 5, \ldots, q(m-1)\}$. Among all $(t, q(t))$-layers created in BL-boxes, the last $(t, q(t))$-layer is called *not-full* and the remaining layers are *full*. Since there is at most one not-full layer of each size, by the choice of n_k we deduce that the sum of volumes of not-full layers is smaller than

$$\left(\frac{1}{5}+\frac{1}{7}+\frac{1}{n_4}\right)\cdot 2^{-3}+\left(\frac{1}{11}+\cdots+\frac{1}{n_5}\right)\cdot 2^{-4}+\cdots$$
$$+\left(\frac{1}{n_{q(m-1)-1}+1}+\cdots+\frac{1}{n_{q(m-1)}}\right)\cdot 2^{-q(m-1)}$$
$$<\frac{1}{2}\cdot 2^{-3}+\frac{1}{2}\cdot 2^{-4}+\cdots+\frac{1}{2}\cdot 2^{-q(m-1)}<\frac{1}{8}.$$

The total volume of basic *BL*-boxes is greater than

$$v_{bl}-(2^{-4}+2^{-5}+\cdots+2^{-q(m-1)})>v_{bl}-1/8.$$

The total volume of created layers is greater than $\frac{3}{5}(v_{bl}-1/8)$. The total volume of full layers is greater than $\frac{3}{5}(v_{bl}-1/8)-\frac{1}{8}$. Since at least 64/125 of any $(t, q(t))$-layer is occupied by cubes of type $(t, 0)$, by Lemma 8 we deduce that the total volume of packed m-cubes is greater than

$$\left(\frac{3}{5}(v_{bl}-1/8)-\frac{1}{8}\right)\cdot\frac{64}{125}=\frac{192}{625}v_{bl}-\frac{64}{625}.$$

□

Lemma 12 *If* $d \geq 5$, *then the total volume of* m *-cubes packed into each closed bin* $\mathscr{B}$ *is greater than* 1/9.

Proof By Lemma 6 we get that

$$v_{bc}+v_{bl}>1-2^{-3}=7/8,$$

where v_{bl} denotes the sum of the volumes of all open *BL*-boxes and v_{bc} denotes the sum of the volumes of all open *BC*-boxes (there is no empty space in $\mathscr{B}$ to open a B_k -box for some $k \geq 3$).

If $v_{bc} > \frac{13}{24}$, then by Lemma 10 we obtain that the total volume of packed m-cubes is greater than

$$\frac{8}{27}\left(\frac{13}{24}-\frac{1}{8}-\frac{1}{32}\right)>\frac{1}{9}.$$

If $v_{bc} \leq \frac{13}{24}$, by Lemmas 10 and 11 we deduce that the total volume of packed items is greater than

$$\frac{8}{27}\left(v_{bc}-\frac{1}{8}-\frac{1}{32}\right)+\frac{192}{625}v_{bl}-\frac{64}{625}>\frac{8}{27}v_{bc}-\frac{5}{108}+\frac{192}{625}\left(\frac{7}{8}-v_{bc}\right)-\frac{64}{625}$$
$$=-\frac{184}{16875}v_{bc}-\frac{5}{108}+\frac{104}{625}\geq-\frac{184}{16875}\cdot\frac{13}{24}-\frac{5}{108}+\frac{104}{625}>\frac{1}{9}.$$

□

3.8 Algorithm *har*(*d*)

Let $10 \le m \le 2^{d-1}$ be an even number. Moreover let $C_1, C_2, \dots$ be a sequence of items (hypercubes). For each C_i from the sequence let K_i^+ be the smallest m-cube containing C_i. For example, if C_i is a hypercube of the edge length greater than 1/2, then K_i^+ is the unit hypercube (a cube of type $(1, 0)$).

An m-cube K_i^+ is *big*, provided its edge length is greater than $1/m$; in that case we say also that the item C_i is *big*. Clearly, any big cube K_i^+ is a cube of type $(t, 0)$ for $t \in \{1, 2, \dots, m-1\}$; we say also that in that case C_i is a *big t-item*.

An m-cube K_j^+ is *small*, provided its edge length is not greater than $1/m$; in that case we say also that the item C_j is *small*.

For example: big 4-cubes have the edge lengths from the set $\{1, 1/2, 1/3\}$ while small 4-cubes have the edge lengths from the set $\{1/4, 1/6, 1/8, 1/12, 1/16, \dots\}$. Moreover, for $m = 4$ each item C_i of the edge length greater than 1/4 is big (the smallest 4-cube containing C_i has the edge length from the set $\{1, 1/2, 1/3\}$) while each item of the edge length not greater than 1/4 is small. Another example: big 6-cubes have the edge lengths from the set $\{1, 1/2, 1/3, 1/4, 1/5\}$ while small 6-cubes have the edge lengths from the set

$$\{1/6, 1/8, 1/10, 1/12, 1/16, 1/20, 1/24, 1/32, 1/40, \dots\}.$$

For $m = 6$ each item C_i of the edge length greater than 1/6 is big.

Lemma 13 *If C_i is a small item, then* $|C_i|/|K_i^+| > \left(m/(m+2)\right)^d$.

Proof Big m-cubes have the edge lengths from the set

$$\left\{1, \frac{1}{2}, \dots, \frac{1}{m/2}, \frac{1}{m/2+1}, \frac{1}{m/2+2}, \dots, \frac{1}{m-1}\right\}.$$

Small m-cubes have the edge lengths of the form

$$\frac{1}{m} = \frac{1}{2 \cdot \frac{m}{2}}, \ \frac{1}{m+2} = \frac{1}{2\left(\frac{m}{2}+1\right)}, \ \frac{1}{m+4} = \frac{1}{2\left(\frac{m}{2}+2\right)}, \ \dots,$$
$$\frac{1}{2m}, \ \frac{1}{2m+4} = \frac{1}{4\left(\frac{m}{2}+1\right)}, \ \frac{1}{2m+6}, \ \dots$$

Since K_i^+ is the smallest small m-cube containing C_i, it follows that

$$\frac{|C_i|}{|K_i^+|} > \frac{(m+2)^{-d}}{m^{-d}} = \left(\frac{m}{m+2}\right)^d.$$

□

Algorithm $har(d)$ **for packing of** C_i **into** $\mathscr{B}$.

- First find the smallest m-cube K_i^+ containing C_i. Then pack K_i^+ together with $C_i \subset K_i^+$ by the algorithm $har^+(d)$.

3.9 Competitive Ratio $O(d/\log D)$

Let $10 \le m \le 2^{d-1}$ be an even number.

Lemma 14 *If* $d \ge 5$, *then the asymptotic competitive ratio of the* $har(d)$ *algorithm is not greater than* $9\Big(\big(\frac{m+2}{m}\big)^d + m - 1\Big)$.

Proof Let S be a sequence of items of the volume v, let λ_j denote the number of big j-items in S and let β be the number of bins used to pack items from S according to the $har(d)$ algorithm. In particular, λ_1 is equal to the number of hypercubes of the edge lengths greater than 1/2 (big 1-items) and λ_2 denotes the number of hypercubes of the edge lengths greater than 1/3 but not greater than 1/2.

Two items of the edge lengths greater than 1/2 cannot be packed into one bin. This implies that $OPT(S) \ge \lambda_1$. Moreover, $2^d + 1$ items of the edge lengths greater than 1/3 cannot be packed into one bin also. For example, if $\lambda_2 = 2^d + 1$, then $OPT(S) > 1$; if $\lambda_2 = 2 \cdot 2^d + 1$, then $OPT(S) > 2$; Thus $OPT(S) \ge \lambda_2/2^d$. Finally, $j^d + 1$ items of the edge lengths greater than $1/(j+1)$ cannot be packed into one bin. Consequently, $OPT(S) \ge \lambda_j \cdot j^{-d}$ for $j \ge 1$.

Let

$$\mu = \max(v, \lambda_1, \lambda_2 \cdot 2^{-d}, \lambda_3 \cdot 3^{-d}, \dots, \lambda_{m-1} \cdot (m-1)^{-d}).$$

Since $OPT(S) \ge v$ as well as $OPT(S) \ge \lambda_j \cdot j^{-d}$ for $j \ge 1$, it follows that

$$OPT(S) \ge \mu.$$

If $\beta - 2\lambda_1 - 1 \le 0$, then

$$\frac{\beta}{OPT(S)} \le \frac{2\lambda_1 + 1}{\mu} \le \frac{2\mu + 1}{\mu} = 2 + \frac{1}{\mu}.$$

Consider the case when $\beta - 2\lambda_1 - 1 > 0$.

Each big t-item was packed into a big cube of type $(t, 0)$. The total volume of big cubes of type $(t, 0)$ containing a big t-item equals $\lambda_t \cdot t^{-d}$ while the total volume of big t-items is greater than $\lambda_t \cdot (t+1)^{-d}$.

The last bin from among β bins can be almost empty. Only one bin (from among β bins) can be almost empty. By the description of the $har(d)$ algorithm (Rule 1) we deduce that in two consecutive bins only one small item and only one 1-big item can be packed. Consequently, in $2\lambda_1$ bins the average occupation can be close to 2^{-d-1}. We will omit the volume of items packed into these $2\lambda_1 + 1$ bins in our computations.

By Lemma 12 we know that the sum of the volumes of packed m-cubes is greater than

$$\frac{1}{9}(\beta - 2\lambda_1 - 1).$$

From among these m-cubes there are big cubes as well as small cubes. The total volume of big m-cubes of edges not greater than 1/2 is equal to

$$v_C = \lambda_2 \cdot 2^{-d} + \lambda_3 \cdot 3^{-d} + \cdots + \lambda_{m-1} \cdot (m-1)^{-d}.$$

If $\beta \leq 9\lambda_1 + 9v_C + 1$, then

$$\begin{aligned}\frac{\beta}{OPT(S)} &\leq \frac{9\lambda_1 + 9v_C + 1}{\mu} \\ &= \frac{9\mu + 9\left(\lambda_2 \cdot 2^{-d} + \lambda_3 \cdot 3^{-d} + \cdots + \lambda_{m-1} \cdot (m-1)^{-d}\right) + 1}{\mu} \\ &\leq \frac{9\mu + 9(\mu + \cdots + \mu) + 1}{\mu} = 9(m-1) + \frac{1}{\mu}.\end{aligned}$$

Consider the case when $\beta > 9\lambda_1 + 9v_C + 1$.

The total volume of big items is greater than

$$v_B = \lambda_1 \cdot 2^{-d} + \lambda_2 \cdot 3^{-d} + \cdots + \lambda_{m-1} \cdot m^{-d}.$$

This value is relatively small compared to $2\lambda_1 + v_C$ and we will omit it in our computations.

The total volume of small m-cubes is greater than

$$\frac{1}{9}(\beta - 2\lambda_1 - 1) - v_C.$$

By Lemma 13 we deduce that the total volume of small items is greater than

$$\left(\frac{m}{m+2}\right)^d \cdot \left(\frac{1}{9}(\beta - 2\lambda_1 - 1) - v_C\right).$$

Consequently, the total volume v of packed items is greater than

$$\begin{aligned}v > & v_B + \left(\frac{m}{m+2}\right)^d \cdot \left(\frac{1}{9}(\beta - 2\lambda_1 - 1) - v_C\right) \\ \geq & \left(\frac{m}{m+2}\right)^d \cdot \left(\frac{1}{9}(\beta - 2\lambda_1 - 1) - \lambda_2 \cdot 2^{-d} - \lambda_3 \cdot 3^{-d} - \cdots - \lambda_{m-1} \cdot (m-1)^{-d}\right) \\ \geq & \left(\frac{m}{m+2}\right)^d \cdot \left(\frac{1}{9} \cdot (\beta - 1) - \lambda_1 - \lambda_2 \cdot 2^{-d} - \lambda_3 \cdot 3^{-d} - \cdots - \lambda_{m-1} \cdot (m-1)^{-d}\right),\end{aligned}$$

i.e.,

$$\beta < 9 \cdot \left(\left(\frac{m+2}{m} \right)^d \nu + \lambda_1 + \lambda_2 \cdot 2^{-d} + \lambda_3 \cdot 3^{-d} + \cdots + \lambda_{m-1} \cdot (m-1)^{-d} \right) + 1.$$

As a consequence,

$$\frac{\beta}{OPT(S)} \leq \frac{\beta}{\mu} \leq \frac{9 \cdot \left(\left(\frac{m+2}{m}\right)^d \mu + \mu + \cdots + \mu \right) + 1}{\mu} = 9 \cdot \left(\left(\frac{m+2}{m}\right)^d + m - 1 \right) + \frac{1}{\mu}.$$

This means that the asymptotic competitive ratio of the *har*(d) algorithm is not greater than $9\left(\left(\frac{m+2}{m}\right)^d + m - 1 \right)$. □

Theorem 2 *If $d \geq 5$, then the asymptotic competitive ratio of the har(d) algorithm is not greater than $9\left(\sqrt{d} + \frac{4d}{\log d} + 1\right)$.*

Proof Similarly as in the proof of Theorem 2.2 in [8] take as m the even number such that

$$4d/\log d < m \leq 2 + 4d/\log d.$$

By Lemma 14 we get that the asymptotic competitive ratio of the *har*(d) algorithm is not greater than

$$\begin{aligned} 9\left(\left(1 + \frac{2}{m}\right)^d + m - 1 \right) &< 9\left(\left(1 + \frac{2\log d}{4d}\right)^d + 2 + \frac{4d}{\log d} - 1 \right) \\ &< 9\left(e^{(\log d)/2} + \frac{4d}{\log d} + 1 \right) \\ &= 9\left((e^{\log d})^{1/2} + \frac{4d}{\log d} + 1 \right) \\ &= 9\left(\sqrt{d} + \frac{4d}{\log d} + 1 \right). \end{aligned}$$

□

Corollary 1 *The asymptotic competitive ratio of the har(d) algorithm is $O(d/\log d)$.*

Appendix 1: The Reverse 2-Cubes Versus 3-Cubes Sequence and Its Increments

Let $n \geq 0$, we define $y(n)$ to be an increasing sequence made from $y_d(2^d - n)$, i.e.,

$$y(0) = 0, \quad y(1) = 1,$$

and for arbitrary $n \geq 2$ let d be the smallest integer greater than or equal to $\log_2 n$, then

$$y(n) = y_d(2^d - n).$$

Let $x(n)$ be the sequence of increments of $y(n)$

$$x(n) = y(n) - y(n-1) \quad \text{for } n \geq 1.$$

We will prove that the sequence $x(n)$ is a Gould's sequence, see [26].

Fact 1 *The sequence $x(n)$ starts with 1 and can be defined inductively:*
When the first 2^d elements of the sequence $x(n)$ are given, the next 2^d elements are defined

$$x(2^d + l) = 2 \cdot x(l), \quad \text{for } 0 < l \leq 2^d.$$

Proof Let $d \geq 0$ and $0 < l \leq 2^d$.

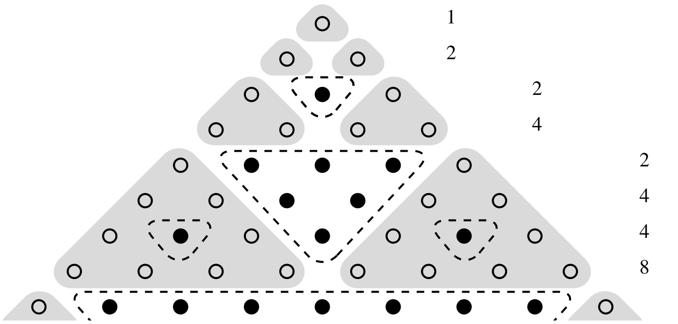


Fig. 14 The parity of elements of Pascal's triangle: a circle indicates an odd number, a filled circle—an even number

$$
\begin{aligned}
x(2^d + l) &= y(2^d + l) - y(2^d + l - 1) \\
&= y_{d+1}(2^{d+1} - (2^d + l)) - y_{d+1}(2^{d+1} - (2^d + l - 1)) \\
&= y_{d+1}(2^d - l) - y_{d+1}(2^d - (l - 1)) \\
&= 3^d + 2 \cdot y_d(2^d - l) - 3^d - 2 \cdot y_d(2^d - (l - 1)) \quad \text{by (1)} \\
&= 2 \cdot (y_d(2^d - l) - y_d(2^d - (l - 1))) = 2 \cdot (y(l) - y(l - 1)) \\
&= 2 \cdot x(l).
\end{aligned}
$$

□

The immediate consequence of the fact is that the sequence $x(n)$ can be created in the following way

1 the first element is 1, multiply by 2 and combine
1 | 2 multiply it by 2 and combine
1 2 | 2 4 multiply by 2 and combine
1 2 2 4 | 2 4 4 8 multiply by 2 and combine
1 2 2 4 2 4 4 8 | 2 4 4 8 4 8 8 16 …

The above pattern can also be recognised in Pascal's triangle as in Fig. 14, see also [26].

References

1. Balogh, J., Békesi, J., Dósa, G., Epstein, L., Levin, A.: A new and improved algorithm for online bin packing. In: 26th Annual European Symposium on Algorithms (ESA 2018). Article No. 5; pp. 5:1–5:14
2. Balogh, J., Békesi, J., Dósa, G., Epstein, L., Levin, A.: A new lower bound for classic online bin packing. arXiv:1807.0554v1
3. Brown, D.J.: A lower bound for on-line one-dimensional bin packing algorithms. Technical Report R-864, Coordinated Sci. Lab, Urbana, Illinois (1979)
4. Christensen, H.I., Khan, A., Pokutta, S., Tetali, P.: Approximation and online algorithms for multidimensional bin packing: a survey. Comput. Sci. Rev. **24**, 63–79 (2017). https://doi.org/10.1016/j.cosrev.2016.12.001
5. Coppersmith, D., Raghavan, P.: Multidimensional on-line bin packing: algorithms and worst-case analysis. Oper. Res. Lett. **8**(1), 17–20 (1989). https://doi.org/10.1016/0167-6377(89)90027-8
6. Csirik, J., Frenk, J., Labbé, M.: Two-dimensional rectangle packing: on-line methods and results. Discrete Appl. Math. **45**(3), 197–204 (1993). https://doi.org/10.1016/0166-218X(93)90009-D
7. Csirik, J., Johnson, D.S.: Bounded space on-line bin packing: best is better than first. Algorithmica **31**(2), 115–138 (2001). https://doi.org/10.1007/s00453-001-0041-7
8. Epstein, L., van Stee, R.: Optimal online algorithms for multidimensional packing problems. SIAM J. Comput. **35**(2), 431–448 (2005). https://doi.org/10.1137/S0097539705446895
9. Grzegorek, P., Januszewski, J.: Online algorithms for 3-space bounded 2-dimensional bin packing and square packing. Rom. J. Inf. Sci. Technol. **17**(2), 190–203 (2014)
10. Grzegorek, P., Januszewski, J.: A note on one-space bounded square packing. Inf. Process. Lett. **115**(11), 872–876 (2015). https://doi.org/10.1016/j.ipl.2015.06.001
11. Grzegorek, P., Januszewski, J.: Drawer algorithms for 1-space bounded multidimensional hyperbox packing. J. Comb. Optim. **37**, 1011–1044 (2019). https://doi.org/10.1007/s10878-018-0338-y
12. Han, X., Chin, F.Y.L., Ting, H.F., Zhang, G., Zhang, Y.: A new upper bound on 2D online bin packing. ACM Trans. Algorithms **7**(4), 50:1–50:18 (2011). https://doi.org/10.1145/2000807.2000818

13. Han, X., Fujita, S., Guo, H.: A two-dimensional harmonic algorithm with performance ratio 2.7834. IPSJ SIG Not. **93**, 43–50 (2001)
14. Heydrich, S., van Stee, R.: Beating the harmonic lower bound for online bin packing. In: 43rd International Colloquium on Automata, Languages, and Programming, LIPIcs—Leibniz International Proceedings in Informatics, vol. 55, pp. 41:1–41:14. Schloss Dagstuhl - LZI, Wadern (2016)
15. Januszewski, J., Lassak, M.: On-line packing sequences of cubes in the unit cube. Geom. Dedic. **67**(3), 285–293 (1997). https://doi.org/10.1023/A:1004953109743
16. Januszewski, J., Zielonka, Ł.: Improved online algorithms for 2-space bounded 2-dimensional bin packing. Int. J. Found. Comput. Sci. **27**(4), 407–429 (2016)
17. Januszewski, J., Zielonka, Ł.: Efficient online packing of 4-dimensional cubes into the unit cube. Stud. Sci. Math. Hung. **55**(3), 305–326 (2018). https://doi.org/10.1556/012.2018.55.3.1400
18. Januszewski, J., Zielonka, Ł.: Online packing of d-dimensional boxes into the unit cube. Period. Math. Hung. (2020). https://doi.org/10.1007/s10998-020-00315-4
19. Johnson, D.S.: Fast algorithms for bin packing. J. Comput. Syst. Sci. **8**(3), 272–314 (1974). https://doi.org/10.1016/S0022-0000(74)80026-7
20. Johnson, D.S., Demers, A., Ullman, J.D., Garey, M.R., Graham, R.L.: Worst-case performance bounds for simple one-dimensional packing algorithms. SIAM J. Comput. **3**(4), 299–325 (1974). https://doi.org/10.1137/0203025
21. Lee, C.C., Lee, D.T.: A simple on-line bin-packing algorithm. J. ACM **32**(3), 562–572 (1985). https://doi.org/10.1145/3828.3833
22. Liang, F.M.: A lower bound for on-line bin packing. Inf. Process. Lett. **10**(2), 76–79 (1980). https://doi.org/10.1016/S0020-0190(80)90077-0
23. Ramanan, P., Brown, D.J., Lee, C., Lee, D.: On-line bin packing in linear time. J. Algorithms **10**(3), 305–326 (1989). https://doi.org/10.1016/0196-6774(89)90031-X
24. Seiden, S.S.: On the online bin packing problem. J. ACM **49**(5), 640–671 (2002). https://doi.org/10.1145/585265.585269
25. Seiden, S.S., van Stee, R.: New bounds for multidimensional packing. Algorithmica **36**(3), 261–293 (2003). https://doi.org/10.1007/s00453-003-1016-7
26. Sloane, N.J.A.: Gould's sequence. https://oeis.org/A001316
27. Ullman, J.D.: The performance of a memory allocation algorithm. Technical Report 100, Princeton University. Department of Electrical Engineering. Computer Sciences Laboratory (1971)
28. van Vliet, A.: An improved lower bound for on-line bin packing algorithms. Inf. Process. Lett. **43**(5), 277–284 (1992). https://doi.org/10.1016/0020-0190(92)90223-I
29. Woeginger, G.: Improved space for bounded-space, on-line bin-packing. SIAM J. Discrete Math. **6**(4), 575–581 (1993). https://doi.org/10.1137/0406045
30. Yao, A.C.C.: New algorithms for bin packing. J. ACM **27**(2), 207–227 (1980). https://doi.org/10.1145/322186.322187
31. Zhang, Y., Chin, F.Y.L., Ting, H.F., Han, X.: Online algorithms for 1-space bounded multi dimensional bin packing and hypercube packing. J. Comb. Optim. **26**(2), 223–236 (2013). https://doi.org/10.1007/s10878-012-9457-z
32. Zhang, Y., Chin, F.Y.L., Ting, H.F., Han, X., Poon, C.K., Tsin, Y.H., Ye, D.: Online algorithms for 1-space bounded 2-dimensional bin packing and square packing. Theor. Comput. Sci. **554**, 135–149 (2014). https://doi.org/10.1016/j.tcs.2014.02.045