S©MR ISSN 1813-3304

СИБИРСКИЕ ЭЛЕКТРОННЫЕ
МАТЕМАТИЧЕСКИЕ ИЗВЕСТИЯ

Siberian Electronic Mathematical Reports
http://semr.math.nsc.ru

Том 20, стр. 853–8644 (2020) УДК 514.174
DOI 10.33048/semi.2020.17.062 MSC 52C17,52C22

PERFECT PACKING OF D-CUBES

A. JOÓS

Abstract. A packing of d-cubes into a d-box of the right area is called perfect packing. Since $\sum_{i=1}^{\infty} 1/i^{dt} = \zeta(dt)$, it can be asked for which t can be found a perfect packing of the d-cubes of edge lengths 1, 2^{-t}, 3^{-t}, ... into a d-box of the right area. In this paper an algorithm will be presented which packs the d-cubes of edge lengths 1, 2^{-t}, 3^{-t}, ... into a d-box of area $\zeta(dt)$ for any t on the interval $[d_0, 2^{d-1}/(d2^{d-1}-1)]$, where d_0 depends on d only.

Keywords: packing, d-cube, tiling.

1. Introduction

Meir and Moser [17] originally noted that since $\sum_{i=2}^{\infty} 1/i^2 = \pi^2/6-1$, it is reasonable to ask whether the squares of side lengths 1/2, 1/3, 1/4, ... can be packed into a rectangle of area $\pi^2/6-1$. Failing that, find the smallest ϵ such that the squares can be packed into a rectangle of area $\pi^2/6-1+\epsilon$. The problem also appears in [8], [6], [5].

A packing into a rectangle of the right (not the right resp.) area is called *perfect* (*imperfect* resp.) packing. In [17], [10], [4], [18], [9] can be found better and better imperfect packings of the squares of side lengths 1/2, 1/3, 1/4, ...
Similarly if the squares of side lengths 1/3, 1/5, 1/7, ... are packed in a rectangle, then in [11], [3], [18], [14], [9] can be found better and better imperfect packings. If the rectangles of dimensions $1\times 1/2$, $1/2\times 1/3$, $1/3\times 1/4$, ... are packed in a rectangle, then in [17], [2], [11], [3], [18], [12], [9] can be found better and better

Joós, A., Perfect packing of d-cubes.

The work is supported by EFOP-3.6.1-16-2016-00003 funds, Establishment of long-term R and D and I processes at the University of Dunaujvaros.
Received July, 8, 2019, published June, 26, 2020.

imperfect packings.
It can be found several papers in this topic e.g. [16], [1], [15].

Chalcraft [7] generalized this question. He packed the squares of side lengths $1, 2^{-t}, 3^{-t}, \ldots$ into a square of the right area. He proved that there is a perfect packing of the squares of side lengths $1, 2^{-t}, 3^{-t}, \ldots$ into a square of the right area for all t in the range $[0.5964, 0.6]$.
Wästlund [19] proved if $1/2 < t < 2/3$, then the squares of side lengths $1, 2^{-t}, 3^{-t}, \ldots$ can be packed into some finite collection of square boxes of the same area $\zeta(2t)$ as the total area of the small squares. In [13] can be found a 3-dimensional generalization of the question of Chalcraft, i.e. a packing of the 3-dimensional cubes of edge lengths $1, 2^{-t}, 3^{-t}, \ldots$ into a 3-box of the right area for all t in the range $[0.36273, 4/11]$.

In this paper this question is generalized to the d-dimensional Euclidean space. A collection of d-cubes can be *packed* into a d-box if the d-cubes can be rotated and translated so that they are subsets of the d-box and that they have pairwise disjoint interiors. In the following arrangements will be considered when the edges of the d-cubes are parallel to the coordinate axes. Similarly as on the plane a packing of d-cubes into a d-box of the right area is called *perfect* packing.
Question: For which t can be found perfect packing of the d-cubes of edge lengths $1, 2^{-t}, 3^{-t}, \ldots$ into a d-box of volume $\zeta(dt)$?

2. The result

Theorem 1. *If*
$$t = \frac{2^{d-1}}{d2^{d-1} - 1},$$
then the d-cubes of edge lengths $1, 2^{-t}, 3^{-t}, \ldots$ *can be packed perfectly into the d-box of dimensions* $1 \times \ldots \times 1 \times \zeta(dt)$.

Theorem 2. *Let* t_0 *be the unique solution of the equation*
$$\zeta(dt) - 1 = \frac{2^{d-1} - 1}{1 - (d-1)t}$$
on the interval $[1/d, 2^{d-1}/(d2^{d-1} - 1)]$. *If*
$$t_0 \le t \le \frac{2^{d-1}}{d2^{d-1} - 1},$$
then the d-cubes of edge lengths $1, 2^{-t}, 3^{-t}, \ldots$ *can be packed perfectly into the d-box of dimensions* $1 \times \ldots \times 1 \times \zeta(dt)$.

3. Notation

If d is fixed ($d \ge 2$), then the constant t is a number between $1/d$ and $1/(d-1)$. As usual, $\zeta(t) = \sum_{i=1}^{\infty} i^{-t}$. Let C_n^t denote the d-cube of edge length n^{-t}. A *d-box* B is the Cartesian product of the intervals $[0, x_1], \ldots, [0, x_d]$, where $x_1 > 0, \ldots, x_d > 0$. If (without loss of generality it may be assumed) $x_1 \le \ldots \le x_d$, then the *d-volume* of B is $v(B) = x_1 \cdot \ldots \cdot x_d$, the *i-th width* of B is $w_i(B) = x_i$ for $i = 1, \ldots, d$ and the *partial surface* of B is $s(B) = w_2(B) \cdot \ldots \cdot w_d(B)$.

Remark 1. *If* B_1 *and* B_2 *are two d-boxes such that the edges of* B_1 *and* B_2 *are parallel to each other and* $B_1 \subset B_2$, *then* $s(B_1) \le s(B_2)$.

Given a set of d-boxes $\mathscr{B} = \{B_1, \ldots, B_n\}$. Let $v(\mathscr{B}) = \sum_{i=1}^{n} v(B_i)$, $s(\mathscr{B}) = \sum_{i=1}^{n} s(B_i)$, $w_1(\mathscr{B}) = \max_{i=1,2\ldots,n} w_1(B_i)$. It is assumed that the empty set is a d-box as well and $v(\emptyset) = s(\emptyset) = w_1(\emptyset) = 0$.

4. The algorithm

The following step will be used several times.

Cutting step **c**

Input: An integer $n \geq 1$ and a d-box B of dimensions $x_1 \times \ldots \times x_d$ such that $w_1(B) \geq n^{-t}$.

Output: The output is two d-boxes, namely $CS_1(n, B)$ and $CS_2(n, B)$. Let $x_{max} = \max\{x_1, \ldots, x_d\}$. If $w_d(B) > n^{-t}$, then $CS_1(n, B)$ is a d-box of dimensions $x_1 \times \ldots \times x_{max-1} \times n^{-t} \times x_{max+1} \times \ldots \times x_d$ and $CS_2(n, B)$ is a d-box of dimensions $x_1 \times \ldots \times x_{max-1} \times (x_{max} - n^{-t}) \times x_{max+1} \times \ldots \times x_d$. If $w_d(B) = n^{-t}$, then $CS_1(n, B) = C_n^t$ and $CS_2(n, B) = \emptyset$.

Action: The step cuts the d-box B into two smaller d-boxes. The cutting hyperplane is perpendicular to one of the longest edge of B.

There is a choice at the Cutting step **c** if the d-box B has more than one longest edge, but the success or failure of the forthcoming algorithms cannot depend on this choice, because the dimensions of the remaining boxes are the same in all cases.

Algorithm **a**

Input: An integer $n \geq 1$ and a d-box B, where $w_1(B) = n^{-t}$.

Output: If the algorithm terminates, then it defines an integer $m_{\mathbf{a}}(n, B)$ (or shortly $m_{\mathbf{a}}$) and a set of d-boxes $\mathscr{B}_{\mathbf{a}}(n, B)$ (or shortly $\mathscr{B}_{\mathbf{a}}$).

Action: If the algorithm terminates, then it packs the d-cubes $C_n^t, \ldots, C_{m_{\mathbf{a}}-1}^t$ into B. Moreover $\mathscr{B}_{\mathbf{a}}$ is the set of d-boxes containing the remaining volume. If it does not terminate, then it packs the d-cubes $C_n^t, C_{(n+1)}^t, \ldots$ into B.

(**a**1) Let $N_{2,1} = n + 1$, $D_d = B$, $\mathscr{B}_2 = \ldots = \mathscr{B}_d = \mathscr{B}_{2,1} = \ldots = \mathscr{B}_{d,1} = \emptyset$.

(**a**2) For $k = d, d-1, \ldots, 2$

(**a**3) Apply the Cutting step **c** with inputs n and D_k. The outputs are $D_{k-1} = CS_1(n, D_k)$ and $F_{k,1} = CS_2(n, D_k)$.

(**a**4) End For.

(**a**5) (Observe the dimensions of D_1 are $n^{-t} \times \ldots \times n^{-t}$, the dimensions of D_j are $n^{-t} \times \ldots \times n^{-t} \times w_2(B) \times \ldots \times w_j(B)$ for $j = 2, \ldots, d-1$, the dimensions of $F_{2,1}$ are $n^{-t} \times \ldots \times n^{-t} \times (w_2(B) - n^{-t})$ (or $F_{2,1} = \emptyset$) and the dimensions of $F_{j,1}$ are $n^{-t} \times \ldots \times n^{-t} \times w_2(B) \times \ldots \times w_{(j-1)}(B) \times (w_j(B) - n^{-t})$ (or $F_{j,1} = \emptyset$) for $j = 3, \ldots, d$.)

(**a**6) Pack the cube C_n^t snugly into D_1.

(**a**7) For $k = 2, 3, \ldots, d$

(**a**8) For $j = 1, 2, \ldots$

(**a**9) (Note: At stage j, the d-cubes $C_n^t, \ldots, C_{(N_{k,j}-1)}^t$ are packed into B. The remaining d-boxes are $\mathscr{B}_2, \ldots, \mathscr{B}_k$, $F_{k,j}$, $F_{(k+1),1}, \ldots, F_{d,1}$. The d-boxes $\mathscr{B}_2, \ldots, \mathscr{B}_k$ will be unused during the remaining part of the algorithm.)

(**a**10) If $w_1(F_{k,j}) < N_{k,j}^{-t}$, then $\mathscr{B}_k = \mathscr{B}_{k,j} \cup \{F_{k,j}\}$, $N_{(k+1),1} = N_{k,j}$ and terminate the For loop.

(**a**11) Apply the Cutting step **c** with inputs $N_{k,j}$ and $F_{k,j}$. The outputs are $G_{k,j} = CS_1(N_{k,j}, F_{k,j})$ and $F_{k,(j+1)} = CS_2(N_{k,j}, F_{k,j})$.

(**a**12) Apply Algorithm **a** recursively to $N_{k,j}$ and $G_{k,j}$. If this terminates, then let $N_{k,(j+1)} = m_{\mathbf{a}}(N_{k,j}, G_{k,j})$ and $\mathscr{G}_{k,j} = \mathscr{B}_{\mathbf{a}}(N_{k,j}, G_{k,j})$.

(**a**13) Let $\mathscr{B}_{k,(j+1)} = \mathscr{B}_{k,j} \cup \mathscr{G}_{k,j}$.

(**a**14) End For.

(**a**15) End For.

(**a**16) Let $m_{\mathbf{a}} = N_{(d+1),1}$ and $\mathscr{B}_{\mathbf{a}} = \mathscr{B}_2 \cup \ldots \cup \mathscr{B}_d$.

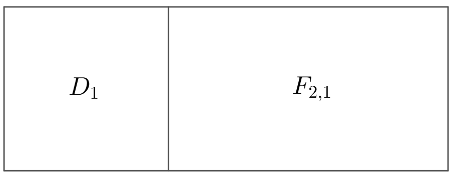

FIG. 1. The 2-boxes D_1 and $F_{2,1}$ at step (**a**4) if $d = 2$, $t = 0.6$, $n = 1$ and $B = 2.65 \times 1$.

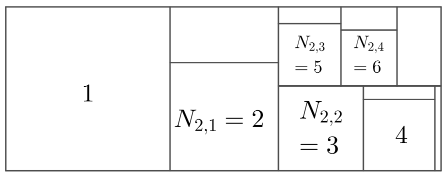

FIG. 2. The terminated Algorithm **a** if $d = 2$, $t = 0.6$, $n = 1$ and $B = 2.65 \times 1$.

Figures 1 and 2 illustrate the running of Algorithm **a** with inputs $n = 1$ and $B = 2.65 \times 1$ assuming $d = 2$ and $t = 0.6$. In Figure 1 can be found the 2-box $B = 2.65 \times 1$ (during this explanation the first (second resp.) dimension means the horizontal (vertical resp.) dimension of the 2-box). At step (**a**4) the dimensions of the 2-box D_1 ($F_{2,1}$ resp.) are 1×1 (1.65×1 resp.). Since $d = 2$ the loop at step (**a**7) runs only one times. The dimensions of the 2-box $G_{2,1}$ at step (**a**11) are $2^{-t} \times 1$ (Figure 2) and $\mathscr{G}_{2,1}$ consists of the rectangle of dimensions $2^{-t} \times (1 - 2^{-t})$, the dimensions of the 2-box $F_{2,2}$ are $(1.65 - 2^{-t}) \times 1$ and $N_{2,2} = 3$. Since $w_1(F_{2,2}) = 1.65 - 2^{-t} < 1$, the dimensions of $G_{2,2}$ ($F_{2,3}$ resp.) at step (**a**11) are $(1.65 - 2^{-t}) \times 3^{-t}$ ($(1.65 - 2^{-t}) \times (1 - 3^{-t})$ resp.) and $N_{2,3} = 5$. Similarly $N_{2,4} = 6$ and Algorithm **a** with inputs $n = 1$ and $B = 2.65 \times 1$ terminated. Observe $m_{\mathbf{a}} = 7$ and $\mathscr{B}_{\mathbf{a}}$ consists of 6 rectangles.

Figures 3 and 4 illustrate the running of Algorithm **a** with inputs $n = 10$ and $B = 1 \times 0.9 \times 10^{-t}$ assuming $d = 3$ and $t = 4/11$. In Figure 3 can be found the 3-box $B = 1 \times 0.9 \times 10^{-t}$ (during this explanation the first (second resp.) dimension means the horizontal (vertical resp.) dimension of the 3-box). At step (**a**4) the dimensions of the 3-box D_1 ($F_{2,1}$, $F_{3,1}$ resp.) are $10^{-t} \times 10^{-t} \times 10^{-t}$ ($10^{-t} \times (0.9 - 10^{-t}) \times 10^{-t}$, $(1 - 10^{-t}) \times 0.9 \times 10^{-t}$ resp.). Since $d = 3$ the loop at step (**a**7) is executed two times. If $k = 3$, then the dimensions of the 3-box $G_{2,1}$ at step (**a**11) are $10^{-t} \times 11^{-t} \times 10^{-t}$ (Figure 4) and $\mathscr{G}_{2,1}$ consists of two 3-boxes of dimensions $10^{-t} \times 11^{-t} \times (10^{-t} - 11^{-t})$ and $(10^{-t} - 11^{-t}) \times 11^{-t} \times 11^{-t}$. Since $w_1(F_{2,2}) < 12^{-t}$, the for loop terminated, $N_{3,1} = 12$ and the dimensions of $G_{3,1}$ ($F_{3,2}$ resp.) at step (**a**11) are $(1 - 10^{-t}) \times 12^{-t} \times 10^{-t}$ ($(1 - 10^{-t}) \times (0.9 - 12^{-t}) \times 10^{-t}$ resp.) and $N_{3,2} = 13$. The dimensions of $F_{3,3}$ are $(1 - 10^{-t} - 13^{-t}) \times (0.9 - 12^{-t}) \times 10^{-t}$.

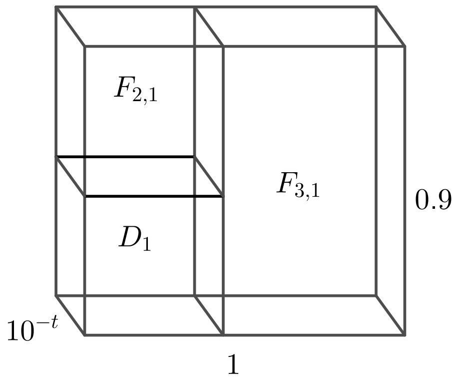

FIG. 3. The 3-boxes D_1, $F_{2,1}$ and $F_{3,1}$ at step (**a**4) if $d = 3$, $t = 4/11$, $n = 10$ and $B = 1 \times 0.9 \times 10^{-t}$.

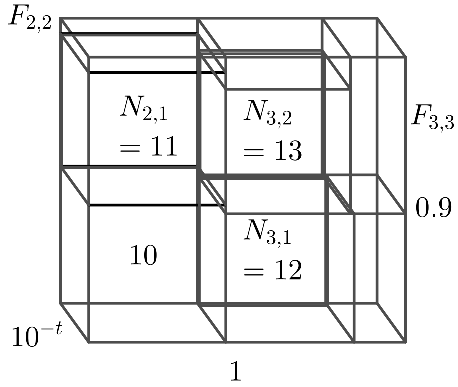

FIG. 4. The terminated Algorithm **a** if $d = 3$, $t = 4/11$, $n = 10$ and $B = 1 \times 0.9 \times 10^{-t}$.

Since $w_1(F_{3,3}) < 14^{-t}$ Algorithm **a** with inputs $n = 10$ and $B = 1 \times 0.9 \times 10^{-t}$ terminated. Observe $m_{\mathbf{a}} = 14$ and $\mathscr{B}_{\mathbf{a}}$ consists of eight 3-boxes.

The Algorithm **a** is used in the following algorithm.

Algorithm **b**

Input: An integer $n \geq 1$ and a set of d-boxes $\mathscr{B}$.

Action: If the Algorithm **b** does not fail, then it packs the d-cubes $C_n^t, C_{(n+1)}^t, \dots$ into $\mathscr{B}$.

(**b**1) Let $n_1 = n$ and $\mathscr{B}_{\mathbf{b}1} = \mathscr{B}$.
(**b**2) For $i = 1, 2, \dots$
(**b**3) (Note: At stage i the d-cubes $C_n^t, \dots, C_{(n_i-1)}^t$ are packed into $\mathscr{B}$. The remaining d-boxes are $\mathscr{B}_{\mathbf{b}i}$.)
(**b**4) If $w_1(\mathscr{B}_{\mathbf{b}i}) < n_i^{-t}$, then fail.
(**b**5) Let $w_{i1} = \min\{w_1(C) | C \in \mathscr{B}_{\mathbf{b}i}, w_1(C) \geq n_i^{-t}\}, w_{i2} = \min\{w_2(C) | C \in \mathscr{B}_{\mathbf{b}i}, w_1(C) = w_{i1}\}, \dots, w_{id} = \min\{w_d(C) | C \in \mathscr{B}_{\mathbf{b}i}, w_1(C) = w_{i1}, \dots, w_{d-1}(C) = w_{i(d-1)}\}$.
(**b**8) Choose any $B_i \in \mathscr{B}_{\mathbf{b}i}$ which satisfies $w_1(B_i) = w_{i1}, \dots, w_d(B_i) = w_{id}$.
(**b**9) If $w_{i1} = \dots = w_{id} = n_i^{-t}$, then
(**b**10) Pack $C_{n_i}^t$ into B_i.
(**b**11) Let $\mathscr{B}_{\mathbf{b}(i+1)} = \mathscr{B}_{\mathbf{b}i} \setminus \{B_i\}$ and $n_{(i+1)} = n_i + 1$.
(**b**12) Else
(**b**13) Apply the Cutting step **c** with inputs n_i and B_i. The outputs are $H_i = CS_1(n_i, B_i)$ and $E_i = CS_2(n_i, B_i)$.
(**b**14) Apply Algorithm **a** with inputs n_i and H_i. If this terminates, then let $n_{(i+1)} = m_{\mathbf{a}}(n_i, H_i)$ and $\mathscr{H}_i = \mathscr{B}_{\mathbf{a}}(n_i, H_i)$.
(**b**15) Let $\mathscr{B}_{\mathbf{b}(i+1)} = \mathscr{B}_{\mathbf{b}i} \setminus \{B_i\} \cup \mathscr{H}_i \cup \{E_i\}$.
(**b**16) End If.

(**b**17) End For.

5. The proof

The following lemmas will be used.

Lemma 1. *If* $\mathscr{B} = \{B_1, \dots, B_n\}$ $(n \geq 1)$*, then* $v(\mathscr{B}) \leq w_1(\mathscr{B})s(\mathscr{B})$.

Proof. It is true

$$v(\mathscr{B}) = \sum_{i=1}^{n} v(B_i) = \sum_{i=1}^{n} w_1(B_i)s(B_i) \leq \sum_{i=1}^{n} w_1(\mathscr{B})s(B_i)$$

$$= w_1(\mathscr{B}) \sum_{i=1}^{n} s(B_i) = w_1(\mathscr{B})s(\mathscr{B}),$$

which completes the proof. □

Lemma 2. *Let* B *be a* d*-box and* $w_1(B) = n^{-t}$*. If Algorithm* **a** *with inputs* n, B *terminates and the outputs are* $m_{\mathbf{a}} = m_{\mathbf{a}}(n, B)$, $\mathscr{B}_{\mathbf{a}} = \mathscr{B}_{\mathbf{a}}(n, B)$*, then*

$$s(\mathscr{B}_{\mathbf{a}}) \leq \left(2^{d-1} - 1\right) \sum_{u=n}^{m_{\mathbf{a}}-1} u^{-(d-1)t}.$$

Proof. It can be assumed that Algorithm **a** and all the recursive calls to Algorithm **a** terminated. Let it be supposed that Algorithm **a** terminated when $j = J_k$ for $k = 2, \dots, d$, so $N_{k,J_k} = N_{(k+1),1}$ for $k = 2, \dots, d$ and $m_{\mathbf{a}} = N_{d,J_d} = N_{(d+1),1}$. Now it will be proved that

$$s(F_{k,J_k}) \leq 2^{k-2} n^{-(d-1)t} \tag{1}$$

for $k = 2, \dots, d$.

If $F_{k,J_k} = \emptyset$, then $s(F_{k,J_k}) = 0 \leq 2^{k-2} n^{-(d-1)t}$ for $k = 2, \dots, d$. Thus it is assumed that $F_{k,J_k} \neq \emptyset$ for $k = 2, \dots, d$.

Since Algorithm **a** terminated without placing the next cube and the longest edge of F_{2,J_2} is smaller than or equal to n^{-t},

$$s(F_{2,J_2}) \leq n^{-(d-1)t}.$$

So (1) is true for $k = 2$. Let k be the first index for which is not true that $s(F_{k,J_k}) \leq 2^{k-2} n^{-(d-1)t}$. Thus $s(F_{k,J_k}) > 2^{k-2} n^{-(d-1)t}$ and $k > 2$. Let $f_1 \times \ldots \times f_d$ be the dimensions of F_{k,J_k} and assume that $f_1 \leq \ldots \leq f_d$. Since Algorithm **a** terminated,

$$f_1 < N_{k,J_k}^{-t} \leq n^{-t}.$$

Before the last Cutting step **c** one of the longest edge of D_k was $f_1 + N_{k,J_k}^{-t}$ and

$$f_d \leq f_1 + N_{k,J_k}^{-t} < 2N_{k,J_k}^{-t} \leq 2n^{-t}.$$

The dimensions of $F_{k,1}$ are $n^{-t} \times \ldots \times n^{-t} \times w_2(B) \times \ldots \times w_{(k-1)}(B) \times (w_k(B) - n^{-t})$ and $F_{k,J_k} \subset F_{k,1}$. After using several times the Cutting step **c** at least one of the edge lengths $w_2(B), \dots, w_{(k-1)}(B), (w_k(B) - n^{-t})$ is less than or equal to n^{-t}. Thus

$$f_2 \leq \ldots \leq f_{(d-k+2)} \leq n^{-t}$$

and

$$f_{(d-k+3)} \leq f_{(d-k+4)} \leq \ldots \leq f_d < 2N_{k,J_k}^{-t} \leq 2n^{-t}.$$

Now

$$s(F_{k,J_k}) = f_2 \dots f_d \le 2^{k-2} n^{-(d-1)t},$$

a contradiction, thus $s(F_{k,J_k}) \le 2^{k-2} n^{-(d-1)t}$ for $k = 2, \dots, d$.

Now the statement of the lemma will be proved by induction on the number of d-cubes packed. If Algorithm **a** terminates with $m_{\mathbf{a}} = n+1$ (only one d-cube is packed), then by (1),

$$s(F_{k,1}) \le 2^{k-2} n^{-(d-1)t}$$

for $k = 2, \dots, d$. Since

$$s(\mathscr{B}_{\mathbf{a}}) = \sum_{k=2}^{d} s(F_{k,1}) < \sum_{k=2}^{d} 2^{k-2} n^{-(d-1)t} = (2^{d-1} - 1) n^{-(d-1)t},$$

the first step is true.

It can be assumed that the lemma is true for all the recursive calls to Algorithm **a**. Now by induction,

$$s(\mathscr{G}_{k,j}) \le \left(2^{d-1} - 1\right) \sum_{u=N_{k,j}}^{N_{k,(j+1)}-1} u^{-(d-1)t} \quad \text{for } k = 2, \dots, d, j = 1, \dots, J_k - 1$$

and

$$\begin{aligned} s(\mathscr{B}_k) &= \sum_{j=1}^{J_k - 1} s(\mathscr{G}_{k,j}) + s(F_{k,J_k}) \\ &\le \sum_{j=1}^{J_k - 1} \left(\left(2^{d-1} - 1\right) \sum_{u=N_{k,j}}^{N_{k,(j+1)}-1} u^{-(d-1)t} \right) + s(F_{k,J_k}) \\ &= \left(2^{d-1} - 1\right) \sum_{u=N_{k,1}}^{N_{k,J_k}-1} u^{-(d-1)t} + s(F_{k,J_k}) \quad \text{for } k = 2, \dots, d. \end{aligned}$$

Thus

$$\begin{aligned} s(\mathscr{B}_{\mathbf{a}}) &= \sum_{k=2}^{d} s(\mathscr{B}_k) \\ &= \sum_{k=2}^{d} \left(\left(2^{d-1} - 1\right) \sum_{u=N_{k,1}}^{N_{k,J_k}-1} u^{-(d-1)t} + s(F_{k,J_k}) \right) \\ &= \left(2^{d-1} - 1\right) \sum_{u=N_{2,1}}^{N_{d,J_d}-1} u^{-(d-1)t} + \sum_{k=2}^{d} s(F_{k,J_k}) \\ &= \left(2^{d-1} - 1\right) \sum_{u=n+1}^{m_{\mathbf{a}}-1} u^{-(d-1)t} + \sum_{k=2}^{d} s(F_{k,J_k}). \end{aligned}$$

By (1),

$$\begin{aligned} s(\mathscr{B}_{\mathbf{a}}) &\le \left(2^{d-1} - 1\right) \sum_{u=n+1}^{m_{\mathbf{a}}-1} u^{-(d-1)t} + \sum_{k=2}^{d} 2^{k-2} n^{-(d-1)t} \\ &= \left(2^{d-1} - 1\right) \sum_{u=n+1}^{m_{\mathbf{a}}-1} u^{-(d-1)t} + \left(2^{d-1} - 1\right) n^{-(d-1)t} \end{aligned}$$

$$= \left(2^{d-1} - 1\right) \sum_{u=n}^{m_{\mathbf{a}}-1} u^{-(d-1)t}$$

which completes the proof. □

Lemma 3. *If a, b are integers ($0 < a < b$) and t is a real number such that $1/d < t < 1/(d-1)$, then*

$$\sum_{j=a}^{b} j^{-(d-1)t} < \frac{b^{1-(d-1)t} - (a-1)^{1-(d-1)t}}{1-(d-1)t}, \tag{2}$$

$$\frac{a^{1-dt} - (b+1)^{1-dt}}{dt-1} < \sum_{j=a}^{b} j^{-dt} < \frac{(a-1)^{1-dt} - b^{1-dt}}{dt-1}. \tag{3}$$

Proof. The proof comes from the inequalities

$$\int_{a}^{b+1} \frac{1}{x^{\alpha}} < \sum_{j=a}^{b} \frac{1}{j^{\alpha}} < \int_{a-1}^{b} \frac{1}{x^{\alpha}}.$$

□

Lemma 4. *Consider Algorithm* **b** *at step (***b**4*) for some value of i. If the following conditions hold*

$$v(\mathscr{B}_{\mathbf{b}i}) \geq \sum_{j=n_i}^{\infty} j^{-dt}, \tag{4}$$

$$s(\mathscr{B}_{\mathbf{b}i}) \leq \frac{n_i^{1-(d-1)t}}{dt-1}, \tag{5}$$

then Algorithm **b** *at step (***b**4*) will not fail for this value of i.*

Proof. It is assumed indirectly, that Algorithm **b** fail, thus $w_1(\mathscr{B}_{\mathbf{b}i}) < n_i^{-t}$. By Lemma 1, (5) and (3),

$$v(\mathscr{B}_{\mathbf{b}i}) \leq w_1(\mathscr{B}_{\mathbf{b}i}) s(\mathscr{B}_{\mathbf{b}i}) < \frac{n_i^{1-dt}}{dt-1} \leq \sum_{j=n_i}^{\infty} j^{-dt} \leq v(\mathscr{B}_{\mathbf{b}i}),$$

a contradiction, which completes the proof of the lemma. □

Lemma 5. *Given an integer $n \geq 1$, a non-empty set of boxes $\mathscr{B}$ and suppose that the following conditions hold*

$$v(\mathscr{B}) \geq \sum_{j=n}^{\infty} j^{-dt}, \tag{6}$$

$$s(\mathscr{B}) \leq \frac{\left(2^{d-1} - 1\right)}{1-(d-1)t} (n-1)^{1-(d-1)t}, \tag{7}$$

$$t \leq \frac{2^{d-1}}{d2^{d-1} - 1}.$$

If the inputs of Algorithm **b** *are* n *and* $\mathscr{B}$*, then the following conditions hold at step* (**b**4) *for all* $i \geq 1$ *for which step* (**b**4) *is executed. The conditions are*

$$v(\mathscr{B}_{\mathbf{b}i}) \geq \sum_{j=n_i}^{\infty} j^{-dt}, \tag{8}$$

$$s(\mathscr{B}_{\mathbf{b}i}) \leq s(\mathscr{B}) + \left(2^{d-1} - 1\right) \sum_{j=n}^{n_i-1} j^{-(d-1)t}. \tag{9}$$

Moreover, Algorithm **b** *will never fail.*

Proof. Observe

$$\frac{1}{d} < \frac{1}{d}\frac{2^{d-1}}{2^{d-1} - 1/d} = \frac{2^{d-1}}{d2^{d-1} - 1} = \frac{1}{d - 1/2^{d-1}} < \frac{1}{d-1}.$$

First, it will be shown that (8) and (9) ensure that Algorithm **b** will not fail. By (9), (2), and (7),

$$s(\mathscr{B}_{\mathbf{b}i}) \leq s(\mathscr{B}) + \left(2^{d-1} - 1\right) \sum_{j=n}^{n_i-1} j^{-(d-1)t}$$

$$< s(\mathscr{B}) + \frac{\left(2^{d-1}-1\right)}{1-(d-1)t}\left((n_i - 1)^{1-(d-1)t} - (n-1)^{1-(d-1)t}\right)$$

$$\leq \frac{\left(2^{d-1}-1\right)}{1-(d-1)t}(n_i - 1)^{1-(d-1)t}.$$

Since $t \leq 2^{d-1}/(d2^{d-1} - 1)$,

$$\frac{2^{d-1}-1}{1-(d-1)t} \leq \frac{1}{dt-1}.$$

Thus

$$s(\mathscr{B}_{\mathbf{b}i}) < \frac{\left(2^{d-1}-1\right)}{1-(d-1)t}(n_i - 1)^{1-(d-1)t} \leq \frac{1}{dt-1}(n_i - 1)^{1-(d-1)t} < \frac{n_i^{1-(d-1)t}}{dt-1}.$$

By Lemma 4, (**b**4) will not fail.

Of course (8) holds for all i.

Now (9) will be proved by induction on i. Of course (9) holds for $i = 1$. Let $(i+1)$ be the smallest value for which (9) is not true. If the condition in (**b**9) was true for i, then $s(\mathscr{B}_{\mathbf{b}(i+1)}) = s(\mathscr{B}_{\mathbf{b}i}) - n_i^{-(d-1)t}$ and $n_{(i+1)} = n_i + 1$. Thus by induction,

$$s(\mathscr{B}_{\mathbf{b}(i+1)}) = s(\mathscr{B}_{\mathbf{b}i}) - n_i^{-(d-1)t}$$

$$\leq s(\mathscr{B}) + \left(2^{d-1} - 1\right) \sum_{u=n}^{n_i-1} u^{-(d-1)t} - n_i^{-(d-1)t}$$

$$< s(\mathscr{B}) + \left(2^{d-1} - 1\right) \sum_{u=n}^{n_i-1} u^{-(d-1)t}$$

$$= s(\mathscr{B}) + \left(2^{d-1} - 1\right) \sum_{u=n}^{n_{i+1}-2} u^{-(d-1)t}$$

$$< s(\mathscr{B}) + \left(2^{d-1} - 1\right) \sum_{u=n}^{n_{i+1}-1} u^{-(d-1)t}.$$

If the condition in (**b**9) was not true for i, then

$$s(\mathscr{B}_{\mathbf{b}(i+1)}) = s(\mathscr{B}_{\mathbf{b}i}) + s(\mathscr{H}_i) - s(B_i) + s(E_i).$$

Observe E_i is a subset of B_i. By Remark 1,

$$s(E_i) \leq s(B_i).$$

Thus

$$s(\mathscr{B}_{\mathbf{b}(i+1)}) \leq (\mathscr{B}_{\mathbf{b}i}) + s(\mathscr{H}_i).$$

By induction and Lemma 2,

$$\begin{aligned} s(\mathscr{B}_{\mathbf{b}(i+1)}) &< s(\mathscr{B}_{\mathbf{b}i}) + s(\mathscr{H}_i) \\ &\leq s(\mathscr{B}) + \left(2^{d-1} - 1\right) \sum_{u=n}^{n_i-1} u^{-(d-1)t} + \left(2^{d-1} - 1\right) \sum_{u=n_i}^{n_{i+1}-1} u^{-(d-1)t} \\ &= s(\mathscr{B}) + \left(2^{d-1} - 1\right) \sum_{u=n}^{n_{i+1}-1} u^{-(d-1)t}, \end{aligned}$$

which completes the proof. □

Proof of Theorem 1. If the d-cube C_1^t is packed into the d-box $B = 1\times\ldots\times1\times\zeta(dt)$ snugly at one end of B, then the remaining d-box is

$$\mathscr{B} = \{1 \times \ldots \times 1 \times (\zeta(dt) - 1)\}.$$

If d is increased, then dt is decreased. If dt is decreased, then $\zeta(dt)$ is increased. Since $dt = 4/3$ if $d = 2$ and $\zeta(4/3) > 2$,

$$\zeta(dt) - 1 > 1$$

and

$$s(\mathscr{B}) = \zeta(dt) - 1.$$

By (3),

$$\begin{aligned} s(\mathscr{B}) = \zeta(dt) - 1 &= \frac{1}{2^{dt}} + \frac{1}{3^{dt}} + \ldots \\ \leq \frac{1}{dt-1} &= \frac{2^{d-1}-1}{1-(d-1)t}(2-1)^{1-(d-1)t}. \end{aligned}$$

By Lemma 5, Algorithm **b** packs perfectly the d-cubes C_n^t $(n \geq 2)$ into $\mathscr{B}$, which completes the proof. □

Proof of Theorem 2. It is easy to prove, that t_0 is well defined. Let I be the interval $\left[t_0, 2^{d-1}/(d2^{d-1}-1)\right]$. If the d-cube C_1^t is packed into the d-box $B = 1\times\ldots\times1\times\zeta(dt)$ snugly at one end of B, then the remaining set of d-boxes is

$$\mathscr{B} = \{1 \times \ldots \times 1 \times (\zeta(dt) - 1)\}.$$

Let $f(t) = s(\mathscr{B})$. Since $\zeta(dt) - 1 > 1$ if $t \in I$,

$$s(\mathscr{B}) = f(t) = \zeta(dt) - 1.$$

Let

$$g(t) = \frac{2^{d-1}-1}{1-(d-1)t}.$$

Since g is an increasing, f is a decreasing function on the interval I and

$$f(t_0) = g(t_0),$$

by Lemma 5, Algorithm **b** packs perfectly the d-cubes C_n^t $(n \geq 2)$ into $\mathscr{B}$, which completes the proof. □

6. Discussion

In dimensions 2 the Chalcraft range for t was $0.5964 \leq t \leq 0.6$ (see [7]). Chalcraft wrote "The more interesting challenge, however, seems to be to increase the bound $t \leq 0.6$."This packing algorithm is different from the algorithm of Chalcraft and the upper bound of t was increased to $2/3$ if $d = 2$.

The inequality (1) is a keystone of the proof. If d is fixed, the inputs of Algorithm **a** are n, B where n is large and the dimensions of B are slightly smaller than $n^{-t} \times (n^{-t} + (n+1)^{-t}) \times (n^{-t} + (n+1)^{-t}) \times \ldots \times (n^{-t} + (n+1)^{-t})$, then $m_{\mathbf{a}} = n+1$ and (1) is close to sharp. The question is open yet whether the best upper bound of t is

$$\frac{2^{d-1}}{d2^{d-1} - 1}$$

or not.

It is easy to find sequences of d-cubes where the d-cubes can not be packed into a d-box of the right area. E.g. if the first few d-cubes are big and the remaining d-cubes are enough small. The idea of Chalcratf was the increasing the edge lengths of the small squares instead of ignoring the big cubes. The question is open yet whether the squares of side lengths $1/n$, $1/(n+1)$, $1/(n+2)$, $\ldots$ for some $n > 0$ can be packed into a rectangle of the right area or not.

References

[1] B. Altunkaynak, *An algebraic approach to rectangle packing Problems*, http://arxiv.org/abs/math/0403194 (2004).

[2] V. Bálint, *A packing problem and the geometrical series*, Proc. of 4th Czechoslov. Symp. on Combinatorics, Graphs and Complexity, Prachatice (1990), Ann. Discrete Math. **51** (1992), 17–21. Zbl 0847.52017

[3] V. Bálint, *Two packing problems*, Discrete Math., **178**:1-3 (1998), 233–236. Zbl 0901.05028

[4] K. Ball, *On packing unequal squares*, J. Combin. Theory Ser. A., **75**:2 (1996), 353–357. Zbl 0857.05019

[5] V. Boju, L. Funar, *The Math Problems Notebook*, Birkhäuser, Basel, 2007. Zbl 1133.00003

[6] P. Brass, W.O.J. Moser, J. Pach, *Research Problems in Discrete Geometry*, Springer, New York, 2005. Zbl 1086.52001

[7] A. Chalcraft, *Perfect square packings*, J. Combin. Theory, Ser. A., **92**:2 (2000), 158–172. Zbl 0977.52023

[8] H.T. Croft, K.J. Falconer, R.K. Guy, *Unsolved Problems in Geometry*, Springer, New York etc., 1991. Zbl 0748.52001

[9] P. Grzegorek, J. Januszewski, *A note on three Moser's problems and two Paulhus' lemmas*, J. Comb. Theory, Ser. A, **162** (2019), 222–230. Zbl 1401.05060

[10] D. Jennings, *On packing unequal rectangles in the unit square*, J. Combin. Theory, Ser. A., **68**:2 (1994), 465–469. Zbl 0808.05033

[11] D. Jennings, *On packings of squares and rectangles*, Discrete Math., **138**:1-3 (1995), 293–300. Zbl 0836.52005

[12] A. Joós, *On packing of rectangles in a rectangle*, Discrete Math., **341**:9 (2018), 2544–2552. Zbl 1392.05093

[13] A. Joós, *Perfect packing of cubes*, Acta Math. Hung., **156**:2 (2018), 375–384. Zbl 1413.05047

[14] A. Joós, V. Bálint, *Packing of odd squares revisited*, J. Geom., **110**:1 (2019), Paper 10. Zbl 1410.52013

[15] Z. Liu, *On continuity properties for infinite rectangle packing*, arXiv:1705.02443 (2017).
[16] G. Martin, *Compactness theorems for geometric packings*, J. Combin. Theory Ser. A., **97**:2 (2002), 225–238. Zbl 1015.52015
[17] A. Meir, L. Moser, *On packing of squares and cubes*, J. Combin. Theory, **5** (1968), 126–134. Zbl 0165.25202
[18] M.M. Paulhus, *An algorithm for packing squares*, J. Combin. Theory Ser. A., **82**:2 (1998), 147–157. Zbl 0913.05034
[19] J. Wästlund, *Perfect packings of squares using the stack-pack strategy*, Discrete Comput. Geom., **29**:4 (2003), 625–631. Zbl 1033.52014

Antal Joós
University of Dunaújváros,
Táncsics Mihály utca 1/A,
2400, Dunaújváros, Hungary
E-mail address: joosa@uniduna.hu