Acta Math. Hungar., **156** (**2**) (2018), 375–384
https://doi.org/10.1007/s10474-018-0858-z
First published online August 7, 2018

PERFECT PACKING OF CUBES

A. JOÓS

University of Dunaújváros, Táncsics Mihály utca 1/A, 2400 Dunaújváros, Hungary
e-mail: joosanti@gmail.com

(Received March 9, 2018; accepted May 22, 2018)

Abstract. It is known that $\sum_{i=1}^{\infty} 1/i^2 = \pi^2/6$. We can ask what is the smallest ε such that all the squares of sides of length 1, 1/2, 1/3, ... can be packed into a rectangle of area $\pi^2/6 + \varepsilon$. A packing into a rectangle of the right area is called perfect packing. Chalcraft [4] packed the squares of sides of length 1, 2^{-t}, 3^{-t}, ... and he found perfect packings for $1/2 < t \leq 3/5$. We generalize this problem and pack the 3-dimensional cubes of sides of length 1, 2^{-t}, 3^{-t}, ... into a right rectangular prism of the right volume. Moreover we show that there is a perfect packing for all t in the range $0.36273 \leq t \leq 4/11$.

1. Introduction

Meir and Moser [10] originally noted that since $\sum_{i=2}^{\infty} 1/i^2 = \pi^2/6 - 1$, it is reasonable to ask whether the set of squares with sides of length 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 $\pi^2/6 - 1 + \varepsilon$. The problem also appears in [4], [3], [6].

A packing into a rectangle of the right (resp. not the right) area is called *perfect* (resp. *imperfect*) packing. In [10], [7], [2], [11] can be found better and better imperfect packings.

Chalcraft [5] generalized this question. He packed the squares of side n^{-t} for $n \in \{1, 2, \dots\}$ into a square of the right area. He proved that for all t in the range $[0.5964, 0.6]$ there is a perfect packing of the squares.

Wästlund [12] proved if $1/2 < t < 2/3$, then the squares of side n^{-t} for $n \in \{1, 2, \dots\}$ can be packed into some finite collection of square boxes of the same area $\zeta(2t)$ as the total area of the tiles.

We can find several papers in this topic e.g. [9], [1], [8].

Very little is known about the generalizations of the above problem to higher-dimensional spaces. We generalize this to the 3-dimensional space in the following way.

Key words and phrases: packing, cube, rectangular prism.
Mathematics Subject Classification: 52C17, 52C22.

Since $\sum_{i=1}^{\infty} i^{-3t} = \zeta(3t)$, it is reasonable to ask whether the set of cubes with sides of length 1, 2^{-t}, 3^{-t}, ... can be packed into a right rectangular prism of volume $\zeta(3t)$. Failing that, find the smallest ε such that the cubes can be packed into a right rectangular prism of volume $\zeta(3t)+\varepsilon$.

2. Notation

We use the constants $1/3 < t < 1/2$. As usual, $\zeta(t) = \sum_{i=1}^{\infty} i^{-t}$.

Let C_n^t denote the cube of side length n^{-t}. A 3-box B is a right rectangular prism of sides x, y and z, where $x > 0$, $y > 0$ and $z > 0$. If (without loss of generality we may assume) $x \le y \le z$, then we define its volume $v(B) = xyz$, its *first width* $w_1(B) = x$, its *second width* $w_2(B) = y$, its *third width* $w_3(B) = z$ and its *partial surface* $s(B) = w_2(B)w_3(B)$.

Given a set of boxes $\mathscr{B} = \{B_1, \dots, B_n\}$, we define

$$v(\mathscr{B}) = \sum_{i=1}^{n} v(B_i), \quad s(\mathscr{B}) = \sum_{i=1}^{n} s(B_i), \quad w_1(\mathscr{B}) = \max_{i=1,2\dots,n} w_1(B_i).$$

Let $v(\emptyset) = s(\emptyset) = w_1(\emptyset) = 0$.

3. The results

THEOREM 1. *For* $t = 4/11$, *the cubes* C_n^t $(n \ge 1)$ *can be packed perfectly into the right rectangular prism of dimensions* $1 \times 1 \times \zeta(3t)$.

THEOREM 2. *For all* t *in the range* $t_0 \le t \le 4/11$, *the cubes* C_n^t $(n \ge 1)$ *can be packed perfectly into the right rectangular prism of dimensions* $1 \times 1 \times \zeta(3t)$ *where* $t_0 = 0.36272\dots$ *is the unique solution of the equation*

$$\zeta(3t) - 1 = \frac{3}{1-2t}$$

in the interval $(1/3, 4/11]$.

4. The algorithm

We generalize the algorithm of Chalcraft [5].

Algorithm **b3d**

Input: An integer $n \ge 1$ and a 3-box B, where $w_1(B) = n^{-t}$.

Output: If the algorithm terminates, then it defines an integer $m_{\mathbf{b3d}} = m_{\mathbf{b3d}}(n, B) > n$ and a set of 3-boxes $\mathscr{B}_{\mathbf{b3d}} = \mathscr{B}_{\mathbf{b3d}}(n, B)$.

Action: If the algorithm terminates, then it packs the cubes $C_n^t, \dots, C_{m_{\mathbf{b3d}}-1}^t$ into B, and $\mathscr{B}_{\mathbf{b3d}}$ is the set of 3-boxes containing the remaining volume. If it does not terminate, then it packs the cubes C^t_n, C^t_{n+1}, … into B.

(**b3d**1) We may assume $B = w_1(B) \times w_2(B) \times w_3(B) = n^{-t} \times w_2(B) \times w_3(B)$. If $w_3(B) > n^{-t}$, then split B into two boxes: one called D with dimensions $n^{-t} \times w_2(B) \times n^{-t}$, and the other called E_1 which is the remainder of B with dimensions $n^{-t} \times w_2(B) \times (w_3(B) - n^{-t})$.

(**b3d**2) Let $n_1 = n + 1$, $y_1 = w_2(B) - n^{-t}$, $\mathscr{D}_1 = \emptyset$.

(**b3d**3) Put the cube C^t_n snugly at one end of D.

(**b3d**4) If $y_1 > 0$, then let D_1 be the remainder of D so that D_1 has dimensions $n^{-t} \times y_1 \times n^{-t}$ (Fig. 1).

(**b3d**5) For $i = 1, 2, \ldots$

(**b3d**6) (Note: At stage i, we have packed C^t_n, …, $C^t_{n_i-1}$ into D. The remaining boxes are $\mathscr{D}_i$, which we never use again in this algorithm, and D_i (as long as $y_i > 0$), which has dimensions $n^{-t} \times y_i \times n^{-t}$.)

(**b3d**7) If $y_i = 0$, then $\mathscr{D} = \mathscr{D}_i$ and terminate the For loop.

(**b3d**8) If $y_i < n_i^{-t}$, then $\mathscr{D} = \mathscr{D}_i \cup \{D_i\}$ and terminate the For loop.

(**b3d**9) Let $y_{i+1} = y_i - n_i^{-t}$.

(**b3d**10) If $y_{i+1} = 0$, then let $D^1_i = D_i$.

(**b3d**11) If $y_{i+1} > 0$, then split D_i into two boxes: one called D^1_i with dimensions $n^{-t} \times n_i^{-t} \times n^{-t}$, and the other called D_{i+1} with dimensions $n^{-t} \times y_{i+1} \times n^{-t}$ (Fig. 2).

(**b3d**12) Apply algorithm **b3d** recursively with inputs n_i and D^1_i. If this terminates, let $n_{i+1} = m_{\mathbf{b3d}}(n_i, D^1_i)$ and $\mathscr{D}^1_i = \mathscr{B}_{\mathbf{b3d}}(n_i, D^1_i)$.

(**b3d**13) Let $\mathscr{D}_{i+1} = \mathscr{D}_i \cup \mathscr{D}^1_i$.

(**b3d**14) End For.

(**b3d**15) Let $N_1 = n_i$, $\mathscr{B}_1 = \mathscr{D}$ and $E_1 = n^{-t} \times Y_1 \times Z_1$.

(**b3d**16) For $j = 1, 2, \ldots$

(**b3d**17) (Note: At stage j, we have packed C^t_n, …, $C^t_{N_j-1}$ into B. The remaining boxes are $\mathscr{B}_j$, which we never use again in this algorithm, and E_j (as long as $Y_j > 0$ and $Z_j > 0$), which has dimensions $n^{-t} \times Y_j \times Z_j$.)

(**b3d**18) If $Y_j = 0$ or $Z_j = 0$, then terminate with $m_{\mathbf{b3d}} = N_j$ and $\mathscr{B}_{\mathbf{b3d}} = \mathscr{B}_j$.

(**b3d**19) If $Y_j < N_j^{-t}$ or $Z_j < N_j^{-t}$, then terminate with $m_{\mathbf{b3d}} = N_j$ and $\mathscr{B}_{\mathbf{b3d}} = \mathscr{B}_j \cup \{E_j\}$.

(**b3d**20) If $Y_j \geq Z_j$, then

(**b3d**21) Let $Y_{j+1} = Y_j - N_j^{-t}$ and $Z_{j+1} = Z_j$.

(**b3d**22) Split E_j into two boxes: one called E^1_j with dimensions $n^{-t} \times N_j^{-t} \times Z_j$, and the other called E_{j+1} with dimensions $n^{-t} \times Y_{j+1} \times Z_{j+1}$ (Fig. 3).

(**b3d**23) Else

(**b3d**24) Let $Z_{j+1} = Z_j - N_j^{-t}$ and $Y_{j+1} = Y_j$.

(**b3d**25) Split E_j into two boxes: one called E_j^1 with dimensions $n^{-t} \times Y_j \times N_j^{-t}$, and the other called E_{j+1} with dimensions $n^{-t} \times Y_{j+1} \times Z_{j+1}$ (Fig. 4).

(**b3d**26) End If.

(**b3d**27) Apply algorithm **b3d** recursively with inputs N_j and E_j^1. If this terminates, let $N_{j+1} = m_{\mathbf{b3d}}(N_j, E_j^1)$ and $\mathscr{E}_j^1 = \mathscr{B}_{\mathbf{b3d}}(N_j, E_j^1)$.

(**b3d**28) Let $\mathscr{B}_{j+1} = \mathscr{B}_j \cup \mathscr{E}_j^1$.

(**b3d**29) End For.

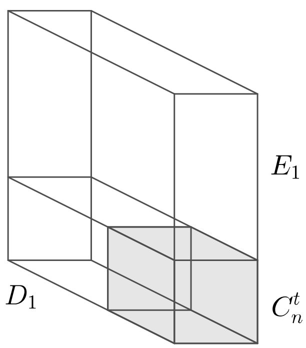

Fig. 1: The 3-box D_1 at step (**b3d**4)

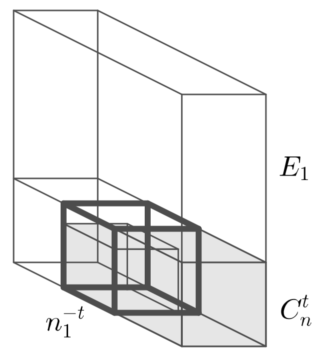

Fig. 2: The 3-box D_1^1 at step (**b3d**11)

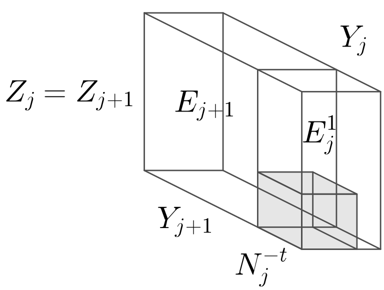

Fig. 3: The 3-boxes at step (**b3d**22)

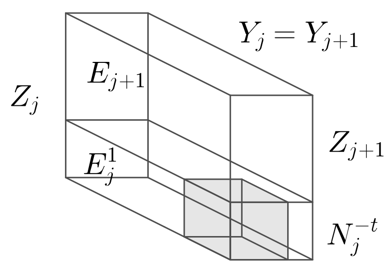

Fig. 4: The 3-boxes at step (**b3d**25)

Figs. 6 and 5 show the result of running algorithm **b3d** for $t = 0.4$, $n = 1$ and B a 3-box of dimensions $1 \times 2.15 \times 2.2$. These parameters illustrate the algorithm **b3d**′s behavior. Observe $n_1 = 2$, $n_2 = 3$ and $y_2 = 2.15 - 1 - 2^{-t} < n_2^{-t}$ thus the first for loop terminated. We have $N_1 = 3$, $N_2 = 4$, $N_3 = 6$, $N_4 = 9$, $N_5 = 11$, $N_6 = 13$ and $Z_6 = 2.2 - 1 - 6^{-t} - 11^{-t} < N_6^{-t}$ thus the algorithm **b3d** terminated.

The subroutine **b3d** is used in the algorithm **c3d**.

Algorithm **c3d**

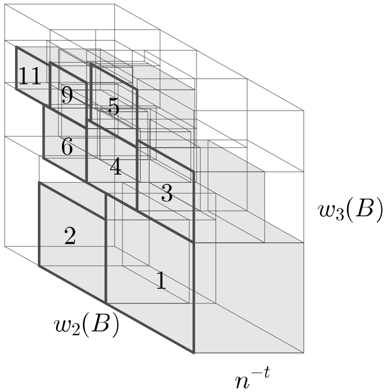

Fig. 5: Algorithm **b3d** *with* $t = 0.4$, $n = 1$ *and a* $1 \times 2.15 \times 2.2$ *box*

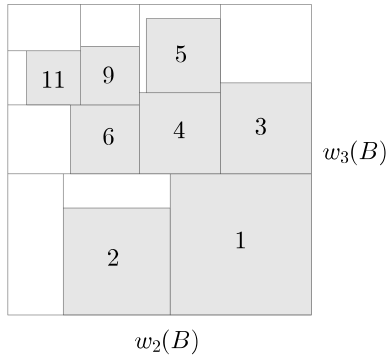

Fig. 6: Algorithm **b3d** *with* $t = 0.4$, $n = 1$ *and a* $1 \times 2.15 \times 2.2$ *box*

Input: An integer $n \geq 1$ and a set of 3-boxes $\mathscr{B}$.

Action: If the algorithm does not fail, then it packs the cubes C_n^t, C_{n+1}^t, ... into $\mathscr{B}$.

(**c3d**1) Let $n_1 = n + 1$ and $\mathscr{B}_1 = \mathscr{B}$.
(**c3d**2) For $i = 1, 2, \ldots$
(**c3d**3) (Note: At stage i, we have packed C_n^t, ..., $C_{n_i-1}^t$ into B. The remaining 3-boxes are $\mathscr{B}_i$.)
(**c3d**4) If $w_1(\mathscr{B}_i) < n_i^{-t}$, then fail.
(**c3d**5) Let $w_{i1} = \min\{w_1(C) \mid C \in \mathscr{B}_i,\ w_1(C) \geq n_i^{-t}\}$.
(**c3d**6) Let $w_{i2} = \min\{w_2(C) \mid C \in \mathscr{B}_i,\ w_1(C) = w_{i1}\}$.
(**c3d**7) Let $w_{i3} = \min\{w_3(C) \mid C \in \mathscr{B}_i,\ w_1(C) = w_{i1},\ w_2(C) = w_{i2}\}$.
(**c3d**8) Choose any $B_i \in \mathscr{B}_i$ which satisfies $w_1(B_i) = w_{i1}$, $w_2(B_i) = w_{i2}$ and $w_3(B_i) = w_{i3}$.
(**c3d**9) If $w_{i1} = w_{i2} = w_{i3} = n_i^{-t}$, then
(**c3d**10) Pack $C_{n_i}^t$ into B_i.
(**c3d**11) Let $\mathscr{B}_{i+1} = \mathscr{B}_i \setminus \{B_i\}$.
(**c3d**12) Let $n_{i+1} = n_i + 1$.
(**c3d**13) Else
(**c3d**14) We may assume that $B = w_{i1} \times w_{i2} \times w_{i3}$. Cut B_i into two 3-boxes: one called C_i of dimensions $w_{i1} \times w_{i2} \times n_i^{-t}$ and the other called D_i of dimensions $w_{i1} \times w_{i2} \times (w_{i3} - n_i^{-t})$.
(**c3d**15) Call algorithm **b3d** with inputs n_i and C_i. If this terminates, then let $n_{i+1} = m_{\mathbf{b3d}}(n_i, C_i)$ and $\mathscr{C}_i = \mathscr{B}_{\mathbf{b3d}}(n_i, C_i)$.
(**c3d**16) Let $\mathscr{B}_{i+1} = \mathscr{B}_i \setminus \{B_i\} \cup \mathscr{C}_i \cup \{D_i\}$.
(**c3d**17) End If.
(**c3d**18) End For.

5. The proof

We use the following lemmas.

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. We have

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

LEMMA 2. *Suppose* $w_1(B) = n^{-t}$ *and algorithm* **b3d** *with inputs* n *and* B *terminates with* $m_{\mathbf{b3d}} = m_{\mathbf{b3d}}(n, B)$ *and* $\mathscr{B}_{\mathbf{b3d}} = \mathscr{B}_{\mathbf{b3d}}(n, B)$. *Then*

$$s(\mathscr{B}_{\mathbf{b3d}}) < 3 \sum_{j=n}^{m_{\mathbf{b3d}}-1} j^{-2t}.$$

PROOF. The proof is by induction on the number of cubes packed. Of course, if **b3d** terminates with $m_{\mathbf{b3d}} = n + 1$, then

$$s(\mathscr{B}_{\mathbf{b3d}}) < 3n^{-2t} = 3 \sum_{j=n}^{m_{\mathbf{b3d}}-1} j^{-2t}.$$

Thus the first step is true.

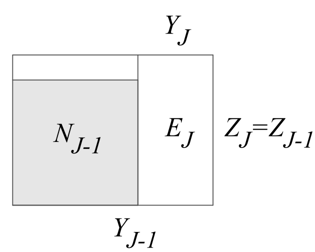

Fig. 7: The boxes E_{J-1} *and* E_J

We can assume that the lemma is true for all the recursive calls to algorithm **b3d**. We can also assume that **b3d** and all the recursive calls to **b3d** terminated. Suppose algorithm **b3d** terminates when $i = I$ and $j = J$, so $m_{\mathbf{b3d}} = N_J$. Since algorithm **b3d** terminated without placing the next cube, the following statements are true

- $y_I < n_I^{-t} < n^{-t}$ and
- $(Y_J < N_J^{-t} < n^{-t}$ or $Z_J < N_J^{-t} < n^{-t})$.

If $Y_{J-1} \geq Z_{J-1}$ (the opposite case is similar) (see Fig. 7), then $n^{-t} > N_J^{-t} > Y_J = Y_{J-1} - N_{J-1}^{-t}$ and $Z_J = Z_{J-1}$. If $Z_J \leq 2n^{-t}$, then

$$s(E_J) = \max(n^{-t}Y_J, n^{-t}Z_J, Y_J Z_J) \leq 2n^{-2t}.$$

If we assume $Z_J > 2n^{-t}$, then

$$2n^{-t} < Z_J = Z_{J-1} \leq Y_{J-1} = Y_J + N_{J-1}^{-t} < N_J^{-t} + N_{J-1}^{-t} < 2n^{-t},$$

a contradiction. Thus $s(E_J) < 2n^{-2t}$. Observe, $s(D_I) = n^{-2t}$. Now by induction,

$$s(\mathscr{D}_i^1) \leq 3 \sum_{k=n_i}^{n_{i+1}-1} k^{-2t} \quad \text{for } i < I, \quad s(\mathscr{E}_j^1) \leq 3 \sum_{k=N_j}^{N_{j+1}-1} k^{-2t} \quad \text{for } j < J,$$

$$s(\mathscr{B}_{\mathbf{b3d}}) = \sum_{m=1}^{I-1} s(\mathscr{D}_m^1) + \sum_{m=1}^{J-1} s(\mathscr{E}_m^1) + s(D_I) + s(E_J)$$

$$\leq 3 \sum_{k=n_1}^{N_J-1} k^{-2t} + 3n^{-2t} = 3 \sum_{k=n}^{m_{\mathbf{b3d}}-1} k^{-2t}. \quad \square$$

Lemma 3. *We have*

$$(1) \qquad (b+1)^{1-2t} - a^{1-2t} < (1-2t) \sum_{k=a}^{b} k^{-2t} < b^{1-2t} - (a-1)^{1-2t},$$

$$(2) \qquad a^{1-3t} - (b+1)^{1-3t} < (3t-1) \sum_{k=a}^{b} k^{-3t} < (a-1)^{1-3t} - b^{1-3t}.$$

Proof. Routine. $\square$

Lemma 4. *Consider step* (**c3d**4) *for some value of i. Suppose the following conditions hold.*

$$(3) \qquad v(\mathscr{B}_i) \geq \sum_{k=n_i}^{\infty} k^{-3t},$$

$$(4) \qquad s(\mathscr{B}_i) \leq \frac{n_i^{1-2t}}{3t-1}.$$

Then step (**c3d**4) *will not fail for this value of i.*

PROOF. We assume that the algorithm fail. Therefore we have $w_1(\mathscr{B}_i) < n_i^{-t}$. By Lemma 1, (4) and (2),

$$v(\mathscr{B}_i) \le w_1(\mathscr{B}_i)s(\mathscr{B}_i) < \frac{n_i^{1-3t}}{3t-1} \le \sum_{k=n_i}^{\infty} k^{-3t} \le v(\mathscr{B}_i),$$

a contradiction, which completes the proof of the lemma. $\square$

LEMMA 5. *Given an integer $n \ge 1$ and a non-empty set of boxes $\mathscr{B}$, suppose the following conditions hold for $t \le \frac{4}{11}$:*

$$v(\mathscr{B}) \ge \sum_{k=n}^{\infty} k^{-3t}, \tag{5}$$

$$s(\mathscr{B}) \le \frac{3}{1-2t}(n-1)^{1-2t}. \tag{6}$$

If we run algorithm **c3d** *with the inputs n and $\mathscr{B}$, then the following conditions hold at step* (**c3d**4) *for all $i \ge 1$ for which step* (**c3d**4) *is executed. The conditions are*

$$v(\mathscr{B}_i) \ge \sum_{k=n_i}^{\infty} k^{-3t}, \tag{7}$$

$$s(\mathscr{B}_i) \le s(\mathscr{B}) + 3\sum_{k=n}^{n_i-1} k^{-2t}. \tag{8}$$

Moreover, the algorithm will never fail.

PROOF. First, we will show that (7) and (8) ensure that the algorithm will not fail. By (8), (1), and (6),

$$s(\mathscr{B}_i) \le s(\mathscr{B}) + 3\sum_{k=n}^{n_i-1} k^{-2t}$$

$$< s(\mathscr{B}) + \frac{3}{1-2t}\left((n_i-1)^{1-2t} - (n-1)^{1-2t}\right) \le \frac{3}{1-2t}(n_i-1)^{1-2t}.$$

Since $t \le 4/11$,

$$\frac{3}{1-2t} \le \frac{1}{3t-1}\,.$$

Thus

$$s(\mathscr{B}_i) < \frac{3}{1-2t}(n_i-1)^{1-2t} \le \frac{1}{3t-1}(n_i-1)^{1-2t} < \frac{n_i^{1-2t}}{3t-1}\,.$$

By Lemma 4, (**c3d**4) will not fail.

Now we prove (7) and (8) by induction on i. Of course they hold for $i = 1$ and (7) holds for all i. Let $i > 1$ be the smallest i for which (8) is not true.

If the condition in (**c3d**9) was true for $i-1$, then $s(\mathscr{B}_i) = s(\mathscr{B}_{i-1}) - 3n_{i-1}^{-2t}$ and $n_i = n_{i-1} + 1$. Thus by induction,

$$s(\mathscr{B}_i) = s(\mathscr{B}_{i-1}) - 3n_{i-1}^{-2t} \le s(\mathscr{B}) + 3\sum_{k=n}^{n_{i-1}-1} k^{-2t} - 3n_{i-1}^{-2t}$$

$$< s(\mathscr{B}) + 3\sum_{k=n}^{n_{i-1}-1} k^{-2t} = s(\mathscr{B}) + 3\sum_{k=n}^{n_i-2} k^{-2t} < s(\mathscr{B}) + 3\sum_{k=n}^{n_i-1} k^{-2t}.$$

If the condition in (**c3d**9) was not true for $i-1$, then

$$s(\mathscr{B}_i) = s(\mathscr{B}_{i-1}) + s(\mathscr{C}_{i-1}) - s(B_{i-1}) + s(D_{i-1}) < s(\mathscr{B}_{i-1}) + s(\mathscr{C}_{i-1}).$$

By induction and Lemma 2,

$$s(\mathscr{B}_i) < s(\mathscr{B}_{i-1}) + s(\mathscr{C}_{i-1})$$

$$\le s(\mathscr{B}) + 3\sum_{k=n}^{n_{i-1}-1} k^{-2t} + 3\sum_{k=n_{i-1}}^{n_i-1} k^{-2t} = s(\mathscr{B}) + 3\sum_{k=n}^{n_i-1} k^{-2t}. \quad \square$$

Proof of Theorem 1. If the cube C_1^t is packed in the 3-box $B = 1 \times 1 \times \zeta(3t)$ snugly at one end of B, then the remaining 3-box is

$$\mathscr{B} = \left\{1 \times 1 \times (\zeta(3t) - 1)\right\}$$

and

$$s(\mathscr{B}) = \zeta(3t) - 1 = 10.58 < 11 = \frac{3}{1-2t}(2-1)^{1-2t}.$$

By Lemma 5, the algorithm **c3d** packs perfectly the cubes C_n^t $(n \ge 2)$ into $\mathscr{B}$. $\square$

Proof of Theorem 2. If the cube C_1^t is packed in the 3-box $B = 1 \times 1 \times \zeta(3t)$ snugly at one end of B, then the remaining 3-box is

$$\mathscr{B} = \left\{1 \times 1 \times (\zeta(3t) - 1)\right\}.$$

Let $f(t) = s(\mathscr{B})$. We have $\zeta(3t) - 1 > 1$ if $t \in [t_0, 4/11]$. Thus $s(\mathscr{B}) = f(t) = \zeta(3t) - 1$. Since $g(t) = \frac{3}{1-2t}$ is an increasing, $f(t)$ is a decreasing function on the interval $[t_0, 4/11]$ and $f(t_0) = g(t_0)$, by Lemma 5, the algorithm **c3d** pack perfectly the cubes C_n^t $(n \ge 2)$ into $\mathscr{B}$. $\square$

6. Discussion

If we increase the number of the packed cubes before we start the algorithm **c3d**, then we can probably decrease the constant t_0. The more interesting challenge, however, seems to be to increase the bound 4/11.

References

[1] B. Altunkaynak, An algebraic approach to rectangle packing problems, arxiv.org/abs/math/0403194 (2004).
[2] K. Ball, On packing unequal squares, *J. Combin. Theory Ser. A.*, **75** (1996), 353–357.
[3] V. Boju and L. Funar, *The Math Problems Notebook*, Birkhäuser (Basel, 2007).
[4] P. Brass, W. O. J. Moser and J. Pach, *Research Problems in Discrete Geometry*, Springer-Verlag (2005).
[5] A. Chalcraft, Perfect square packings, *J. Combin. Theory Ser. A.*, **92** (2000), 158–172.
[6] H. T. Croft, K. J. Falconer, and R. K. Guy, *Unsolved Problems in Geometry*, Springer-Verlag (1991).
[7] D. Jennings, On packing unequal rectangles in the unit square, *J. Combin. Theory Ser. A.*, **68** (1994), 465–469.
[8] Z. Liu, On continuity properties for infinite rectangle packing, arxiv:1705.02443 (2017).
[9] G. Martin, Compactness theorems for geometric packings, *J. Combin. Theory Ser. A.*, **97** (2002), 225–238.
[10] A. Meir and L. Moser, On packing of squares and cubes, *J. Combin. Theory*, **5** (1968), 126–134.
[11] M. M. Paulhus, An algorithm for packing squares, *J. Combin. Theory Ser. A.*, **82** (1997), 147–157.
[12] J. Wästlund, Perfect packings of squares using the stack-pack strategy, *Discrete Comput. Geom.*, **29** (2003), 625–631.