PERFECT PACKING OF SQUARES

ANTAL JOÓS

Communicated by Ioan Tomescu

It is known that $\sum_{i=1}^{\infty} 1/i^2 = \pi^2/6$. Meir and Moser asked 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 $\pi^2/6 + \epsilon$. A packing into a rectangle of the right area is called perfect packing. Chalcraft packed the squares of sides of length 1, 2^{-t}, 3^{-t}, ... and he found perfect packing for $1/2 < t \leq 3/5$. We will show, based on an algorithm by Chalcraft, that there are perfect packings if $1/2 < t \leq 2/3$. Moreover, we show that there is a perfect packing for all t in the range $\log_3 2 \leq t \leq 2/3$.

AMS 2020 Subject Classification: 52C15, 52C20.

Key words: packing, square, rectangle.

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 in a rectangle of area $\pi^2/6 - 1 + \epsilon$. The problem also appears in [6], [4], [3].

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

Chalcraft [5] generalized this question. He packed the squares of side n^{-t} for $n = 1, 2, \ldots$ 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. In [5], it can be read that "Other packings will work for other ranges of t. We can probably make the t_0 in Theorem 8 as close to 1/2 as desired in this way. The more interesting challenge, however, seems to be to increase the bound $t \leq 3/5$." Our aim is to increase this bound.

Wästlund [12] proved if $1/2 < t < 2/3$, then the squares of side n^{-t} for $n = 1, 2, \ldots$ can be packed into some finite collection of square boxes of the

MATH. REPORTS **25(75)** (2023), *2*, 221–229
doi: 10.59277/mrar.2023.25.75.2.221

same area $\zeta(2t)$ as the total area of the tiles. This is an increase of the bound $t \leq 3/5$, but we have many enclosing rectangles.

There are several papers in this topic, e.g. [9], [1], [8].

2. PERFECT PACKING

THEOREM 1. *For $t = 2/3$, the squares S_n^t $(n \geq 1)$ can be packed perfectly into the rectangle of dimensions $\zeta(2t) \times 1$.*

THEOREM 2. *For all t in the range $\log_3 2 \leq t \leq 2/3$, the squares S_n^t $(n \geq 1)$ can be packed perfectly into the rectangle of dimensions $\zeta(2t) \times 1$.*

3. NOTATION

We use the Chalcraft's algorithm in [5] and we modify the proof of Chalcraft. For the sake of simplicity, we use the Chalcraft's notation. For the completeness, we recall these.

Throughout the paper, the width of a rectangle will always refer to the shorter side and the height will always refer to the longer side. We use the constant $1/2 < t \leq 2/3$. As usual, $\zeta(t) = \sum_{i=1}^{\infty} i^{-t}$. Let S_n^t denote the square of side length n^{-t}. A box is a rectangle of sides $x, y > 0$. Let $x \times y$ denote the box B of sides x and y. We define its area $a(B) = xy$, its semi-perimeter $p(B) = x + y$, its width $w(B) = \min(x, y)$ and its height $h(B) = \max(x, y)$.

Given a set of boxes $\mathscr{B} = \{B_1, \ldots, B_n\}$, we define $a(\mathscr{B}) = \sum_{i=1}^{n} a(B_i)$, $h(\mathscr{B}) = \sum_{i=1}^{n} h(B_i)$ and $w(\mathscr{B}) = \max_{i=1,\ldots,n} w(B_i)$. Let $a(\emptyset) = h(\emptyset) = w(\emptyset) = 0$.

4. CHALCRAFT'S ALGORITHM

For the completeness, we repeat the description of Chalcraft's algorithm.

First, we recall the subroutine of Chalcraft, which we call Algorithm **b** as in [5].

Algorithm **b**

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

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

Action: If the algorithm terminates, then it packs the squares $S_n^t, \ldots, S_{m_{\mathbf{b}}-1}^t$ into B, and $\mathscr{B}_{\mathbf{b}}$ is the set of boxes containing the remaining area. If it does not terminate, then it packs the squares $S_n^t, S_{n+1}^t, \ldots$ into B.

(**b**1) Let $n_1 = n + 1$, $x_1 = h(B) - n^{-t}$ and $\mathscr{B}_1 = \emptyset$.
(**b**2) Put the square S_n^t snugly at one end of B.
(**b**3) If $x_1 > 0$, then let B_1 be the remainder of B, so that B_1 has dimensions $x_1 \times n^{-t}$.
(**b**4) For $i = 1, 2, \ldots$
(**b**5) (Note: At stage i, we have packed $S_n^t, \ldots, S_{n_i-1}^t$ into B. The remaining boxes are $\mathscr{B}_i$, which we never use again in this algorithm, and B_i (as long as $x_i > 0$), which has dimensions $x_i \times n^{-t}$.)
(**b**6) If $x_i = 0$, then terminate with $m_{\mathbf{b}} = n_i$ and $\mathscr{B}_{\mathbf{b}} = \mathscr{B}_i$.
(**b**7) If $x_i < n_i^{-t}$, then terminate with $m_{\mathbf{b}} = n_i$ and $\mathscr{B}_{\mathbf{b}} = \mathscr{B}_i \cup \{B_i\}$.
(**b**8) Let $x_{i+1} = x_i - n_i^{-t}$.
(**b**9) If $x_{i+1} = 0$, then let $C_i = B_i$.
(**b**10) If $x_{i+1} > 0$, then split B_i into two boxes: one called C_i with dimensions $n_i^{-t} \times n^{-t}$, and the other called B_{i+1} with dimensions $x_{i+1} \times n^{-t}$.
(**b**11) Apply Algorithm **b** recursively with inputs n_i and C_i. If this terminates, let $n_{i+1} = m_{\mathbf{b}}(n_i, C_i)$ and $\mathscr{C}_i = \mathscr{B}_{\mathbf{b}}(n_i, C_i)$.
(**b**12) Let $\mathscr{B}_{i+1} = \mathscr{B}_i \cup \mathscr{C}_i$.
(**b**13) End For.

The subroutine **b** is used in the Chalcraft's algorithm **c**.

Algorithm **c**

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

Action: If the algorithm does not fail, then it packs the squares $S_n^t, S_{n+1}^t, \ldots$ into $\mathscr{B}$.

(**c**1) Let $n_1 = n + 1$ and $\mathscr{B}_1 = \mathscr{B}$.
(**c**2) For $i = 1, 2, \ldots$
(**c**3) (Note: At stage i, we have packed $S_n^t, \ldots, S_{n_i-1}^t$ into B. The remaining boxes are $\mathscr{B}_i$.)
(**c**4) If $w(\mathscr{B}_i) < n_i^{-t}$, then fail.
(**c**5) Let $w_i = \min\{w(C) | C \in \mathscr{B}_i, w(C) \geq n_i^{-t}\}$.
(**c**6) Let $h_i = \min\{h(C) | C \in \mathscr{B}_i, w(C) = w_i\}$.
(**c**7) Choose any $B_i \in \mathscr{B}_i$ which satisfies $w(B_i) = w_i$ and $h(B_i) = h_i$.
(**c**8) If $w_i = h_i = n_i^{-t}$, then
(**c**9) Put $S_{n_i}^t$ snugly into B_i.
(**c**10) Let $\mathscr{B}_{i+1} = \mathscr{B}_i \setminus \{B_i\}$.

(**c**11) Let $n_{i+1} = n_i + 1$.
(**c**12) Else
(**c**13) Cut B_i into two boxes: one called C_i of dimensions $w_i \times n_i^{-t}$ and the other called D_i of dimensions $w_i \times (h_i - n_i^{-t})$.
(**c**14) Call Algorithm **b** with inputs n_i and C_i. If this terminates, then let $n_{i+1} = m_{\mathbf{b}}(n_i, C_i)$ and $\mathscr{C}_i = \mathscr{B}_{\mathbf{b}}(n_i, C_i)$.
(**c**15) Let $\mathscr{B}_{i+1} = \mathscr{B}_i \setminus \{B_i\} \cup \mathscr{C}_i \cup \{D_i\}$.
(**c**16) End If.
(**c**17) End For.

5. THE PROOF

The key lemma of Chalcraft is Lemma 1 in [5]. We modify that in the following way.

LEMMA 1. *If* $\mathscr{B} = \{B_1, \ldots, B_n\}$ $(n \geq 1)$, *then* $a(\mathscr{B}) \leq w(\mathscr{B})h(\mathscr{B})$.

Proof. We have

$$a(\mathscr{B}) = \sum_{i=1}^{n} a(B_i) = \sum_{i=1}^{n} w(B_i)h(B_i) \leq \sum_{i=1}^{n} w(\mathscr{B})h(B_i)$$

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

which completes the proof. □

We prove the modified Chalcraft's lemmas in which we use the height instead of the semi-perimeter.

LEMMA 2. *Suppose* $w(B) = n^{-t}$ *and Algorithm* **b** *with inputs* n *and* B *terminates with* $m_{\mathbf{b}} = m_{\mathbf{b}}(n, B)$ *and* $\mathscr{B}_{\mathbf{b}} = B_{\mathbf{b}}(n, B)$. *Therefore,*

$$h(\mathscr{B}_{\mathbf{b}}) \leq \sum_{j=n}^{m_{\mathbf{b}}-1} j^{-t}.$$

Proof. The proof is similar to the proof of Lemma 2 in [5]. For completeness, we write it again.

The proof is by induction on the number of squares packed. Of course, if **b** terminates with $m_{\mathbf{b}} = n + 1$, then $h(\mathscr{B}_{\mathbf{b}}) \leq n^{-t} = \sum_{j=n}^{m_{\mathbf{b}}-1} j^{-t}$. We can assume that the lemma is true of all the recursive calls to Algorithm **b**. We can also assume that **b** and all the recursive calls to **b** terminated.

Suppose Algorithm **b** terminates when $i = k$, so $m_{\mathbf{b}} = n_k$. Since Algorithm **b** terminated without placing the next square, $x_k < n_k^{-t} < n^{-t}$, so $h(B_k) = n^{-t}$. Now by induction,

$$h(\mathscr{C}_i) \leq \sum_{j=n_i}^{n_{i+1}-1} j^{-t} \quad \text{for } i < k,$$

$$\sum_{i=1}^{k-1} h(\mathscr{C}_i) \leq \sum_{j=n_1}^{n_k-1} j^{-t} = \sum_{j=n+1}^{m_{\mathbf{b}}-1} j^{-t}.$$

If the condition in (**b**6) was true, then

$$h(\mathscr{B}_{\mathbf{b}}) = \sum_{i=1}^{k-1} h(\mathscr{C}_i) \leq \sum_{j=n+1}^{m_{\mathbf{b}}-1} j^{-t} < \sum_{j=n}^{m_{\mathbf{b}}-1} j^{-t}.$$

If the condition in (**b**7) was true, then

$$h(\mathscr{B}_{\mathbf{b}}) = \sum_{i=1}^{k-1} h(\mathscr{C}_i) + h(B_k) \leq \sum_{j=n+1}^{m_{\mathbf{b}}-1} j^{-t} + n^{-t} = \sum_{j=n}^{m_{\mathbf{b}}-1} j^{-t},$$

which completes the proof. □

LEMMA 3. *We have*

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

$$a^{1-2t} - (b+1)^{1-2t} < (2t-1)\sum_{j=a}^{b} j^{-2t} < (a-1)^{1-2t} - b^{1-2t}. \tag{2}$$

We omit the proof.

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

$$a(\mathscr{B}_i) \geq \sum_{j=n_i}^{\infty} j^{-2t}, \tag{3}$$

$$h(\mathscr{B}_i) \leq \frac{n_i^{1-t}}{2t-1}. \tag{4}$$

Therefore, step (**c**4) *will not fail for this value of* i.

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

$$a(\mathscr{B}_i) \leq w(\mathscr{B}_i)h(\mathscr{B}_i) < \frac{n_i^{1-2t}}{2t-1} \leq \sum_{j=n_i}^{\infty} j^{-2t} \leq a(\mathscr{B}_i),$$

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

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

$$a(\mathscr{B}) \geq \sum_{j=n}^{\infty} j^{-2t}, \tag{5}$$

$$h(\mathscr{B}) \leq \frac{1}{1-t}(n-1)^{1-t}, \tag{6}$$

$$t \leq \frac{2}{3}.$$

If we run Algorithm **c** *with the inputs n and $\mathscr{B}$, then the conditions*

$$a(\mathscr{B}_i) \geq \sum_{j=n_i}^{\infty} j^{-2t}, \tag{7}$$

$$h(\mathscr{B}_i) \leq h(\mathscr{B}) + \sum_{j=n}^{n_i-1} j^{-t}. \tag{8}$$

hold at step (**c**4) *for all $i \geq 1$ for which step* (**c**4) *is executed. 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),

$$\begin{aligned} h(\mathscr{B}_i) &\leq h(\mathscr{B}) + \sum_{j=n}^{n_i-1} j^{-t} \\ &< h(\mathscr{B}) + \frac{1}{1-t}((n_i-1)^{1-t} - (n-1)^{1-t}) \\ &\leq \frac{1}{1-t}(n_i-1)^{1-t}. \end{aligned}$$

Since $t \leq 2/3$,

$$\frac{1}{1-t} \leq \frac{1}{2t-1}.$$

Thus

$$h(\mathscr{B}_i) < \frac{1}{1-t}(n_i-1)^{1-t} \leq \frac{1}{2t-1}(n_i-1)^{1-t} < \frac{n_i^{1-t}}{2t-1}.$$

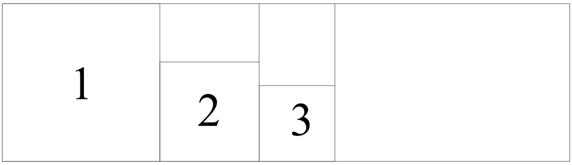

Figure 1 – The squares S_1^t, S_2^t, S_3^t and the set of boxes $\mathscr{B}$.

By Lemma 4, (**c**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 (**c**8) was true for $i-1$, then $h(\mathscr{B}_i) = h(\mathscr{B}_{i-1}) - n_{i-1}^{-t}$ and $n_i = n_{i-1}+1$. Thus by induction,

$$h(\mathscr{B}_i) = h(\mathscr{B}_{i-1}) - n_{i-1}^{-t} \le h(\mathscr{B}) + \sum_{j=n}^{n_{i-1}-1} j^{-t} - n_{i-1}^{-t}$$

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

If the condition in (**c**8) was not true for $i-1$, then we distinguish two cases.

If $w_{i-1} \ge h_{i-1} - n_{i-1}^{-t}$ (that is $h(D_{i-1}) = w_{i-1}$), then

$$\begin{aligned} h(\mathscr{B}_i) &= h(\mathscr{B}_{i-1}) + h(\mathscr{C}_{i-1}) - h(B_{i-1}) + h(D_{i-1}) \\ &= h(\mathscr{B}_{i-1}) + h(\mathscr{C}_{i-1}) - h_{i-1} + w_{i-1} \le h(\mathscr{B}_{i-1}) + h(\mathscr{C}_{i-1}). \end{aligned}$$

If $w_{i-1} < h_{i-1} - n_{i-1}^{-t}$ (that is $h(D_{i-1}) = h_{i-1} - n_{i-1}^{-t}$), then, similarly,

$$h(\mathscr{B}_i) \le h(\mathscr{B}_{i-1}) + h(\mathscr{C}_{i-1}).$$

By induction and Lemma 2,

$$h(\mathscr{B}_i) \le h(\mathscr{B}_{i-1}) + h(\mathscr{C}_{i-1})$$

$$\le h(\mathscr{B}) + \sum_{j=n}^{n_{i-1}-1} j^{-t} + \sum_{j=n_{i-1}}^{n_i-1} j^{-t} = h(\mathscr{B}) + \sum_{j=n}^{n_i-1} j^{-t},$$

which completes the proof. □

Proof of Theorem 1. If the first three squares are packed in the box $B = \zeta(2t) \times 1$ as in Fig. 1 (this is Paulhus's algorithm [11]), then the remaining boxes are

$$\mathscr{B} = \left\{ \left(\zeta(2t) - 1 - 2^{-t} - 3^{-t}\right) \times 1, 2^{-t} \times \left(1 - 2^{-t}\right), 3^{-t} \times \left(1 - 3^{-t}\right) \right\}$$

and

$$h(\mathscr{B}) = \zeta(2t) - 2 \cdot 3^{-t} = 2.639$$
$$< 4.327 = \frac{1}{1-t}(4-1)^{1-t}.$$

By Lemma 5, the Algorithm **c** pack perfectly the squares S_n^t $(n \geq 4)$ into $\mathscr{B}$, which completes the proof. □

Remark 1. The squares S_n^t $(n \geq 1)$ in Theorem 1 can be packed similarly in a square of the right area.

Proof of Theorem 2. If the first three squares are packed in the box $B = \zeta(2t) \times 1$ as in Fig. 1, then the remaining boxes are

$$\mathscr{B} = \left\{ \left(\zeta(2t) - 1 - 2^{-t} - 3^{-t}\right) \times 1, 2^{-t} \times \left(1 - 2^{-t}\right), 3^{-t} \times \left(1 - 3^{-t}\right) \right\}.$$

Observe that $\zeta(2t) - 1 - 2^{-t} - 3^{-t} > 1$, $2^{-t} > 1 - 2^{-t}$ and $1 - 3^{-t} \geq 3^{-t}$ if $t \in [\log_3 2, 2/3]$. Let $f(t) = h(\mathscr{B})$. Thus

$$h(\mathscr{B}) = f(t) = \zeta(2t) - 2 \cdot 3^{-t}.$$

Since

$$g(t) = \frac{1}{1-t} 3^{1-t}$$

is an increasing, $f(t)$ is a decreasing function on the interval $[\log_3 2, 2/3]$ and

$$f(\log_3 2) = 3.41 < 4.06 = g(\log_3 2),$$

the Algorithm **c** pack perfectly the squares S_n^t $(n \geq 4)$ into $\mathscr{B}$, which completes the proof. □

6. DISCUSSION

If we increase the number of the packed squares before we start the Algorithm **c** and do detailed analysis of the height of the boxes, then we can decrease the constant $\log_3 2$. It remains an interesting question to increase the bound 2/3.

Acknowledgments. 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.

REFERENCES

[1] B. Altunkaynak, *An Algebraic Approach to Rectangle Packing Problems.* http://arxiv.org/abs/math/0403194 (2004).

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

[3] V. Boju and L. Funar, *The Math Problems Notebook.* Birkhäuser Basel, 2007, p. 125.

[4] P. Brass, W.O.J. Moser, and J. Pach, *Research Problems in Discrete Geometry.* Springer-Verlag, New York, 2005, p. 499.

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

[6] H.T. Croft, K.J. Falconer, and R.K. Guy, *Unsolved Problems in Geometry.* Springer-Verlag, New York, 1991, pp. 112–113.

[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 Ser. A. **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.

Received September 16, 2019

University of Dunaújváros,
Dunaújváros, Hungary 2400
joosa@uniduna.hu