Proceedings of the Federated Conference on Computer Science and Information Systems pp. 171–175 DOI: 10.15439/2019F88 ISSN 2300-5963 ACSIS, Vol. 18

An algorithm for 1-space bounded cube packing

Łukasz Zielonka
Institute of Mathematics and Physics
UTP University of Science and Technology
Al. Prof. S. Kaliskiego 7, 85-789 Bydgoszcz, Poland
Email: Lukasz.Zielonka@utp.edu.pl

***Abstract*—In this paper, we present a 1-space bounded cube packing algorithm with asymptotic competitive ratio 10.872.**

***Index Terms*—Online algorithms, bin packing, cube, one-space bounded**

I. Introduction

IN THE bin packing problem, we receive a sequence 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 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.

In the online version of packing a crucial parameter is the number of bins available for packing, i.e., *active bins*. Each incoming item is packed into one of the active bins; the remaining bins are not available at this moment. If we close one of the current active bins, we open a new active bin. Once an active bin has been closed, it can never become active again. When the method allows at most t active bins at the same time, it is called *t-space bounded*. Unbounded space model does not impose any limits on the number of active bins. It is natural to expect a packing method to be less efficient with fewer number of active bins. In this paper, we study 1-space bounded 3-dimensional cube packing.

Let S be a sequence of cubes. Denote by $A(S)$ the number of bins used by the algorithm A to pack items from S. Furthermore, denote by $OPT(S)$ the minimum possible number of bins used to pack items from S by the optimal offline algorithm. By the asymptotic competitive ratio for the algorithm A we mean:

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

A. Related work

The one-dimensional case of the space bounded bin packing problem has been extensively studied and the best possible algorithms are known: the Next-Fit algorithm [5] for the one-space bounded model and the Harmonic algorithm [6] when the number of active bins goes to infinity. The questions concerning t-space bounded d-dimensional packing ($d \geq 2$) have been studied in a number of papers. For large number of active bins, Epstein and van Stee [1] presented a $(\Pi_\infty)^d$-competitive space bounded algorithm, where $\Pi_\infty \approx 1.69103$ is the competitive ratio of the one-dimensional algorithm Harmonic. Algorithms for 2-dimensional bin packing with only one active bin were explored for the first time in [8], where the authors give 8.84-competitive algorithm for 2-dimensional bin packing. An improved result of that case can be found in the paper [7], where a 5.155-competitive method is presented. The last article also contains an algorithm for packing squares with competitive ratio at most 4.5. In [4], a 4.84-competitive 1-space bounded 2-dimensional bin packing algorithm was presented. Grzegorek and Januszewski [3] presented a 3.5^d-competitive as well as a $12{\cdot}3^d$-competitive online d-dimensional hyperbox packing algorithm with one active bin. The d-dimensional case of 1-space bounded hyper-cube packing was discussed in [9], where a 2^{d+1}-competitive algorithm was described. The aim of this paper is to improve the upper bound (2^{3+1}) in the 3-dimensional case. We present 10.872-competitive 1-space bounded cube packing algorithm.

B. Our results

The algorithm presented in this article considers packing items (cubes of edges not greater than 1) into one active cube of edge 1. The main packing method is a bit like the classic computer game Tetris. The packing method which we describe is similar to the method presented by Grzegorek and Januszewski in [2]. The algorithm distinguishes types of items what determines a method for packing a specific item in a bin. Items that are considered big enough are packed from top to bottom. Different types of small items are packed from bottom upwards. The algorithm handles small items in a Tetris manner: to determine a place to pack an item a part of a bin is temporarily divided into congruent cuboids of appropriate size. Then an item is packed as low as possible inside a carefully chosen cuboid.

In Section II we give a 1-space bounded cube packing algorithm with the ratio 10.872 .

II. The *one-space*-algorithm

Let S be a sequence of cubes $Q_1, Q_2, \ldots$. Denote by a_i the edge length of Q_i.

- an item Q_i is *huge*, provided $a_i > 1/2$;
- an item Q_i is *big*, provided $1/4 < a_i \leq 1/2$;
- an item Q_i is *small*, provided $a_i \leq 1/4$; a small item Q_i is of *type* k provided $2^{-k-1} < a_i \leq 2^{-k}$.

Let $\mathcal{B}$ be the active bin. To shorten the notation, a cuboid whose edges have lengths $a \times a \times b$ will be called an (a, b)-cuboid.

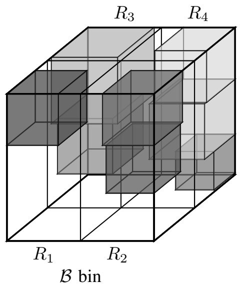

Fig. 1. *Big items – the darker an item's colour, the later it arrived*

A. Description of the *one-space-algorithm*

(a) In packing items we distinguish coloured and white (not coloured) space. Items are placed only in the white space. Each newly opened bin is white.

(b) We divide each freshly opened bin into $(1/2, 1)$-cuboids. These cuboids are named R_1, R_2, R_3, R_4 in an arbitrary order.

(c) Huge items (edge $> 1/2$) are packed alone into a bin, i.e., if Q_i is huge, then we close the active bin and open a new bin to pack this item. After packing Q_i we close the bin and open a new active bin.

(d) If Q_i is big ($1/4 <$ edge $\leq 1/2$) we find the highest indexed R_j such that Q_i can be packed into it. We pack Q_i into R_j along the edge of $\mathcal{B}$ as high as it is possible (see Figs. 1 and 3). If such a packing is not possible, we close the active bin, open a new active bin and pack Q_i into it.
When a big item is packed, it colours the space covered by itself.

(e) If Q_i is a small item of type k ($2^{-k-1} <$ edge $\leq 2^{-k}$) (see Figs. 2 and 3) we find the lowest indexed R_j such that Q_i can be packed into it. Since j is fixed now, we will write R instead of R_j.
We temporarily divide R into $(2^{-k}, 1)$-cuboids called $R(1), \ldots, R(4^{k-1})$. Denote by $t(n)$ the distance between the top of $R(n)$ and the top of the topmost item packed in $R(n)$ for $n = 1, \ldots, 4^{k-1}$ (see Fig. 5, right) and let η be an integer such that $t(\eta) = \max\{t(1), \ldots, t(4^{k-1})\}$.
We pack Q_i into $R(\eta)$ as low as possible. The result of packing Q_i is the colouring of the $(2^{-k}, 1 - t(\eta) + a_i)$-cuboid contained in the bottom of $R(\eta)$ (see Fig. 5, right, where $\eta = 2$ before Q_{14} was packing).
If such a packing is not possible, then we close the active bin and open a new active bin to pack Q_i.

B. Competitive ratio

Let P_j for $j = 1, \ldots, 16$ be $(1/4, 1)$-cuboids with pairwise disjoint interiors. Each cuboid R_i for $i \in \{1, 2, 3, 4\}$ is divided into four cuboids $P_{4i-3}, \ldots, P_{4i}$ (see Fig. 4).

Lemma 1. *Assume that only small items were packed into $\mathcal{B}$. Assume that $j \in \{1, 2, \ldots, 16\}$. Denote by n the number of items packed into P_j and by t_n the distance between the bottom of $\mathcal{B}$ and the top of the topmost item packed into P_j. The total volume v_n of small items packed into P_j is greater than*

$$f(t_n) = \frac{19}{2048} \cdot t_n - \frac{13}{16384}.$$

Moreover, if the topmost packed item is of type 2, then

$$v_n > f_+(t_n) = \frac{19}{2048} \cdot t_n.$$

Proof. Without loss of generality we can assume that $P_j = [0, 1/4] \times [0, 1/4] \times [0, 1]$. We will prove the result using induction over the number n of packed items.

First assume that only one item Q_b was packed into P_j. Obviously, $t_1 = a_b$. Let

$$\varphi(a) = a^3 - \frac{19}{2048} a.$$

The function $\varphi(a)$ for $a > 0$ has a minimum at

$$a_0 = \sqrt{\frac{19}{6144}}.$$

A computation shows that

$$\varphi(a_0) > -\frac{1}{2} \cdot \frac{13}{16384} \tag{1}$$

(this lower bound will be useful in the last part of the proof). We get

$$v_1 = a_b^3 > \frac{19}{2048} \cdot t_1 - \frac{1}{2} \cdot \frac{13}{8192} = f(t_1).$$

Moreover, if $1/8 < a_b \leq 1/4$, then $v_1 = a_b^3 > \frac{19}{2048} a_b = f_+(t_1)$.

Now assume that the statement holds for at most n items packed into P_j (this is our inductive assumption). Let Q_u be the $(n+1)$st item packed into P_j and let t_{n+1} be the distance between the bottom of P_j and the top of the topmost item (from among $n+1$ items $Q_b, \ldots, Q_u$) packed into P_j.

If $a_u > 1/8$, then $t_{n+1} = t_n + a_u$. Using the inductive assumption,

$$v_{n+1} = v_n + a_u^3 > f(t_n) + a_u^3 = \frac{19}{2048} \cdot t_n - \frac{13}{16384} + a_u^3.$$

Since

$$\varphi'(a) = 3a^2 - \frac{19}{2048} > 3 \cdot \frac{1}{64} - \frac{19}{2048} > 0$$

for $a > 1/8$, we get

$$\varphi(a) > \varphi(\frac{1}{8}) = \frac{13}{16384}$$

for $a > 1/8$. Consequently,

$$\begin{aligned} v_{n+1} > f(t_n) + a_u^3 &= \frac{19}{2048}(t_n + a_u) + \varphi(a_u) - \frac{13}{16384} \\ &\geq \frac{19}{2048}(t_n + a_u) = f_+(t_n + a_u) = f_+(t_{n+1}). \end{aligned}$$

Finally, consider the case when $a_u \leq 1/8$. First, we choose the topmost packed item Q^1 with edge greater than $1/8$ and denote by τ the distance between the bottom of P_j and the top of Q^1 (see Fig. 6, left). If there is no such item, then we

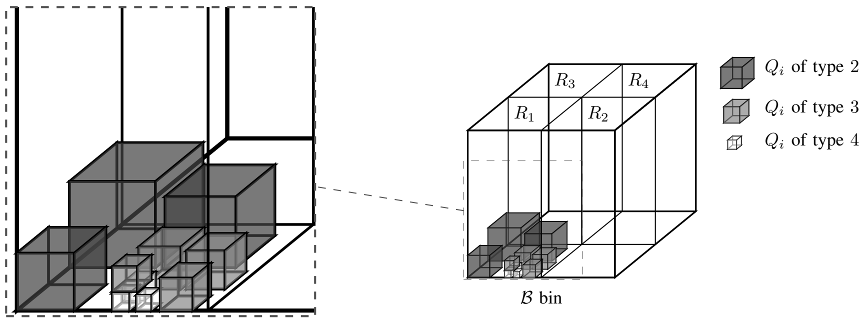

Fig. 2. *Small items*

take $\tau = 0$. The total volume of items packed up to τ, by the inductive assumption, is not smaller than $f_+(\tau)$. Above Q^1 we divide P_j into four $(1/8, 1-\tau)$-cuboids $P_j^1, P_j^2, P_j^3, P_j^4$. Denote by $Q_1^l, \ldots, Q_{v_l}^l$ the items from among $Q_b, \ldots, Q_{u-1}$ packed into P_j^l above Q^1 (if any) for each $l = 1, 2, 3, 4$. Moreover, denote by t_n^l the distance between the bottom of P_j and the top of the topmost item from among $Q_b, \ldots, Q_{u-1}$ packed into P_j^l and let $t_n^* = \min(t_n^1, t_n^2, t_n^3, t_n^4)$ (see Fig. 6, right). Clearly, $t_n^* \geq \tau$ and $t_n^* \leq t_n$.

If $t_n^* + a_u \leq t_n$, then $t_{n+1} = t_n$. Consequently,

$$v_{n+1} \geq v_n + a_u^3 = f(t_n) + a_u^3 = f(t_{n+1}) + a_u^3 > f(t_{n+1}).$$

If $t_n^* + a_u > t_n$, then $t_{n+1} = t_n^* + a_u$. Items $Q_1^l, \ldots, Q_v^l$ were packed into $(1/8, t_n^l - \tau)$-cuboid P_j^l. Let $h(P_j^l) = [0, 1/4] \times [0, 1/4] \times [0, 2t_n^l - 2\tau]$ be the image of P_j^l in a homothety h of ratio 2. By the inductive assumption, the total volume of cubes $h(Q_1^l), \ldots, h(Q_v^l)$ is not smaller than $\frac{19}{2048}(2t_n^l - 2\tau) - \frac{13}{16384} = f(2t_n^l - 2\tau)$. Since the volume of each $h(Q_i^l)$ is 8 times greater than the volume of Q_i^l, it follows that the total volume of cubes $Q_1^l, \ldots, Q_v^l$ is not smaller than $\frac{1}{8} f(2t_n^l - 2\tau)$.

Consequently,

$$\begin{aligned} v_{n+1} &\geq f_+(\tau) + 4 \cdot \frac{1}{8} f\left(2t_n^* - 2\tau\right) + a_u^3 \\ &= a_u^3 + \frac{19}{2048} t_n^* - \frac{1}{2} \cdot \frac{13}{16384}. \end{aligned}$$

By (1) we know that

$$\varphi(a_0) > -\frac{1}{2} \cdot \frac{13}{16384}.$$

Consequently,

$$\begin{aligned} v_{n+1} &\geq \varphi(a_u) + \frac{19}{2048}(t_n^* + a_u) - \frac{1}{2} \cdot \frac{13}{16384} \\ &> \frac{19}{2048}(t_n^* + a_u) - \frac{13}{16384} = f(t_{n+1}). \end{aligned}$$

□

Lemma 2. *Define* $V_3 = 101/1024$. *Let* S *be a finite sequence of cubes and let* ν *be the number of bins used to pack items from* S *by the one-space-algorithm. Moreover, let* m *be the number of huge items in* S. *The total volume of items in* S *is greater than* $2^{-3} \cdot m + V_3(\nu - 2m - 1)$.

Proof. Among ν bins used to pack items from S by the *one-space*-algorithm the first $\nu - 1$ bins will be called *full*. Let Q_z be the first item from S which cannot be packed into a full bin $\mathcal{B}$ by the *one-space*-algorithm. Clearly, Q_z is the first item packed into the next bin.

Denote by $v_{\mathcal{B}}$ the sum of volumes of items packed into $\mathcal{B}$.

If the incoming item Q_z is huge, then the average occupation ratio in both bins $\mathcal{B}_j$ and the next bin $\mathcal{B}_{j+1}$ into which Q_z was packed is greater than $1/2^4$. Obviously, there are $2m$ such bins.

It is possible that the last bin is almost empty.

To prove Lemma 2 it suffices to show that if Q_z is not huge and if no huge item was packed into $\mathcal{B}$, then $v_{\mathcal{B}} > V_3$ (the number of such bins equals $\nu - 2m - 1$).

Case 1: Q_z is small and all items packed into $\mathcal{B}$ are small.

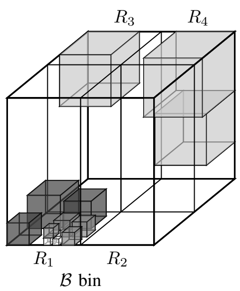

Fig. 3. *one-space-algorithm*

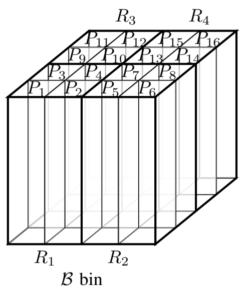

Fig. 4. $(1/4, 1)$-*cuboids* P_j

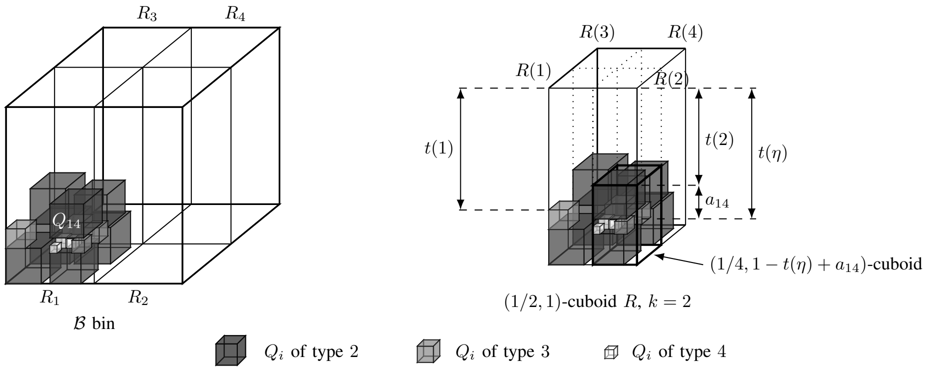

Fig. 5. *Packing small items into* $(2^{-k}, 1)$*-cuboids*

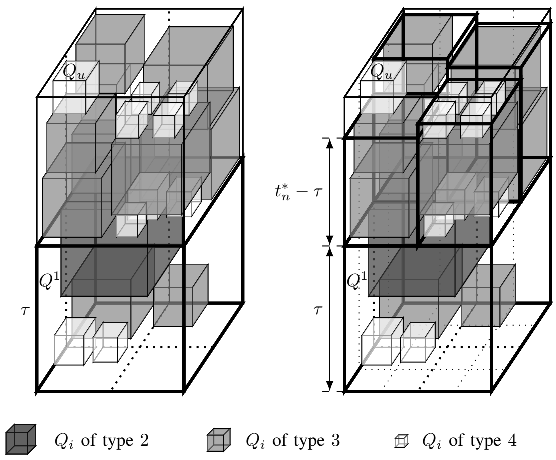

Fig. 6. *The division*

Since $a_z \le 1/4$, it follows that each P_i is packed up to height at least $3/4$. By Lemma 1 we deduce that

$$v_{\mathcal{B}} > 4^2 f\Big(\frac{3}{4}\Big) = 16 \cdot \Big(\frac{19}{2048} \cdot \frac{3}{4} - \frac{13}{4 \cdot 16384}\Big) = V_3.$$

Case 2: Q_z *is small and a big item was packed into* $\mathcal{B}$.

The volume of a big item Q_b with edge t is equal to $t^3 > t \cdot \left(\frac{1}{4}\right)^2$. In considerations presented in Case 1 we accept that the total volume of small items packed into R_j up to height t equals $4f(t)$. It is easy to see that

$$4f(t) < \frac{1}{16} \cdot t.$$

As a consequence, $v_{\mathcal{B}} > V_3$.

Case 3: Q_z *is a big item and all items packed into* $\mathcal{B}$ *are small*

Assume that there is $(2^{-2}, 1)$-cuboid $R_j(n)$ $(j \in \{1,2,3\},\ n \in \{1,2,3,4\})$ such that the distance between its top and the top of the topmost item packed into it is greater than 1/8 and denote by R_+ first such cuboid. The total volume of items packed into R_+ is greater than $f(3/4)$. The total volume of items packed into each cuboid preceding R_+ is greater than $f(7/8)$. The total volume of items packed into

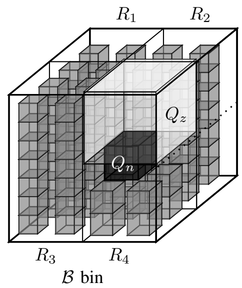

Fig. 7. *Case 3*

each of remaining cuboids is greater than $\frac{3}{4} \cdot \frac{1}{8^2} > f(7/8)$ (in such a cuboid only items greater than $1/8$ were packed).

Denote by Q_n the topmost small item packed in R_4 (as in Fig. 7). Since $a_z \le 1/2$ and Q_z cannot be packed in R_4, it follows that

$$v_{\mathcal{B}} > \big(16 - 5\big) f\Big(\frac{7}{8}\Big) + f\Big(\frac{3}{4}\Big) + 4f\Big(\frac{1}{2} - a_n\Big) + a_n^3.$$

Denote by $\gamma(a_n)$ the function on the right-hand side of this formula. This function for positive a has a minimum at $a_0 = \sqrt{\frac{19}{1536}}$.

A computation shows that $\gamma(a_0) > V_3$. Consequently, $v_{\mathcal{B}} > V_3$.

If there is no $(2^{-2}, 1)$-cuboid $R_j(n)$ $(j \in \{1,2,3\},\ n \in \{1,2,3,4\})$ such that the distance between its top and the top of the topmost item packed into it is greater than $1/8$, then

$$v_{\mathcal{B}} > \big(16 - 4\big) f\Big(\frac{7}{8}\Big) + 4f\Big(\frac{1}{2} - a_n\Big) + a_n^3.$$

Since $f(7/8) > f(3/4)$, we get $v_{\mathcal{B}} > \gamma(a_0) > V_3$.

Case 4: Q_z *is big and a big item was packed into* $\mathcal{B}$

Similarly as in Case 2 we get

$$4f(t) < t^3.$$

We deduce by Case 3 that $v_{\mathcal{B}} > V_3$.

□

Theorem 1. *The asymptotic competitive ratio for the one-space-algorithm is not greater than* $1098/101 \approx 10.8713$.

Proof. Let S be a sequence of items of total volume v, let m denote the number of huge items in S and let μ be the number of bins used to pack items from S using the one-$space$-algorithm. Obviously, $OPT(S) \geq v$ as well as $OPT(S) \geq m$.

By Lemma 2 we get $v > \frac{1}{2^3} \cdot m + V_3 \cdot (\mu - 2m - 1)$, i.e.,

$$\mu < \frac{v}{V_3} + m\Big(2 - \frac{1}{2^3 V_3}\Big) + 1.$$

It is easy to check that $2 - \frac{1}{8V_3} > 0$.

If $m < v$, then

$$\frac{\mu}{OPT(S)} \leq \frac{\mu}{v} < \frac{\frac{v}{V_3} + v\big(2 - \frac{1}{2^3 V_3}\big) + 1}{v} = \frac{2^3 - 1}{2^3 V_3} + 2 + \frac{1}{v}.$$

If $v \leq m$, then

$$\frac{\mu}{OPT(S)} \leq \frac{\mu}{m} \leq \frac{\frac{m}{V_3} + m\big(2 - \frac{1}{2^3 V_3}\big) + 1}{m} = \frac{2^3 - 1}{2^3 V_3} + 2 + \frac{1}{m}.$$

Consequently, the asymptotic competitive ratio for the one-$space$-algorithm is not greater than

$$\frac{7}{8} \cdot \frac{1024}{101} + 2 = \frac{1098}{101} < 10.872.$$

□

References

[1] L. Epstein and R. van Stee. Optimal online algorithms for multidimensional packing problems. *SIAM Journal on Computing*, 35(2):431–448, 2005.

[2] P. Grzegorek and J. Januszewski. A note on one-space bounded square packing. *Information Processing Letters*, 115(11):872–876, 2015.

[3] P. Grzegorek, J. Januszewski. Drawer algorithms for 1-space bounded multidimensional hyperbox packing. *Journal of Combinatorial Optimization*, 37(3): 1011-1044, 2019.

[4] J. Januszewski and Ł. Zielonka. Online packing of rectangular items into square bins. In R. Solis-Oba and R. Fleischer, editors, *Approximation and Online Algorithms. WAOA 2017*, volume 10787 of *Lecture Notes in Computer Science*, pages 147–163, Cham, 2018. Springer.

[5] D. S. Johnson. Fast algorithms for bin packing. *Journal of Computer and System Sciences*, 8(3):272–314, 1974.

[6] C.-C. Lee and D.-T. Lee. A simple on-line bin-packing algorithm. *J. ACM*, 32(3):562–572, July 1985.

[7] Y. Zhang, J. Chen, F. Y. L. Chin, X. Han, H.-F. Ting, and Y. H. Tsin. Improved online algorithms for 1-space bounded 2-dimensional bin packing. In O. Cheong, K.-Y. Chwa, and K. Park, editors, *Algorithms and Computation*, pages 242–253, Berlin, Heidelberg, 2010. Springer.

[8] Y. Zhang, F. Y. L. Chin, and H.-F. Ting. One-space bounded algorithms for two-dimensional bin packing. *International Journal of Foundations of Computer Science*, 21(06):875–891, 2010.

[9] Y. Zhang, F. Y. L. Chin, H.-F. Ting, and X. Han. Online algorithms for 1-space bounded multi dimensional bin packing and hypercube packing. *Journal of Combinatorial Optimization*, 26(2):223–236, 2013.