Online Square Packing

Sándor P. Fekete* Tom Kamphans* Nils Schweer*

Abstract

We analyze the problem of packing squares in an online fashion: Given an semi-infinite strip of width 1 and an unknown sequence of squares with side lengths in $[0, 1]$ that arrive from above, one at a time. The objective is to pack these items as they arrive, minimizing the resulting height. Just like in the classical game of Tetris, each square must be moved along a collision-free path to its final destination; in addition, we may have to account for gravity in both motion and position (i.e, squares are not allowed to move up and any final destination has to be supported from below). This problem has been considered before; the best previous result is by Azar and Epstein, who gave a 4-competitive algorithm in a setting without gravity, based on ideas of shelf-packing, with the possibility of letting squares "hang in the air" in order to assign them to different levels, allowing an analysis that is reminiscent of some bin-packing arguments.

We present an algorithm with competitive factor $\frac{34}{13} \approx 2.6154$, with or without the presence of gravity.

1 Introduction

Packing problems arise in many different situations, either concrete (where actual physical objects have to be packed), or abstract (where the space is virtual, e.g., in scheduling). Even in a one-dimensional setting, computing an optimal set of positions in a container for a known set of objects is a classical, hard problem. Having to deal with two-dimensional objects adds a variety of difficulties; one of them is the more complex structure of feasible placements; see, for example, Fekete et al. [8]. Another one is actually moving the objects into their final locations without causing collisions or overlap along the way.

A different kind of difficulty may arise from a lack of information: in many settings, objects have to be assigned to their final locations one by one, without knowing future items. Obviously, this makes the challenge even harder.

In this paper, we consider online packing of squares into a vertical strip of unit width. Squares arrive from above in an online fashion, one at a time, and have to be moved to their final positions. On this path, a square may move only through unoccupied space; in allusion to the well-known computer game, this is called the *Tetris constraint*. In addition, an item is not allowed to move upwards and has to be supported from below when reaching its final position; these conditions are called *gravity constraints*. The objective is to minimize the total height of the occupied part of the strip.

1.1 Related Work

There is a considerable amount of work of online rectangle packing without the Tetris constraint, see Coffman et al. [4] for a quick overview. Most notably, shelf-packing approaches [2] (in which the strip is subdivided into shelves of certain heights, and no two rectangles within the same shelf get placed on top of each other) cannot do better than an asymptotic worst-case ratio of 1.691..., which can be achieved by Harmonic shelf packing, see Csirik and Woeginger [5].

Every reader is certainly familiar with the classical game of Tetris: Given a strip of fixed width, find online placements for a sequence of objects falling down from above, such that space is utilized as best as possible[1]. In this process, no item can ever move upward or collide with another object. An item will come to a stop only if it is supported from below, and each placement has to be fixed before the next item arrives. Even when disregarding the difficulty of ever-increasing speed, Tetris is notoriously difficult: As was shown by Breukelaar et al. [3], Tetris is PSPACE-hard, even for the original, limited set of different objects.

Tetris-like online packing has been studied before. Most notably, Azar and Epstein [1] considered online packing of rectangles into a strip; just like in Tetris, they considered the situation with or without rotation of objects. For the case without rotation, they showed that no constant competitive ratio is possible, unless there is a fixed-size lower bound of ε on the size of the objects, in which case there is an upper bound of $O(\log \frac{1}{\varepsilon})$. For the case in which rotation is possible, they showed a 4-competitive strategy, based on shelf-packing methods, with all rectangles being rotated to be placed on their narrow sides; until now, this is also the best deterministic upper bound for squares.

*Braunschweig University of Technology, Computer Science, Algorithms Group, 38106 Braunschweig, Germany

[1]Obviously, there is a slight difference in the objective function, because Tetris aims at filling rows. In actual optimization scenarios, this is less interesting, as it is not critical whether a row is used to precisely 100%—in particular, as full rows do not magically disappear in real life.

In this strategy, gravity is not taken into account, as items are allowed to be placed at appropriate levels, even if they are unsupported.

More recently, Coffmann, Downey, and Winkler [4] considered probabilistic aspects of online rectangle packing with Tetris constraint, without allowing rotations. If n rectangle sizes are chosen uniformly at random from the interval $[0, 1]$, they showed that there is a lower bound of $(0.31382733...)n$ on the expected height of the strip; using another kind of level-type strategy, which arises from the bin-packing–inspired *Next Fit Level*, they established an upper bound of $(0.36976421...)n$ on the expected height. Epstein and van Stee gave an optimal bounded space algorithm for online hypercube bin-packing for any dimension $d \geq 2$, and unbounded-space, competitive algorithms for square and cube packing [6, 7].

2 Problem and Definitions

We are given a strip, S, of width 1 which is closed at the bottom and has infinite height, as well as a sequence, $A_1, \ldots, A_n$, of squares with side length $|A_i| \in [0, 1]$. The squares are presented one by one; the next square always arrives above all previously placed squares. Our goal is to find a nonoverlapping placement of squares in the strip that keeps the height of the occupied area as low as possible. The sides of the squares in the placement are parallel to the sides of the strip. A packing has to fulfill two additional constraints:

Gravity constraint: A square must be packed on top of a square that has been packed before (i.e., the intersection of the upper square's bottom and the lower square's top must be a line segment); in addition, no square may ever move up.

Tetris constraint: At the time a square is placed, there is a collision-free path from the top of the strip to the square's final position.

We consider the online problem; that is, the sequence $\mathcal{A}$ is not known in advance. Our strategy gets the squares one by one and has to place a square before it gets the next.

We call a square A_j a *bottom neighbor* of a square A_i if the top side of A_j and the bottom side of A_i overlap in more than one point. For every square A_i we define the *bottom sequence* as follows: A_i is the first element of this sequence and the next element is chosen as an arbitrary bottom neighbor of the previous element. The sequence ends if no such neighbor exists.

3 Algorithm

Consider two vertical lines of infinite length going upwards from the bottom side of S and parallel to the left and the right side of S. We call the area between these lines a *slot*, the lines the *left boundary* and the *right boundary* of the slot, and the distance between the lines the *width* of the slot.

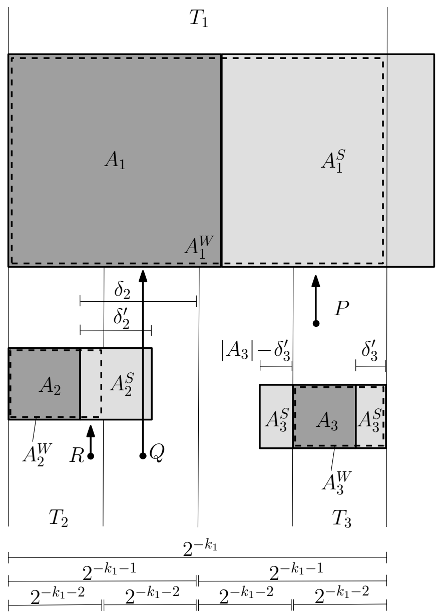

Figure 1: Squares A_i with their *shadows* A_i^S and their *widening* A_i^W. δ_2' is equal to $|A_2|$ and δ_3' is equal to δ_3. The points P and Q are charged to A_1. R is not charged to A_1, but to A_2.

Now our algorithm works as follows: We divide the strip S of width 1 into slots of different widths; for every $j = 0, 1, 2\ldots$ we create 2^j slots of width $\frac{1}{2^j}$ side by side; that is, we divide S into one slot of width 1, two slots of width $\frac{1}{2}$, four slots of width $\frac{1}{4}$ and so on. Note that a slot of width 2^{-i} contains 2 slots of width 2^{-i-1}; see Fig. 1.

For every square A_i we round the side length $|A_i|$ to the smallest number $\frac{1}{2^{k_i}}$ that is larger than or equal to $|A_i|$. We place A_i in the slot of width 2^{-k_i} that allows A_i to be placed as near to the bottom of S as possible by moving A_i down along the left boundary of the chosen slot until another square is reached. We call this algorithm *SlotAlgorithm*. It clearly satisfies the Tetris and the Gravity constraints and next we show that the produced height is at most 2.6154 times the height of an optimal packing.

4 Analysis

Let A_i be a square placed by the SlotAlgorithm in a slot T_i of width 2^{-k_i}. Let δ_i be the distance between the right side of A_i and the right boundary of the slot of width 2^{-k_i+1} that contains A_i and $\delta_i' := \min\{|A_i|, \delta_i\}$. We call the area obtained by enlarging A_i by δ_i' to the right and by $|A_i| - \delta_i'$ to the left the *shadow* of A_i and denote it by A_i^S. Thus, A_i^S is an area of the same size as A_i and lies completely inside a slot of twice the width of A_i's slot. Moreover, we define the *widening* of A_i as $A_i^W = (A_i \cup A_i^S) \cap T_i$; see Fig. 1.

Now, consider a point P in T_i that is not inside an A_j^W for any square A_j and that does not lie on the left or the right boundary of any slot. We charge P to the square A_i if A_i^W is the first widening that intersects the vertical line going upwards from P. We denote by F_{A_i} the set of all points charged to A_i and by $|F_{A_i}|$ its area. The set of points lying on the left or the right boundary of any slot has area zero and is therefore neglected in the rest of this section. For the analysis, we place a closing square, A_{n+1}, of side length 1 on top of the packing.[2] Therefore, every point in the packing that does not lie inside an A_j^W is charged to a square. Because A_i and A_i^S have the same area, we can bound the height, ALG, of the packing produced by the algorithm as follows:

$$ALG \leq 2\sum_{i=1}^{n} |A_i|^2 + \sum_{i=1}^{n+1} |F_{A_i}|$$

Theorem 1 *The SlotAlgorithm is competitive with factor 2.6154.*

Proof. The height of an optimal packing is at least $\sum_{i=1}^{n} |A_i|^2$ and, therefore, it suffices to show that $|F_{A_i}| \leq 0.6154 \cdot |A_i|^2$ holds for every square A_i. We construct for every A_i a sequence of squares $B_1^i, B_2^i, \ldots, B_m^i$ with $B_1^i = A_i$ (to ease notation, we omit the superscript i in the following). We denote by E_{B_j} the extension of the bottom side of B_j to the left and to the right (Fig. 2). We will show that by an appropriate choice of the sequence we can bound the area of the part of F_{B_1} that lies between a consecutive pair of extensions, E_{B_j} and $E_{B_{j+1}}$, in terms of B_{j+1} and the slot widths. From this we will derive the upper bound on the area of F_{B_1}. We assume throughout the proof that the square B_j, $j \geq 1$, is placed in a slot, T_j, of width 2^{-k_j}. Note that F_{B_1} is completely contained in T_1.

A slot is called *active (with respect to E_{B_j} and B_1)* if there is a point in the slot that lies below E_{B_j} and that is charged to B_1 and *nonactive* otherwise. If it is clear from the context we leave out the B_1.

The sequence of squares is chosen as follows: B_1 is the first square and the square B_{j+1}, $j = 1, \ldots, m-1$ is chosen as the smallest one that intersects or touches E_{B_j} in an active slot (w.r.t. E_{B_j} and B_1) of width 2^{-k_j} and that is not equal to B_j. The sequence ends if all slots are nonactive w.r.t. to an extension E_{B_m}. We claim the following:

(i) B_{j+1} exists for $j+1 \leq m$ and $|B_{j+1}| \leq 2^{-k_j-1}$ for $j+1 \leq m-1$.

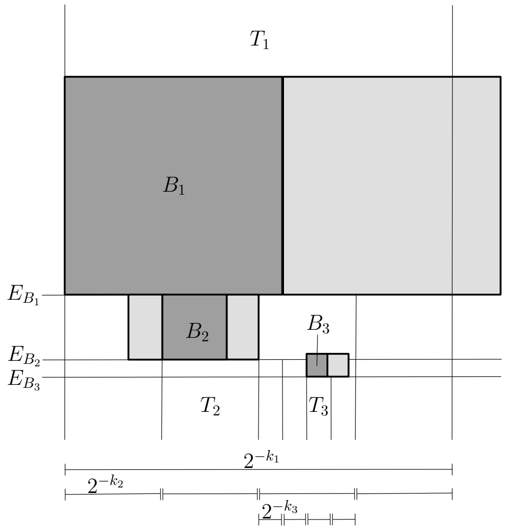

Figure 2: The first three squares of the sequence. In this example, B_2 is the smallest square that bounds B_1 from below. B_3 is the smallest one that intersects E_{B_2} in an active slot (w.r.t. E_{B_2}) of width $\frac{1}{2^{k_2}}$. T_2 is nonactive (w.r.t. E_{B_2}) and, of course, also w.r.t. all extension E_{B_j}, $j \geq 3$

(ii) The number of active slots (w.r.t. E_{B_j}) of width 2^{-k_j} is at most

$$\begin{cases} 1 & , \text{for } j = 1 \\ \prod_{i=2}^{j} \left(\frac{1}{2^{k_{i-1}}} 2^{k_i} - 1\right) & , \text{for } j \geq 2 \end{cases}$$

(iii) The area of the part of F_{B_1} that lies in an active slot of width 2^{-k_j} between E_{B_j} and $E_{B_{j+1}}$ is at most $2^{-k_j}|B_{j+1}| - 2|B_{j+1}|^2$.

We prove the claims by induction. If B_1 is placed on the bottom of S, F_{B_1} has size 0 and B_1 is the last element of the sequence. Otherwise, the square B_1 has at least one bottom neighbor, which is a candidate for the choice of B_2. If $|B_2| > 2^{-k_1-1}$, then B_2 is a bottom neighbor of B_1 of side length greater than or equal to B_1 and, thus, all points below are charged to B_2. Hence, slot T_1 is nonactive and F_{B_1} is of size zero.

If (i) is fulfilled T_1 is the only slot of width 2^{-k_1} that is active. Moreover, we conclude that the area of the part of F_{B_1} that lies between E_{B_1} and E_{B_2} is at most $2^{-k_1}|B_2| - 2|B_2|^2$ (Fig. 2). Note that we can subtract the area of B_2 twice, because B_2^S was defined to lie completely inside a strip of width $2^{-k_2+1} \leq 2^{-k_1}$ and is of same area as B_2.

Now suppose for a contradiction that the $(j+1)$th element does not exist for $j+1 \leq m$. Let T' be an active slot in T_1 (w.r.t. E_{B_j}) of width 2^{-k_j} where E_{B_j} is not intersected by a square in T'. If there is an ε such that for every point $P \in (T' \cap E_{B_j})$ there is a point P' at a distance of ε below P that is charged to B_1 we

[2]This does not affect the competitive factor asymptotically.

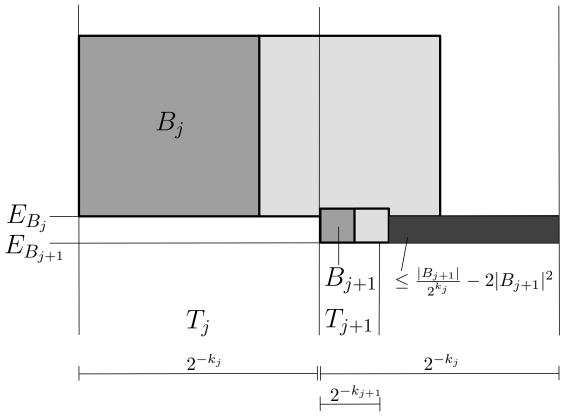

Figure 3: The part of F_{B_1} (darkest gray) that lies between E_{B_j} and $E_{B_{j+1}}$ in an active slot of width $\frac{1}{2^{k_j}}$ is at most $\frac{1}{2^{k_j}}|B_{j+1}|-2|B_{j+1}|^2$ because points in B^W_{j+1} are not charged to B_1.

conclude that there would have been a better position for B_j. Hence, there is at least one point, Q, below E_{B_j} that is not charged to B_1. Consider the bottom sequence of the square Q is charged to. This sequence has to intersect E_{B_j} outside of T' (by choice of T'). But then one of its elements has to intersect the left or the right boundary of T' and we can conclude that this square has at least the width of T', because (by the algorithm) a square with rounded side length $2^{-\ell}$ cannot cross a slot's boundary of width larger than $2^{-\ell}$. In turn, a square larger than T' completely covers T' and T' cannot be active w.r.t. to E_{B_j} and B_1. Thus, all points in T' below E_{B_j} are charged to this square; a contradiction. This proves the existence of B_{j+1}. Because we chose B_{j+1} to be of minimal side length, $|B_{j+1}| \geq 2^{-k_j}$ would imply that all slots inside T are nonactive (w.r.t. E_{B_j}). Therefore, if B_{j+1} is not the last element of the sequence, $|B_{j+1}| \leq 2^{-k_j-1}$ holds.

By the induction hypothesis there are at most $(\frac{1}{2^{k_1}}2^{k_2}-1)\cdot(\frac{1}{2^{k_2}}2^{k_3}-1)\cdot\ldots\cdot(\frac{1}{2^{k_{j-2}}}2^{k_{j-1}}-1)$ active slots of width $2^{-k_{j-1}}$ (w.r.t. $E_{B_{j-1}}$). Each of these slots contains $2^{k_j-k_{j-1}}$ slots of width 2^{-k_j} and in every active slot of width $2^{-k_{j-1}}$ at least one slot of width 2^{-k_j} is nonactive because we chose B_j to be of minimum side length. Hence, the number of active slots (w.r.t. E_{B_j}) is a factor of $(\frac{1}{2^{k_{j-1}}}2^{k_j}-1)$ larger than the number of active slots (w.r.t. $E_{B_{j-1}}$).

Again by the choice of B_{j+1} and by the fact that in every active slot of width 2^{-k_j} there is at least one square, B, intersecting E_{B_j} (points below B^W are not charged to B_1) we conclude that the area of F_{B_1} between E_{B_j} and $E_{B_{j+1}}$ is at most $2^{-k_j}|B_{j+1}|-2|B_{j+1}|^2$ in every active slot of width 2^{-k_j} (Fig. 3).

Altogether, we get an upper bound on $|F_{B_1}|$ of

$$\frac{|B_2|}{2^{k_1}}-2|B_2|^2+\sum_{j=2}^{m}\left[\left(\frac{|B_{j+1}|}{2^{k_j}}-2|B_{j+1}|^2\right)\prod_{i=1}^{j-1}\left(\frac{2^{k_{i+1}}}{2^{k_i}}-1\right)\right]$$

This expression is maximized if we choose $|B_{i+1}| = \frac{1}{2^{k_i+2}}$ for $i = 1,\ldots,m$. This implies $k_i = k_1 + 2(i-1)$ and we get the upper bound

$$|F_{B_1}| \leq \sum_{i=0}^{\infty} \frac{3^i}{2^{2k_1+4i+3}} .$$

The fraction $\frac{|F_{B_1}|}{|B_1|^2}$ is maximized if we choose $|B_1|$ as small as possible; that is, $B_1 = 2^{-(k_1+1)} + \varepsilon$. We conclude:

$$\begin{aligned}\frac{|F_{B_1}|}{|B_1|^2} &\leq \sum_{i=0}^{\infty} \frac{2^{2k_1+2}\cdot 3^i}{2^{2k_1+4i+3}} = \sum_{i=0}^{\infty}\frac{3^i}{2^{4i+1}} \\ &= \frac{1}{2}\cdot\sum_{i=0}^{\infty}\left(\frac{3}{16}\right)^i = \frac{8}{13} = 0.6154...\end{aligned}$$

□

5 Conclusion

We presented an algorithm that is 2.6154-competitive. We believe that our algorithm can be improved; at this point, the best known lower bound is 1.25. We believe that our approach can be extended to higher dimensions. Rectangles may require a slightly different analysis. These topics will be the subject of future research. It is an open question, whether our analysis is tight or can be improved. The best lower bound for *SlotAlgorithm* known to us is 2.

References

[1] Y. Azar and L. Epstein. On two dimensional packing. *J. Algorithms*, 25:290–310, 1997.

[2] B. S. Baker and J. S. Schwarz. Shelf algorithms for two-dimensional packing problems. *SIAM J. Comput.*, 12:508–525, 1983.

[3] R. Breukelaar, E. D. Demaine, S. Hohenberger, H. J. Hoogeboom, W. A. Kosters, and D. Liben-Nowell. Tetris is hard, even to approximate. *Internat. J. Comput. Geom. Appl.*, 14:41–68, 2004.

[4] E. G. Coffman Jr., P. J. Downey, and P. Winkler. Packing rectangles in a strip. *Acta Inform.*, 38:673–693, 2002.

[5] J. Csirik and G. J. Woeginger. Shelf algorithms for online strip packing. *Inform. Proc. Let.*, 63:171–175, 1997.

[6] L. Epstein and R. van Stee. Optimal online bounded space multidimensional packing. In *Proc. 15th ACM-SIAM Sympos. Discrete Algorithms*, pages 214–223, 2004.

[7] L. Epstein and R. van Stee. Online square and cube packing. *Acta Inform.*, 41:595–606, 2005.

[8] S. P. Fekete, J. Schepers, and J. van der Veen. An exact algorithm for higher-dimensional orthogonal packing. *Operations Research*, 55:569–587, 2007.