Online Packing of Equilateral Triangles

Shahin Kamali* Alejandro López-Ortiz* Zahed Rahmati*

Abstract

We investigate the online triangle packing problem in which a sequence of equilateral triangles with different sizes appear in an online, sequential manner. The goal is to place these triangles into a minimum number of squares of unit size. We provide upper and lower bounds for the competitive ratio of online algorithms. In particular, we introduce an algorithm which achieves a competitive ratio of at most 2.474. On the other hand, we show that no online algorithm can have a competitive ratio better than 1.509.

1 Introduction

The classic 1-dimensional bin packing problem asks for assignment of a set of items of different sizes into a minimum number of bins of unit capacity. For convenience, it is often assumed that bins have capacity 1 and items' sizes are in the range $(0, 1]$. In the online version, the items are revealed in a sequential manner, and an algorithm has to place an item into a bin without any information about forthcoming items. Online algorithms are often compared according to their competitive ratio, which is the maximum ratio between the cost of an online algorithm and that of an optimal offline algorithm, denoted by Opt, for serving the same sequence. For bin packing, we are particularly interested in *asymptotic* competitive ratio which only considers sequences for which the cost of Opt is arbitrary large.

Online bin packing has many applications in practice, from server consolidation to cutting stock. In the latter application, the goal is to cut patterns of given sizes from stocks of unit size. Clearly, this application can be extended into two dimensions. In the 2-dimensional bin packing problem, bins are typically squares of unit size while items are similar objects of different sizes. Two studied variations are box packing and square packing in which items are boxes (rectangles) and squares, respectively, of different sizes.

In this paper, we consider the *equilateral triangle packing problem*, which is stated as follows. The problem can be thought as a two-dimensional version of the classic bin packing problem.

Let $\sigma = \langle x_1, x_2, \ldots, x_n \rangle$ be an online sequence of equilateral triangles, where $x_i \in (0, 1.035]$ indicates the side length of the i^{th} equilateral triangle, $1 \leq i \leq n$. The online equilateral triangle packing problem is to place these equilateral triangles into a minimum number of squares of unit size so that no two triangles overlap. Upon receiving an equilateral triangle, an online algorithm makes an irrevocable decision for placing the triangle into a square. For that, the algorithm does not have any information about the (sizes of) forthcoming triangles. Triangles are allowed to be rotated.

The assumption $x_i \in (0, 1.035]$ comes from the fact that no equilateral triangle of length larger than $1/\cos\frac{\pi}{12} \approx 1.035$ fits in a square of unit size. In the rest of the paper by "triangle of size x" we mean "an equilateral triangle whose side length is x". We interchangeably use terms "bin" for "square of unit size", and "items" for incoming "equilateral triangles of different sizes".

Related work. The 1-dimensional bin packing problem has been extensively studied in the past few decades (see [2, 3] for excellent surveys). The most practical online algorithms are First Fit and Best Fit, which are greedy in the sense that they avoid opening a new bin if possible. The competitive ratio of both algorithms is 1.7 [11]. The Harmonic family of algorithms is based on classifying items by their sizes [12]. A member of this family is Harmonic++ of Seiden [13] with a competitive ratio of 1.588 which is the best among online bin packing algorithms.

For online square packing, the first set of results included an upper bound of 2.6875 and a lower bound of 4/3 [4]. The upper bound was later improved a few times ([5, 6, 9]). The best existing upper bound is given by an algorithm of competitive ratio 2.1187 [10]. In [14], a lower bound of 1.62176 was proved for the competitive ratio of any online algorithm. This lower bound was later improved to 1.6406 [6].

The best existing online algorithm for the box packing problem has a competitive ratio of 2.5545 [8] while there is a lower bound of 1.907 for the competitive ratio of any online box packing algorithm [1].

To our knowledge, there is no previous work addressing online packing of triangles. While we currently have no particular application in mind, we believe that triangle packing is of inherent and compelling interest.

*School of Computer Science, University of Waterloo {s3kamali, alopez-o, zrahmati}@uwaterloo.ca

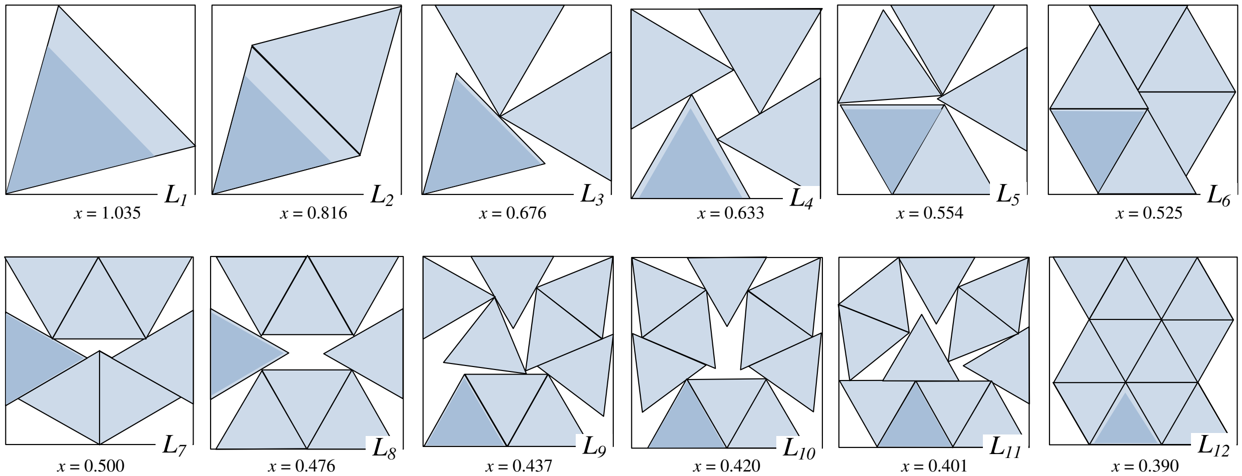


Figure 1: The triangle spots of a classification of large triangles of size x. The dark triangles indicate the lower bound for the size of the triangles.

Our contributions. We provide an online algorithm for packing equilateral triangles, which achieves a competitive ratio of 2.474. Our algorithm is *bounded-space* in the sense that it only keeps a constant number of squares opened at any given time. We prove a lower bound of 1.509 for *any* online equilateral triangle packing algorithm.

Our algorithm, in Section 2, classifies the triangles based on their sizes into different classes, and places triangles of the same class into the same bin. It requires a careful classification, which is more detailed compared to prior packing algorithms of boxes and squares.

For lower bound, in Section 3, we consider a sequence formed by triangles of three different sizes. To achieve a competitive ratio, we provide a linear program which captures the requirements for different subsequences of the original sequence.

2 Algorithm

In this section, we introduce our algorithm for packing of equilateral triangles, which has a competitive ratio of 2.474.

Similar to Harmonic family of algorithms, we classify triangles by their sizes. We refer to a triangle as being *large* if its size is larger than $1/3$, *medium* if its size is in the range $(1/20, 1/3]$ and *small* if it is at most $1/20$. Triangles that belong to any of these three groups are classified further into smaller classes, and members of each class are packed separately from others. In what follows, we describe the classification and packing for each group separately.

2.1 Large Triangles

The large triangles are classified into 12 groups, denoted by L_c, $c = 1, \ldots, 12$, based on their sizes. A large triangle of size x belongs to class L_c if at most c items of size x fit into a bin. Figure 1 shows the (best known) way of placing c ($1 \leq c \leq 12$) equal triangles of maximum size into a square [7]. If $0.816 < x \leq 1.035$, at most one item of size x can fit in a unit square and the triangle belongs to class L_1. If $0.676 < x \leq 0.816$, at most two items of size x fit in the same bin and it belongs to class L_2. Similarly, we can obtain the boundaries $(l_c, r_c]$ for any class L_c; see Table 1.

Triangles of each class are treated separately from other classes. For each class L_c ($1 \leq c \leq 12$) with boundaries $(l_c, r_c]$, the algorithm has at most one *active bin* of type c. When a bin of type L_c is opened, it is declared as the active bin of the class and c triangle spots of size r_c are reserved in that (this is feasible by definition of classes). Upon arrival of an item of type L_c, it is placed in one of the c spots of the active bin. If all these spots are occupied by previous items, a new bin of type L_c is opened. This ensures that all bins of type L_c, except potentially the current active bin, include c items of size greater than l_c.

2.2 Medium Triangles

The size of a medium triangle is in the range $(1/20, 1/3]$. Similar to that of Section 2.1, medium triangles are classified and items of each class are treated separately. Here, the classification is performed in a more regulated manner. We define 34 groups with ranges $(1/20, 1/19.5], (1/19.5, 1/19], \ldots, (1/3.5, 1/3]$ as boundaries of the classes. In particular, we say a triangle of

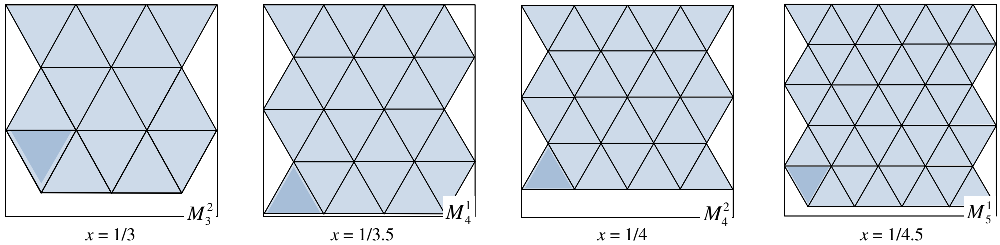


Figure 2: The triangle spots of a classification of medium triangles of size x. The dark triangles indicate the lower bound for the size of the triangles.

size x belongs to class M_k^1 if $x \in (1/k, 1/(k-0.5)]$, and belongs to class M_k^2 if $x \in (1/(k+0.5), 1/k]$, where $3 \leq k \leq 20$; classes M_3^1 and M_{20}^2 are not defined.

The item placement is similar to large triangles. As before, there is at most one active bin for each class L_c with boundaries $(l_c, r_c]$. When an active bin of type M_c is opened, X_c spots of size r_c are reserved in it. Upon arrival of an item of type M_c, it is placed in one of the spots in the active bin (and a new bin is opened if required). The following lemma gives the value of X_c.

Lemma 1 *It is possible to place $2k^2 - 2k$ triangles of class M_k^1 in the same bin. Similarly, $2k^2 - k$ triangles of class M_k^2 can be placed in the same bin ($3 \leq k \leq 20$).*

Proof. Consider a horizontal triangle strip constructed by r connected equilateral triangles of size x, for an integer $r \geq 5$. The vertical width of the triangle strip is the height of an equilateral triangle, *i.e.*, $\sqrt{3}x/2$. Assume the horizontal width of the triangle strip is 1. Therefore, $x = 2/(r+1)$.

For a triangle of size $x = 1/k$, which is associated with class M_k^2, we obtain $r = 2k - 1$. Note that, without overlappings, we can place at most k copies of the corresponding strip in a bin, where $k \geq 3$; see M_3^2 and M_4^2 in Figure 2. Therefore, there exist $2k^2 - k$ triangles of class M_k^2 in the same bin.

In a similar way, for a triangle with $x = 1/(k-0.5)$ of class M_k^1, we obtain $r = 2k - 2$. Since we can place at most k ($k \geq 4$) copies of the corresponding strip in a bin, there exist $2k^2 - 2k$ triangles of class M_k^1 in the same bin; see M_4^1 and M_5^1 in Figure 2. $\square$

2.3 Small Triangles

A small triangle has size at most to 1/20. We maintain at most one active bin for placing small items. When a bin is opened for these items, we reserve four *triangle spots* of size 0.633, *i.e.*, the four triangles of class L_4 in Figure 1. These triangle spots are used as bins for placing small items.

Epstein and van Stee [5] provide an approach to assign small squares into unit squares. The algorithm is to find an appropriate sub-bin for an item, which can be found by partitioning a (sub-)bin into four identical sub-bins (see [5], for more details). A similar approach to that of Epstein and van Stee can be used for assigning equilateral triangles into unit equilateral triangle bins. The analysis of the algorithm for *the case when both the items and bins are equilateral triangles* is the same to *the case when both the items and bins are squares.* Therefore, we can use Claim 3 of [5] for our case. That is,

Claim 1 Given an online sequence of equilateral triangles of sizes no more than $1/M$, for some integer M, one can pack items into equilateral triangle bins of unit size so that the total occupied space in each bin is at least $\frac{M^2-1}{(M+1)^2}$.

Lemma 2 *The occupied area of each bin opened for small items, except possibly a constant number of them, is more than* 0.585.

Proof. Note that the side length ($\leq 1/20$) of a small triangle is within a factor of at most 1/12.66 of the side length (0.633) of the triangle spots. Assuming $M = 12$, by Claim 1, a fraction of at least $(12^2-1)/13^2 = 0.846$ of all triangle spots, except potentially a constant number of them, is occupied by small triangles.

Since the area of each bin covered by four triangle spots is $4 \times 0.173 = 0.692$ (see L_4 of Figure 1), the occupied area of each bin opened for small items is more than $0.692 \times 0.846 = 0.585$. $\square$

2.4 Analysis

For analyzing our algorithm, we use a weighting argument. Corresponding to each triangle of size x, we define a weight $w(x)$ as follows.

Recall that a large triangle of a class L_c ($1 \leq c \leq 12$) is placed in a square which has c spots for items of this class. All bins opened for these triangles, except possibly the last active bin, include c items of this class. We define the weight of items of class c to be $1/c$. This

Class	Side length x	Occupied Area	Weight	Density
L_1	$(0.816, 1.035]$	$> 1(0.288) = 0.288$	1	< 3.472
L_2	$(0.676, 0.816]$	$> 2(0.197) = 0.395$	1/2	< 2.538
L_3	$(0.633, 0.676]$	$> 3(0.173) = 0.520$	1/3	< 1.926
L_4	$(0.554, 0.633]$	$> 4(0.132) = 0.531$	1/4	< 1.893
L_5	$(0.525, 0.554]$	$> 5(0.119) = 0.596$	1/5	< 1.680
L_6	$(0.500, 0.525]$	$> 6(0.108) = 0.649$	1/6	< 1.543
L_7	$(0.476, 0.500]$	$> 7(0.098) = 0.686$	1/7	< 1.457
L_8	$(0.437, 0.476]$	$> 8(0.082) = 0.661$	1/8	< 1.524
L_9	$(0.420, 0.437]$	$> 9(0.076) = 0.682$	1/9	< 1.461
L_{10}	$(0.401, 0.420]$	$> 10(0.069) = 0.696$	1/10	< 1.449
L_{11}	$(0.390, 0.401]$	$> 11(0.065) = 0.724$	1/11	< 1.398
L_{12}	$(1/3, 0.390]$	$> 12(0.048) = 0.577$	1/12	< 1.732
M_3^2	$(1/3.5, 1/3]$	$> 15(0.035) = 0.530$	1/15	< 1.886
M_4^1	$(1/4, 1/3.5]$	$> 24(0.027) = 0.649$	1/24	< 1.540
M_4^2	$(1/4.5, 1/4]$	$> 28(0.021) = 0.598$	1/28	< 1.672
M_5^1	$(1/5, 1/4.5]$	$> 40(0.017) = 0.692$	1/40	< 1.445
...	...	...	...	...
M_{19}^2	$(1/19.5, 1/19]$	$> 703(0.00113) = 0.800$	1/703	< 1.250
M_{20}^1	$(1/20, 1/19.5]$	$> 760(0.00108) = 0.822$	1/760	< 1.216
Small	$(0, 1/20]$	> 0.585	$\frac{\sqrt{3}/4x^2}{0.585}$	< 1.708

Table 1: A summary of classes: range of the side length x, minimum occupied area, weights, and densities.

way, the total weight of items in bins opened for large triangles, except possibly 12 of them, is exactly 1.

For medium triangles of class M_k^1, we define the weight to be $1/(2k^2 - 2k)$, where $4 \leq k \leq 20$. By Lemma 1, the algorithm places $2k^2 - 2k$ triangles of this class in each open bin (except possibly the last active bin), which implies the weight of all bins opened for this class is exactly 1. Similarly, the weight of triangles of class M_k^2 is defined as $1/(2k^2 - k)$, where $3 \leq k \leq 19$. This implies a weight of 1 for all bins (except possibly the last one) opened for this class. To summarize, the total weight of triangles in all bins opened for medium items, except possibly 34 of them (one for each class), is 1.

For a small triangle Δ of size x, we define its weight as $area(\Delta)/0.585$, where $area(\Delta) = \sqrt{3}x^2/4$ is the area of Δ. By Lemma 2, the occupied area of all bins opened for small items (except a constant number of them) is at least 0.585 which implies their total weight is at least $0.585/0585 = 1$.

Table 1 gives a summary of the weights of items in different classes. From the above argument, we conclude the following:

Lemma 3 *The total weight of triangles in each bin opened by the algorithm, except possibly a constant number of then, is at least 1.*

Next, we provide an upper bound for the total weight of items in a bin of the optimal offline algorithm (Opt).

Lemma 4 *The total weight of items in a bin of* Opt *is less than 2.474.*

Proof. Define the *density* of an item as the ratio between its weight and its area. An upper bound for the density of items of each class is reported in Table 1. For large and medium triangles, these values are simply the ratio between the weight and minimum area of items in each bin. For a small triangle Δ, the density is less than $1/0.585 = 1.708$, which is the ratio between the weight $((\sqrt{3}x^2/4)/0.585)$ and the area of Δ.

Note that the density of all triangles except those of types L_1 and L_2 cannot be more than 1.926 (by Table 1). In the following, we consider different cases, and show that no bin B of Opt can have a total weight more than 2.474.

First, assume there are no triangles of type L_1 or L_2 in B. The density of all triangles is less than 1.926, which implies the total weight of items in B is less than 1×1.926 (which is < 2.474).

Assuming there is no item of type L_1, there exist two cases: (I) If there are two items of type L_2, the total weight of these two items is $2 \times 1/2 = 1$. Since the remaining area is at most $1 - (2 \times 0.197) < 0.606$, the total weight of the items that fit in the remaining area would be at most $0.606 \times 1.926 < 1.168$. Therefore, the total weight of all items in B is less than $1+1.168$ (which is < 2.474). (II) If there is only one item of type L_2, it would have a weight of $1/2$ and the remaining area in B is at most $1 - 0.197 = 0.803$. The total weight of other items that fit in the remaining area of B would be at most $0.803 \times 1.926 < 1.547$. Therefore, the total weight of items in the bin will be less than $1/2 + 1.547$ (which is < 2.474).

Note that no two items of type L_1 fit in B. Assume there is one item of type L_1. The remaining area is at most $1 - 0.288 = 0.712$. In such case, we can only place at most one item of type L_2 in B. If there is no item of type L_2, the total weight of items that fit in the remaining area of B is at most $0.712 \times 1.926 < 1.372$. Thus the total weight of items in B would be less than $1 + 1.372$ (which is < 2.474).

The only remaining case is when B contains a triangle Δ_1 of type L_1 and a triangle Δ_2 of type L_2 (see Figure 3). Assume the remaining area is tightly covered by triangles of smaller sizes. It easy to check that when B contains Δ_1 and Δ_2, there is no way to place triangles of smaller sizes of class L_3 in B. The total weights of Δ_1 and Δ_2 is $1 + 1/2 = 1.5$. The remaining area in B has size at most $1 - 0.288 - 0.197 = 0.515$. Since the density of items (except those of types $L_1, \ldots, L_3$) is at most 1.893 (by Table 1), the total weight of the items would be at most $0.515 \times 1.893 < 0.974$. Therefore, the total weight of items in B is less than $1.5+0.974$ (which is equal to 2.474). □

Now, we give the main result of this section.

Theorem 5 *There exists an online algorithm for packing equilateral triangles into squares with a competitive ratio less than 2.474.*

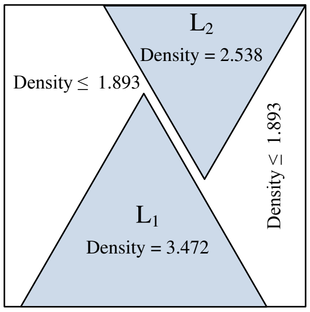


Figure 3: The maximum weight for a bin of OPT is achived when it contains an item of class L_1 and an item of class L_2. Other items that fit in the bin will have denisity smaller than 1.893.

Proof. For an input σ, let $A(\sigma)$ and $\text{OPT}(\sigma)$ denote the cost of the presented algorithm and OPT, respectively. Let $w(\sigma)$ denote the total weight of items of σ. Lemmas 3 and 4 imply that $A(\sigma) \leq w(\sigma) + c$, where c is a constant independent of the length of σ, and $\text{OPT}(\sigma) \geq w(\sigma)/2.474$. From these inequalities we conclude $A(\sigma) \leq 2.474\,\text{OPT}(\sigma) + c$, which proves an upper bound 2.474 for the competitive ratio of the algorithm. □

3 General Lower Bound

In this section, we show that no online algorithm for triangle packing can achieve a competitive ratio better than $80/53 \approx 1.509$. In our proof, we build sequences containing only triangles of sizes $x = 0.554$, $y = 0.676 + \epsilon$, and $z = 0.816 + \epsilon$, where ϵ is a sufficiently small constant. Note that triangles of size x are the largest triangles of the class L_5, and triangles of sizes y and z belong to the classes L_2 and L_1, respectively.

Let $\sigma = \sigma_1\sigma_2\sigma_3$ be a sequence of triangles, where σ_1, σ_2, and σ_3 are n replicas of the triangles of sizes x, y, and z, respectively. Assume n is sufficiently large. We compare the cost of any online algorithm A with that of OPT after serving sequences σ_1, $\sigma_1\sigma_2$, and $\sigma_1\sigma_2\sigma_3$.

Lemma 6 *$\text{OPT}(\sigma_1) = n/5 + 1$, $\text{OPT}(\sigma_1\sigma_2) \leq n/2 + 1$, and $\text{OPT}(\sigma_1\sigma_2\sigma_3) \leq n + 1$.*

Proof. For σ_1, OPT places five triangles in one bin; see Figure 4a. Thus each bin, except potentially the last one, contains five triangles, which gives a total of $n/5 + 1$ bins for placing σ_1. For $\sigma_1\sigma_2$, OPT can place two triangles of size x with two triangles of size y in the same bin (see Figure 4b), so all bins (except potentially the last one), contain four triangles, which gives a total of $2n/4+1$ bins. For $\sigma_1\sigma_2\sigma_3$, OPT can place one triangle of each size x, y, z in the same bin; see Figure 4c. Thus all bins include three triangles, which results a total of at most $3n/3 + 1$ bins for placing $\sigma_1\sigma_2\sigma_3$. □

Now, we obtain the main result of this section.

Theorem 7 *The competitive ratio of any online algorithm A for triangle packing is at least $\frac{80}{53} \approx 1.509$.*

Proof. Let a_{ij} be the number of bins which include i replicas of size x and j replicas of size y, where $1 \leq i \leq 5$ and $0 \leq j \leq 2$.

Denote by $\text{A}(\sigma_1)$ the number of bins opened by the algorithm A for σ_1. Note that if a square contains four or five triangles of size x, then it cannot contain a triangle of size y. Similarly, if a square contains three triangles of size x, then it cannot contain more than one triangle of size y. In summary,

$$\text{A}(\sigma_1) = a_{10} + a_{11} + a_{12} + a_{20} + a_{21} + a_{22} + a_{30} + a_{31} + a_{40} + a_{50}.$$

Assume A has a competitive ratio of at most r. By Lemma 6, for some constant c_1,

$$\text{A}(\sigma_1) \leq r \times n/5 + c_1. \quad (1)$$

By counting the number of triangles of size x, we obtain

$$a_{10} + a_{11} + a_{12} + 2a_{20} + 2a_{21} + 2a_{22} + 3a_{30} + 3a_{31} + 4a_{40} + 5a_{50} = n. \quad (2)$$

Let b_1 be the number of bins that include only one replica of size y, and also let b_2 be the number of bins that include only two replicas of size y and no replica of size x. Consider a packing of A after serving $\sigma_1\sigma_2$, meaning that A has placed n triangles of size x and n triangles of size y. In a similar way to that of $\text{A}(\sigma_1)$, we can obtain $\text{A}(\sigma_1\sigma_2)$, the number of bins opened by A for $\sigma_1\sigma_2$:

$$\text{A}(\sigma_1\sigma_2) = a_{10} + a_{11} + a_{12} + a_{20} + a_{21} + a_{22} + a_{30} + a_{31} + a_{40} + a_{50} + b_1 + b_2,$$

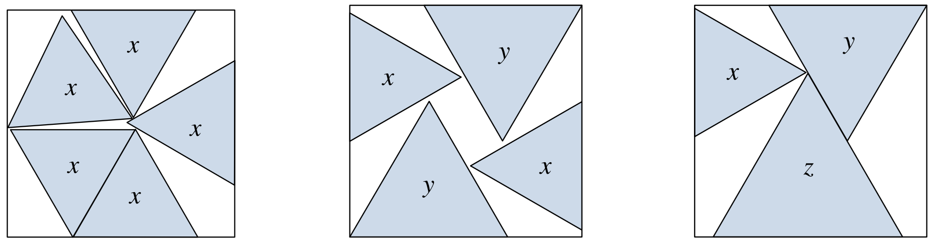
which can be bounded as follows (by Lemma 6, for some constant c_2):

$$\text{A}(\sigma_1\sigma_2) \leq r \times n/2 + c_2. \quad (3)$$

Counting the number of triangles of size y gives

$$a_{11} + 2a_{12} + a_{21} + 2a_{22} + a_{31} + b_1 + 2b_2 = n. \quad (4)$$

Now, consider a packing of A after placing triangles of size z. The algorithm A can place triangles of size z in bins which either `(I)` include at most two triangles of size x and no triangle of size y (*i.e.*, $a_{10} + a_{20}$ bins), or `(II)` include one triangle of size x and one triangle of size y (*i.e.*, a_{11} bins), or `(III)` include only one triangle



(a) A bin of OPT(σ_1) (b) A bin of OPT($\sigma_1\sigma_2$) (c) A bin of OPT($\sigma_1\sigma_2\sigma_3$)

Figure 4: OPT uses different packings for different prefix sequences of $\sigma_1\sigma_2\sigma_3$.

of size y (*i.e.*, b_1 bins). Except these bins, A has to open one bin for each triangle of size y, which implies $n - a_{10} - a_{20} - a_{11} - b_1$ new bins. In summary,

$$\mathrm{A}(\sigma_1\sigma_2\sigma_3) = a_{12} + a_{21} + a_{22} + a_{30} + a_{31} + a_{40} + a_{50} + b_2 + n$$

By Lemma 6, for some constant c_3, the number of bins opened by A for $\sigma_1\sigma_2\sigma_3$ bounds as follows:

$$\mathrm{A}(\sigma_1\sigma_2\sigma_3) \leq r \times n + c_3. \quad (5)$$

Equations 1-5 should hold for a competitive ratio of r. If we scale these equations by $1/n$, the constants c_1, c_2, and c_3 can be ignored. Let $\alpha_{ij} = a_{ij}/n$, $\beta_1 = b_1/n$, and $\beta_2 = b_2/n$. The following linear program summarizes the above discussion.

minimize r subject to

$$\alpha_{10} + \alpha_{11} + \alpha_{12} + \alpha_{20} + \alpha_{21} + \alpha_{22} + \alpha_{30} + \alpha_{31} + \alpha_{40} + \alpha_{50} \leq r/5;$$

$$\alpha_{10} + \alpha_{11} + \alpha_{12} + \alpha_{20} + \alpha_{21} + \alpha_{22} + \alpha_{30} + \alpha_{31} + \alpha_{40} + \alpha_{50} + \beta_1 + \beta_2 \leq r/2;$$

$$\alpha_{12} + \alpha_{21} + \alpha_{22} + \alpha_{30} + \alpha_{31} + \alpha_{40} + \alpha_{50} + \beta_2 + 1 \leq r;$$

$$\alpha_{10} + \alpha_{11} + \alpha_{12} + 2\alpha_{20} + 2\alpha_{21} + 2\alpha_{22} + 3\alpha_{30} + 3\alpha_{31} + 4\alpha_{40} + 5\alpha_{50} = 1;$$

$$\alpha_{11} + 2\alpha_{12} + \alpha_{21} + 2\alpha_{22} + \alpha_{31} + \beta_1 + 2\beta_2 = 1;$$

$$\alpha_{ij}, \beta_1, \beta_2 \geq 0.$$

This linear program obtains the optimal value $r = 80/53$. Therefore, we conclude the actual competitive ratio is at least $80/53$. □

References

[1] D. Blitz, A. van Vliet, and G. J. Woeginger. Lower bounds on the asymptotic worst-case ratio of online bin packing algorithms. Unpublished manuscript, 1996.

[2] E. G. Coffman, M. R. Garey, and D. S. Johnson. Approximation algorithms for bin packing: A survey. In D. Hochbaum, editor, *Approximation algorithms for NP-hard Problems*. PWS Publishing Co., 1997.

[3] E. G. Coffman Jr., J. Csirik, G. Galambos, S. Martello, and D. Vigo. Bin packing approximation algorithms: survey and classification. In P. M. Pardalos, D.-Z. Du, and R. L. Graham, editors, *Handbook of Combinatorial Optimization*, pages 455–531. Springer, 2013.

[4] D. Coppersmith and P. Raghavan. Multidimensional online bin packing: Algorithms and worst case analysis. *Oper. Res. Lett.*, 8:17–20, 1989.

[5] L. Epstein and R. van Stee. Optimal online bounded space multidimensional packing. In *Proc. 15th Symp. on Discrete Algorithms (SODA)*, pages 214–223, 2004.

[6] L. Epstein and R. van Stee. Online square and cube packing. *Acta Inform.*, 41(9):595–606, 2005.

[7] E. Friedman. Triangles in squares. URL: http://www.stetson.edu/ efriedma/triinsqu/.

[8] X. Han, F. Y. L. Chin, H.-F. Ting, G. Zhang, and Y. Zhang. A new upper bound 2.5545 on 2d online bin packing. *ACM Trans. Algorithms*, 7(4):1–18, Sept. 2011.

[9] X. Han, D. Ye, and Y. Zhou. Improved online hypercube packing. In *Proc. 4th International Workshop in Approximation and Online Algorithms (WAOA)*, pages 226–239, 2006.

[10] X. Han, D. Ye, and Y. Zhou. A note on online hypercube packing. *Cent. Eur. J. Oper. Res.*, 18(2):221–239, 2010.

[11] D. S. Johnson, A. J. Demers, J. D. Ullman, M. R. Garey, and R. L. Graham. Worst-case performance bounds for simple one-dimensional packing algorithms. *SIAM J. Comput.*, 3:256–278, 1974.

[12] C.-C. Lee and D.-T. Lee. A simple online bin packing algorithm. *J. ACM*, 32:562–572, 1985.

[13] S. S. Seiden. On the online bin packing problem. *J. ACM*, 49:640–671, 2002.

[14] S. S. Seiden and R. van Stee. New bounds for multidimensional packing. *Algorithmica*, 36(3), 2003.