Online strip packing with modifiable boxes *

Csanád Imreh †

Abstract

In the strip packing problem the goal is to pack a set of rectangles into a vertical strip so as to minimize the total height of the strip needed. We consider a modified version of the strip packing problem. In this version it is allowed to change the form of the rectangles by lengthening them, keeping the area fixed. We introduce online algorithms to solve this modified problem. Moreover a lower bound is presented, as well.

Keywords: strip packing, online algorithm, competitive analysis

1 Introduction

Two dimensional packing problems arise in many situations. There are two basic models to pack two dimensional boxes. One possibility is to pack the boxes into unit cubes so as to minimize the number of boxes needed. The offline problem is investigated in [6]. The first online algorithms are defined for this problem in

*Research supported by the Ministry of Culture and Education of Hungary, Grant TO30074.

† `cimreh@inf.u-szeged.hu` Department of Informatics, University of Szeged, Árpád tér 2, H-6720 Szeged, Hungary.

[10], and these algorithms are improved in [7]. Another possibility is to pack the rectangles into a vertical strip. In this paper we investigate a generalized version of strip packing. In the strip packing problem there is a set of two dimensional boxes, defined by their widths and heights, and the task is to pack them without rotation into a vertical strip of width W by minimizing the total height of the strip. This problem appears in many situations. Usually, scheduling of tasks with shared resource involves two dimensions, the resource and the time. We can consider the widths as the resource and the heights as the time. Our goal is to minimize the total amount of time used. Some applications can be found in computer scheduling problems.

The offline problem is NP-hard (cf. [4]), and some offline approximation algorithms are developed for it (cf. [2], [4]). These algorithms are based on the preordering of the boxes by decreasing heights or decreasing widths. An approximation scheme for the problem is given in [12]. Only a few online algorithms are presented for this problem, the first algorithms, the shelf algorithms are developed in [5]. The presented algorithms are 7.46 and 6.99 competitive. A lower bound 2 for the competitive ratio of any online strip packing algorithm are presented in [3]. An improved shelf algorithm and a generalized definition of shelf algorithms are defined in [8], furthermore, a matching lower bound 1.69 for the asymptotic competitive ratio of the shelf algorithms is also presented. Two generalized versions of this problem are considered. A 3-dimensional version is examined in [13], and the case where rotations are allowed is investigated in [1]. One can find more details about the previous works in the area of strip packing in the survey [9].

Let us suppose that the two dimensions of the boxes are the required resource and time. The given parameters, however, show only one possible configuration: one can also satisfy the task by using less resource. Of course, using less resource means that it takes more time to satisfy the task. We can give a mathematical model for this extended problem by slightly changing the strip packing model. In the modified model the width of the box gives the maximal resource which can be used to satisfy the task and the height of the box gives the time which is necessary using this amount of resource. The fact that we can use less resource for more time means that we can lengthen the box, keeping the area fixed. A similar question with different model is investigated in [11]. The main differences between the two models, that in the work [11] the resource is measured by the number of the used processors and the geometrical structure of the processor's network is also considered.

We examine the online version of the problem where the boxes arrive from a list and we have to lengthen and pack each box without any knowledge on further boxes. We use competitive analysis to measure the performance of the presented algorithms. In the next section we give the basic definitions, and we present some simple observations. We show an easy optimal offline algorithm, and we show that the asymptotic competitive analysis can not be used for this problem even if we suppose that the heights of the boxes are bounded by 1. After the basic definitions and observations, in section 3 we present and analyse two online algorithms for the general problem. After the presentation of the algorithms, in section 4, we show a lower bound which is valid for the competitive ratio of any online algorithm.

2 Premilinaries

The mathematical model of the problem described in the introduction can be given as follows. There is a list of rectangles $p_i = (w_i, h_i)$, $i = 1, \ldots$, where each rectangle is defined by two parameters: by its width and its height. Our goal is to pack the rectangles into a vertical strip so as to minimize the total height of the strip needed. Before packing the rectangles we can lengthen them, we can increase the height and decrease the width, keeping the area fixed. After lengthening the rectangle we have to pack it into the strip, the rectangles cannot be rotated and they cannot overlap. We call this problem strip packing with modifiable boxes. In the following part of this paper we will assume that the width of the strip is 1.

The offline version of this problem, where the whole list L is known in advance is not interesting. As the following statement shows it is easy to calculate the optimal total height (which is denoted by $OPT(L)$). An algorithm which lengthens each rectangle to make them $OPT(L)$ high and packs them to the bottom of the strip can reach the total height of $OPT(L)$.

Proposition 2.1 *The optimal solution of the offline strip packing problem with modifiable boxes has*

$$OPT(L) = \max\{S, H\}$$

total height, where $S = \sum_{i \in L} w_i \cdot h_i$ *and* $H = \max_{i \in L} h_i$.

Proof: First consider the case when $S > H$. Then, it is possible to lengthen each rectangle so as to have S height. Packing the lengthened rectangles into the bottom of the strip we get a packing with S total height. On the other hand, the sum of the areas of the rectangles does not change, thus it is not possible to pack the rectangles so that they have smaller total height than S. The other case is similar, the only difference is that we have to lengthen each rectangle to have the same height as the highest one.

Therefore, the interesting question is what we can say about the online version of the problem. In the online version we obtain the rectangles from the list one by one, and we have to decide for each rectangle how to lengthen it and where to pack it without any knowledge of further rectangles. There are two basic methods to measure the performance of an online algorithm, using the absolute and the asymptotic competitive ratios. *The absolute competitive ratio* of an algorithm A is

$$C_A = \sup_L \{A(L)/OPT(L)\},$$

where $A(L)$ denotes the total height of the strip packing made by A, on the list L of rectangles. Usually, the *asymptotic competitive ratio* allows more freedom for the algorithm if only a few rectangles arrive. This ratio is defined by the formula

$$C_A^{\infty} = \lim_{k\to\infty} \sup_L \{A(L)/OPT(L) \ \ |OPT(L) \geq k.\}$$

Usually in packing problems the asymptotic competitive ratio is used. On the other hand, in the problem considered we can reach very large optimal value for the total height by using only a few very high rectangles. Therefore, the use of asymptotic competitive ratio gives freedom for the case of few rectangles only if we have an upper bound on the heights of the rectangles. Thus, as it often happens in the case of asymptotic competitive ratio, we assume that each rectangle has at most the height of 1. We will show that under this assumption there is an algorithm for every $\varepsilon > 0$, with an asymptotic competitive ratio smaller than $1 + \varepsilon$. Before presenting these algorithms we define a general class of algorithms, i. e. the class of the shelf algorithms. Our definition is a generalized version of the definition from [8].

One basic way of packing into the strip, is to define shelves and pack the rectangles into the shelves. By shelf we mean a rectangular part of the strip with the width of 1. Shelf packing algorithms place each rectangle into one of the shelves. If the algorithm decides which shelf will contain the rectangle, then first the rectangle is lengthened to have the same height as the shelf has. The lengthened rectangle is placed into the shelf as much to the left as it is possible without overlapping the other rectangles placed earlier into the shelf considered. Therefore, after the arrival of a rectangle, the algorithm has to make two decisions. The first decision is whether to create a new shelf or not. If the algorithm creates a new shelf it also has to decide the height of the new shelf. The created shelves always start from the top of the previous shelf. The first shelf is placed to the bottom of the strip. The algorithm also has to choose the shelf to which it puts the rectangle. In what follows, we will say that it is possible to pack a rectangle into a shelf, if there is enough room for the lengthened rectangle in the shelf. It is worth noting that if a rectangle is higher than a shelf we cannot place it into the shelf.

As an example we can define the following shelf algorithm, which depends only on parameter x.

Algorithm xS

Initialization: Create a shelf with height $x > 1$. Let this shelf be the active shelf.

Packing part: When a rectangle arrives, pack it into the active shelf if it is possible (first lengthen to have height x). Otherwise, create a new shelf with height x, let this new shelf be the active shelf and pack the rectangle into it.

Proposition 2.2 *The asymptotic competitive ratio of xS is at most* $1+1/(x-1)$.

Proof: Suppose that a list L of rectangles arrived and the algorithm created k shelves. Examine the sum of the areas of the rectangles packed. Whenever the algorithm creates a new active shelf, it means that a rectangle arrives which is too large to pack it into the previous active shelf. On the other hand, no rectangle has larger area than 1, thus the width of a lengthened rectangle cannot be larger than $1/x$. Therefore, when a new active shelf was created, at least

$1-1/x$-th part of the previous active shelf was loaded. This yields that for each shelf except the last one the sum of the rectangles' areas packed into the shelf is at least $x-1$. This means that the total sum of the areas cannot be less than $(k-1)(x-1)$. Therefore the value of $x(L)/OPT(L)$ is at most

$$\frac{kx}{(k-1)(x-1)}.$$

As k tends to infinity, this bound tends to $1+1/(x-1)$, and the result follows.

Choosing the value $x = 1 + 1/\varepsilon$ we obtain the following corollary:

Corollary 2.1 *There is an algorithm for every $\varepsilon > 0$ with asymptotic competitive ratio at most $1 + \varepsilon$.*

These results yield that using the asymptotic competitive ratio we can easily get near to optimal algorithms. Therefore, for the problem considered, the absolute competitive ratio can be a more appropriate measure. In the following parts of this paper we always mean absolute competitiveness by competitiveness, and we do not assume any upper bounds on the heights of the rectangles.

3 Upper Bounds

The first algorithm which we investigate is an extended version of the NFS_r (next fit shelf r) algorithm which is presented in [5]. The original algorithm has the competitive ratio 7.46 We denote also the extended version by NFS_r. This algorithm depends on a parameter $r > 1$ and it is defined as follows. Before the description of the algorithm remark that this algorithm also uses active shelves similarly to algorithm xS, the main difference that it uses more active shelves, it can have one active shelf with height r^{k+1} for every k.

Algorithm NFS_r

When a rectangle $p_i = (w_i, h_i)$ arrives, choose a value for k that satisfies $r^k < h_i \leq r^{k+1}$. Lengthen the rectangle to the form $(w_i \cdot h_i/r^{k+1}, r^{k+1})$. If there is an active shelf with height r^{k+1} and it is possible to pack the rectangle into it, then pack it there. If there is no active shelf with height r^{k+1}, or it is not possible to pack the rectangle into the active shelf with height r^{k+1}, then create a new shelf with height r^{k+1}, put the rectangle into it, and let this new shelf be the active shelf with height r^{k+1}.

Theorem 3.1 *The competitive ratio of algorithm* NFS_r *is* $2 + \frac{r^2}{r-1}$.

Proof: First we prove that the algorithm is $2 + \frac{r^2}{r-1}$ competitive. Consider an arbitrary list L of rectangles. Let H_A denote the sum of the heights of the shelves which are active at the end of the packing, and let H_D be the sum of the heights of the other shelves. Let h be the height of the highest active shelf, and let H be the height of the highest rectangle. Since the algorithm created a shelf with height h, we have $H > h/r$. As there is at most 1 active shelf for each height

$$H_A \leq h \sum_{i=0}^{\infty} r^{-i} = \frac{hr}{r-1} \leq \frac{Hr^2}{r-1}.$$

Consider the shelves which are not active at the end. Let S be one of these shelves with height r^k. The next shelf S' with height r^k contains one rectangle which (after the lengthening) would not fit into S. Therefore, the total sum of the rectangles' areas packed into S and S' is at least r^k. If for each k we pair the shelves of height r^k in this way, using the active shelf if the number of the shelves of the considered height is odd, we obtain that H_D is not larger than twice the sum of the rectangles' areas. This yields that $H_D \leq 2OPT(L)$. Using this inequality and the fact that $H \leq OPT(L)$ we obtain the desired statement.

Now we prove that the upper bound on the competitive ratio is tight. Let $\varepsilon > 0$ be arbitrary, and let n be an integer with the property $r^{-n} < \varepsilon$. Consider the following list L_ε of rectangles. Let the first $2 \cdot \lfloor r^n \rfloor$ rectangles be $(1/2+\varepsilon/r^n, r^{-n})$, and let the second part of the list be $p_i = (\varepsilon, r^{-i} + \varepsilon)$, $i = n-1, \ldots, 0$. Using these lists a simple calculation shows that the ratio $NFS_r(L_\varepsilon)/OPT(L_\varepsilon)$ tends to $2 + \frac{r^2}{r-1}$ as ε tends to 0.

Choosing the value of r optimal we obtain the following corollary:

Corollary 3.1 *The competitive ratio of algorithm NFS_2 is 6.*

Remark 3.1 The NFS_r algorithm is called next fit shelf algorithm since it can be interpreted as an algorithm which consider the shelves like bins and uses the well known bin packing algorithm to pack them. There are similar algorithms for the original strip packing problem with the difference that they are using more efficient bin packing algorithms to pack the shelves. Many of these algorithms can be modified for this problem. On the other hand we present a more efficient algorithm below so we do not examine further the improved versions of this algorithm.

The following algorithm uses similar idea like the scheduling algorithm from [14]. It can be defined as follows.

Algorithm DS

After the arrival of the first rectangle create a shelf with height h_1, and let this shelf be the active shelf. Later, when a rectangle arrives use the following iteration to pack it.

Step 1. If it is possible to pack the rectangle into the active shelf, pack it (first lengthen it), otherwise go to step 2.

Step 2. Create a new shelf which is twice higher than the active shelf, let the new shelf be the active shelf and go to step 1.

Theorem 3.2 *The competitive ratio of algorithm DS is* 4.

Proof: First we show that the algorithm is 4 competitive. Consider an arbitrary list of rectangles. Denote the list by L. Let H denote the height of the shelf which is active at the end. When the algorithm created this shelf it was not possible to pack the rectangle into the previous active shelf, hence we obtain that the height of the previous active shelf is smaller than $OPT(L)$. Therefore $H \leq 2 \cdot OPT(L)$. On the other hand the heights of the other shelves, are $H/2, H/4, \ldots, H/2^i$. Thus the total height of the packing is smaller than $2H$, and the result follows. To see that the upper bound is tight, consider the following list L_i of rectangles. Let the first i rectangles be $(1/4, 2^j)$, $j = 1, \ldots, i$, and let the last rectangle be $(1/4, 2^i + 1)$. A simple calculation shows that the ratio $DS(L_i)/OPT(L_i)$ tends to 4 as i tends to infinity, which yields the tigthness of the upper bound.

Remark 3.2 Algorithm DS has better competitive ratio than algorithm NFS_r. On the other hand, NFS_r has some advantage, it does not create much higher shelves, than the maximal height of the rectangles. Therefore it can be also useful in some more restricted model.

Remark 3.3 In packing problems one can often have better results if the problem is not completely online, if there is some information about the list of rectangles. If we consider the case where it is assumed that the list of rectangles is preordered by decreasing height, then we can get a simple algorithm which is 2 competitive. One can prove easily that the following algorithm is 2 competitive in this case:

Algorithm FFS1

Let H be the height of the first rectangle. Create first a shelf with height H. Later, when a rectangle arrives check whether exists a shelf to which it is possible to pack it. If there are such shelves put the rectangle into the lowest one (first lengthen it). If there is no such shelf, create a new shelf with height H and put the rectangle into it.

4 Lower Bound

First we introduce the following definitions. For any rectangle p_i we denote the lengthened rectangle by p_i'. Furthermore, two rectangle p_i and p_j are called to be colateral in a packing if there is a horizontal line which intersects both p_i and p_j. Using these definitions we can prove the following lower bounds.

Theorem 4.1 *There is no online algorithm for the strip packing with modifiable boxes that has smaller competitive ratio than C where $C \approx 1.73$ is the solution of the equation $e^{-(c+1/c-2)} = c-1$.*

Proof: Suppose by contrary, that we have an algorithm with smaller competitive ratio than C. Consider such algorithm and denote it by A, further denote its competitive ratio by d. Examine the following list L_n of rectangles. Let the first rectangle be $p_1 = (1,1)$. This rectangle is lengthened to $p_1' = (1/x, x)$, with $1 \leq x \leq d$. Furthermore, $x > 1$ since otherwise a second rectangle $(\varepsilon, 1)$ with a small ε implies that the algorithm is not better than 2 competitive. Let the second rectangle be $p_2 = (1-1/x, x)$, and the $i+2$-th rectangle be $(1/n, x(\frac{n}{n-1})^i)$ for $i = 1, 2 \ldots$. The sequence of rectangles is stopped when the first rectangle is packed which is not colateral with the others. Suppose that the sequence is finished after the $j+2$-th rectangle. Then by Proposition 2.1 we obtain that the optimal offline cost is

$$x(1 + \sum_{i=1}^{j} 1/n \cdot (\frac{n}{n-1})^i) = x(\frac{n}{n-1})^j.$$

Since the first rectangle is lengthened to x and all the others are higher than x the algorithms cost is at least

$$x + x(\frac{n}{n-1})^j.$$

Therefore

$$A(L)/OPT(L) \geq \frac{x + x(\frac{n}{n-1})^j}{x(\frac{n}{n-1})^j} = 1 + (\frac{n-1}{n})^j.$$

On the other hand it follows by Proposition 2.1, that the optimal offline cost after the $i+2$-th rectangle, $i = 0, \ldots, j-1$ is $x(\frac{n}{n-1})^i$, thus the d competitive algorithm can lengthen these rectangles at most d times higher. Therefore the sum of the widths of first $j+1$ lengthened rectangle is at least

$$1/x + 1/d(1 - 1/x + \sum_{i=1}^{j-1} 1/n).$$

Since these rectangles are packed colateral, this total width is not greater than 1. Therefore $j \leq n(1-1/x)(d-1)+1$. Using this bound we obtain that

$$A(L_n)/OPT(L_n) \geq 1 + (\frac{n-1}{n})^{n(1-1/x)(d-1)+1}.$$

The function on the right side of the inequality is decreasing in x on the interval $[1, d]$, therefore it is minimal when $x = d$. Furthermore $(\frac{n-1}{n})^n$ tends to e^{-1} as n goes to infinity, and $d \geq A(L_n)/OPT(L_n)$ for each n. This yields the following inequality

$$d \geq 1 + e^{-(1-1/d)(d-1)} = 1 + e^{-(d+1/d-2)}.$$

On the other hand taking the derivative it can be easily seen, that the function $d - 1 - e^{-(d+1/d-2)}$ is strictly increasing in d in the interval $d \geq 1$ and it is 0 if $d = c$, therefore the above inequality cannot be true for any $d < c$. This means that we obtained a contradiction, which finish the proof of the theorem.

5 Conclusions and Related Question

We have investigated that version of strip packing in which it is allowed to lengthen the rectangles before packing. We presented a 4 competitive algorithm for the general problem, and we showed that 1.73 is a lower bound for the problem. Like in other cases of high dimensional packing problems, there is a relatively large gap between the upper and lower bounds. It would be nice to find the optimal online algorithm for this problem or at least to decrease the gap between the lower and upper bounds.

Some other interesting open questions arise related to the topic. We can consider other models of strip packing with modifiable boxes. The first model is very similar to the investigated one, with the difference that it is not allowed to lengthen the rectangles for arbitrary height. There are two possible ways to define the problem in this case. One possibility is the following model: There is a given constant c, each box has smaller height than c, and no rectangle is allowed to lengthen higher than c. Another possibility is to allow to lengthen every rectangle at most c times higher than their original height, where c is a given constant.

There is a more general question. We can consider a model, where it is allowed to lengthen and press the rectangles to arbitrary size but changing the rectangle causes some penalty. There are two basic ways to define penalties. One possibility is that we define some outside penalties which depend in some way on the transformation we performed on the rectangle. In this case the total cost is the sum of the penalties plus the total height of the strip. Another possibility is to define inside penalties - this means that changing the form of a rectangle may yield a rectangle with larger area.

References

[1] Azar Y., Epstein L: On two dimensional packing. *Journal of Algorithms*, **25** (2), (1997) 290-310. (Also in Proc. SWAT'96 321-332.)

[2] Baker B. S., Brown D. J., Katseff H. P.: A 5/4 algorithm for two dimensional packing. J. Algorithms, **2**, (1981) 348-368

[3] Baker B. S., Brown D. J., Katseff H. P.: Lower bounds for two dimensional packing algorithms, Acta Informatica **8**, (1982) 207-225

[4] Baker B. S., Coffman E. G., Rivest R. L.: Orthogonal packings in two dimensions. SIAM J. Computing, **9**, (1980) 846-855

[5] Baker B. S., Schwartz J. S.: Shelf algorithms for two dimensional packing problems. SIAM J. Computing, **12**, (1983) 508-525

[6] Chung F. R. K., Garey M. R., Johnson D. S.: On packing two dimensional bins. SIAM J. Algebraic Discrete Methods **3**, (1982) 66-76

[7] Csirik J., Frenk J. B. G., Labbe M.:, Two-dimensional rectangle packing: on-line methods and results. Discrete Applied Mathematics, **45**, (1993) 197-204

[8] Csirik J., Woeginger G.; Shelf algorithms for on-line strip packing. Information Processing Letters, **63**, (1997) 171-175

[9] Csirik J., Woeginger G.: Online packing and covering problems. In Online Algorithms: The State of the Art LNCS 1442, Springer Verlag, (1998) 147-178.

[10] Coppersmith D., Raghavan P.: Multidimensional on-line bin packing: algorithms and worst case analysis. Operation Research Letters, **8**, (1989) 17-20

[11] Feldman A., Sgall J., Teng S.H.: Dynamic scheduling on parallel machines. Theoretical Comput. Sci., **130** (1), (1994) 49-72 (Also in Proc. FOCS'91 111-120.)

[12] Kenyon C., Remila E.: Approximate strip packing, to appear in Mathematics of Operation Research.

[13] Li K., Cheng K. H.: Heuristic algorithms for on-line packing in three dimensions. J. Algorithms, **13** (1992) 589-605

[14] Shmoys D. B., Wein J., Williamson D. P., Scheduling parallel machines online. SIAM J. Computing, **24** (1995) 1313-1331