REVERSE-FIT: A 2-OPTIMAL ALGORITHM FOR PACKING RECTANGLES

Ingo Schiermeyer
Lehrstuhl C für Mathematik
Technische Hochschule Aachen
D-52056 Aachen, Germany

Abstract

We describe and analyze a "level-oriented" algorithm, called "Reverse-Fit", for packing rectangles into a unit-width, infinite-height bin so as to minimize the total height of the packing. For L an arbitrary list of rectangles, all assumed to have width no more than 1, let h_{OPT} denote the minimum possible bin height within the rectangles in L can be packed, and let $RF(L)$ denote the height actually used by Reverse-Fit. We will show that $RF(L) \leq 2 \cdot h_{OPT}$ for an arbitrary list L of rectangles.

Key words: level-oriented packing algorithm, bin-packing, two-dimensional packing, k-rectangle packing problem, rectangle packing conjecture

1 Introduction

We consider the following two-dimensional packing problem, first proposed in [1]: Given a collection of rectangles, and a bin with fixed width and unbounded height, pack the rectangles into the bin so that no two rectangles overlap and so that the height to which the bin is filled is as small as possible. We shall assume that the given rectangles are oriented, each having a specified side that must be parallel to the bottom of the bin. We also assume, with no loss of generality, that the bin width has been normalized to 1.

This problem is a natural generalization of the one-dimensional bin-packing problem. Indeed, if all rectangles are required to have the same height, then the two problems coincide. On the other hand, the case in which all rectangles have the same width corresponds to the well-known makespan minimization problem of combinatorial scheduling theory. Both these restricted problems are known to be NP-complete (cf. [4]), from which it follows that the two-dimensional packing problem is also NP-complete.

However, several approximation algorithms have been established which run in polynomial time, among them the so-called *level-oriented packing algorithms*, which can be described as follows: Assume that the rectangles of an arbitrary list L are ordered by decreasing (actually, nonincreasing) height, then the rectangles are packed in the order given by L so as to form a sequence of *levels*. All rectangles will be placed with their bottoms resting on one of these levels. The first level is simply the bottom of the bin. Each subsequent level is defined by a horizontal line drawn through the top of the first (and hence maximum height) rectangle placed on the previous level. Notice how this corresponds with one-dimensional bin-packing; the horizontal slice determined by two adjacent levels can be regarded as a bin (lying on its side) whose width is determined by the maximum height rectangle placed in that bin. The following two level-oriented algorithms are suggested by analogous algorithms studied for one-dimensional bin-packing:

1. **Next-Fit (NF)**. With this algorithm, rectangles are packed left-justified on a level until there is insufficient space at the right to accomodate the next rectangle. At that point, the next level is defined, packing on the current level is discontinued, and packing proceeds on the new level.

2. **First-Fit (FF)**. At any point in the packing sequence, the next rectangle to be packed is placed left-justified on the first (i.e., lowest) level on which it will fit. If none of the current levels will accomodate this rectangle, a new level is started as in the NF algorithm.

Now for L an arbitrary list of rectangles, all assumed to have width no more than 1, let h_{OPT} denote the minimum possible height within the rectangles in L can be packed. Let $NF(L)$ and $FF(L)$ denote the height actually used by these algorithms when applied to L. The following upper bounds for NF and FF have been shown in [3].

Theorem 1.1 *For any list L ordered by nonincreasing height,*

$$NF(L) \leq 3 \cdot h_{OPT},$$

$$FF(L) \leq 2,7 \cdot h_{OPT}.$$

Note that it is easy to verify that $FF(L) \leq NF(L)$ for all lists L.

In [5] Sleator presents an algorithm that packs in $2,5 \cdot h_{OPT}$ which has been the best *absolute performance bound* known so far. This algorithm can be considered as a modification of First-Fit. In the first step all rectangles of width greater than $1/2$ are stacked on top of one another in the bottom of the bin. If the height to which they reach is called H_0, then all subsequent packing will occur above H_0. Actually, this first step has turned out to be of major importance in order to prove the performance bound.

In this paper we present a packing algorithm, called **Reverse-Fit (RF)**, which will be described in section 2. The basic idea of this algorithm is that, instead of packing every block in a left-to-right manner, it alternately packs blocks from left-to-right and then from right-to-left. In section 3 we will show that Reverse-Fit packs in $2 \cdot h_{OPT}$ followed by a discussion and two rectangle packing conjectures in section 4.

2 The Packing Algorithm

In order to describe the algorithm, the following additional notations will be useful. We shall associate with the bin all points (x, y) of the plane having coordinates $0 \leq x \leq 1$ and $y \geq 0$. Thus the left and the right corner of the bin will be associated with $(0,0)$ and $(1,0)$, respectively. Let the list L be given as $r_1, r_2, \ldots, r_n$. For a particular rectangle r_i of width w_i and height h_i, its coordinates of the lower left, lower right, upper right and upper left corner will be given by $(a_i, c_i), (b_i, c_i), (b_i, d_i)$ and (a_i, d_i). Analogously, for a particular rectangle r we shall speak of $w(r), h(r), a(r), b(r), c(r)$ and $d(r)$.

Reverse-Fit

1. Stack all the rectangles of width greater than 1/2 on top of one another in the bottom of the bin. Call the height to which they reach H_0, and the total area of these rectangles A_0. All subsequent packing will occur above H_0.

2. Sort the remaining rectangles in order of decreasing height. The rectangles will be packed into the bin in this order. Let h_{max} be the height of the tallest of these rectangles.

3. Now pack rectangles from left to right with their bottoms along the line of height H_0 with the first rectangle adjacent to the left wall of the bin, and each subsequent rectangle adjacent to the one just packed. Continue until there is no more room or there are no rectangles left to pack (Figure 1).

4. Let d_1 be the height of the tallest of the remaining rectangles. Now pack rectangles from right to left with their tops along the line of height $H_0 + h_{max} + d_1$ with the first rectangle adjacent to the right wall of the bin. Let this be the second *reverse-level* and now pack each subsequent rectangle according to First-Fit. Hence, each subsequent rectangle is packed left-justified on the first level, if it fits there, or right-justified on the second reverse-level, respectively. Continue until the total width of the rectangles packed on the second reverse-level is at least 1/2 or there are no rectangles left. In the latter case the algorithm stops. Next all rectangles from the second reverse-level are dropped (each of them by the same amount, say e_1), until (at least) one of them touches some rectangle below. Set $H_1 := h_{max} + d_1 - e_1$. Let r_p and r_q be the right most pair of touching rectangles with r_p placed on the first level and r_q placed on the second level, respectively. Let $m_1 := max(a_p, a_q), m_2 := min(b_p, b_q)$. Then the 'touching-line' $T(r_p, r_q)$ of r_p and r_q is given by $T(r_p, r_q) = \{(x, d_p) \mid m_1 \le x \le m_2\}$.

   If $m_2 \ge 1/2$, then set $v_p := max(1/2, m_1)$ and goto step 5.

   If $m_2 < 1/2$, then note that there are at least two rectangles placed on the second reverse-level because of step 1. Let r_k and r_j be the last and the last but one rectangle packed on the second reverse-level, respectively. Now all rectangles from the second reverse-level, except for r_k, are dropped again (each of them by the same amount, say H_2) until (at least) one of them touches some rectangle below (Figure 6). Again we determine the right most pair of touching rectangles r_p and r_q. Since $H_2 > 0$ we have $r_q \neq r_k$ and thus $m_2 \geq m_1 = max(a_p, a_q) \geq a_j > 1/2$. Now, if $H_2 = h_k$, then the third level is defined by a horizontal line drawn through the top of r_j. Next r_k is moved leftwards along this line as far as possible. Hence, it either touches the left wall or a rectangle placed on the first level. Continue with step 5.

If $H_2 > h_k$ (Figure 7), then (again) the third level is defined by a horizontal line drawn through the top of r_j. This time r_k is packed left-justified on the third level, touching a rectangle placed on the first level, since $H_2 > h_k$.

5. We now continue packing rectangles by a modified First-Fit algorithm. Each subsequent level (starting with the third one) is defined by a horizontal line drawn through the top of the first (and hence maximum height) rectangle packed on the previous level. Note that on such a new level, the first rectangle, which is packed left-justified, need not touch the left wall but a rectangle from the first level.

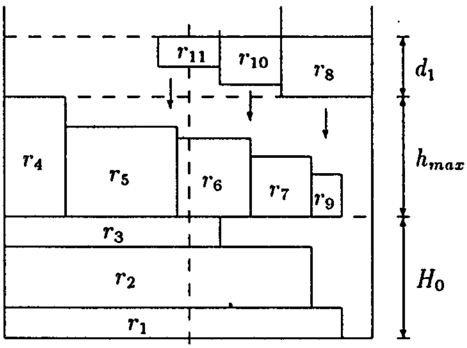


Figure 1

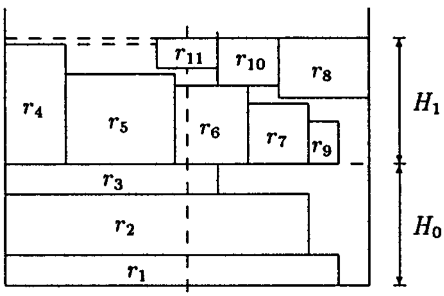


Figure 2

3 The Performance Bound

Some additional notation will be useful for showing the desired performance bound of Reverse-Fit. Packings will be regarded as a sequence of blocks $B_0, B_1, \ldots, B_N$, where the index increases from the bottom to the top of the packing. Let R_i denote the total area of the rectangles in block B_i, and let H_i denote the height of block $B_i, i = 0$ or $3 \le i \le N$. The (partial) area of R_i between two vertical lines drawn through $(x, 0)$ and $(y, 0)$ for a pair x, y such that $0 \le x \le y \le 1$ will be denoted by $R_i(x, y)$. Let s_i be the first rectangle placed in block B_i, and let r_{ij} be the last rectangle which has been packed in block B_i at the time r_j is packed.

Theorem 3.1 *Let $RF(L)$ be the height of a packing given by Reverse-Fit. Then*

$$RF(L) \le 2 \cdot h_{OPT}$$

for any list L ordered by nonincreasing height.

Proof: Since over half of the area of the bin below height H_0 is filled with rectangles we have

$$A_0 \ge \frac{1}{2} H_0.$$

Next observe that $H_0 \le h_{OPT}$ and thus $H_0 + h_{max} \le 2 \cdot h_{OPT}$. Hence we may assume that $N \ge 2$. For A_1 we have

$$\begin{aligned}
A_1 &= v_p \cdot h_p + (1 - v_p) \cdot h_q \\
&= \left(\frac{1}{2} + (v_p - \frac{1}{2})\right)\left(\frac{h_p + h_q}{2} + \frac{h_p - h_q}{2}\right) + \left(\frac{1}{2} - (v_p - \frac{1}{2})\right)\left(\frac{h_p + h_q}{2} - \frac{h_p - h_q}{2}\right) \\
&= 2\left(\frac{1}{2} \cdot \frac{h_p + h_q}{2} + (v_p - \frac{1}{2}) \cdot \frac{h_p - h_q}{2}\right) \\
&\ge 2 \cdot \frac{1}{2} \cdot \frac{H_1}{2} = \frac{1}{2} H_1,
\end{aligned}$$

since $h_p \ge h_q, v_p \ge \frac{1}{2}$ and $H_1 = h_p + h_q$.

For A_2 we obtain

$$A_2 \geq (\frac{1}{2} - a_k)H_2 + (a_k - 0)(h_q + H_2 - h_k) \geq \frac{1}{2}H_2,$$

since $h_k \leq h_q$ (Figure 6). Note that $H_2 = 0$ if $v_p = \frac{1}{2}$.

Let r_m be the first rectangle placed on the third level (i.e., $r_m = s_3$). Now suppose that $v_p = \frac{1}{2}$. If $b_m \leq \frac{1}{2}$ (Figure 3 and 6), then

$$> \frac{1}{2}h_m = \frac{1}{2}H_3,$$

since $h_m \leq h_{1m}, h_m \leq h_k, w_m \leq b_m$ and r_m does not fit on the first level.

If $b_m > \frac{1}{2}$ (Figure 4 and 7), then

$$\begin{aligned} A_3 &\geq (a_m - 0)h_k + (\frac{1}{2} - a_m)h_m + (b_{1m} - \frac{1}{2})h_{1m} \\ &\geq b_{1m}h_m > \frac{1}{2}h_m = \frac{1}{2}H_3, \end{aligned}$$

since $b_{1m} > \frac{1}{2}$.

Next suppose that $v_p > \frac{1}{2}$. If $0 \leq H_2 \leq h_k$ (Figure 6), then (as above)

$$\begin{aligned} A_3 &\geq (a_m - 0)h_k + (min(\frac{1}{2}, b_m) - a_m)h_m + R_2(\frac{1}{2}, v_p) + (b_{1m} - v_p)h_{1m} \\ &\geq min(\frac{1}{2}, b_m)h_m + (b_{1m} - \frac{1}{2})h_m \\ &\geq (w_m + b_{1m} - \frac{1}{2})h_m > \frac{1}{2}h_m = \frac{1}{2}H_3. \end{aligned}$$

If $H_2 > h_k$ (Figure 7), then

$$\begin{aligned} A_3 &\geq (a_k - 0)h_j + (\frac{1}{2} - a_k)h_k + (b_{1k} - v_p)h_{1k} \\ &\geq (\frac{1}{2} + b_{1k} - v_p)h_k > \frac{1}{2}h_k = \frac{1}{2}H_3, \end{aligned}$$

since $b_{1k} - v_p > 0$.

If $a(s_4) = 0$, then

$$\begin{aligned} A_3 &\geq (a_m - 0)h_k + w_m h_m + (b_{1m} - \frac{1}{2})h_{1m} \\ &\geq (b_m + b_{1m} - \frac{1}{2})h_m \end{aligned}$$

$$\begin{aligned} A_4 &\geq w_m h_m + (b_{3m} - \frac{1}{2})h_{3m} \\ &\geq (b_{3m} - \frac{1}{2} + w_m)h_m > \frac{1}{2}h_m = \frac{1}{2}H_4, \end{aligned}$$

since w_m does not fit on the third level.

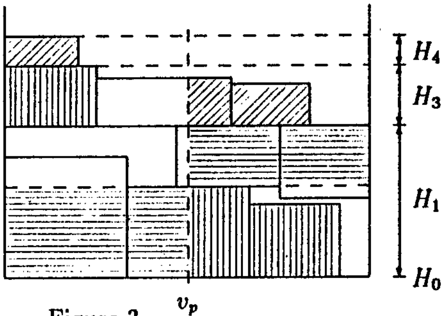


Figure 3

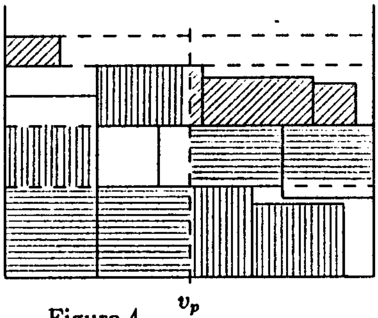


Figure 4

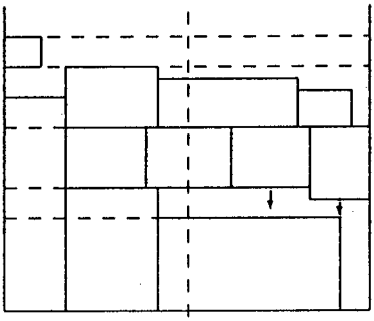


Figure 5

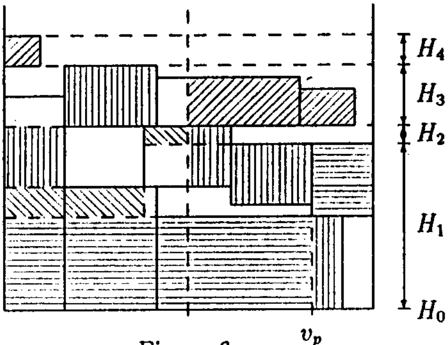


Figure 6

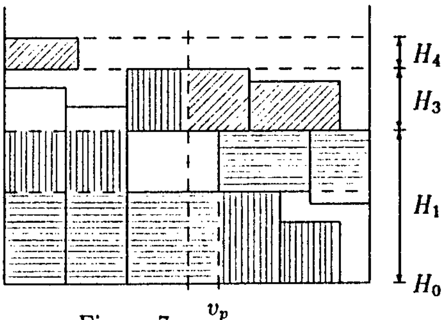
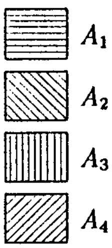


Figure 7



If $a(s_4) > 0$, then (as above)

$$\begin{aligned} A_4 &\geq (a_m - 0)h_k + (min(\frac{1}{2}, b_m) - a_m)h_m + (b_{3m} - \frac{1}{2})h_{3m} \\ &\geq (min(\frac{1}{2}, b_m) + b_{3m} - \frac{1}{2})h_m \\ &\geq (b_{3m} - \frac{1}{2} + w_m)h_m > \frac{1}{2}h_m = \frac{1}{2}H_4. \end{aligned}$$

In the same way, for each pair of blocks B_i and B_{i+1} with $2 \leq i \leq N-1$, there is an area A_{i+1} such that

$$A_{i+1} \geq \frac{1}{2}H_{i+1}.$$

Altogether we obtain

$$\sum_{i=0}^{N} R_i \geq \sum_{i=0}^{N} A_i \geq \frac{1}{2}\sum_{i=0}^{N} H_i,$$

where $RF(L) = H_0 + H_1 + \ldots + H_N$. This completes the proof. □

4 Discussion and Rectangle Packing Conjectures

First note that by Reverse-Fit at most once rectangles are packed from right-to-left. Hence, with a repeat of this approach, one might expect that the performance bound could be reduced below 2. However, the proof of the performance bound in the previous section already indicates, that it might not be that easy. Especially, step 1 of Reverse-Fit could not be applied any longer in its present form.

Next we consider the following restricted rectangle packing problem which has been suggested by Brucker [2]. We shall call it the **k-rectangle packing problem.**

k-RECTANGLE PACKING PROBLEM

INSTANCE: A set of positive integers w, k, n, w_i, h_i, H such that $n = \sum_{i=1}^{k} n_i, w_i \leq w$ for $1 \leq i \leq k$, a bin of width w and infinite height, k pairwise different rectangles r_i of width w_i and height h_i ($(w_i, h_i) \neq (w_j, h_j)$ for $1 \leq i < j \leq k$), each of them occuring n_i times.

QUESTION: Is there a packing of the n rectangles into the bin of height H or less?

For $k = 1$, Next-Fit, First-Fit and Reverse-Fit always find an optimal packing of the n rectangles which can be easily verified. Hence, the **1-rectangle packing problem** can be determined in polynomial time. However, for $k \geq 2$, the complexity of the k-rectangle packing problem seems not to be known so far, as has been mentioned by Brucker and Grötschel [2].

Analyzing several instances for the k-rectangle packing problem we have recognized that the minimum height for a particular instance often depends on the number theoretic properties of the set of integers w, w_i, h_i and n_i. Especially, the difference $h_{OPT} - \frac{1}{w}\sum_{i=1}^{k} n_i w_i h_i$ of h_{OPT} and the (theoretical) lower bound for the height of a packing can be arbitrarily large. On the other hand, because of the (often) 'complicated structures' of optimal packings for particular instances, it seems not to be that easy to reduce another NP-complete or NP-hard problem to the k-rectangle packing problem in order to show that it is NP-complete. All this gives reason for the following two conjectures.

Weak rectangle packing conjecture. The 2-rectangle packing problem can be solved in polynomial time depending on n.

We even believe that this is true for any fixed $k \geq 2$.

Strong rectangle packing conjecture. For any fixed $k \geq 2$ the k-rectangle packing problem can be solved in polynomial time depending on n and k.

References

[1] B. S. Baker, E. G. Coffman, Jr. and R. L. Rivest, Orthogonal packings in two dimensions, SIAM J. Comput., Vol. 9, No. 4 (1980) 846 - 855.

[2] P. Brucker and M. Grötschel, personal communication, January 1994.

[3] E. G. Coffman, Jr., M. R. Garey, D. S. Johnson and R. E. Tarjan, Performance bounds for level-oriented two-dimensional packing algorithms, SIAM J. Comput., Vol. 9, No. 4 (1980) 808 - 826.

[4] M. R. Garey and D. S. Johnson, *Computers and Intractability, A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, San Francisco, 1979.

[5] D. D. K. D. B. Sleator, A 2.5 times optimal algorithm for packing in two dimensions, Inform. Process. Lett. 10, no. 1 (1980) 37 - 40.