A Note on Packing Rectangles in Groups

E.G. Coffman, Jr.
Bell Laboratories/Lucent Technologies†
Peter J. Downey
University of Arizona
Peter Winkler
Bell Laboratories/Lucent Technologies†

TR 97-07

ABSTRACT

Rectangles from a list of length n are packed into a unit width strip. The rectangles have dimensions independently chosen from a uniform distribution on $[0, 1]$, and the packing objective is to minimize the expected height of the packing of n items. The packing algorithms of interest must operate on-line, as well as adhere to a constraint reminiscent of the *Tetris*™ game: rectangles arrive from the top and must be moved at arrival without overlap within the strip to reach their final placement. This paper assumes no rotation of rectangles. The Group Packing algorithm GP_3 packs rectangles densely in groups of 3 at a time, starting a new level at the highest point reached by any rectangle in the group. The GP_3 algorithm achieves an asymptotic expected height of $(0.38541...)n$. This is slightly worse than the bound $(0.38134...)n$ achieved by *Next Fit Level (NFL)* packing.

May 15, 1997

Department of Computer Science
The University of Arizona
Tucson, Arizona 85721

---

†Bell Laboratories, Lucent Technologies, 600 Mountain Ave, Murray Hill NJ 07974-2070

A Note on Packing Rectangles in Groups

1. Introduction

In the problem called two-dimensional strip packing, rectangles of width and height bounded by 1 are packed into a semi-infinite strip of width 1. Packings must be such that no rectangle overlaps another's area, and the sides of the rectangles are parallel to the strip sides. The objective is to minimize the height of the packing in the strip, for a given sequence of *n* rectangles. For a full discussion of this problem see [CS90, CS93].

This note analyzes an algorithm for strip-packing under the assumption that rectangle heights and widths are sampled independently from a uniform distribution on $[0,1]$. In addition, this algorithm meets the *on-line Tetris*™ *constraint:*

(1) The packing algorithm is *on-line*: the algorithm must inspect rectangles (also referred to as *items*) one at a time and make a placement decision for each rectangle at the time it is inspected. The placement is fixed at that time and cannot be altered later.

(2) The packing algorithm obeys *Tetris*-like constraint [AE96]: rectangles descend from the top of the strip, and must be moved within the strip horizontally and vertically to reach their final placement, and may not overlap the area of any other rectangle during this movement. In this paper, rotation of rectangles during placement is not allowed.

Constraints similar to the Tetris packing restriction arise in warehousing and cargo container loading applications.

More information on on-line algorithms and Tetris-constrained algorithms can be found in [CDW97].

Requiring an algorithm to adhere to a Tetris constraint eliminates from consideration many of the strip-packing algorithms studied heretofore (such as shelf algorithms [BS83, CW97] and the First Fit Level algorithm [Hof80, CL91].) Packings possible by sliding the rectangles ''outside'' the strip boundaries are thus ruled out. See Figure 1 for an example.

The objective of the strip packing problem is to pack a list L_n of *n* rectangles in such a way as to minimize the height of the packing. After all rectangles in L_n have been placed, the *height* of the packing is the maximum distance from the strip bottom to the top of any packed rectangle.

™Tetris is a registered trademark of The Tetris Company.

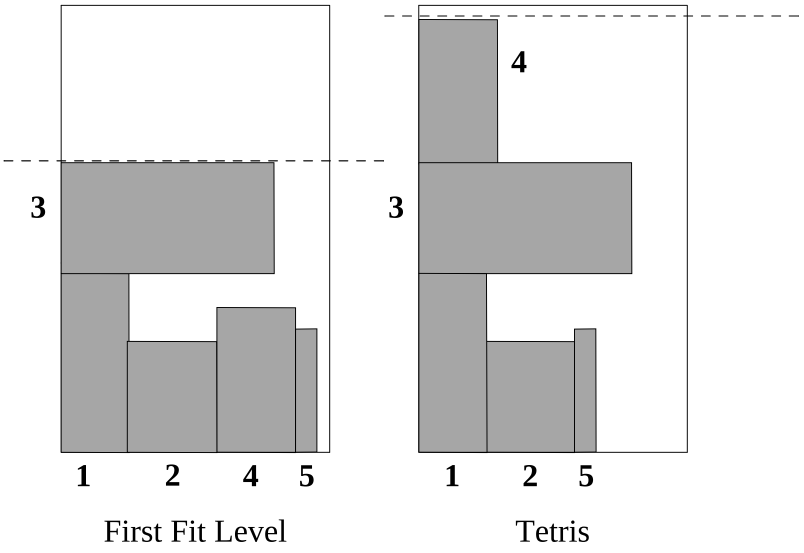

Figure 1: If the items depicted arrive on-line in the order $1, 2, ...$, they can be packed on-line using a *First-Fit Level* heuristic [Hof80, CL91] as shown at left. Under the Tetris constraint, the best possible on-line packing is shown at right.

The packing problem can be studied under deterministic or probabilistic assumptions about the rectangle widths and heights. Here we analyze one packing algorithm under the *uniform model*, in which in which the n given rectangle widths and heights (W_i, H_i), $i = 1, 2, \ldots, n$ are $2n$ independent draws from a uniform distribution on $[0, 1]$. We analyze the exact expected height of the packing, and the expectation asymptotically for large n.

The algorithm analyzed in this note is the **group packing** algorithm with groups of size 3, denoted GP_3 for short. In this algorithm, the input list L_n is divided into groups of exactly size 3 (and possibly a final group of size 1 or 2). Packing begins along a level line drawn across the strip (initially the line is the bottom of the strip). For each group of size 3, items are packed in the order they arrive using heuristic rules that attempt to keep the height of the group small. The packing rules are not known to be optimal even for a group of size 3, but are reasonable. The rules are illustrated by Figure 2, and detailed in the next section. When all three items have been packed, a new level line is drawn at the top of the highest item in the group, and the group packing process is repeated for the next group of 3.

The last group in L_n may be of size 1 or 2 (or zero). If so, this group is packed optimally. The height of the packing is the height of a level line drawn through the top of the highest item in this final group.

For the uniform model, the the algorithm called *Next Fit Level (NFL)* that meets the on-line Tetris constraint has been analyzed [Hof80] that has expected height $(0.381338...)n$ for large n. The GP_3 algorithm analyzed here has expected height asymptotic to $(0.38541...)n$. This is not as good as *NFL*, but the simplicity of the algorithm suggests that group packing for larger groups may result in improved bounds.†

In section 2, we describe the GP_3 algorithm in greater detail. Section 3 contains the analysis of its expected height.

2. The Group Packing Algorithm GP_3

Suppose we are packing rectangles, starting with a level line drawn across the strip as a floor. Then the probability that exactly 1, 2 or 3 items will fit side-by-side across the level is easily seen to be $1/2 + 1/3 + 1/8 = 23/24$. This motivates trying to pack exactly three pieces at a time, in a reasonably efficient way, trying to keep the resulting height as small as possible in expectation.

GP_3 packs each group of 3 items starting on a new level line across the strip. Initially the level line is the strip bottom. When 3 items have been packed, a new level line is drawn across the top of that item having the highest top in the group of 3. The algorithm then repeats this process starting at the new level line.

Let the three items in the next group of three have widths W_1, W_2 and W_3. Figure 2 depicts the decision tree for packing the three items. The resulting packing patterns are shown at the leaves of the tree.

We can easily see that this decision tree constitutes an on-line packing procedure, by phrasing the packing rules as in Figure 3. This procedure is followed for every group of three items. If there are two items left at the end of the list of items, they are packed either side-by-side on a new level (if they will fit), or one on top of the other. If there is only one item left at the end of the list of items, it is packed alone on a new level.

Figure 2 shows all the resulting packings of 3 items that can occur, along with the conditions on widths that lead to these packings. In the subtree for $W_1 + W_2 \leq 1$, packings are depicted for the case where item 1 is the shorter of the first two items ($H_1 < H_2$). This loses no generality, because if $H_1 \geq H_2$, the resulting packings are simply mirror images of those shown in leaves *b* and *c* of Figure 2.

If we assume without loss of generality that $H_1 < H_2$, then note that the packing decisions depend only on the widths W_1, W_2, W_3 of the three items in a group. Consequently, the heights of the items H_1, H_2, H_3 completely determine the resulting height of each packed group. Expressions for the heights of the packings for leaves a, b, c—given in the Lemma below—will be the same whether $H_1 < H_2$ or its symmetric case holds.

---

†The Compression Algorithm, described in [CDW97] improves on *NFL*, but is more difficult to describe and analyze.

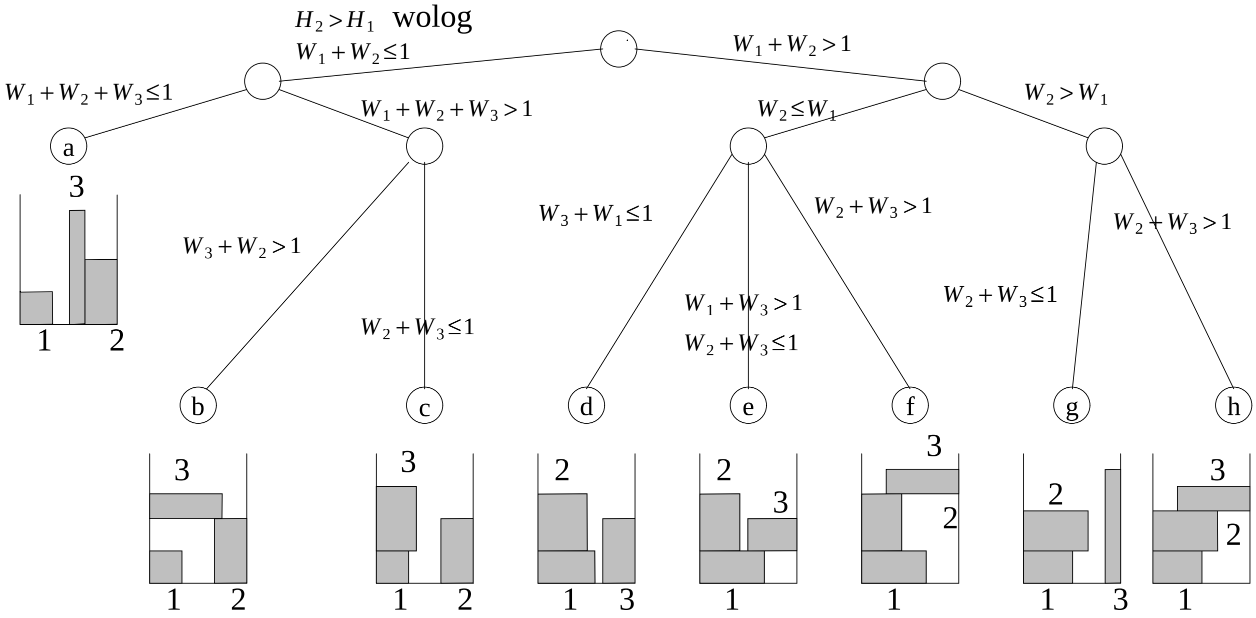

Figure 2: The Group Packing Algorithm for $k = 3$, showing possible packings (at leaves a through h) and the events that lead to those packings. The figure is also a decision tree showing how each group is packed, depending on widths W_1, W_2, W_3. Without loss of generality, we have shown packings assuming $H_2 > H_1$.

3. Expected Height Analysis

For each of the events depicted at the leaves in Figure 2, we calculate its probability of occurrence. We then compute the expected height of each packing of 3 items that results. Finally, at the end of this section, we compute the total expectation of the height of a group of three items, and then use this result to compute the asymptotic expected height of the packing for the Group Packing Algorithm with groups of size 3.

3.1. Event Probabilities

All variates W have density 1 on $[0, 1]$. Denote by A_a, A_b, ..., A_h the events shown in leaves a, b, ..., h of Figure 2. A leaf event is obtained by conjoining the events on the path from the root to that leaf.

For events A_a, A_b and A_c, we have depicted the resulting packings assuming $H_2 > H_1$. We may do this without loss of generality, as the outcomes for $H_2 \le H_1$ are simply mirror images of the packings shown. Of course, the probability of the leaf events shown is not affected by the heights of the items. However, when we later

Algorithm GP_3: Packing a group of 3:

```
Pack item 1 left justified on the level line;
if $W_1 + W_2 \le 1$
    {   Pack item 2 right justified on the level line;
        if $W_1 + W_2 + W_3 \le 1$
            {   Pack item 3 justified against item 2   }
        else
            {   Pack item 3 above the shorter of items 1 and 2,
                justified against the nearest strip wall,
                and slid down as far as it will go   }
    }
else
    {   Pack item 2 left justified on top of item 1;
        Pack item 3 right justified, and slid down as far as it will go }
```

Figure 3: Algorithm for packing a group of size 3 under GP_3

compute the expected heights of these packing patterns, the item heights—conditioned by the assumption $H_2 > H_1$—will come into play.

$$\mathbf{P}[A_a] = \mathbf{P}[W_1 + W_2 \le 1,\ W_1 + W_2 + W_3 \le 1] \tag{1a}$$
$$= \int_0^1 dw_1 \int_0^{1-w_1} dw_2 \int_0^{1-w_1-w_2} dw_3$$
$$= \frac{1}{6}\ .$$

$$\mathbf{P}[A_b] = \mathbf{P}[W_1 + W_2 \le 1,\ W_1 + W_2 + W_3 > 1,\ W_2 + W_3 > 1] \tag{1b}$$
$$= \int_0^1 dw_1 \int_0^{1-w_1} dw_2 \int_{1-w_2}^{1} dw_3$$
$$= \frac{1}{6}\ .$$

$$\mathbf{P}[A_c] = \mathbf{P}[W_1 + W_2 \le 1,\ W_1 + W_2 + W_3 > 1,\ W_2 + W_3 \le 1] \tag{1c}$$
$$= \int_0^1 dw_1 \int_0^{1-w_1} dw_2 \int_{1-w_1-w_2}^{1-w_2} dw_3$$
$$= \frac{1}{6}\ .$$

In events A_d, A_e and A_f, the conjunction of $W_1 > 1 - W_2$ and $W_1 \geq W_2$ implies that $W_1 \geq 1/2$ in order for the inner integrals to be non-zero.

$$\mathbf{P}[A_d] = \mathbf{P}[W_1 + W_2 > 1,\ W_2 \leq W_1,\ W_1 + W_3 \leq 1] \tag{1d}$$

$$= \int_{\frac{1}{2}}^{1} dw_1 \int_{1-w_1}^{w_1} dw_2 \int_{0}^{1-w_1} dw_3$$

$$= \frac{1}{24}\ .$$

$$\mathbf{P}[A_e] = \mathbf{P}[W_1 + W_2 > 1,\ W_2 \leq W_1,\ W_3 > 1 - W_1,\ W_3 \leq 1 - W_2] \tag{1e}$$

$$= \int_{\frac{1}{2}}^{1} dw_1 \int_{1-w_1}^{w_1} dw_2 \int_{1-w_1}^{1-w_2} dw_3$$

$$= \frac{1}{12}\ .$$

$$\mathbf{P}[A_f] = \mathbf{P}[W_1 + W_2 > 1,\ W_2 \leq W_1,\ W_3 > 1 - W_2] \tag{1f}$$

$$= \int_{\frac{1}{2}}^{1} dw_1 \int_{1-w_1}^{w_1} dw_2 \int_{1-w_2}^{1} dw_3$$

$$= \frac{1}{8}\ .$$

In events A_g and A_h, the conjunction of $W_2 > 1 - W_1$ and $W_2 > W_1$ implies that $W_2 > \max(W_1, 1 - W_1)$ in order for the innermost integral to be non-zero.

$$\mathbf{P}[A_g] = \mathbf{P}[W_1 + W_2 > 1,\ W_2 > W_1,\ W_3 \leq 1 - W_2] \tag{1g}$$

$$= \int_{0}^{1} dw_1 \int_{\max(w_1,\, 1-w_1)}^{1} dw_2 \int_{0}^{1-w_2} dw_3$$

$$= \frac{1}{24}\ .$$

$$\mathbf{P}[A_h] = \mathbf{P}[W_1 + W_2 > 1,\ W_2 > W_1,\ W_3 > 1 - W_2] \tag{1h}$$

$$= \int_{0}^{1} dw_1 \int_{\max(w_1,\, 1-w_1)}^{1} dw_2 \int_{1-w_2}^{1} dw_3$$

$$= \frac{5}{24}\ .$$

3.2. Expected Heights

The expected heights of the packings in events A_a through A_h depend entirely on the independent $U[0,1]$ heights H_i, $i = 1, 2, 3$.

Lemma

Denote by H_a the height of the packing for event A_a, by H_b the height of the packing for event A_b, and so forth. Let R_{12} denote the range of a uniform 2-sample:

$$R_{12} = \max(H_1, H_2) - \min(H_1, H_2) \ .$$

Then

$$H_a = \max(H_1, H_2, H_3) \ . \tag{2a}$$

$$H_b = \max(H_1, H_2) + H_3 \ . \tag{2b}$$

$$\begin{aligned} H_c &= \max(H_2, H_1 + H_3) \mid H_2 > H_1 \\ &= \max(R_{12}, H_3) + \min(H_1, H_2) \ . \end{aligned} \tag{2c}$$

$$H_d = \max(H_1 + H_2, H_3) \ . \tag{2d}$$

$$H_e = H_1 + \max(H_2, H_3) \ . \tag{2e}$$

$$H_f = H_1 + H_2 + H_3 \ . \tag{2f}$$

$$H_g = \max(H_1 + H_2, H_3) \ . \tag{2g}$$

$$H_h = H_1 + H_2 + H_3 \ . \tag{2h}$$

Proof: These expressions for the heights are for the most part self-evident from the packings depicted in Figure 2. Only (2c), (2d) and (2g) require discussion.

In the case of (2c), it is evident from the figure that $H_c = \max(H_2, H_1 + H_3)$. However, we have conditioned on the event $H_2 > H_1$, so that by subtracting the minimum $H_1 = \min(H_1, H_2)$ we get

$$H_c = \max(H_2 - H_1, H_3) + H_1 = \max(R_{12}, H_3) + \min(H_1, H_2) \ .$$

In cases (2d) and (2g), there is no conditioning on H_1 and H_2, and the packing height is the unconditioned maximum of a sum of two independent $U[0,1]$ variates, and yet a third independent $U[0,1]$ variate. □

Corollary

The following expected packing heights follow from the assumption that H_1, H_2, H_3 are independent $U[0,1]$ variates:

$$\mathbf{E}H_a = \frac{3}{4} \ . \tag{3a}$$

$$\mathbf{E}H_b = \frac{7}{6} \ . \tag{3b}$$

$$\mathbf{E}H_c = \frac{11}{12} \ . \tag{3c}$$

$$\mathbf{E}H_d = \frac{25}{24} . \tag{3d}$$

$$\mathbf{E}H_e = \frac{7}{6} . \tag{3e}$$

$$\mathbf{E}H_f = \frac{3}{2} . \tag{3f}$$

$$\mathbf{E}H_g = \frac{25}{24} . \tag{3g}$$

$$\mathbf{E}H_h = \frac{3}{2} . \tag{3h}$$

Proof: The expectation of the maximum of m independent $U[0, 1]$ variates is $m/(m + 1)$ [Dav81], and the expectation of their sum is $m/2$. These facts along with (2a), (2b), (2e), (2f) and (2h) establish (3a), (3b), (3e), (3f) and (3h).

For (3c) we need to compute $\mathbf{E}[\max(R_{12}, H_3) + \min(H_1, H_2)]$, where all the H_i are independent and uniform. Now the distribution of R_{12} is $2t - t^2$ [Dav81], and so

$$\mathbf{P}[\max(R_{12}, H_3) \le t] = 2t^2 - t^3 ,$$

from which by integration we obtain $\mathbf{E}\max(R_{12}, H_3) = 7/12$. The expectation of a minimum of m uniforms is $1/(m + 1)$ [Dav81]. Adding $\mathbf{E}\min(H_1, H_2) = 1/3$ to the foregoing yields (3c).

For (3d) and (3g) we need to compute $\mathbf{E}\max(H_1 + H_2, H_3)$ for three independent uniforms. The sum of two uniforms has d.f.

$$\mathbf{P}[H_1 + H_2 \le t] = \begin{cases} t^2/2 & 0 \le t \le 1 \\ t^2/2 - (t - 1)^2 & 1 < t \le 2 \end{cases} .$$

The d.f. of the desired maximum is the product of this with t on $[0, 1]$ and 1 on $[1, 2]$:

$$\mathbf{P}[\max(H_1 + H_2, H_3) \le t] = \begin{cases} t^3/2 & 0 \le t \le 1 \\ t^2/2 - (t - 1)^2 & 1 < t \le 2 \end{cases} .$$

Using the above distribution to compute the expectation results in

$$\begin{aligned} \mathbf{E}\max(H_1 + H_2, H_3) &= \int_0^1 t \cdot \frac{3t^2}{2}\, dt + \int_1^2 t \cdot (2 - t)\, dt \\ &= \frac{3}{8} + \frac{2}{3} = \frac{25}{24} . \end{aligned}$$

□

3.3. Expected Height of a Group

Weighting the expected heights of individual packings in Figure 2 with their respective probabilities of occurrence yields, via (1a-h) and (3a-h),

$$\begin{aligned}\mathbf{E}[\text{height of a packed group of } 3] &= \sum_{x \in \{a \ldots h\}} \mathbf{P}[A_x] \cdot \mathbf{E} H_x \\ &= \frac{37}{32} \ .\end{aligned} \tag{4}$$

3.4. Expected Height of the GP_3 Algorithm

Knowing the expected height of each group of 3, we can now state the exact expected performance of GP_3 in packing a list L_n of n items:

Theorem

$$\mathbf{E}[GP_3(L_n)] = \frac{37}{32} \left\lfloor \frac{n}{3} \right\rfloor + \begin{cases} 0 & n \bmod 3 = 0 \\ 1/2 & n \bmod 3 = 1 \\ 5/6 & n \bmod 3 = 2 \end{cases} \tag{5}$$

$$\sim \frac{37}{96} n \sim (0.38541\ 66666\ldots) n \qquad (n \to \infty) \ .$$

Proof: Each of the groups of 3 from the n items contributes an expected height of 37/32 to the total. There are $\lfloor n/3 \rfloor$ such full groups. If one item is left over, its group packing height is expected to be 1/2. If two items are left over, the expected group packing height for two items is

$$\tfrac{1}{2} \mathbf{E}[\max(H_1, H_2)] + \tfrac{1}{2} \mathbf{E}[H_1 + H_2] = \frac{5}{6} \ .$$

□

Comparison of the Theorem with results for the *NFL* algorithm [Hof80] shows that GP_3 is slightly worse than *NFL* by an amount $(0.00407\ldots)n$.

The integrals in this section were verified by the *Mathematica* System [Wol91].

4. References

[AE96] Azar, Y. and L. Epstein, On two dimensional bin-packing, *Proc. SWAT '96: 5th Scandinavian Workshop on Algorithm Theory*, Reykjavik, Iceland, July 1996, 321-332.

[BS83] Baker, B.S. and J.S. Schwarz, Shelf algorithms for two-dimensional packing problems, *SIAM J. on Computing* **12**, (1983) 508-525.

[CDW97] Coffman, E.G. Jr., P.J. Downey, and P. Winkler, Packing rectangles in a strip, Technical Report TR97-04, Department of Computer Science, University of Arizona (April 1997).

[CL91] Coffman, E.G. Jr. and G.S. Lueker, *Probabilistic Analysis of Packing and Partitioning Algorithms*, Wiley, New York, 1991.

[CS90] Coffman, E.G. Jr. and P.W. Shor, Average-case analysis of cutting and packing in two dimensions, *European Journal of Operational Research* **44**, (1990) 134-144.

[CS93] Coffman, E.G., Jr. and P.W. Shor, Packing in two dimensions: asymptotic average-case analysis of algorithms, *Algorithmica* **9** (1993) 253-277.

[CW97] Csirik, J. and G.J. Woeginger, Shelf algorithms for on-line strip packing, Department of Computer Science, University of Szeged, H-6720 Szeged, Hungary.

[Dav81] David, H.A., *Order Statistics, 2nd Ed.*, Wiley, New York, 1981.

[Hof80] Hofri, M., Two dimensional packing: expected performance of simple level algorithms, *Information and Control* **45** (1980) 1-17.

[Wol91] Wolfram, S. *Mathematica: A System for Doing Mathematics by Computer, 2nd Edition*, Addison-Wesley, Reading, MA, 1991.