# Data Set Generation for Rectangular Placement Problems

Christine L. Valenzuela (Mumford)
School of Computer Science & Informatics
Cardiff University
5 The Parade
Cardiff CF24 3AA
United Kingdom
C.L.Mumford@cs.cardiff.ac.uk

Pearl Y. Wang*
Department of Computer Science MS 4A5
George Mason University
Fairfax, VA 22030-4444
U.S.A
pwang@cs.gmu.edu

Published in 2001

## Abstract

This paper describes a recursive process for generating data sets of rigid rectangles that can be placed into rectangular regions with zero waste. The generation procedure can be modified to guarantee that the aspect and area ratios of the rectangles in the generated data sets satisfy user-specified parameters. This recursive process can thus be employed to create a variety of data sets that can be used to evaluate the efficiency and scalability of rectangular cutting and packing algorithms.

**Keywords: cutting, packing, rectangular placement**

## 1 Introduction

Many rectangular cutting and packing algorithms have appeared in the literature in the last three decades. These solution procedures have often been evaluated by using a variety of test data sets. For cutting problems, several popular benchmark data sets [1, 4] have been utilized for solving constrained and unconstrained problems. The number of rectangles appearing in these data sets typically ranges from the tens to hundreds of pieces, and the optimal solution for each data set may or may not be known. Similarly, many rectangular bin packing heuristics have relied on data sets for demonstrating their effectiveness. For example, bin packing data sets have been used in [2, 5, 6, 7]; these contain at most hundreds of rectangles and several share the property that an optimal solution for the data set is known. Other bin packing data sets (e.g. [3]) contain rectangles whose heights and widths have been randomly generated and whose optimal solution has waste that is unknown but can be bounded below by summing the areas of the rectangles to be packed.

The rectangles appearing in the published data sets for both cutting and packing display a range of properties. Some data sets contain rectangles that appear to be "nearly" square. Others contain rectangles that are mostly tall and thin or short and fat, while other data sets contain both types of rectangles. Additionally, many contain rectangles that are either very large or very small in area, while others contain rectangles that all have similar area.

This variety of rectangle sizes and areas enables researchers to determine if their proposed algorithms are biased towards any particular types of data. However, the small sizes of these data sets do not enable a determination to be made of whether cutting and packing algorithms will scale to large problem sizes, and often, the quality of the solution can only be approximated since the optimal solution is not known.

Due to this sparsity of large benchmark data sets for the problem of cutting or packing rectangles into rectangular regions, we have developed a recursive routine for generating data sets of rigid rectangles which can be packed into a zero-waste rectangular region. More importantly, this procedure permits the user to specify a *range of variation* in the dimensions and areas of the generated rectangles.

Section 2 describes the basic approach used by the data set generation algorithm which, simply stated, recursively cuts a user specified input rectangle into smaller subrectangles. This method can be modified so that the height-to-width ratios of the resulting rectangles is controlled as proven in section 3. Further, section 4 illustrates how the imposition of restrictions governing the choice of which subrectangles can be recursively sliced will yield data sets where

*This work was partially supported by the NASA Goddard Space Flight Center (NAG–5–4868)

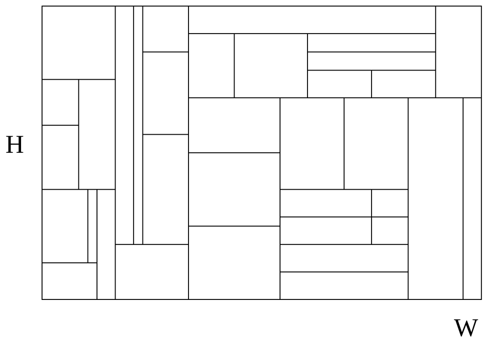

Figure 1: A zero–waste packing

the maximum-to-minimum area of resulting rectangles can be limited. Next, we show how these procedures can be combined to produce data sets containing rectangles whose height-to-width and area ratios are controlled as described in section 5. Section 6 characterizes some sample data sets that were generated by these procedures. A description of how these algorithms and data sets can be used is presented in section 7. Finally, areas of ongoing research are described in section 8.

# 2 Generating unconstrained rectangles

A basic procedure can be formulated which generates data sets containing $n$ rectangles with no restrictions being placed on the relative height $h_i$ and width $w_i$ of each rectangle. Recursive slicing of a large (stock) rectangle is performed by applying vertical or horizontal cuts with equal probability. At each step, slicing positions are chosen with uniform probability. By reversing the slicing process, the rectangles can be reassembled into a zero-waste packing. In this manner, rectangles such as those shown in Figure 1 can easily be generated.

At the start of the data set generation process, the user is asked to input the dimensions of the stock rectangle and also the precise number of rectangular pieces desired. The input parameters consist of $n$, the number of desired rectangles, and $H$ and $W$, the height and width of the stock rectangle being cut. Algorithm 1 describes the basic technique for generating a data set.

It is clear that this $\Theta(n)$ process generates a set of rectangles with real-valued dimensions that can be reassembled into a stock rectangle of size $H \times W$ with zero waste.

**Algorithm 1** Generating Unconstrained Rectangles

Input: $n$, $H$, and $W$
**while** $n$ rectangles have not yet been generated **do**
  choose a rectangle $R$ randomly
  choose a vertical or horizontal slicing direction randomly
  choose a random position to cut $R$ in the chosen direction
  perform the cut, generating two subrectangles
  replace $R$ in the list with the two new subrectangles
**end while**

# 3 Generating rectangles satisfying the aspect ratio constraint

The *aspect ratio* of a rectangle with height $h_i$ and width $w_i$ is defined to be the ratio $\frac{h_i}{w_i}$. Algorithm 1 can be modified to produce a set of rectangles whose aspect ratios fall within a user-specified range of $[1/\rho, \rho]$ where $\rho \geq 2$. To ensure this, additional constraints must be satisfied during the generation process. First, the input stock rectangle must satisfy an initial condition based on the value of the $\rho$ parameter. Next, positions at which successive random cutting of the initial stock piece and the intermediate subrectangles must be restricted. To prove that the final set of generated rectangles have the desired aspect ratio, the following theorems are noted. For convenience, a rectangle $R$ of height $H$ and width $W$ is said to "have" aspect ratio $\rho$ if $1/\rho \leq H/W \leq \rho$.

## 3.1 Mathematical conditions for aspect ratio cutting

**Lemma 1** *Let $R$ be a rectangle having height $H$ and width $W$ that is sliced vertically into two subrectangles $R_1$ and $R_2$. If $W > 2\rho H$ for a given $\rho$, then $R_1$ and $R_2$ cannot both have aspect ratio $\rho$.*

**Proof.** Suppose $R$ is sliced at position $x$ to form two subrectangles $R_1$ and $R_2$ as shown in Figure 2. Let $W > 2\rho H$ and assume that $R_1$ has aspect ratio $\rho$.

It follows that $x < W/2$ and $W - x > W/2$ because

$$\begin{aligned}
&\implies H/x \geq 1/\rho \\
&\implies x \leq \rho H < \rho W/(2\rho) = W/2 \\
&\implies x < W/2 \\
&\implies W - x > W/2
\end{aligned}$$

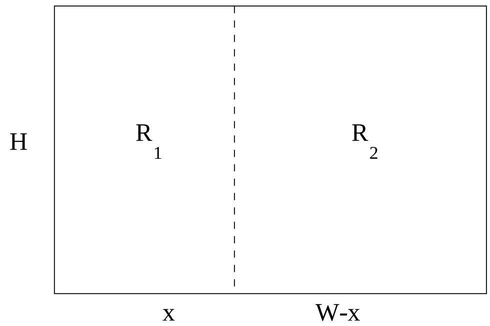


Figure 2: Vertical slicing of an $H \times W$ rectangle

from which

$$\implies H/(W-x) < H/(W/2)$$
$$\implies H/(W-x) < 2H/W < 2H/(2\rho H) = 1/\rho$$
$$\implies H/(W-x) < 1/\rho$$

which implies that $R_2$ does not have aspect ratio $\rho$. A similar argument can be used to establish that $R_1$ will not have aspect ratio $\rho$ if $W > 2\rho H$ and $R_2$ has aspect ratio $\rho$. $\square$

**Lemma 2** *Let $R$ be a rectangle having height $H$ and width $W$ that is sliced vertically into two subrectangles $R_1$ and $R_2$. If $W < 2H/\rho$ for a given $\rho$, then $R_1$ and $R_2$ cannot both have aspect ratio $\rho$.*

**Proof.** Let $W < 2H/\rho$ and assume that $R_1$ has aspect ratio $\rho$. Then it can be shown that $x > W/2$ and $W - x < W/2$:

$$\implies H/x \leq \rho$$
$$\implies H \leq \rho x$$
$$\implies \rho W/2 < H \leq \rho x$$
$$\implies W/2 < x$$
$$\implies W - x < W/2$$

so that

$$\implies H/(W-x) \geq 2H/W > 2H/(2H/\rho) = \rho$$
$$\implies H/(W-x) > \rho$$

which implies that $R_2$ does not have aspect ratio $\rho$. Similarly, if $W < 2H/\rho$ and $R_2$ has aspect ratio $\rho$, then it can be shown that $R_1$ does not have aspect ratio $\rho$. $\square$

**Lemma 3** *Let $R$ be a rectangle having height H and width W that is sliced horizontally into two subrectangles $R_1$ and $R_2$. If $H > 2\rho W$ or if $H < 2W/\rho$ for a given $\rho$ value, then $R_1$ and $R_2$ cannot both have aspect ratio $\rho$.*

**Proof.** This lemma can be established by first observing that $1/\rho \leq H/W \leq \rho$ implies that $1/\rho \leq W/H \leq \rho$ and then applying the same arguments used in the proofs of Lemmas 1 and 2 with $H$ and $W$ interchanged. $\square$

The proof techniques used for the above lemmas provide clues for obtaining some conditions which guarantee that a rectangle can be vertically (or horizontally) sliced into two subrectangles, each having an aspect ratio of $\rho$. For example, note that Lemmas 1 and 2 have indicated that if there is to be a chance that a rectangle can be cut vertically into two acceptable subrectangles, then it should probably have height $H$ and width $W$ satisfying $\frac{2H}{\rho} \leq W \leq 2\rho H$. We now show that this must be the case.

**Theorem 1** *A rectangle with height $H$ and width $W$ can be sliced vertically into two subrectangles with aspect ratio $\rho$ if $W$ satisfies $\frac{2H}{\rho} \leq W \leq 2\rho H$.*

**Proof.** If the height $H$ and width $W$ of the rectangle to be cut satisfy any of the conditions in Lemmas 1, 2 or 3, then the two resulting subrectangles cannot both have aspect ratio $\rho$. Thus, suppose that the width $W$ of a rectangle satisfies $\frac{2H}{\rho} \leq W \leq 2\rho H$. (Note that this is equivalent to $\frac{H}{\rho} \leq W/2 \leq \rho H$.)

Now observe that any vertical cut at position $x$ definitely dictates that $H/\rho \leq x \leq \rho H$: if not, then $x < H/\rho$ implies that $H/x > \rho$ and $x > \rho H$ implies that $H/x < 1/\rho$. These conditions cause the left subrectangle, $R_1$, which is formed by the cut at $x$, to lack the desired aspect ratio property.

However, it is not clear that cutting the rectangle at position $x$ where $H/\rho \leq x \leq \rho H$ will guarantee that $R_1$ has aspect ratio $\rho$. To verify this, first assume that $H/\rho \leq x \leq W/2$. If this is true, then $H/x \geq 2H/W \geq 1/\rho$ and $H/x \leq H/(H/\rho) = \rho$, and so $R_1$ will have aspect ratio $\rho$.

For a cut position $x$, $W/2 < x \leq \rho H$, $H/x < H/(W/2) \leq \rho$ so $H/x < \rho$. Also $H/x \geq H/\rho H = 1/\rho$. Thus the resulting $R_1$ will again have aspect ratio $\rho$.

Cutting the rectangle at position $x$ where $H/\rho \leq x \leq \rho H$ does not, however, necessarily guarantee that the subrectangle created to the right of the $x$ cut will also have aspect ratio $\rho$. In order for this to be true, the $x$ cut will have to be limited to the range $W - \rho H \leq x \leq W - H/\rho$ as seen in Figure 3. Arguments similar to those used above can be applied to prove this restriction, since cutting at these positions is symmetric with respect to the midpoint $W/2$ of the rectangle width.

It follows that in order to cut the rectangle so that both subrectangles have aspect ratio $\rho$, the vertical cut must be

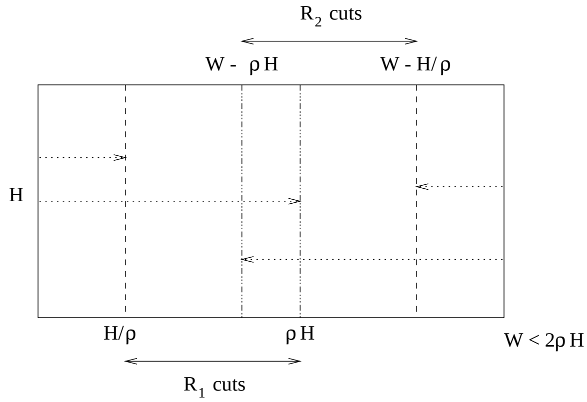


Figure 3: Legal vertical slicing positions of an $H \times W$ rectangle

restricted to positions

$$\max(H/\rho, W - H\rho) \leq x \leq \min(H\rho, W - H/\rho).$$

Note that if $W = 2H/\rho$ or $W = 2\rho H$, it can easily be shown that only one vertical cut position of $x = W/2$ can be used which will produce two subrectangles with aspect ratios of $\rho$. □

**Corollary 1** *If an $H \times W$ rectangle with $\frac{2H}{\rho} \leq W \leq 2\rho H$ is sliced vertically at any position $x$ where*

$$\max(H/\rho, W - H\rho) \leq x \leq \min(H\rho, W - H/\rho)$$

*then the resulting two subrectangles have aspect ratio $\rho$.*

Equivalent results can also be derived for horizontally cutting a rectangle of height $H$ and width $W$ so that two resulting subrectangles with aspect ratio $\rho$ are obtained. These results can be obtained by interchanging $H$ and $W$ in the above proof, hence Theorem 2 and Corollary 2.

**Theorem 2** *A rectangle with height $H$ and width $W$ can be sliced horizontally into two subrectangles with aspect ratio $\rho$ if $H$ satisfies $\frac{2W}{\rho} \leq H \leq 2\rho W$.*

**Corollary 2** *If an $H \times W$ rectangle with $\frac{2W}{\rho} \leq H \leq 2\rho W$ is sliced horizontally at any position $y$ where*

$$\max(W/\rho, H - W\rho) \leq y \leq \min(W\rho, H - W/\rho)$$

*then the resulting two subrectangles have aspect ratio $\rho$.*

We can combine these two theorems to obtain a condition that will guarantee that both horizontal and vertical slicing will yield two subrectangles with aspect ratio $\rho$.

**Theorem 3** *If the height $H$ and width $W$ of a rectangle satisfies the relation $2H/\rho \leq W \leq H\rho/2$ where $\rho \geq 2$, then it can be cut horizontally and vertically to yield two subrectangles with aspect ratio $\rho$.*

**Proof.** Using the inequalities $2H/\rho \leq W \leq H\rho/2$ and $1/2 < 2$, it follows that

$$H/(2\rho) \leq 2H/\rho \leq W \leq H\rho/2 \leq 2\rho H.$$

which meets the conditions of both Theorems 1 and 2. □

Theorem 3 can be used to ensure that the initial stock rectangle to be recursively cut by our revised algorithm will generate two resulting rectangles with aspect ratio $\rho$. What remains to be proved is that the recursive cutting of these resulting rectangles will continue to generate subrectangles with aspect ratio $\rho$. We now show that this is the case if $\rho$ is chosen so that $\rho \geq 2$.

**Theorem 4** *Suppose a rectangle $R$ has aspect ratio $\rho$ where $\rho \geq 2$. If $R$ cannot be sliced vertically (horizontally) to produce subrectangles with aspect ratio $\rho$, then it can be sliced horizontally (vertically) to yield subrectangles with aspect ratio $\rho$.*

**Proof.** Suppose $R$ has aspect ratio $\rho \geq 2$ and $R$ cannot be sliced vertically to yield two subrectangles with aspect ratio $\rho$. By Theorem 1, its dimensions satisfy either $W < 2H/\rho$ or $W > 2H\rho$. The only possibility is $W < 2H/\rho$ since the second inequality would contradict the assumption that $R$ has aspect $\rho$: $W > 2H\rho$ implies that $H/W < 1/(2\rho) < 1/\rho$.

Now assuming that $W < 2H/\rho$ and $\rho \geq 2$, we have $\rho^2 \geq 4$ so that $W \geq 4W/\rho^2$. Combining this with the first inequality, then $2H/\rho > 4W/\rho^2$ or $H > 2W/\rho$. We saw earlier in Theorem 2 that if $2W/\rho \leq H \leq 2\rho W$ then $R$ can be cut horizontally to give two subrectangles with aspect ratio $\rho$. So it remains to show that $H \leq 2\rho W$. If $H > 2\rho W$, then $H/W > 2\rho > \rho$ contradicting the initial assumption that $R$ has aspect ratio $\rho$, thus $R$ satisfies the conditions of Theorem 2 and can be cut horizontally to produce subrectangles with aspect ratio $\rho$.

Analogously, it can be shown if that if $R$ has aspect ratio $\rho$ and it cannot be sliced horizontally, then it can be sliced vertically. If $R$ cannot be cut horizontally, then $H < 2W/\rho$ or $H > 2\rho W$. The case where $H > 2\rho W$ conflicts with the assumption that $H/W \leq \rho$.

Now assuming that $H < 2W/\rho$ and $\rho \geq 2$, we have $\rho^2 \geq 4$ so that $H \geq 4H/\rho^2$. Combining this with the first inequality, then $2W/\rho > 4H/\rho^2$ or $W > 2H/\rho$. We saw earlier in Theorem 1 that if $2H/\rho \leq W \leq 2\rho H$ then $R$ can be cut horizontally to give two subrectangles with aspect ratio $\rho$. So it remains to show that $W \leq 2\rho H$. If $W > 2\rho H$, then $H/W < 1/(2\rho) < 1/\rho$ contradicting the initial assumption that $R$ has aspect ratio $\rho$, thus $R$ satisfies the conditions of Theorem 1 and can be cut vertically to produce subrectangles with aspect ratio $\rho$.

Thus, any rectangle $R$ with aspect ratio $\rho \geq 2$ can be sliced vertically or horizontally, (or both ways) if the conditions of Theorem 4 are met. $\square$

## 3.2 Algorithm for generating aspect ratio data sets

It is now possible to design an algorithm that slices a stock rectangle into a set of subrectangles having the same aspect ratio $\rho \geq 2$. First, start with a rectangle $R$ having aspect ratio $\rho$ whose height and width satisfy the conditions of Theorem 3. $R$ can be sliced either vertically or horizontally to yield two subrectangles with the same aspect ratio $\rho$: after selecting a direction randomly, choose a random slicing position dictated by the appropriate Corollary (1 or 2). The resulting two subrectangles will have aspect ratio $\rho$.

Next, randomly select a subrectangle and determine a slicing direction: (horizontal, vertical, or either if possible). Having chosen the direction to slice, select an appropriate random position and cut the subrectangle. Replace the rectangle in the list with these two subrectangles. This process is then re-applied to the list of subrectangles: since each subrectangle has aspect ratio $\rho$, its subrectangles will also have aspect ratio $\rho$ (Theorem 4) and these slicing steps can be repeated. The process terminates when the list contains the desired number of subrectangles.

The second $\Theta(n)$ data generation procedure can be written as expressed in Algorithm 2.

**Algorithm 2** Controlling the Aspect Ratio

```
Input the parameters n, ρ ≥ 2, H, and then W where
2H/ρ ≤ W ≤ ρH/2
while n rectangles not yet generated do
   choose a rectangle R at random {Theorem 4 guarantees that it can be cut in at least one direction}
   randomly choose a vertical or horizontal slicing direction, if possible;
   otherwise select the vertical or horizontal direction as appropriate {Theorem 2 or 3}
   randomly choose a cutting position within the legal range of slicing positions {Corollary 1 or 2}
   perform the cut on R, generating two subrectangles
   replace R in the list with the two subrectangles
end while
```

# 4 Generating rectangles satisfying the area ratio constraint

The data sets generated by Algorithm 2 consist of rectangles $h_i \times w_i$ with aspect ratio $\rho$ (i.e. $1/\rho \leq h_i/w_i \leq \rho$). The areas of these rectangles, however, often vary as widely as those produced by the basic algorithm alone. To control the range of areas, a second parameter, $\gamma$ is now introduced. A modification to Algorithm 1 can be made to generate data sets whose rectangles satisfy a user-specified area ratio $\gamma \geq 2$. That is, the ratio of the areas of any two rectangles in the data set must fall in the interval $[1/\gamma, \gamma]$. To ensure that the generated rectangles satisfy this constraint, the following properties are noted.

**Theorem 5** *Let $\gamma \geq 2$ and $\{R_i\}$ be a set of $n$ rectangles with area ratio $\gamma$ which are ordered by non-increasing areas:*

$$\text{area}(R_0) \geq \text{area}(R_1) \geq \text{area}(R_2) \geq \dots \geq \text{area}(R_{n-1})$$

*Any $R_j$ where $\text{area}(R_j) \geq 2\,\text{area}(R_0)/\gamma$ can be sliced vertically into two subrectangles so that the resulting set of $n+1$ subrectangles will have area ratio $\gamma$.*

**Proof.** Assume that rectangle $R_j$ is selected for cutting, and let the vertical slicing of $R_j$ take place at position $x$. Since cuts at $x$ and $w_j - x$ yield symmetric subrectangles, we limit the choice of $x$ to $x \leq w_j/2$.

To ensure that the two subrectangles resulting from this slicing both have areas of at least $\text{area}(R_0)/\gamma$ when $\gamma \geq 2$, we restrict $x$ further so that $x \geq \frac{\text{area}(R_0)}{\gamma h_j}$. It is possible to restrict $x$ this way because $\text{area}(R_j) = h_j w_j \geq 2\,\text{area}(R_0)/\gamma$, i.e.

$$w_j/2 \geq \frac{\text{area}(R_0)}{\gamma h_j}. \tag{1}$$

The initial set of $n$ sorted rectangles were assumed to have area ratio $\gamma$; in particular

$$1/\gamma \leq \text{area}(R_p)/\text{area}(R_q) \leq \gamma \text{ for all } p, q \neq j$$

and this expression still holds after $R_j$ has been sliced.

Let $A$ and $B$ denote the resulting subrectangles from the slicing of $R_j$ as shown in Figure 4. The remaining area

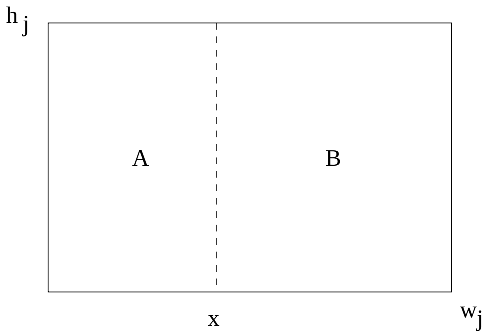


Figure 4: Vertical slicing of rectangle $R_j = h_j \times w_j$ rectangle

ratios that must be examined are: (i) $\text{area}(R_i)/\text{area}(A)$ and $\text{area}(R_i)/\text{area}(B)$ for all $i \neq j$ and
(ii) $\text{area}(A)/\text{area}(B)$

case(i) Since $\text{area}(A) \leq \text{area}(R_j)$ and
$\text{area}(R_j)/\text{area}(R_{n-1}) \leq \gamma$, then
$\text{area}(A)/\text{area}(R_{n-1}) \leq \gamma$ and
$\text{area}(R_{n-1})/\text{area}(A) \geq 1/\gamma$.
But $\text{area}(R_i) \geq \text{area}(R_{n-1})$,
so $\text{area}(R_i)/\text{area}(A) \geq 1/\gamma$ for all $i \neq j$.

Furthermore, the vertical cut position $x$ was chosen so that $\text{area}(A) \geq \text{area}(R_0)/\gamma$, implying that

$\text{area}(A)/\text{area}(R_0) \geq 1/\gamma$ so that $\text{area}(R_0)/\text{area}(A) \leq \gamma$.

Since $\text{area}(R_i) \leq \text{area}(R_0)$,
we have $\text{area}(R_i)/\text{area}(A) \leq \gamma$ for all $i \neq j$.

Thus,

$1/\gamma \leq \text{area}(R_i)/\text{area}(A) \leq \gamma$ for all $i \neq j$

and similarly,

$1/\gamma \leq \text{area}(R_i)/\text{area}(B) \leq \gamma$ for all $i \neq j$.

case (ii) The vertical cut position guarantees that $\text{area}(A) \geq \text{area}(R_0)/\gamma$ so

$\text{area}(A)/\text{area}(B) \geq \text{area}(R_0)/(\gamma\,\text{area}(B))$.

Further, $\text{area}(B) \leq \text{area}(R_j) \leq \text{area}(R_0)$
yields $\text{area}(R_0)/\text{area}(B) \geq 1$, so $\text{area}(A)/\text{area}(B) \geq 1/\gamma$.

Similarly,

$\text{area}(A) \leq \text{area}(R_j) \leq \text{area}(R_0)$

and

$\text{area}(B) \geq \text{area}(R_0)/\gamma$

combine to yield $\text{area}(A)/\text{area}(B) \leq \gamma$, and so

$1/\gamma \leq \text{area}(A)/\text{area}(B) \leq \gamma$.

This completes the proof that the set of $n + 1$ resulting subrectangles satisfies the area ratio constraint. □

**Corollary 3** *Let $m$ denote the area of the rectangle having the maximum area in a list of $n$ rectangles with area ratio $\gamma$. Select any rectangle $R_j$ where $\text{area}(R_j) \geq 2m/\gamma$ and slice it vertically at position $x$ where $\frac{m}{\gamma h_j} \leq x \leq w_j/2$. The resulting set of $n + 1$ rectangles has area ratio $\gamma$.*

It can be shown that a similar condition for horizontal cutting exists. For brevity, we state only the corresponding corollary.

**Corollary 4** *Let $m$ denote the area of the rectangle having the maximum area in a list of $n$ rectangles with area ratio $\gamma$. Select any rectangle $R_j$ where $\text{area}(R_j) \geq 2m/\gamma$ and slice it horizontally at position $y$ where $\frac{m}{\gamma w_j} \leq y \leq h_j/2$. The resulting set of $n + 1$ rectangles has area ratio $\gamma$.*

By incorporating these observations into the basic algorithm, an $O(n^2)$ data generation procedure that creates a set of rectangles satisfying the area ratio constraint can be written as expressed in Algorithm 3.

**Algorithm 3** Controlling the Area Ratio

```
Input the parameters n, γ ≥ 2 , H, and W
while n rectangles not yet generated do
    let m be the area of the largest rectangle in the current set
    choose a rectangle R from all subrectangles whose areas are greater than 2m/γ
    randomly choose a vertical or horizontal slicing direction
    randomly choose a cutting position within the legal range of slicing positions {Corollary 3 or 4}
    perform the cut on R, generating two subrectangles
    replace R in the list with the two subrectangles
end while
```

# 5 Combining aspect and area ratio constraints

The algorithms developed thus far generate data sets where the sizes and areas of the rectangles can be constrained by either aspect ratio or area ratio. In many instances, it is desirable to employ data sets where both the aspect ratio and the maximum-to-minimum area ratio are bounded.

To accomplish this, the algorithms derived in sections 3 and 4 can be combined. However, merging the two methods requires that the conditions in Corollaries 1, 2, 3 and 4 be met. To prove that these conditions do not conflict, the following theorem is established.

**Theorem 6** *Let $\rho, \gamma \geq 2$ and $\{R_i\}$ be a set of $n$ rectangles with aspect ratio $\rho$ and area ratio $\gamma$ which are ordered by non-increasing areas:*

$\text{area}\,(R_0) \geq \text{area}\,(R_1) \geq \text{area}\,(R_2)$
$\geq \ldots \geq \text{area}\,(R_{n-1})$

*Any $R_j$ where $\text{area}\,(R_j) \geq 2\,\text{area}\,(R_0)/\gamma$ can be sliced into two subrectangles so that the resulting set of $n+1$ subrectangles will have aspect ratio $\rho$ and area ratio $\gamma$.*

**Proof.** Assume that rectangle $R_j = h_j \times w_j$ is selected for cutting because $R_j$ meets the condition $\text{area}\,(R_j) \geq 2\,\text{area}\,(R_0)/\gamma$. Note this implies that inequality (1) in Section 4 holds.

From Theorem 4, we know that $R_j$ can also be sliced either vertically or horizontally. (If Theorem 3 is satisfied, $R_j$ can be sliced in either direction.) Specifically, if the conditions of Theorem 1 hold, then $R_j$ can be sliced vertically; if the conditions of Theorem 2 hold, then $R_j$ can be sliced horizontally.

Suppose that the conditions of Theorem 1 hold:

$$\frac{2h_j}{\rho} \leq w_j \leq 2\rho h_j, \tag{2}$$

and so Corollary 1 defines the positions $x$ for slicing vertically so that the resulting two subrectangles will have aspect ratio $\rho$:

$$\max(h_j/\rho, w_j - h_j\rho) \leq x \leq \min(h_j\rho, w_j - h_j/\rho) \tag{3}$$

Similarly, in order to ensure that the $n+1$ subrectangles resulting from the cut will have area ratio $\gamma$, Corollary 3 dictates the slicing positions as

$$\frac{m}{\gamma h_j} \leq x \leq w_j/2 \tag{4}$$

where $m = \text{area}\,(R_0)$.

If the vertical slicing position $x$ can be selected to satisfy both inequalities 3 and 4, then the resulting set of rectangles will have aspect ratio $\rho$ and area ratio $\gamma$. A proof by contradiction establishes that $x$ can be so chosen.

Suppose there is no $x$ that satisfies both inequalities (3) and (4). Then either

(i) $w_j/2 < \max(h_j/\rho, w_j - h_j\rho)$

or

(ii) $\frac{m}{\gamma h_j} > \min(h_j\rho, w_j - h_j/\rho)$.

case(i) If $w_j/2 < \max(h_j/\rho, w_j - h_j\rho)$, then either (a) $w_j/2 < h_j/\rho$ or (b) $w_j/2 < w_j - h_j\rho$. Inequality (a) implies that $w_j < 2h_j/\rho$ and (b) implies that $2\rho h_j < w_j$ which both contradict inequality (2).

case(ii) If $\frac{m}{\gamma h_j} > \min(h_j\rho, w_j - h_j/\rho)$, then either (a) $\frac{m}{\gamma h_j} > h_j\rho$ or (b) $\frac{m}{\gamma h_j} > w_j - h_j/\rho$. Inequality (a) implies that

$\frac{m}{\gamma h_j w_j} > \frac{h_j\rho}{w_j}$

and since $\gamma \geq \frac{m}{h_j w_j}$ and $\frac{h_j}{w_j} \geq \frac{1}{\rho}$, this leads to the contradiction that $1 > 1$.

Since $w_j - h_j/\rho \geq w_j/2$ using inequality (2), inequality (b) simplifies to $\frac{m}{\gamma h_j} > w_j/2$ which contradicts inequality (1) in section 4. Thus, $x$ can be chose to satisfy both conditions of Corollaries 1 and 3.

Using similar techniques, a proof for the case where $R_j$ is to be sliced horizontally can be derived by applying Theorem 2 and Corollaries 2 and 4. □

**Corollary 5** *Let $m$ denote the area of the rectangle having the maximum area in a list of $n$ rectangles with aspect ratio $\rho$ and area ratio $\gamma$. Select any rectangle $R_j$ where $\text{area}\,(R_j) \geq 2m/\gamma$ and slice it vertically at position $x$ where*

$\max(h_j/\rho, w_j - h_j\rho, \frac{m}{\gamma h_j}) \leq x$
$\leq \min(h_j\rho, w_j - h_j/\rho, w_j/2)$.

*The resulting set of $n+1$ rectangles has aspect ratio $\rho$ and area ratio $\gamma$.*

**Corollary 6** *Let $m$ denote the area of the rectangle having the maximum area in a list of $n$ rectangles with aspect*

| Data Set | | Height | | | Width | | |
|---|---|---|---|---|---|---|---|
| Name | $n$ | max | min | avg | max | min | avg |
| path_10 | 10 | 100 | 8.35837 | 60 | 66.1978 | 3.05587 | 38.5064 |
| path_20 | 20 | 74.8535 | 0.37062 | 22.9127 | 181.903 | 0.0047569 | 43.3106 |
| path_30 | 30 | 63.5651 | 0.16178 | 20.3529 | 126.544 | 0.0980349 | 48.8760 |
| path_50 | 50 | 73.4758 | 1.24164e-05 | 7.3563 | 136.987 | 0.4282280 | 41.4501 |
| path_100 | 100 | 78.9366 | 0.00165 | 7.68243 | 145.203 | 0.0109578 | 26.8371 |
| path_200 | 200 | 37.2576 | 2.10877e-07 | 5.56013 | 198.444 | 0.0008622 | 32.9201 |
| path_500 | 500 | 76.5074 | 2.36039e-05 | 1.41293 | 171.899 | 6.76548e-06 | 4.5372 |
| path_1t | 1000 | 35.4326 | 2.23302e-07 | 1.83857 | 127.698 | 1.17497e-06 | 10.1380 |
| path_2t | 2000 | 90.6677 | 2.39115e-09 | 3.81625 | 157.532 | 2.4667e-10 | 5.0222 |
| path_5t | 5000 | 54.8126 | 6.44536e-09 | 2.11321 | 139.143 | 4.55262e-08 | 2.3374 |
| nice_10 | 10 | 69.0438 | 19.1295 | 44.6252 | 70.5507 | 18.8939 | 47.0551 |
| nice_20 | 20 | 58.9809 | 11.0813 | 33.2061 | 62.7279 | 16.8828 | 29.8496 |
| nice_30 | 30 | 100 | 14.6891 | 27.0852 | 94.2618 | 11.5792 | 24.9402 |
| nice_50 | 50 | 60.9188 | 8.69446 | 21.4948 | 37.1099 | 8.88438 | 20.4661 |
| nice_100 | 100 | 32.4315 | 5.10983 | 14.6936 | 30.8926 | 4.80933 | 14.1599 |
| nice_200 | 200 | 28.0957 | 3.67168 | 10.721 | 25.7108 | 3.55487 | 10.2708 |
| nice_500 | 500 | 20.4512 | 2.33988 | 6.63364 | 18.8291 | 2.58266 | 6.56829 |
| nice_1t | 1000 | 12.1489 | 1.60008 | 4.55755 | 13.7578 | 1.6738 | 4.76367 |
| nice_2t | 2000 | 10.4493 | 1.24654 | 3.32311 | 11.4757 | 1.22907 | 3.29283 |
| nice_5t | 5000 | 6.91479 | 0.780134 | 2.07974 | 7.18023 | 0.784575 | 2.11803 |

Table 1: Height and width statistics for example data sets

*ratio $\rho$ and area ratio $\gamma$. Select any rectangle $R_j$ where* area $(R_j) \geq 2m/\gamma$ *and slice it horizontally at position $y$ where*

$$\max(w_j/\rho, h_j - w_j\rho, \tfrac{m}{\gamma w_j}) \leq y$$
$$\leq \min(w_j\rho, h_j - w_j/\rho, h_j/2).$$

*The resulting set of $n+1$ rectangles has aspect ratio $\rho$ and area ratio $\gamma$.*

A fourth $O(n^2)$ data generation procedure can now be written as expressed in Algorithm 4.

**Algorithm 4** Controlling the Aspect and Area Ratio

Input the parameters $n$, $\{\gamma, \rho \geq 2\}$, $H$, and then $W$, where $2H/\rho \leq W \leq \rho H/2$
**while** $n$ rectangles not yet generated **do**
  let $m$ be the area of the largest rectangle in the current set
  choose a rectangle $R$ from all subrectangles whose areas are greater than $2m/\gamma$
  if possible, randomly choose a vertical or horizontal slicing direction;
  otherwise select the vertical or horizontal direction as appropriate $\{$Theorem 2 or 3$\}$
  randomly choose a cutting position within the legal range of slicing positions $\{$Corollary 5 or 6$\}$
  perform the cut on $R$, generating two subrectangles
  replace $R$ in the list with the two subrectangles
**end while**

# 6 Sample Data Sets

To illustrate the differences in the data sets that can be generated by the approaches described in this paper, we examine some sample data sets that were produced by Algorithms 1 and 4. For each of these sets, an initial rectangle of size $100 \times 200$ was recursively sliced as discussed in sections 2 and 5. The characteristics of the resulting data sets are summarized in this section.

The first group of data sets shown in Table 1 were generated by the basic routine given in Algorithm 1. Recall that no restrictions are placed on aspect ratio or area ratio in this case– rectangles are randomly selected for cutting, and slices are equally likely to be made in random directions. This procedure results in generating the rectangles whose height and width dimensions are shown in Figure 5 for $n = 50$, $n = 500$, and $n = 5000$.

We refer to these sets as "pathological" data sets because there is a large variance in the heights and widths of the generated rectangles. As the data sets get larger in size,

| Data Set | | Aspect ratio = Height/Width | | Area | | |
|---|---|---|---|---|---|---|
| Name | $n$ | max | min | max | min | $\gamma$ = max/min |
| path_10 | 10 | 32.723862 | 0.177596 | 6619.78 | 305.587 | 21.6625 |
| path_20 | 20 | 10240.3 | 0.034009 | 8319.75 | 0.124119 | 67030.6 |
| path_30 | 30 | 340.701 | 0.00127842 | 3466.24 | 3.27442 | 1058.58 |
| path_50 | 50 | 3.37028 | 2.40351e-06 | 7847.19 | 6.41424e-05 | 1.2234e+08 |
| path_100 | 100 | 532.073 | 6.24847e-05 | 11461.8 | 0.000493591 | 2.32212e+07 |
| path_200 | 200 | 15628.6 | 6.01994e-09 | 2546.71 | 1.83324e-06 | 1.38918e+09 |
| path_500 | 500 | 23600.3 | 4.14669e-05 | 6141.94 | 2.6104e-09 | 2.35288e+12 |
| path_1t | 1000 | 685712 | 1.36322e-08 | 2428.64 | 6.33723e-08 | 3.83234e+10 |
| path_2t | 2000 | 9.76744e+07 | 1.1043e-07 | 3659.14 | 4.72855e-13 | 7.73839e+15 |
| path_5t | 5000 | 2.9487e+07 | 1.09439e-09 | 1146.07 | 1.11438e-14 | 1.02843e+17 |
| nice_10 | 10 | 2.447984 | 0.271145 | 3555.91 | 873.88 | 4.06911 |
| nice_20 | 20 | 3.113818 | 0.276488 | 2975.18 | 436.494 | 6.81608 |
| nice_30 | 30 | 3.888324 | 0.272336 | 2571.8 | 372.526 | 6.90369 |
| nice_50 | 50 | 3.433568 | 0.260070 | 1132 | 169.914 | 6.66221 |
| nice_100 | 100 | 3.993231 | 0.320414 | 545.908 | 78.5245 | 6.95208 |
| nice_200 | 200 | 3.970387 | 0.252198 | 285.179 | 42.3561 | 6.73289 |
| nice_500 | 500 | 3.999504 | 0.250588 | 150.389 | 21.5188 | 6.98872 |
| nice_1t | 1000 | 3.975254 | 0.250259 | 67.7191 | 9.67665 | 6.9982 |
| nice_2t | 2000 | 3.996923 | 0.250161 | 41.0826 | 5.86941 | 6.99945 |
| nice_5t | 5000 | 3.999353 | 0.250004 | 16.2659 | 2.32397 | 6.99918 |

Table 2: Height/Width and area ratios for sample data sets

either the height or width of the rectangles also seem to get very small.

The second group of data sets shown in Table 1 were generated using Algorithm 4 where the slicing positions are controlled so that resulting rectangles will have aspect ratio $\rho$ and the data set will have area ratio $\gamma$. The values $\rho = 4$ and $\gamma = 7$ were selected for these data sets. Thus each generated rectangle has a height/width ratio lying between [0.25,4]. Similarly, the ratio of the largest area to smallest rectangle area in any data set does not exceed 7.

Figure 6 plots the rectangles for the data sets of size $n = 50$, $n = 500$, and $n = 5000$. The sets can be thought of as "nice" data sets because the rectangles' characteristics fall within specified ranges: there are no long flat or tall thin rectangles and the areas of the rectangles are of the same magnitude.

The characteristics of these two types of data sets are further illustrated by regarding their observed values of $\rho$ and $\gamma$. Note that in Table 2 there tends to be at least two magnitudes of difference in the rectangles' height/width ratios within every pathological data set where $n > 30$. For the nice data sets generated by Algorithm 4, each rectangle's aspect ratios is at most 4 and no less than 0.25 for all data sets.

Figure 7 plots the sorted height/width ratios for rectangles in the pathological and nice data sets where $n = 50$. The graphs shows how the height/width ratios are distributed within each data set and reflects the difference in magnitude shown in Table 2. The distribution of height/width ratios for the pathological rectangles range from many small ratios to several larger ratios. The aspect ratios for the nice data sets fall only between 0.25 and 4.

The differences in the areas of the rectangles belonging to the pathological and nice data sets are also shown in Table 2. For the pathological data sets, the ratio of the largest rectangle area to smallest rectangle area appears to increase by orders of magnitude as the number of rectangles in the sets grows larger. For the nice data sets, this ratio is at most 7, the value specified as the input parameter $\gamma$.

In addition to finding the maximum and minimum area values for each data set, the average rectangle area could be calculated, but this value will always equal the total area of the initial rectangle divided by the number of rectangles generated and so does not provide much any additional information about the data set. However, the areas of the rectangles in each data set can easily be plotted to show the range of their distribution.

In Figure 8, the distribution of area values includes many small and some very large rectangles for the pathological data set. As expected, the area variance in the nice data set is far smaller due to its being bounded by a maximum area ratio equal to 7.

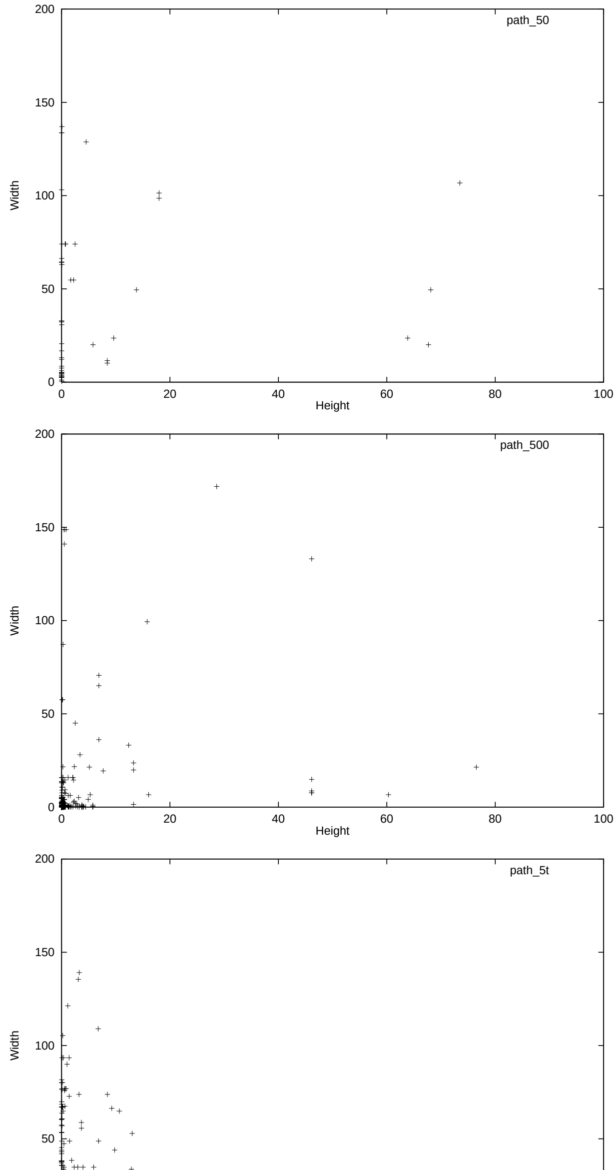


Figure 5: Height and width of rectangles in pathological data sets

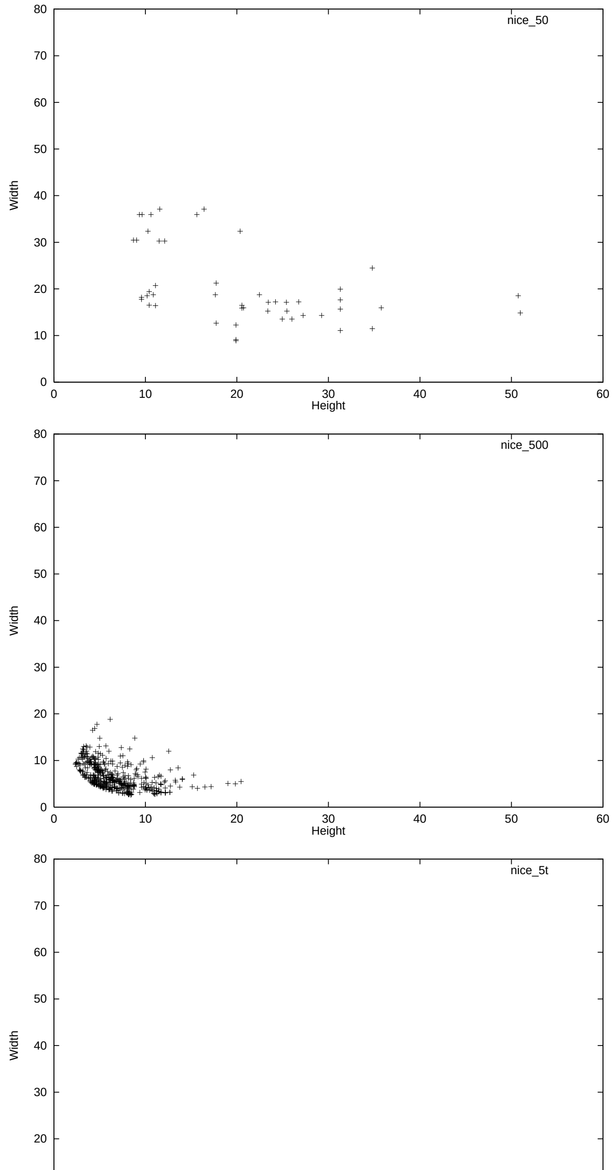


Figure 6: Height and width of rectangles in some nice data sets

# 7 Using the Algorithms to Generate Test Data Sets

The procedures outlined in this report permit the generation of sets of $n$ rigid rectangles which can be packed or cut from rectangular regions with zero waste. The sizes and areas of these rectangles may have large variance as produced by Algorithm 1 or can be restricted to satisfy user specified aspect and maximum-to-minimum area ratios using Algorithms 2, 3 or 4. Data sets of any size can be obtained in this manner.

These data sets can be used to evaluate the performance of algorithms which are intended to solve the problem of packing a given set of rectangles into a single larger two-dimensional rectangular region with minimum total waste, or equivalently, minimal total height in the case of two-dimensional strip packing. Several data sets can also be replicated or combined to yield test suites for solving cutting stock problems. Individual rectangles can be rotated, or different stock sizes could be used when generating the data. In these cases, a range of aspect or area ratios might also be specified. The optimal solution would still have zero

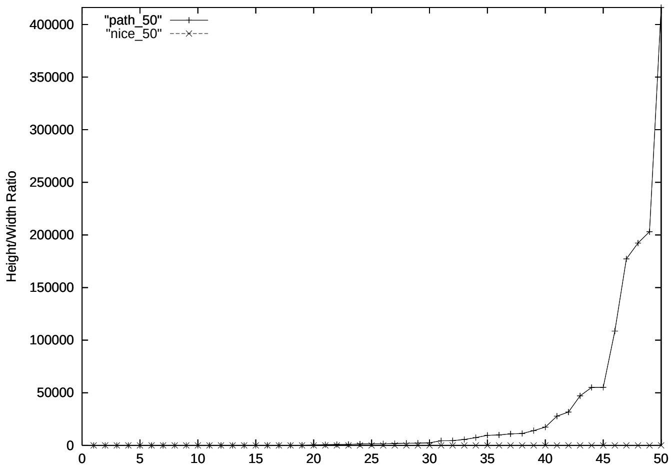


Figure 7: Height/Width comparisons for data sets of size $n = 50$

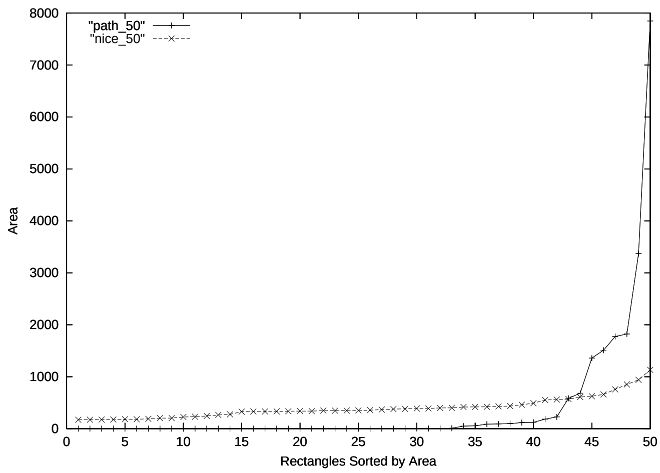


Figure 8: Area comparisons for data sets of size $n = 50$

waste as it would consist of all the original rectangles that were sliced to generate the data sets. The zero waste property, while not reflective of real-world cutting problems, still provides researchers with a means for comparing the quality of their solutions against a known optimal.

One characteristic of the example data sets described in the previous section is clearly that the dimensions of the rectangles are given as real values. The precision of these values is dictated by the default arithmetic precision of the programming language and computer system on which the algorithms are implemented and executed. By limiting the precision of the output statements to the number of desired decimal places (i.e. by rounding the values), we have found that we can usually overcome the difficulties inherent with finite precision arithmetic.

It is also possible to utilize our algorithms to generate data

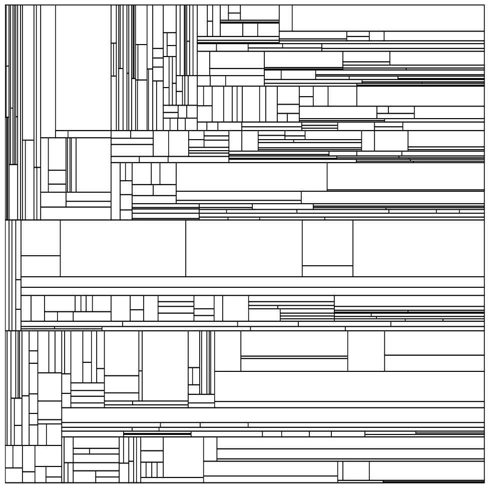

Figure 9: A pathological data set of 500 rectangles with integer dimensions

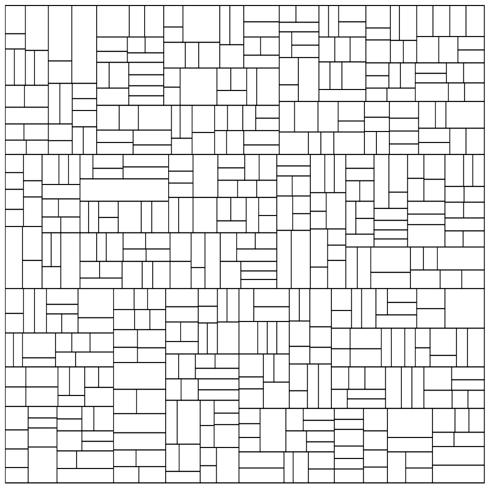

Figure 10: A nice data set of 500 rectangles with integer dimensions

sets containing integer valued rectangles, i.e. rectangles whose heights and widths are integers. One way to accomplish this is by rounding the cut positions to the nearest integer. Two examples of data sets that were generated by rounding the cut positions are shown in Figures 9 and 10 which were generated using Algorithms 1 and 4, respectively. However, it is not guaranteed that a rounded cut position will satisfy the conditions needed to ensure that the aspect and area ratios of the final rectangles will fall within the specified ranges.

For example, if the input rectangle to be sliced has size $10 \times 50$, the maximum number of integer dimension rectangles that could be generated by arbitrarily slicing this rectangle would be $n = 500$ rectangles of size $1 \times 1$. Requesting that the procedure generate 1000 rectangles with integer dimensions would result in at least $500$ final rectangles having a height or width of zero. Thus, the choice of the size of the input rectangle must be carefully considered along with the corresponding $n$ value.

In Figures 9 and 10, a $1000 \times 1000$ rectangle was sliced to generate the data sets of $n = 500$ rectangles. The $\rho$ and $\gamma$ values that were input to Algorithm 4 to obtain the data in Figure 10 were $4$ and $7$, respectively. The algorithms successfully generated rectangles having nonzero integer dimensions. However, because of the rounding process, the data set generated by Algorithm 4 has an aspect ratio of $4.18$ and an area ratio of $7.32$ which are greater than the specified values. Nevertheless, the generated data sets still have bounded aspect and area ratios, and we believe that some additional analysis of our methods will show that the aspect and area ratios of the generated data sets can be predicted when the rectangles are rounded to integer dimensions.

# 8 Ongoing Research

Current versions of our algorithms give the user control over the range of variation within data sets for aspect ratio and area, but no control over the distribution of shapes and sizes. It is possible that small alterations to our data generation algorithms would result in different types of data sets which a researcher may prefer. There are many possibilities here: for example, the largest rectangle at any stage could be the one chosen to be sliced, or slicing positions could be restricted to favor more or less extreme aspect ratios. The distribution of rectangle sizes and areas generated by our algorithms depends on the probability distributions which govern the random choices for the rectangles to be recursively sliced as well as the direction and the position of the slice, to the extent dictated by the appropriate theorems and corollaries. A probabilistic analysis of the size distribution of the final rectangles, or an analysis of the impact of changing the slicing probabilities would be interesting to pursue.

Extensions of our general method can also be made so that data sets can be generated (with bounded aspect and/or area ratios) whose optimal packing (or cutting) solutions consist of non-slicing (i.e. non-guillotine) patterns. This research area is currently under study. In addition we are also considering modifications to our software that will make it possible to generate data sets with known optima for testing algorithms for VLSI floorplanning.

Finally, a public domain software package consisting of the algorithms described in this paper as well as visualization tools for drawing the data sets are under development and will be distributed in the near future.

# Acknowledgements

We would like to thank Yun You for modifying the software and the PLT drawing tool which were used to create the illustrations used in this paper. The authors would also like to thank the reviewers for their insightful comments and suggestions for improving the paper.

# References

[1] J.E. Beasley. OR-Library: distributing test problems by electronic mail. *Journal of the Operational Research Society*, 41(11):1069–1072, 1990. url: `http://people.brunel.ac.uk/~mastjjb/jeb/info.html`

[2] Bengt-Erik Bengtsson. Packing Rectangular Pieces – A Heuristic Approach. *The Computer Journal*, 25(3):353–357, 1982.

[3] J.O. Berkey and P.Y. Wang. Two-Dimensional Finite Bin-Packing. *Journal of the Operational Research Society*, 38(5):423–439, 1987.

[4] ESICUP- The Special Interest Group on Cutting and Packing. `http://paginas.fe.up.pt/~esicup/tiki-index.php` .

[5] S.S. Israni and J.L. Sanders. Performance testing of rectangular parts-nesting heuristics. *International Journal of Production Research*, 23:437–456, 1985.

[6] Stefan Jakobs. On Genetic Algorithms for the Packing of Polygons. *European Journal of Operational Research*, 88:165–181, 1996.

[7] Dequan Liu and Hongfei Teng. An Improved BL-algorithm for Genetic Algorithm of the Orthogonal Packing of Rectangles. *European Journal of Operational Research*, 112:412–420, 1999.