Two-dimensional strip packing problem

A Preprint

Introduction

In 1980, B. S. Baker and E. G. Coffman proposed the problem [1]: Given a vertical strip of width W, bounded below but not above, and a list L of rectangular regions $R_1 \ldots R_n$, pack the regions into the strip so that the final height is a minimal. We also assume each region R_i, $1 \le i \le n$, is to be defined by its width w_i and height h_i. The rectangles are allowed neither to overlap nor to rotate. This type of packing is called a *two-dimensional strip packing*.

Definition

Let us consider the vertical strip

$$S = \{(x, y) \in \mathbb{Q}^2 \mid 0 \le x \le W,\ y \ge 0\},\ W \in \mathbb{Q}_+$$

and the list $L = \{(w_i, h_i)\}_{i=1}^n \subset \mathbb{Q}_+^2$, $n \in \mathbb{N}$. Let also $O := \{(x_i, y_i)\}_{i=1}^n \subset \mathbb{Q}^2$ and $R_i := [x_i; x_i + w_i] \times [y_i; y_i + h_i] \cap \mathbb{Q}^2$. Let us note that elements of L can duplicate.

A pair $P = (S, R)$, $R = \{R_i\}_{i=1}^n$, is called a *strip packing w.r.t. S and L* if the following conditions hold

1. $R_i \subset S$, for any i, $1 \le i \le n$.
2. $\mathrm{Int}(R_i) \cap \mathrm{Int}(R_j) = \varnothing$, for any i, j, $1 \le i, j \le n$, $i \ne j$.

Here $\mathrm{Int}(R_i) := (x_i; x_i + w_i) \times (y_i; y_i + h_i) \cap \mathbb{Q}^2$.

A value

$$H(P) := \max_{1 \le i \le n} (y_i + h_i)$$

is called a *height* of strip packing P.

Let $\mathcal{P}$ be a family of all strip packings w.r.t. S and L. A packing $P^* \in \mathcal{P}$ is called *optimal* if it satisfies the optimum condition

$$H(P^*) = \min_{P \in \mathcal{P}} H(P).$$

A height $H^* := H(P^*)$ is called an *optimal height*.

The objective of the *strip packing problem* is to find P^*. This problem is strongly NP-hard.

There are numerous ways to model and solve this problem (see [2]). We will focus on the Steinberg approximation algorithm and two heuristics.

Steinberg algorithm

Let us focus on the Steinberg algorithm. See [3]. It is an example of approximating algorithms with a predefined ratio of the upper bound. That is, there is $\alpha > 1$ such that

$$\tilde{H} \le \alpha H^*,$$

where $\tilde{H}$ stands for the height obtained with the algorithm.

For the Steinberg approximation, $\alpha = 2$. Nowadays, the best algorithm provides $\alpha = \frac{5}{4} + \varepsilon$ [4].

At the end of this section, we also provide two simple modifications of the Steinberg algorithm and obtain empirical ratios for them. Let

$$S = \{(x, y) \in \mathbb{Q}^2 \mid 0 \leq x \leq W,\ y \geq 0\},\ W \in \mathbb{Q}_+,$$

and

$$L = \{(w_i, h_i)\}_{i=1}^n \in \mathbb{Q}_+^2,\ n \in \mathbb{N}.$$

Let $S = \sum w_i h_i$, $w = \max\{w_i\}$, $h = \max\{h_i\}$.

The Steinberg algorithm provides packing into a container

$$Q = [0; W] \times [0; H] \cap \mathbb{Q}^2,\ W, H \in \mathbb{Q}_+$$

where W, H satisfy the conditions

$$w \leq W,\ h \leq H,\ 2S \leq WH - \max\{2w - W, 0\} \cdot \max\{2h - H, 0\}.$$

If these conditions hold, then there exists the Steinberg packing.

For any rectangle $R = [x_1; x_2] \times [y_1; y_2] \cap Q^2$, let us provide the notations

$$_*[R] := (x_1, y_1),\ ^*[R] := (x_1, y_2),\ [R]_* := (x_2, y_1),\ [R]^* := (x_2, y_2)$$

The Steinberg algorithms is the following:

Step 1 We need to compute H to satisfy sufficient conditions. Put $H := \max\{H', h\}$, where

$$H' := \begin{cases} \frac{S + 4wh - Wh}{2w} & S \leq Wh \leq 2wh \\ \frac{2S}{W} & \text{otherwise} \end{cases}$$

Step 2 We recursively perform the reduction process for Q and L. The process then stops after packing all rectangles. Given an arbitrary (Q, L), we use an appropriate procedure from the following list

P1 This procedure can be applied to the (Q, L) if the condition holds:

$$2w \geq W$$

First, we order and re-number the rectangles of list L by decreasing width $w_1 \geq \cdots \geq w_n$. Let m, $1 \leq m \leq n$, be the maximal index such that $2w_m \geq W$. Place the rectangles $R_1, \ldots, R_m$ so that

$$_*[R_1] = {}_*[Q] \text{ and } {}_*[R_i] = {}^*[R_{i-1}],\ 2 \leq i \leq m$$

If $m = n$, procedure P1 solves the problem. Suppose that $m < n$. Let

$$h' = H - \sum_{i=1}^{m} h_i.$$

We order and re-number the rectangles $R_{m+1}, \ldots, R_n$ by decreasing height $h_{m+1} \geq \cdots \geq h_n$.
If $h_{m+1} \leq h'$, we form a problem (Q', L'), where $L' = \{(w_i, h_i)\}_{i=m+1}^n$ and the container Q' is defined as

$$_*[Q'] = {}^*[R_m],\ [Q']^* = [Q]^*$$

If $h_{m+1} > h'$, denote by k, $m + 1 \leq k \leq n$, the maximal index for which $b_k > h'$ and place $R_{m+1}, \ldots, R_k$ in the following way

$$[R_1]^* = [Q]^* \text{ and } [R_i]^* = {}^*[R_{i-1}],\ m + 2 \leq i \leq k$$

If $k = n$, procedure P1 solves the problem. Suppose that $k < n$, we form a problem (Q', L'), where $L' = \{(w_i, h_i)\}_{i=k+1}^n$ and the container Q' is defined as

$$_*[Q'] = {}^*[R_m],\ [Q']^* = {}^*[R_k]$$

Pm1 This procedure can be applied to the (Q, L) if the condition holds:

$$2h \geq H$$

This procedure is transformed from P1 by interchanging the horizontal and vertical directions.

P3 First, we order and re-number the rectangles of list L by decreasing width $w_1 \geq \cdots \geq w_n$.
This procedure can be applied to the (Q, L) if the conditions hold:

$$2w \leq W,\ 2h \leq H,\ n > 1,$$

and

$$S - \frac{1}{4}WH \leq \sum_{i=1}^{m} w_i h_i \leq \frac{3}{8}WH,\ w_{m+1} \leq \frac{1}{4}W,$$

for some index m, $1 \leq m < n$.
We set $W' := \max\{\frac{1}{2}W, \frac{2\sum_{i=1}^{n} w_i h_i}{H}\}$, $W'' := W - W'$. Then we cut Q into two containers Q' and Q'' with widths W' and W'' and the same height H, and we consider now two problems (Q', L') and (Q'', L'') with $L' = \{(w_i, h_i)\}_{i=1}^{m}$, $L'' = \{(w_i, h_i)\}_{i=m+1}^{n}$.

Pm3 First, we order and re-number the rectangles of list L by decreasing height $h_1 \geq \cdots \geq h_n$.
This procedure can be applied to the (Q, L) if the conditions hold:

$$2w \leq W,\ 2h \leq H,\ n > 1,$$

and

$$S - \frac{1}{4}WH \leq \sum_{i=1}^{m} w_i h_i \leq \frac{3}{8}WH,\ h_{m+1} \leq \frac{1}{4}H$$

for some index m, $1 \leq m < n$.
This procedure is transformed from P3 by interchanging the horizontal and vertical directions.

P2 This procedure can be applied to the (Q, L) if the conditions hold:

$$2w \leq W,\ 2h \leq H,\ n > 1,$$

and there exist two different indices i and k such that

$$w_i, w_k \geq \frac{1}{4}W,\ h_i, h_k \geq \frac{1}{4}H,\ 2(S - w_i \cdot h_i - w_k \cdot h_k) \leq (W - \max\{w_i, w_k\})H.$$

Assuming that $w_i \geq w_k$, we place R_i and R_k so that

$${}_*[R_i] = {}_*[Q],\ {}_*[R_k] = {}^*[R_i].$$

If $n > 2$, we consider a new problem (Q', L'), where $L' = L \setminus \{(w_i, h_i), (w_k, h_k)\}$ and Q' is the container such that

$${}_*[Q'] = [R_i]_*,\ [Q']^* = [Q]^*.$$

Pm2 This procedure can be applied to the (Q, L) if the conditions hold:

$$2w \leq W,\ 2h \leq H,\ n > 1,$$

and there exist two different indices i and k such that

$$w_i, w_k \geq \frac{1}{4}W,\ h_i, h_k \geq \frac{1}{4}H,\ 2(S - w_i \cdot h_i - w_k \cdot h_k) \leq (H - \max\{h_i, h_k\})W.$$

This procedure is transformed from P2 by interchanging the horizontal and vertical directions.

P0 This procedure can be applied to the (Q, L) if the (remaining) conditions hold:

$$2w \leq W,\ 2h \leq H,\ \text{and } S - \frac{1}{4}WH \leq w_i h_i$$

for some index i, $1 \leq i \leq n$.
We place R_i so that

$${}_*[R_i] = {}_*[Q].$$

If $n > 1$, we consider a new problem (Q', L'), where $L' = L \setminus \{(w_i, h_i)\}$ and Q' is the container such that

$${}_*[Q'] = [R_i]_*,\ [Q']^* = [Q]^*.$$

The Steinberg algorithm time complexity is $O(\frac{n \log^2 n}{\log \log n})$.

Examples

Example 1. Let $W = 2$ and $L = \{(1,1),\ (1,1),\ (2,1)\}$. Obviously, $H^* = 2$. Steinberg's packing provides $H = 3 > H^*$. Respective packing is pictured below

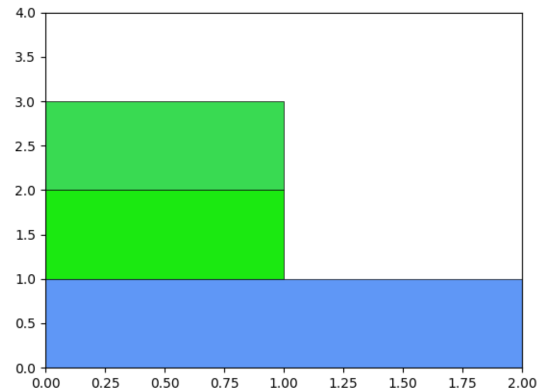

Figure 1: Example 1, original Steinberg algorithm

Example 2. Let $W = 30$ and $L = \{(20,6),\ (3,10),\ (7,10),\ (20,12),\ (10,8),\ (30,10)\}$. We have that $H^* = 28$:

$$O^* = \{(0,22),\ (20,10),\ (23,10),\ (0,10),\ (20,20),\ (0,0)\}$$

Steinberg's packing provides $H = 38$. Respective packing is pictured below

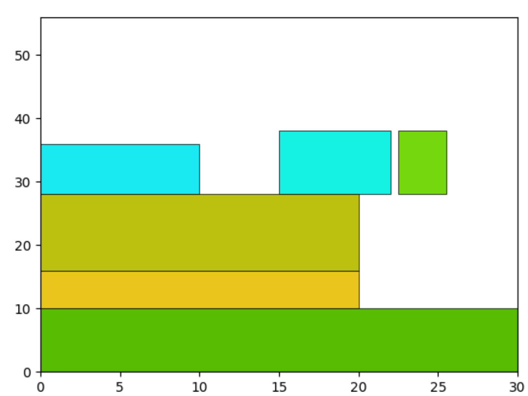

Figure 2: Example 2, original Steinberg algorithm

Example 3. Let $W = 30$ and

$$L = \{(5,3),\ (5,3),\ (2,4),\ (30,8),\ (10,20),\ (20,10),\ (5,5),\ (5,5),$$
$$(10,10),\ (10,5),\ (6,4),\ (1,10),\ (8,4),\ (6,6),\ (20,14)\}$$

Steinberg's packing provides $H = 71$. Respective packing is pictured below

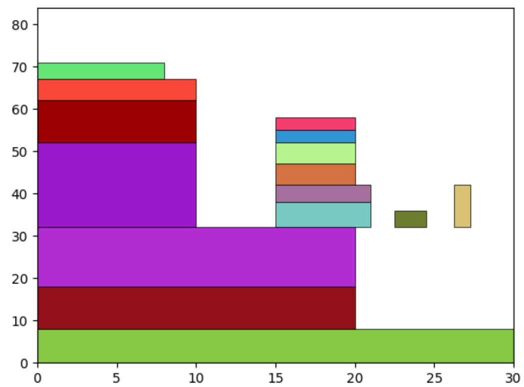

Figure 3: Example 3, original Steinberg algorithm

Example 4. Let $W = 12$ and $L = \{(4,3),\ (4,9),\ (1,12),\ (2,3),\ (2,7),\ (2,2),\ (5,2),\ (5,6),\ (5,4)\}$. We have that $H^* = 12$:

$$O^* = \{(1,9),\ (1,0),\ (0,0),\ (10,0),\ (10,3),\ (10,10),\ (5,0),\ (5,2),\ (5,8)\}$$

Steinberg's packing provides $H = 23.46$. Respective packing is pictured below

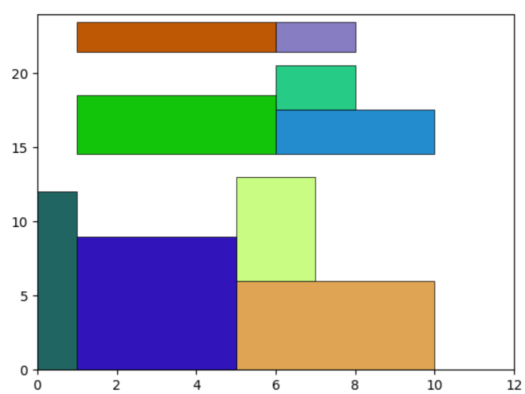

Figure 4: Example 4, original Steinberg algorithm

Example 5. Let $W = 25$ and $L = \{(10,8),\ (10,8),\ (12,4),\ (12,4),\ (25,3)\}$. We have that $H^* = 15$:

$$O^* = \{(0,0),\ (10,0),\ (0,8),\ (12,8),\ (0,12)\}$$

Steinberg's packing provides $H = 23.8$. Respective packing is pictured below

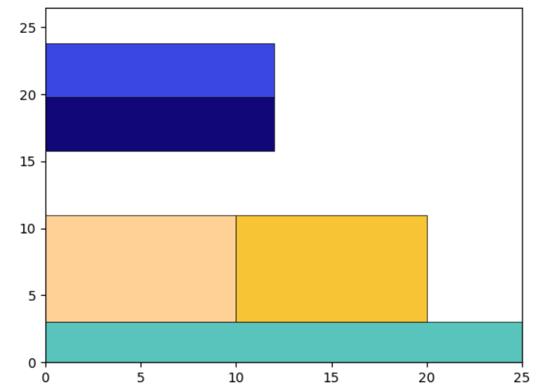

Figure 5: Example 5, original Steinberg algorithm

Example 6. Let $W = 10$ and $L = \{(3,3),\ (7,2),\ (7,1),\ (9,3),\ (5,4),\ (4,4),\ (7,1)\}$. We have that $H^* = 11$:

$$O^* = \{(0,0),\ (3,1),\ (3,0),\ (0,4),\ (5,7),\ (0,7),\ (0,3)\}$$

Steinberg's packing provides $H = 18$. Respective packing is pictured below

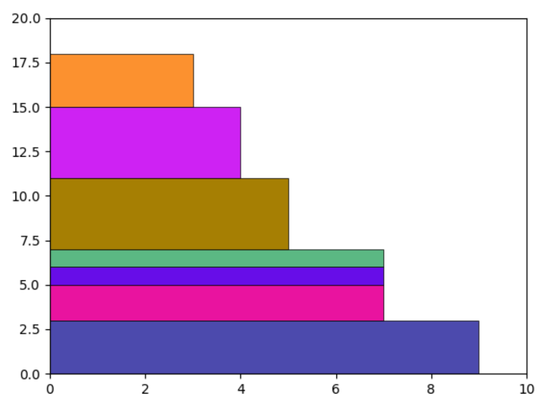

Figure 6: Example 6, original Steinberg algorithm

Example 7. Let $W = 10$ and $L = \{(1,1),\ (1,1),\ (10,8),\ (3,1),\ (9,1),\ (2,1),\ (1,1),\ (3,1)\}$. We have that $H^* = 10$:

$$O^* = \{(9,9),\ (8,9),\ (0,0),\ (5,9),\ (0,8),\ (3,9),\ (9,8),\ (0,9)\}$$

Steinberg's packing provides $H = 18.25$. Respective packing is pictured below

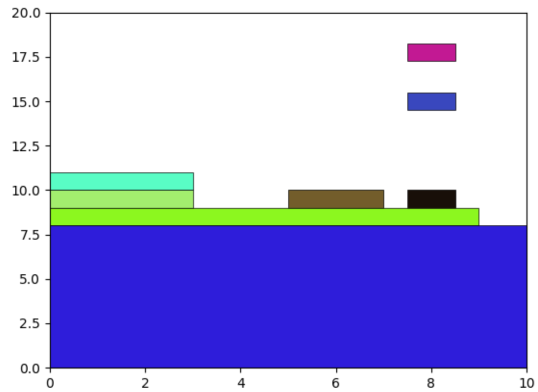

Figure 7: Example 7, original Steinberg algorithm

Modifications of the Steinberg algorithm

As we see in Examples 2–5, 7, Steinberg's algorithm provides "loose" packing. Let us provide two modifications and then analyze them.

Remove all gaps

In Examples 4, 5, 7, we see that the packings involve gaps. Let us remove them. Formally, let

$$S = \{(x, y) \in \mathbb{Q}^2 \mid 0 \leq x \leq W,\ y \geq 0\},\ W \in \mathbb{Q}_+,$$

and

$$L = (w_i, h_i)_{i=1}^n \subset \mathbb{Q}_+^2,\ n \in \mathbb{N}.$$

Let P be a packing w.r.t. L and S. Let $H = H(P)$. A region $G = [0; W] \times [y_1; y_2] \cap Q^2$ is said to be a *gap* if $[y_1; y_2] \subset [0; W] \times [0; H]$ and $\mathrm{Int}(R_i) \cap G = \varnothing$, for every i, $1 \leq i \leq n$.

$C = \{R\}_{i \in I}$, $I \subset \{1, \ldots, n\}$, is said to be a *component* if, for any pair $i, j \in I$, there exists $i_1, \ldots, i_k \in I$ such that

$$\begin{array}{l} [y_i; y_i + h_i] \cap [y_{i_1}; y_{i_1} + h_{i_1}] \neq \varnothing, \\ [y_{i_1}; y_{i_1} + h_{i_1}] \cap [y_{i_2}; y_{i_2} + h_{i_2}] \neq \varnothing, \\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots \\ [y_{i_k}; y_{i_k} + h_{i_k}] \cap [y_j; y_j + h_j] \neq \varnothing \end{array}$$

Let $y_0^C = \min\limits_{i \in I} y_i$ and $y_1^C = \max\limits_{i \in I} y_i + h_i$ be the "bottom" and "top" of a component C.

The "removing gaps" procedure involves the following steps:

1. We initialize the list of lists $\{\{R_1\}, \ldots \{R_n\}\}$.
2. While it is possible, we merge lists if they form a component.
3. After Step 2, we have a list $\{C\}$ of maximal components. We order and re-number this list by increasing the "bottoms". We successively lower the components down so that the "top" of a component coincides with the "bottom" of the next component.

The time complexity of this procedure is not greater than $O(n^2)$. For Examples 4, 5 and 7, we obtain reduced heights $H = 21$, $H = 19$ and $H = 13$ respectively. Let us again picture modified packing.

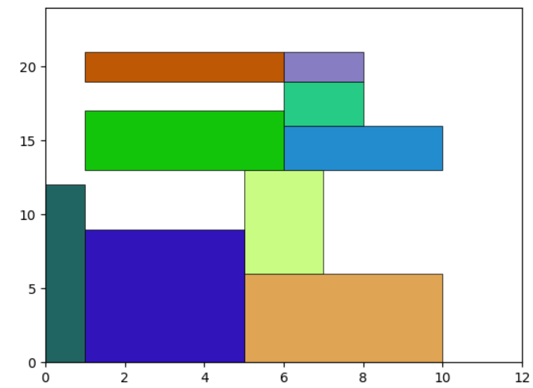

Figure 8: Example 4, gaps removed

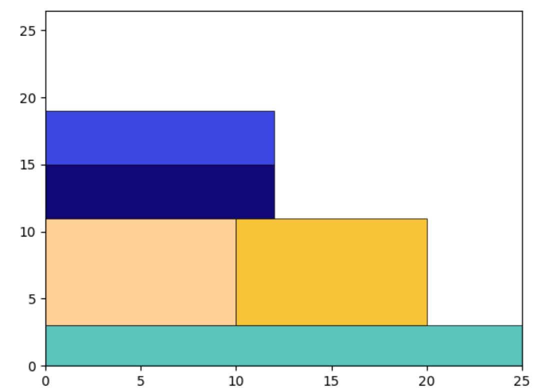

Figure 9: Example 5, gaps removed

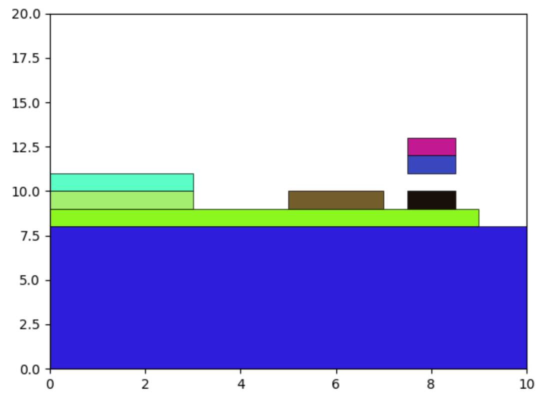

Figure 10: Example 7, gaps removed

Drop hanging rectangles

Previous modification could be improved. For instance, in Example 2, it does nothing. Let us "drop" all hanging rectangles. Formally, let

$$S = \{(x, y) \in \mathbb{Q}^2 \mid 0 \leq x \leq W,\ y \geq 0\},\ W \in \mathbb{Q}_+,$$

and

$$L = \{(w_i, h_i)\}_{i=1}^n \subset \mathbb{Q}_+^2,\ n \in \mathbb{N}.$$

Let P be a packing w.r.t. L and S. For each rectangle R_i, $1 \leq i \leq n$, let us define the downstrip

$$S_i := [x_i; x_i + w_i] \times [0; y_i].$$

R_i is said to be *hanging* if one of the conditions holds

1. $S_i \in \mathrm{Int}(R_j) = \varnothing$, for every $j \neq i$, and $y_i > 0$.

2. There exists $j \neq i$ such that $S_i \cap \mathrm{Int}(R_j) \neq \varnothing$ and

$$y_i > \max\{y_j + h_j \mid j \neq i,\ S_i \cap \mathrm{Int}(R_j) \neq \varnothing\}$$

If R_i is hanging, let us denote

$$\eta_i := \begin{cases} 0 & \text{condition 1 holds} \\ \max\{y_j + h_j \mid j \neq i,\ S_i \cap \mathrm{Int}(R_j) \neq \varnothing\} & \text{otherwise} \end{cases}$$

Therefore, *to drop* R_i means that we replace y_i with η_i.

The "dropping rectangles" procedure is to sequentially drop hanging rectangles as long as possible.

The time complexity of this procedure is not greater than $O(n^2)$.

For Examples 2–5 and 7, it provides the heights: 38, 71, 21, 19 and 12. For Example 7, this modification provides better result than the "removing gaps" modification does. Let us picture an implementation of the modification to Examples 2–4, 7.

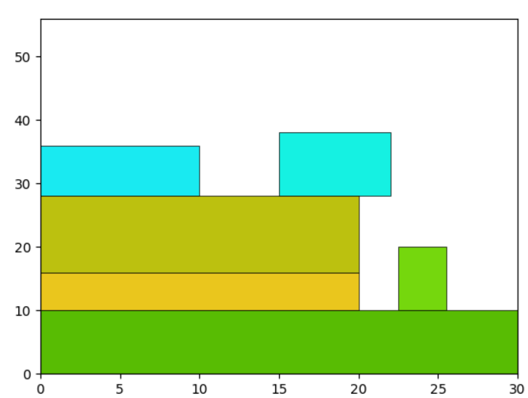

Figure 11: Example 2, rectangles dropped

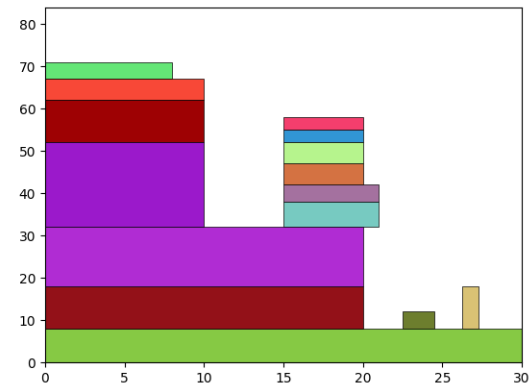

Figure 12: Example 3, rectangles dropped

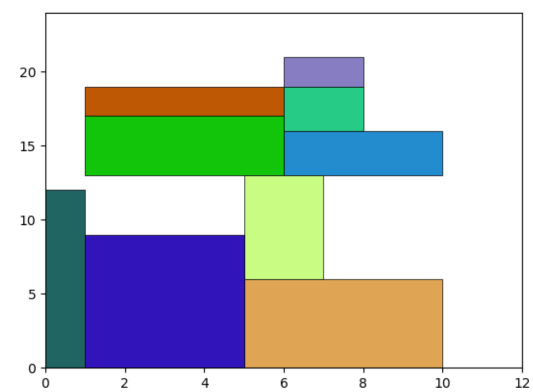

Figure 13: Example 4, rectangles dropped

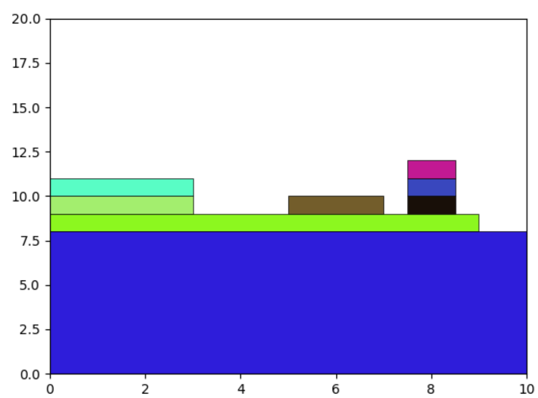

Figure 14: Example 7, rectangles dropped

Tests

Let us compare the Steinberg algorithm, the "removing gaps" and the "dropping hanging rectangles" modifications on random data. We consider two scenarios

1. **Test with the predefined optimal height**
   Let us take random values $10 \leq W$, $H \leq 100$ and $3 \leq n \leq 100$. Then we cut n times (if possible) a rectangle of width W and height H into smaller rectangles. The cuts are also random: we choose a random orientation (horizontal or vertical) and then cut in a random place. Thus, we obtain a strip of width W and a list of rectangles L with predefined optimal height H ("simply connected packing").
2. **Test without the predefined optimal height**
   Let us take random values $3 \leq W \leq 100$ and $3 \leq n \leq 100$. Then we take n times random values $1 \leq w \leq W$ and $1 \leq h \leq 100$. Thus, we obtain a strip of width W and a list of rectangles L with unknown optimal height.

The Steinberg algorithm, the "removing gaps" and the "dropping rectangles" modifications were tested for $N = 10000$ random examples, for each of the scenarios.

We consider the statistics

- Average approximation ratio α_0: let H_K be the height obtained for Example K, and H_K^* be respective optimal height, $1 \leq K \leq N$. Then
$$\alpha_0 := \frac{1}{N} \sum_{K=1}^{N} \frac{H_K}{H_K^*}$$
- Modification success frequency ω and average efficiency δ: let $\mathcal{K}$ be the set of examples for which the modification provides better height H'_K than the original algorithm does (i.e. $H'_K < H_K$). Then
$$\omega = \frac{\text{card}(\mathcal{K})}{N},\ \delta = \frac{1}{\text{card}(\mathcal{K})} \sum_{K \in \mathcal{K}} \frac{H_K}{H'_K}$$
- Average specific time τ: let n_K be the count of rectangles and t_K be the execution time (in seconds) for Example K. Then
$$\tau = \frac{1}{N} \sum_{K=1}^{N} \frac{t_K}{n_K}$$

	α_0	ω	δ	τ
original	1.921	–	–	10^{-5}
removing gaps	1.872	0.556	1.051	10^{-4}
dropping rectangles	1.692	0.815	1.178	10^{-4}

Table 1: Results of scenario 1

	ω	δ	τ
original	–	–	10^{-5}
removing gaps	0.392	1.166	10^{-4}
dropping rectangles	0.461	1.201	10^{-4}

Table 2: Results of scenario 2

Conclusion

Due to the complexity of direct algorithms, we have to use approximating algorithms, which provide non-optimal solutions but are still acceptable. It is worth highlighting the algorithms with an upper estimate for the ratio between heights. We considered in this paper the Steinberg algorithm and proposed two modifications: removing gaps and dropping rectangles. It is noticeable that being more efficient, they both run in an adequate time. When the "simple connected" packing exists, dropping rectangles provides an average ratio of 1.692 and works in $> 80\%$ of cases. In the case of more general packing, it is 1.2 times better than the original algorithm and works in $> 45\%$ of cases.

References

[1] B. S. Baker, E. G. Coffman Jr., R. L. Rivest. Orthogonal Packings in Two Dimensions. *SIAM J. Comput.*, 9(4):846–855, 1980.

[2] A. Lodi, S. Martello, M. Monaci. Two-dimensional packing problems: A survey. *Eur. J. Oper. Res.*, 141:241–252, 2002.

[3] A. Steinberg. A strip-packing algorithm with absolute perfomance bound 2. *SIAM J. Comput.*, 26(2):401–409, 1997.

[4] K. Jansen, M. Rau. Closing the Gap for Pseudo-Polynomial Strip Packing. *27th Annual European Symposium on Algorithms (ESA 2019)*, 144(62):1–14, 2019.