Two-dimensional strip packing problem

Introduction

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

Definition

Let us consider the following vertical strip

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

and the following 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 the L can be duplicated.

A pair $P = (S, R)$, $R = \{R_i\}_{i=1}^n$, is called a *strip packing with respect to 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 with respect to S and L. A packing $P^* \in \mathcal{P}$ is called *optimal*, if it satisfies the following optimum condition

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

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

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

There are several approaches to modeling and solving this problem (see also [2]). We will focus on the Steinberg approximation algorithm and two heuristics.

The Steinberg algorithm

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

$$H_{approx} \leq \alpha H^*,$$

where H_{approx} 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].

In 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

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

Let $S = \sum_{i=1}^n w_i h_i$, $w = \max\{w_i\}$, $h = \max\{h_i\}$.

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 satisfies the following 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 $R = [x_1; x_2] \times [y_1; y_2] \cap \mathbb{Q}^2$ let us provide the following 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 reduction the following reduction process for Q and L. Process then stops if all rectangles are packed. 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 following conditions 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 $R_{m+1}, \dots, R_n$ by decreasing height $h_{m+1} \geq \dots h_n$.

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 by

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

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

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

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

$$2h \geq H$$

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

P3 First, we order and re-number the rectangles of list L by decreasing width $w_1 \geq \dots \geq w_n$.

This procedure can be applied to the (Q, L) if the following conditions holds:

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

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 form 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 $h_1 \geq \dots \geq h_n$.

This procedure can be applied to the (Q, L) if the following conditions holds:

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

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 obtained from P3 by interchanging the horizontal and vertical directions.

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

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

and there exists two different indices i and k such that

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

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 form 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 following conditions holds:

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

and there exists two different indices i and k such that

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

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

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

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

for some index i, $1 \leq i \leq n$.

We place R_i so that

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

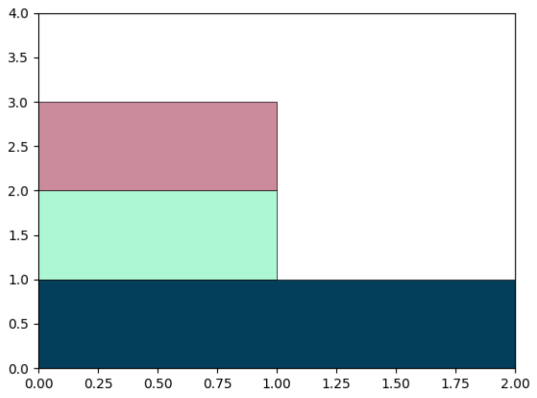
If $n > 1$, we form a new problem (Q', L'), where $L' = L \setminus \{(w_i, h_i)\}$ and Q' is the container such that

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

The Steinberg algorithm can be implemented to run in time $O(\frac{n \log^2 n}{\log \log n})$.

The python implementation (Steinberg.py) is attached. Using this implementation we obtained the solutions for the following examples.

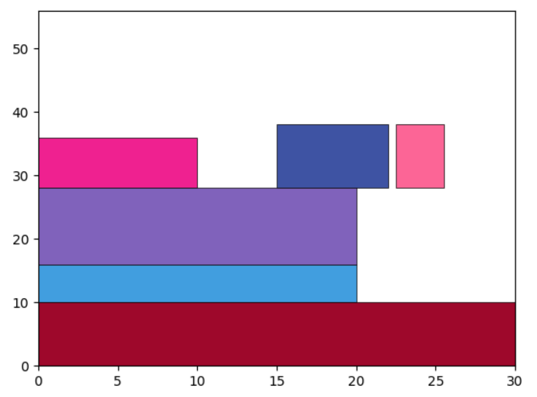
Example 1. Let $W = 2$ and $L = \{(1,1),\ (1,1),\ (2,1)\}$. Obviously, that $H^* = 2$. Steinberg's packing provides $H = 3 > H^*$. Respective packing is pictured below



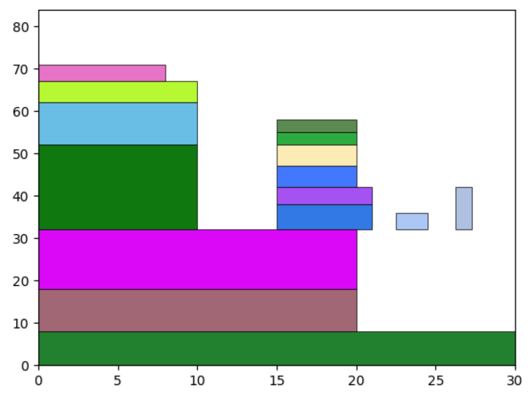
Example 2. Let $W = 30$ and $L = \{(20,6),\ (3,10),\ (7,10),\ (20,12),\ (10,8),\ (30,10)\}$. We have that $H^* = 28$:

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

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



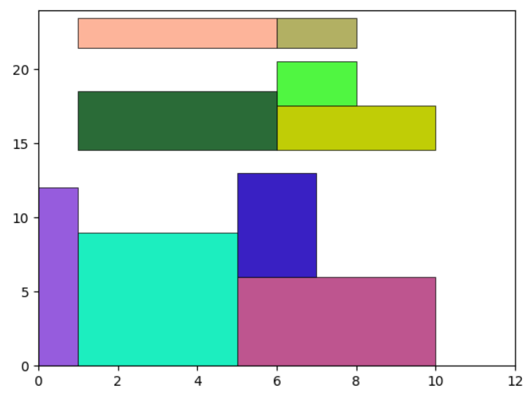
Example 3. Let $W = 30$ and $L = \{(5,3),\ (5,3),\ (2,4),\ (30,8),\ (10,20),\ (20,10),\ (5,5)$
Steinberg's packing provides $H = 71$. Respective packing is pictured below



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 $H^* = 12$:

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

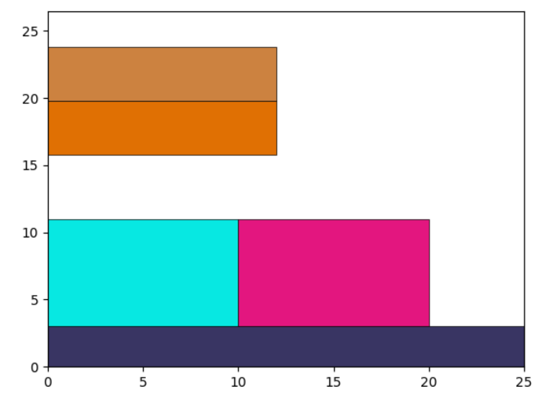
Steinberg's packing provides $H = 23.46$. Respective packing is pictured below



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

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

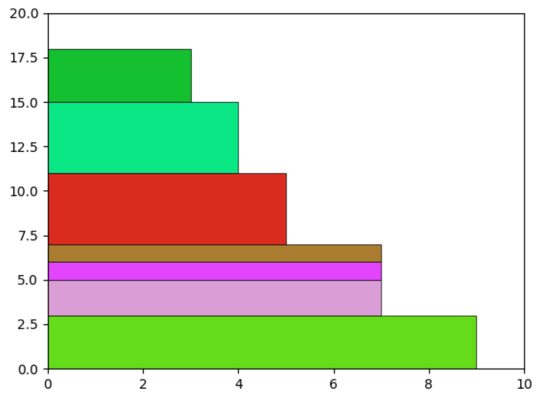
Steinberg's packing provides $H = 23.8$. Respective packing is pictured below



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

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

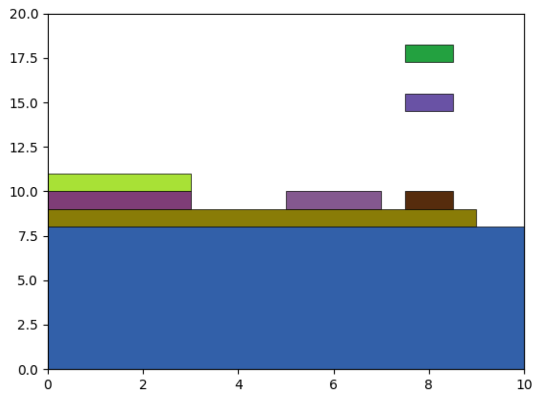
Steinberg's packing provides $H = 18$. Respective packing is pictured below



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



Modifications of the Steinberg algorithm

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

- First, let us remove all "gaps" as in Examples 4, 5, 7.

  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 with respect to L and S. Let $H = H(P)$. A region $G = [0; W] \times [y_1; y_2] \cap \mathbb{Q}^2$ is said to be a *gap*, if $[y_1; y_2] \subset [0; W] \times [0; H]$ and $\text{Int}(R_i) \cap G = \varnothing$, 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

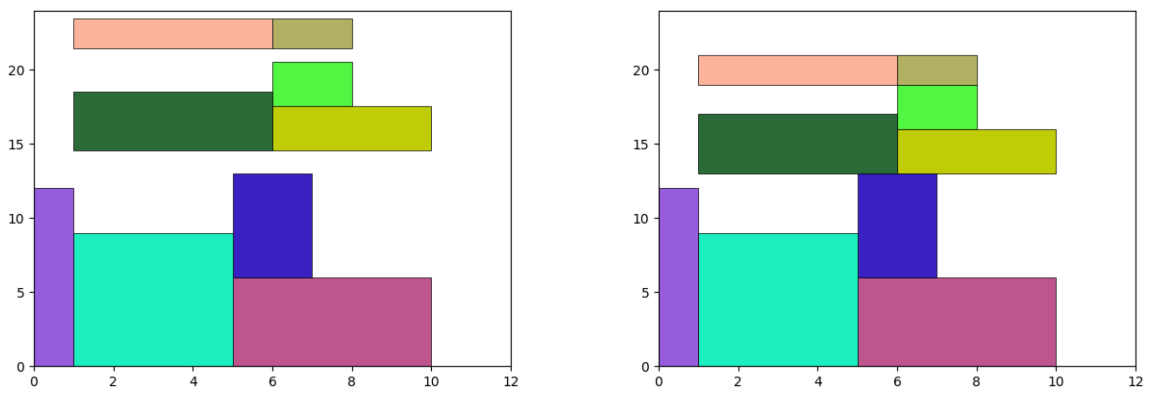
  $[y_i; y_i + h_i] \cap [y_{i_1}; y_{i_1} + h_{i_1}] \neq \varnothing,\ [y_{i_1}; y_{i_1} + h_{i_1}] \cap [y_{i_2}; y_{i_2} + h_{i_2}],\ \ldots,\ [y_{i_k}; y_{i_k} + h_{i_k}] \cap [y_j; y_j + h_j]$

  Let $y_0^C = \min_{i \in I} y_i$ and $y_1^C = \max_{i \in I} y_i + h_i$ — "bottom" and "top" of a component C.

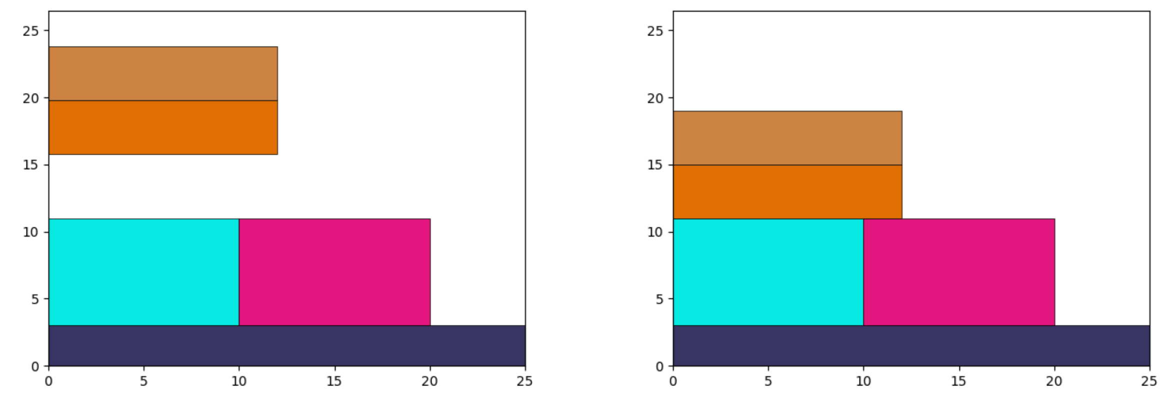
  Removing gaps procedure involves the following steps:

  1. We initialize list of arrays $\{\{R_1\}, \ldots \{R_n\}\}$.
  2. While it is possible we merge arrays, 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 "bottoms". We successively lower the components down so that the "top" of component coincides with the "bottom" of the next component.

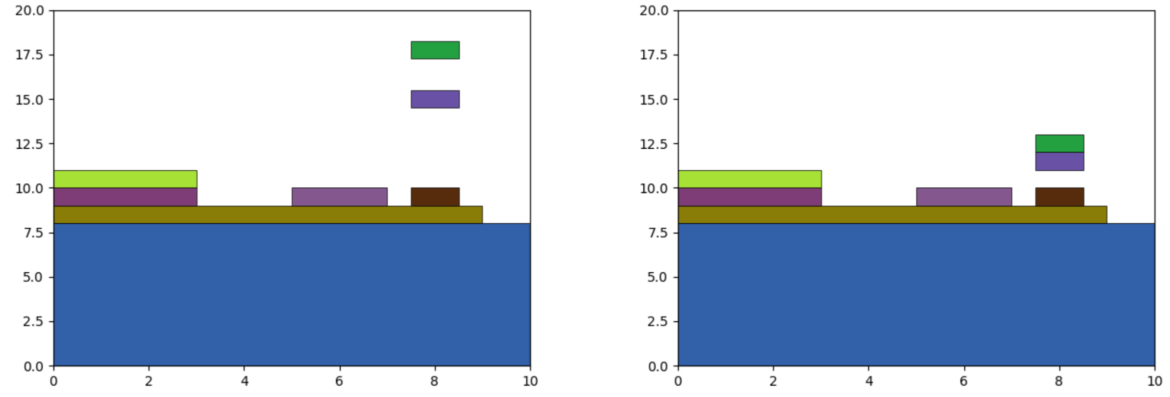
  The time complexity of this procedure is not greater than $O(n^2)$. The python implementation (Steinberg.py) is attached. Using this implementation we obtain reduced heights $H = 21$, $H = 19$ and $H = 13$ for Examples 4, 5 and 7 respectively. Let us again picture Steinberg's packing (left) and modified packing (right)



Example 4



Example 5



Example 7

- This modification could be improved. For instance, in Example 2 it does nothing. Let us "drop" all rectangles.

  Formally, let
  $$S = \{(x, y) \in \mathbb{Q}^2 \mid 0 \le x \le W,\ y \ge 0\}, W \in \mathbb{Q}_+,$$
  and
  $$L = \{(w_i, h_i)\}_{i=1}^n \subset \mathbb{Q}_+^2, n \in \mathbb{N}.$$
  Let P be a packing with respect to L and S. For each rectangle R_i, $1 \le i \le n$, let us define the downstrip
  $$S_i := [x_i; x_i + w_i] \times [0; y_i].$$
  R_i is said to *droppable*, if one of the following conditions holds

  1. $S_i \cap \mathrm{Int}(R_j) = \varnothing$, for every $j \ne i$, and $y_i > 0$.
  2. There exists $j \ne i$ such that $S_i \cap \mathrm{Int}(R_j) \ne \varnothing$ and
  $$y_i > \max\{y_j + h_j \mid j \ne i,\ S_i \cap \mathrm{Int}(R_j) \ne \varnothing\}$$

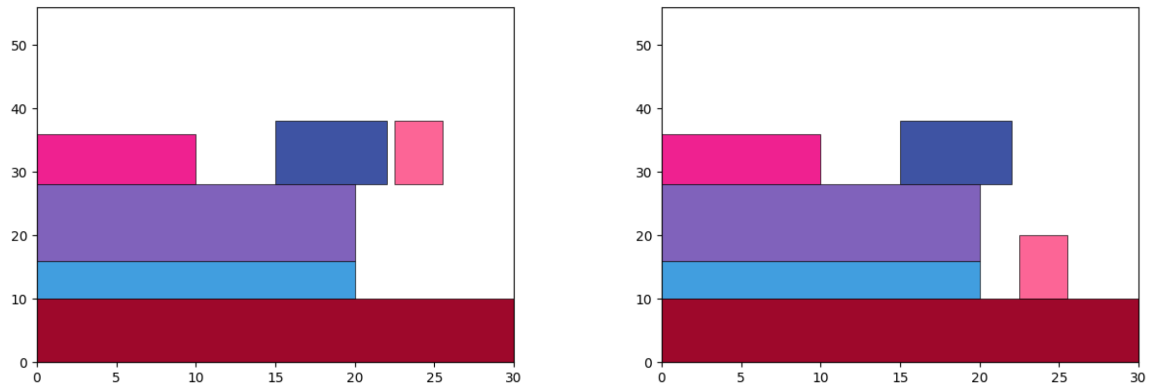
  If R_i is droppable, let us denote
  $$\eta_i := \begin{cases} 0, & \text{condition 1 holds} \\ \max\{y_j + h_j \mid j \ne i,\ S_i \cap \mathrm{Int}(R_j) \ne \varnothing\}, & \text{otherwise} \end{cases}$$
  Therefore, *to drop* R_i means that we change y_i to η_i.

Dropping rectangles procedure is to sequentially drop rectangles as long as possible.

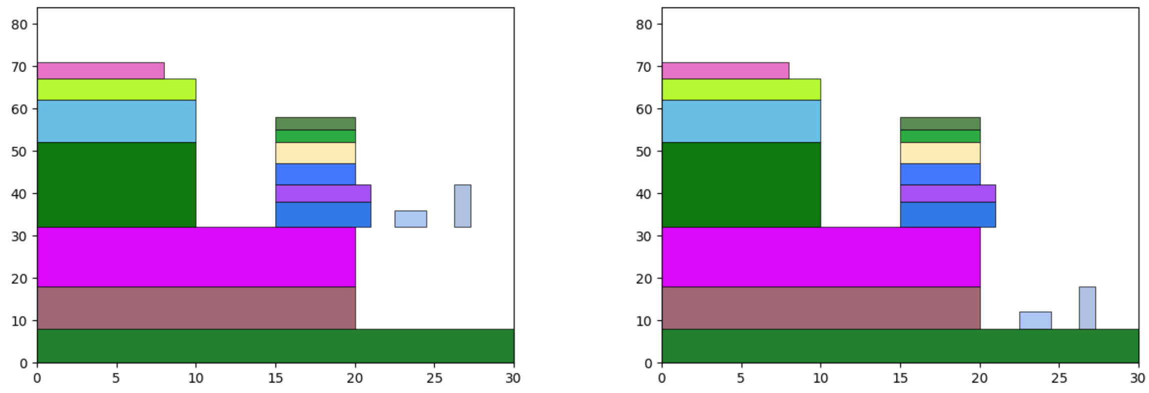
The time complexity of this procedure in not greater than $O(n^2)$. The python implementation (Steinberg.py) is attached.

For Examples 2–5, and 7 it provides the following heights: 38, 71, 21, 19 and 15. For Example 7 this modification provides better result, than removing gaps modification does.

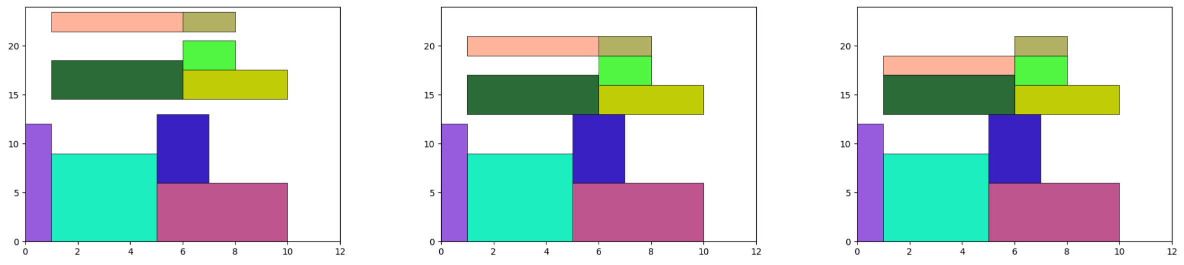
Let us picture implementations of original Steinberg's algorithm, removing gaps modification, dropping rectangles modification to the Examples 2–4, 7.



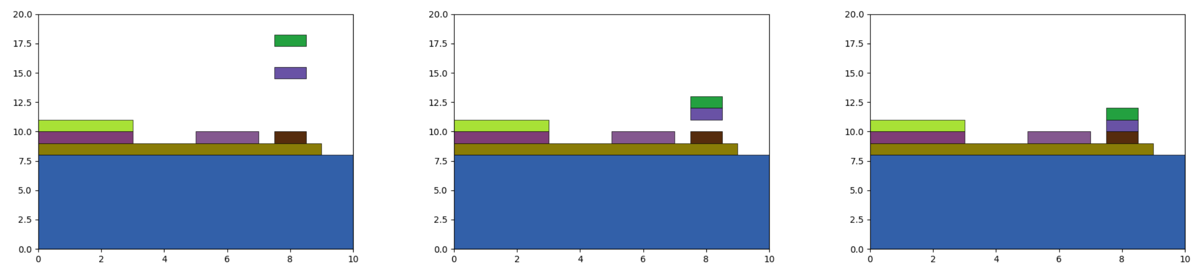
Example 2: original and dropping rectangles



Example 3: original and dropping rectangles



Example 4: original, removing gaps and dropping rectangles


Example 7: original, removing gaps and dropping rectangles

Tests

Let us compare Steinberg's algorithm, removing gaps and dropping rectangles modifications on random data.

1. **Test with predefined optimal height**

   Let us take random values $10 \le W, H \le 100$ and $3 \le n \le 100$. Then we cut n times (if it possible) rectangle of width W and height H into smaller rectangles. Each rectangle is cut randomly: we choose random orientation (horizontal or vertical) and then cut in random place. Thus, we obtain a strip of width W and list of rectangles L with predefined optimal height H ("simply connected packing").

   Steinberg's algorithm, removing gaps and dropping rectangles modifications were tested in $N = 10000$ of such examples. We consider the following statistics

   - Average approximation ratio α_0 for each algorithm: let H_K be an obtained height of Example K, and H_K^* be an optimal height of Example K, $1 \le K \le N$. 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 modification provides better height H'_K than original algorithm does (i.e. $H'_K < H_K$). Then

     $$\omega = \frac{\operatorname{card}(\mathcal{K})}{N}, \quad \delta = \frac{1}{\operatorname{card}(\mathcal{K})} \sum_{K \in \mathcal{K}} \frac{H_K}{H'_K}$$

   - Average specific time τ for each algorithm: let n_K be a count of rectangles and t_K be an execution time (in seconds) of Example K. Then

     $$\tau = \frac{1}{N} \sum_{K=1}^{N} \frac{t_K}{n_K}$$

   We obtained the following results

	α_0	ω	δ	τ
original	1.916	–	–	10^{-5}
removing gaps	1.866	0.554	1.055	10^{-4}
dropping rectangles	1.686	0.813	1.180	10^{-4}

2. **Test without 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 list of rectangles L with unknown optimal height.

   Steinberg's algorithm, removing gaps and dropping rectangles modifications were tested in $N = 10000$ of such examples. We again consider the following statistics ω, δ and τ. We obtained the following results

	ω	δ	τ
original	–	–	10^{-5}
removing gaps	0.385	1.169	10^{-4}
dropping rectangles	0.456	1.200	10^{-4}

Conclusion

Due to the complexity of direct algorithms, we have to use approximate algorithms. Provided with these algorithms solutions are not optimal, but they are useful anyway. It is worth highlighting the algorithms in which there is 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, they both run at the same time, with the other being more efficient. In a case when the "simple connected" packing exists dropping rectangles provides us an average ratio 1.686 and it works in $> 80\%$ cases. In a case of more general packing it 1.2 times better than original algorithm and it works in $> 45\%$ cases.

References

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

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

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

4. K. Jansen, M. Rau *Closing the Gap for Pseudo-Polynomial Strip Packing* // 27th Annual European Symposium on Algorithms (ESA 2019). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. **144**:62 1–14.