Constraint programming: box wrapping

Cristina Luengo Agulló

This document contains an explanation about the strategy used in order to implement the project, and the variables and constraints defined during the implementation.

1 Strategy

We followed three main strategies when specifying variables, constraints and branching:

1. Try to fit the biggest boxes first.
2. Make the boxes take up as much horizontal space as possible.
3. Put the boxes as close to the beginning of the paper (vertically) as possible.

All this statements are defined in the shape of constraints and branchings, which will be explained in the following section.

Moreover, we identified some symmetries that could help prune the search space for the algorithm. We constrained two of such symmetries:

1. It does not matter in which horizontal half of the paper we place the first box. Therefore, we arbitrarily choose to place it in the left horizontal half of the paper.
2. Given two boxes A and B, if A contains B then B contains A as well.

2 Variables and constraints

We defined the following variables for the problem:

x_{tl}: Array of top-left horizontal coordinates for all the boxes.

y_{tl}: Array of top-left vertical coordinates for all the boxes.

x_{br}: Array of bottom-right horizontal coordinates for all the boxes.

y_{br}: Array of bottom-right vertical coordinates for all the boxes.

$length$: Length of the paper roll needed to wrap the boxes.

Both x_{tl} and y_{tl} define the top-left corner of each box, and x_{br} and y_{br} the bottom right ones. First of all, in order to apply strategy 1, we sorted the boxes in decreasing order by size.

Given B a set of box identifiers and W the width of the paper, $width_b$ represents the width of a box (either rotated or not), $height_b$ the height of a box (either rotated or not) and $area(b)$ the total area or size of a box. Considering these constants and variables, the following constraints are posted:

(1) $\quad x_{tl_0} \leq \frac{1}{2} \times (W - width_0)$

(2) $\quad \forall b \in B, x_{tl_b} \leq W - width_b$

(3) $\quad \forall b \in B, x_{tl_b} \leq x_{br_b}$

(4) $\quad \forall b \in B, y_{tl_b} \leq y_{br_b}$

(5) $\quad \forall b \in B, x_{br_b} = x_{tl_b} + width_b - 1$

(6) $\quad \forall b \in B, y_{br_b} = y_{tl_b} + height_b - 1$

if $\text{area}(b_1) = \text{area}(b_2)$

(7) $\quad \forall b_1 \in B, b_2 \in B, b_2 > b_1, x_{tl_{b_2}} > x_{br_{b_1}}$ or $y_{tl_{b_2}} > y_{br_{b_1}}$

if $\text{area}(b_1) \neq \text{area}(b_2)$

(8) $\quad \forall b_1 \in B, b_2 \in B, b_2 > b_1, x_{tl_{b_2}} > x_{br_{b_1}}$ or $y_{tl_{b_2}} > y_{br_{b_1}}$ or $x_{br_{b_2}} < x_{tl_{b_1}}$ or $y_{br_{b_2}} < y_{tl_{b_1}}$

(9) $\quad max(y_{br}) + 1 = length$

On one hand, constraint (1) is used in order to break symmetry 1, since it states that the top-left horizontal coordinate of the first box falls into the left half of the paper. Moreover, for all other boxes, constraint (2) states that their top left horizontal coordinate will always be smaller than the total width of the paper minus their width.

On the other hand, constraints (3), (4), (5) and (6) define the relationship between top-let and bottom-right corner coordinates of all boxes. These depend on the values that the width and height for every box take, since they can be rotated.

Constraints (7) and (8) state that boxes can not overlap when placed in the paper. Particularly, constraint (7) is posted for boxes that have the same size. Since we try to maximize the horizontal space occupied and minimize the length of the total paper roll, when a box is placed and another box of the same size has to be placed, such box will be placed either at the right of the first box or under it. This is based on the search strategy that we follow.

Moreover, the fact that constraints (7) and (8) are posted between boxes only once breaks symmetry 2.

Finally, constraint (9) states that the length of the paper is equal to the maximum bottom-right vertical coordinate.

Regarding the branching strategies followed, we posted three branchers for variables in x_{tl}, y_{tl} and y_{br}. All branchers follow a *first unassigned* strategy to choose the variables, since we want bigger boxes to be assigned first and we sort them beforehand.

Following strategy 2, we put the brancher for x_{tl} first, and we defined the strategy to choose new values such that boxes are placed as close to each other as possible and occupy the maximum horizontal space they can. Moreover, branchers for y_{tl} and y_{br} follow strategy 3 when choosing values for the variables, since they choose the minimum value of their domain.