Combinatorial Decision Making and Optimization

Project Report

VLSI Design

Gee Jun Hui Leonidas Yunani
Anna Fabris

leonidas.gee@studio.unibo.it
anna.fabris@studio.unibo.it

Academic Year: 2020 / 2021

1 SAT

1.1 Considerations on the Height

The problem was solved assuming that no gaps are allowed between the circuits, the plate height is thus obtained by taking the sum of the circuits' areas divided by the plate width.

To solve cases for which this does not apply, the code could be modified by adding a for loop that tries to solve the problem with increasing heights, starting from the calculated height.

1.2 Variable Encoding

Each circuit corresponds to a boolean variable indicating if the given circuit is or not located in that specific position. Each position in the plate is indicated by lists of boolean variables to indicate the horizontal and vertical coordinates.

This creates a grid with true values only where the given circuit is placed ($gird[i][j][c]$ will be true if some parts of circuit c occupies position (i, j), false otherwise).

1.3 Constraints

For the following constraints $width$ and $height$ will be used to indicate the width and the height of the plate while c_width and c_height will be used to indicate the width and the height of a given circuit. To indicate the list of all circuits C will be used, $grid$ is the variable that is used to indicate the model.

- **At most one** circuit can occupy each place of the grid.

$$\bigwedge_{i=1}^{width} \bigwedge_{j=1}^{height} \bigwedge_{0<c1<c2\leq C} \neg(grid_{i,j,c1} \wedge grid_{i,j,c2})$$

- Every circuit must be placed in its entire form in one of its possible positions. The possible positions of each circuit are indicated with $positions_n$, and **at least one** of them must be placed in the plate.

$$positions_{i,j} = \bigwedge_{c1=1}^{c} \bigwedge_{ii=i}^{i+c1\_width} \bigwedge_{jj=j}^{j+c1\_height} grid_{c1,ii,jj},$$

$$\bigwedge_{c1=1}^{c} \bigwedge_{i=1}^{width_{temp}} \bigwedge_{j=1}^{height_{temp}} \bigvee positions_{i,j},$$

$$width_{temp} = width - c1\_width + 1,$$

$$height_{temp} = height - c1\_height + 1$$

In practice there's no need to check that each circuit is in **exactly one** of its possible positions, because it would always lead to a worse solution.

1.4 Rotation

A simple idea to allow the model the possibility of rotation is doubling each circuit swapping its width and height coordinates and adding a constraints to avoid placing the same circuit twice.

A possibly better solution could consist in a list of boolean variables that is the same length as the number of circuits and keeps track of whether the circuit is rotated or not. Given this list, it's easy to find the actual width and height of each circuit and using those in the constraints.

1.5 Final Considerations

The current model was able to solve 20/40 instances with the time constraint of 300 seconds.

To improve this results another enconding of the variables was also tested: the circuits were represented with their left-bottom corners in the plate grid (as it was then used in SMT), but it had worse performance than this current model explained above.

Changing from the exactly one to the at least one constraint in the second constrain explained above helped improving the performances.

Test concerning the implied constraints seemed to generally worsen the solving time (although in some specif case they do help), so they were removed from the code and the model.