CPS - SAT

Gonzalo Solera

June 2020

1 Optimal solution bounds

I applied the same preprocessing step as in the previous assignment. I refer the reader to its respective document.

2 Optimizer

The program encodes and runs the SAT solver for a fixed and guessed L inside the mentioned bounds. It tries different values of L until it finds the optimal L for which encoding has a valid assignment. This search of an optimal L can be done in different ways. For example, it could start with a lower bound of L and try with increasing values of L until it finds a valid assignment. It could also perform a binary search between the lower and upper bound of L. However, these two strategies might require a large amount of calls to the SAT solver with an unsatisfiable formula. Such calls might require more execution time since in general, it's easier to find an existing satisfiable assignment than to conclude there is no such assignment.

This is why I do a linear search starting from the upper bound of L. This way, I will only call the SAT solver with an unsatisfiable formula at most once, although it might call it more times than strictly necessary.

3 Model

I have modeled the problem with 1 boolean variable d_i for each box i to model its orientation. I also have used W variables $p_i^1, p_i^2, \ldots, p_i^W$ to encode the x-coordinate and L variables $q_i^1, q_i^2, \ldots, p_i^L$ to encode the y-coordinate of each box i. To obtain such encoding, it was necessary to enforce that exactly one of the variables for each axis is set to true. This is accomplished with a trivial ALO constraint and a naive AMO constraint, using $O(W^2)$ and $O(L^2)$ clauses and no auxiliary variables. The AMO constraints could be removed by using a simple post-processing step to choose just one value for each coordinate (such that the combination of coordinates is valid), but for this kind of instances the overhead wasn't considerable and the naive encoding is good enough.

In order to encode the overlapping constraints, I have used WL auxiliary variables for each box i in the form of a bitmap. These variables are set to true if the box i is over them. For that, I needed to encode the implication that for any orientation and position of each box, the corresponding variables from the bitmap are set to true. Hence, this adds $O(NWL)$ variables and $O(NW^2L^2)$ clauses. Note that this last bound is not tight, since it assumes that each box is as big as the available paper, which would decrease the total number of boxes.

I also added constraints to enforce that the position of each box is inside the given bounds taking into account their orientation.

Lastly, to encode the overlapping constraints I have used $O(N^2WL)$ clauses to enforce that in any position of the mentioned bitmap there is only one variable set to true. This could be optimised with a more efficient encoding of an AMO constraint of N variables for each of the WL positions of the bitmap, although the extra effort wasn't worth it, since the asymptotic cost was already dominated by other constraints.

4 Results

My proposed model and optimizer of the problem offers good results. From the 108 given instances, it successfully finds the optimal solution of 106 instances within 60 seconds each. It also obtains the same result with a much sorter timeout. If the search of the optimal L was reversed as explained previously, there would be one extra timeout, supporting my assumptions.