ilkercan kaya

CSP Representation:

Some terminology:

A block is a group of white boxes that is betweek two black boxes, either row of column.

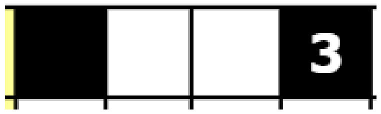

Figure 1:Row block (The two white boxes are called a block)

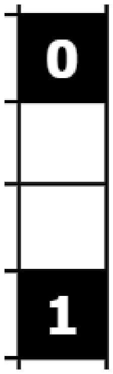

Figure 2: Column block (White boxes are called a block)

We will have 3 Constraints.

Constraint One (White Box Constraint)

Starting from each white box, there has to at least 1 light source that is on the way to closest Column block and a Row block if they exist:

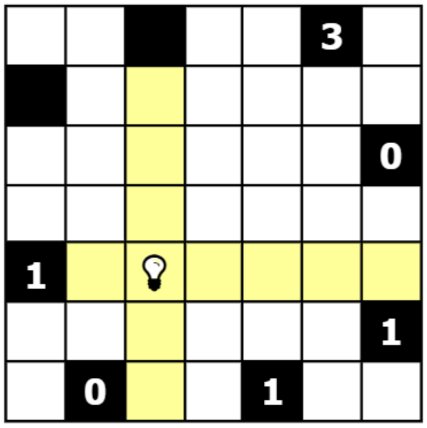

Figure 3: Illustration (Consider Light Bulb As A White Bulb)

The light bulb in Figure 3 is put there to illusturate the white box line to each closest block and row block if they exist. For the sake of illustration consider the light bulb as a white block. There has to at least one light bulb along the box x-axis and y-axis to light the white box. We do this with *MinSumConstraint.*

Constraint Two (Block Constraint)

For each row block and column blocks there has to be at maximum one light bulb among each of these blocks. Meaning a row block can have 0 or 1 light bulbs among it and same goes for the column block. This contraint combined with *Constraint One* makes sure that there is only one light bulb among the shared rows and columns so that none of the light bulbs intersects with each other. We do this with *MaxSumConstraint.*

Constraint Three (Black Box Constraint)

There has to be light bulbs around the black boxes that has a number in them. We check the avaible neighbours and assign a *ExactSumConstraint* to the sum of the neighbours to obtain this constraint.

A* Or CSP

For this problem, CSP would be more accurate and easy way of solving this search. The reason is that due to the nature of Akari, it is easy to represent the goal as a set of constraints whereas the A* would require us to have a goal state which we cant determine by a state expression. We would need to do several function calls to make sure that the state we are in is satisfied to be a goal which would require a huge amount of time due to do since the problem is complex enough. It would also be hard to declare a heuristic according to our needs since a heuristic needs to be admissible we cant really know the cost to a goal step. Hence, we can overestimate.

Examples Solved by the algorithm: