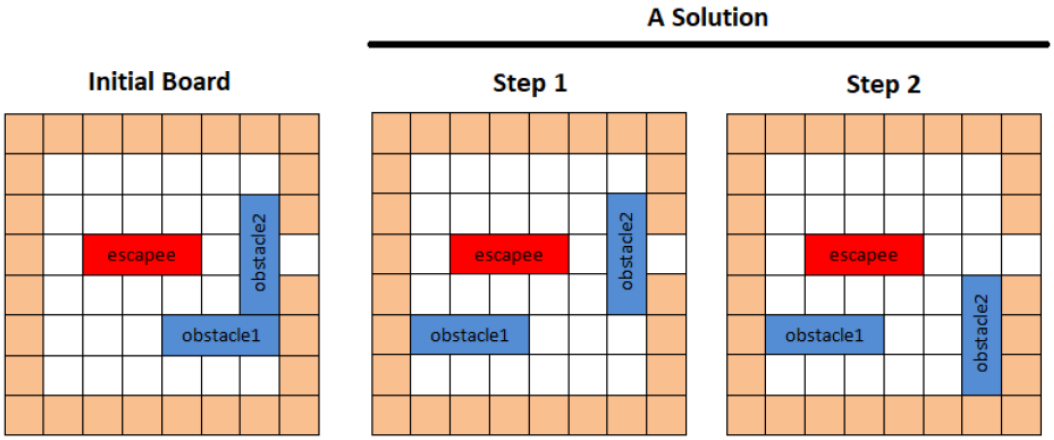

Figure 1: A simple example

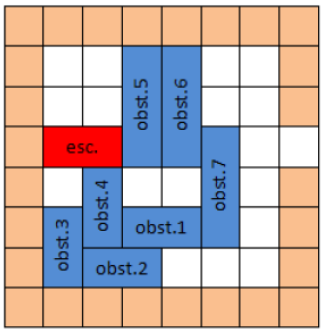

Figure 2: A more complex example

The idea of the default problem is depicted in Figure 1 with a board containing the escapee (In red) and the other two obstacles (in blue). The board has surrounding wall cells (in light brick color) and only one door (in white) so that the escapee object can escape. The wall cells are not changeable, and the empty cells (in white) can be occupied by the objects. Each object can only move along its long-axis any distance without collision with the wall an the other objects. e.g. the escapee can move left or right, obstacle1 can move left one, two or three cells and then move can move right, obstacle3 can move up and move down when obstacle 1 had moved left.
An input board configuration is on the left and a solution for the escapee object to escape from the board including two steps on the middle and the right. A more complex board configuration is depicted in Figure 2.
Figure 3 demonstrates the encoding method applied for the example shown in Figure 1. The initial board with the escapee and other obstacles are all declared in a configuration file (.pl).

config0.pl

```
% 0 for empty cell, 1 for wall, 2 for cells of the object to be escaped,
% other values for cells of the obstacles
% each object only move along its spine/long-axis
board([[1,1,1,1,1,1,1,1],
       [1,0,0,0,0,0,0,1],
       [1,0,0,0,0,0,3,1],
       [1,0,2,2,2,0,3,0],
       [1,0,0,0,0,0,3,1],
       [1,0,0,0,4,4,4,1],
       [1,0,0,0,0,0,0,1],
       [1,1,1,1,1,1,1,1]]).

% [CV, L, D] : [cell_value, length, 0-for_horizon/1-for_vertical movement]
escapee([2, 3, 0]).
obstacles([[3, 3, 1], [4, 3, 0]]).
```

Figure 3: Encoding