Assignment

Urban planning requires careful placement and distribution of commercial and residential lots. Too many commercial lots in one area leave no room for residential shoppers. Conversely, too many residential lots in one area leave no room for shops or restaurants.

The 5x5 grid depicted in the figure bellow shows a sample configuration of residential and commercial lots. Your job is to place a mix of residential and commercial lots to maximize the quality of the layout. All lots are occupied, either residential or commercial (for each row/column: number_of_residential + number_of_commercial = N, where N is the length of the row/column). The quality of the layout is determined by a points system, where all rows and columns adds points based on the number of residential/commercial lots. In the example picture bellow points are awarded as follows (the point distribution to use in the test cases is part of the data file):

- Any column or row that has 5 Residential lots = +5 points (0 Commercial lots)
- Any column or row that has 4 Residential lots = +4 points (1 Commercial lots)
- Any column or row that has 3 Residential lots = +3 points (2 Commercial lots)
- Any column or row that has 2 Residential lots = -3 points (3 Commercial lots)
- Any column or row that has 1 Residential lots = -4 points (4 Commercial lots)
- Any column or row that has 0 Residential lots = -5 points (5 Commercial lots)

The layout displayed in the figure bellow has a total of 9 points:

- Points for each column, from left to right = -3, -5, +3, +4, +3
- Points for each row, from top to bottom = +3, +3, +3, +3, -5.

Task 1: Write a CP-program to solve this optimisation problem. The solution must be found using a constraint programming solver based on DFS-search (JaCoP or Gecode, do not use COIN-BC or Chuffed). The problem contains many equivalent symmetrical solution and to make search efficient you need to add symmetry breaking constraints. The symmetry breaking constraints may only remove symmetries, at least one symmetric solutions must still be valid. For example, the constraint "lot[1,1]=commercial" will remove some symmetries, but when "n_commercial = 0" it will invalidate the only solution. Write a comment in your program explaining how you break symmetries. Your program must be flexible, all data must be parameters, i.e. the data from the files linked bellow must be accessed using the parameter names and may not be hard coded in your program.