

Intelligent Systems

Assignment 6. CSP Formulation



Exercise description

The assignment involves to construct an agent that can solve a specific Constraint Satisfaction Problem (CSP). For this, you should formulate and implement your formulation as Python code to solve a specified Puzzle game.

Team members

Write the student id, name, and campus of each member in a different line.

1: Carlos Hinojosa A01137566, Campus Monterrey

2: Eider Diaz A00828174, Campus Monterrey

3: Miguel Cortes A01270966, Campus Monterrey

Skyscraper Puzzle

Skyscraper puzzles combine the row and column constraints of Sudoku with external clue values that re-imagine each row or column of numbers as a road full of skyscrapers of varying height. Higher numbers represent higher buildings. To understand Skyscraper puzzles, you must imagine that each value you place into the grid represents a skyscraper of that number of floors. So a 1 is a 1-floor skyscraper, while a 4 is a 4floor skyscraper.

Now imagine that you go and stand outside the grid where one of the clue numbers is and look back into the grid. That clue number tells you how many skyscrapers you can see from that point, looking only along the row or column where the clue is, and from the point of view of the clue. Taller buildings always obscure lower buildings, so in other words higher numbers always conceal lower numbers. In the image below, you can see two buildings if they are standing at the left end of the line and three buildings if they are standing at the right end.

Every Skyscraper puzzle only ever has one possible solution, and it can always be reached via reasonable logical deduction. In other words, guessing is never required.

CSP Formulation

Variables:

Each cell on the grid (apart from the most outer ones, which indicate the number of buildings a bystander would be able to see from the spot). For example, in a 4x4 grid:

$\{Building_{11}, Building_{12}, Building_{13}, Building_{14}, Building_{21}, Building_{22}, \dots, Building_{44}\}$

An alternative way to formulate the variables on this problem is to consider each row and column a separate variable. Then, you would be dealing with lists of numbers as their possible values.

Domains:

The domain for each variable is the height of the building that might be placed there. It ranges from 1, the minimum height, to the total number of buildings in each row or column. In a 4x4 grid the domain of each variable would be as follows:

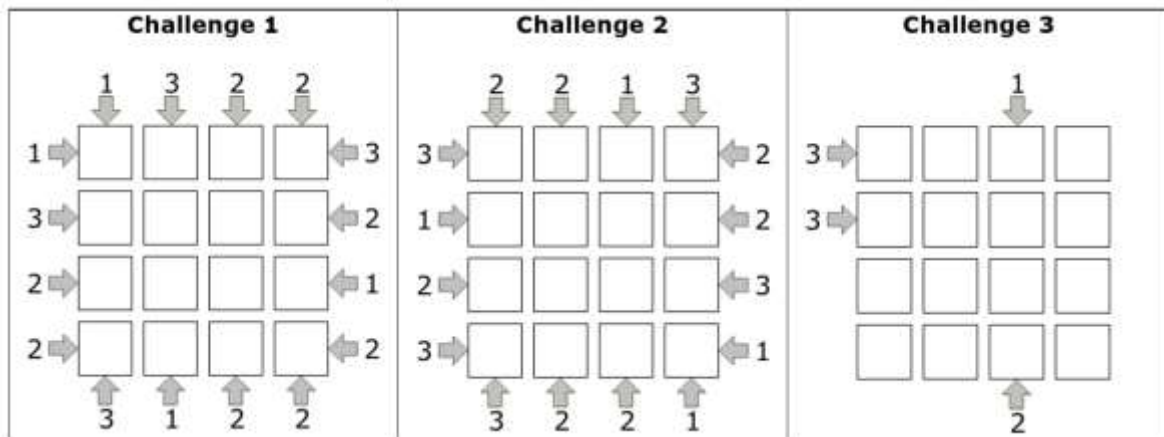
$\{1, 2, 3, 4\}$

On the other hand, considering each row and column a different variable, the domains change to be all possible permutation of the individual values the cells can have. In a 4x4 example, with 4 possible values for each cell, every row and column would have a domain of 256 possible values.

Constraints:

- The same building height (digit) can not appear more than once in a row.
- The same building height (digit) can not appear more than once in a column.
- Each column has an upper value N, which states the amount of buildings that should be visible to a bystander on the spot. The column should have an increasing sequence of N elements from top to bottom.
- Each column has a lower value N, which states the amount of buildings that should be visible to a bystander on the spot. The column should have an increasing sequence of N elements from the bottom to the top.
- Each row has a left value N, which states the amount of buildings that should be visible to a bystander on the spot. The row should have an increasing sequence of N elements from left to right.
- Each row has a right value N, which states the amount of buildings that should be visible to a bystander on the spot. The row should have an increasing sequence of N elements from right to left.

Illustrate this formulation for the 2 challenges below



Challenge 1 - Formulation

Constraints:

Clues_up: {1,3,2,2} Clues_down = {3,1,2,2}

Clues_left = {1,3,2,2} Clues_right = {3,2,1,2}

Domains and variables:

Domains for each variable (cell on the grid) after taking into consideration the restrictions from the external 'clue' values are shown below:

\	1	3	2	2	/
1	4	1,2	3	1,2	3
3	1,2	1,2,3	4	1,2,3	2
2	1,2,3	1,2,3	1,2	4	1
2	1,2,3	4	1,2	1,2,3	2
/	3	1	2	2	\

Solution

	1	3	2	2	
1	4	1	3	2	3
3	2	3	4	1	2
2	3	2	1	4	1
2	1	4	2	3	2
	3	1	2	2	

Challenge 2 – Formulation

Constraints:

Clues_up: {2,2,1,3} Clues_down = {3,2,2,1}

Clues_left = {3,1,2,3} Clues_right = {2,2,3,1}

Domains and variables:

Domains for each variable (cell on the grid) after taking into consideration the restrictions from the external 'clue' values are shown below:

\	2	2	1	3	/
3	1,2,3	1,2,3	4	1,2	2
1	4	1,2	1,2	3	2
2	3	4	1,2	1,2	3
3	1,2	1,2	3	4	1
/	3	2	2	1	\

Solution

	2	2	1	3	
3	1	3	4	2	2
1	4	2	1	3	2
2	3	4	2	1	3
3	2	1	3	4	1
	3	2	2	1	

Challenge 3 - Formulation

Constraints:

Clues_up: {?,?,1,?} Clues_down = {?,?,2,?}

Clues_left = {3,3,?,?} Clues_right = {?,?,?,?}

Domains and variables:

Domains for each variable (cell on the grid) after taking into consideration the restrictions from the external 'clue' values are shown below:

\	?	?	1	?	/
3	1,2	1,2,3	4	1,2,3	?
3	3,4	1,2,3,4	1,2	1,2,3,4	?
?	3,4	1,2,3,4	1,2	1,2,3,4	?
?	1,2	1,2,4	3	1,2,4	?
/	?	?	2	?	\

Solution

```
      ?   ?   1   ?
3 | 1 | 2 | 4 | 3 | ?
3 | 2 | 3 | 1 | 4 | ?
? | 3 | 4 | 2 | 1 | ?
? | 4 | 1 | 3 | 2 | ?
      ?   ?   2   ?
```

-