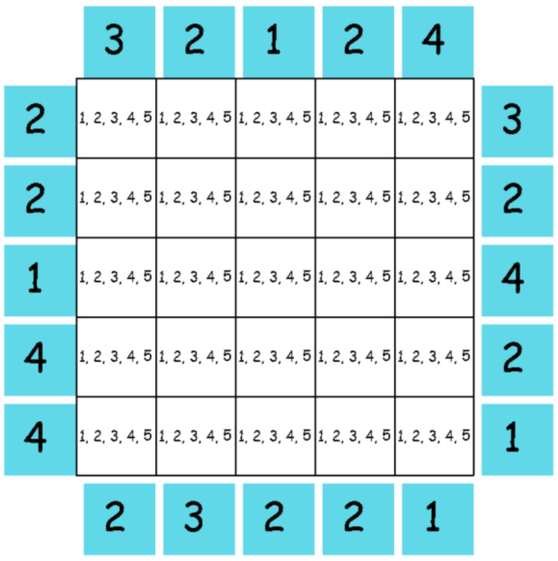
Skyscraper Puzzle Solving Using Domain Contraction Techniques

By Vinay Detani **(B18CSE061)**

INTRODUCTION

Skyscraper is a unique Japanese puzzle game which introduced in 1992 world Puzzle championship. It is Somehow similar to Sudoku and different also. Here we have an NxN grid, where each cell has to be fill with numbers 1 to N such that each row and column have every number from 1 to N, just like Sudoku. This number N represent the height of a building. But difference is the clues that is at the both side of each row and column. Clue is a number from [1,2,3….N] which tells us the
maximum increasing sequence starts from the nearest cell of the clue.



	3	2	1	2	4	
2	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	3
2	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	2
1	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	4
4	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	2
4	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1
	2	3	2	2	1	

The numbers in the light-blue color boxes are clues. The sequence in the first row is has to such that when we saw from the left, there should be exactly 2 building visible and from the right exactly 3. A building is visible only if building before it in the side of view are smaller than it.

Example:

2	**1**	**2**	**5**	**4**	**3**	3

PROBLEM DEFINATION

The Skyscraper problem is classified under CSP (constraint satisfaction problem).
Formal definition: *Variables* – Cell values (Cij)
Domain - [1,2,3 … N]
Clues - H(row/col, side(left/right/top/bottom))

row/col belongs to [1,2,3 …. N]

1<=H<=N or may be Zero(0) no hint/clue

Constraints –

1. 1<= Cij <=N for every i(row) and j(column).
2. Cij != Ckl if, i == k or j == l.
3. Each clue should be satisfied.

Goal – Each cell is filled satisfying the constraints.

As it is CSP, it can be solved by DFS, BFS or backtracking (with filtering or without filtering). The thing matters, how efficiently we can solve this problem using minimum computational resources. For 4 x 4 grid 880 different puzzle possible. For N equal to 5 it is ~ 10^ 7. For N = 6, it is 10^32. So, for large values of N, it is not possible to solve it in valid computational time with DFS or BFS. For backtracking also worst case leads to high resources. Hence, we used a technique that based on domain contraction and constraint propagation to solve the puzzle.

ALGORITHM

This technique, try to decrease the possible domain size of each cell to 1, using some steps that leads to one possible solution. We thought of like a normal human, how he/she approach the problem and mimic it for machines. Let we have a set(domain) containing 1 to N numbers for cell. In subsequent steps, we omit the entries that cannot be possible.

1. **CLUE ELIMINATION** – Here we eliminate the numbers from the domain of a cell that cannot be satisfy the clues in any case. Take 5x5 puzzle.

Example-

Distance	**1**	**2**	**3**	**4**	**5**
3	**4 or 5**	**5**			

If clue is 3. Then in the cell at a distance 1, if we put 5. Then maximum building visible will be one as 5 is the tallest building and hide any buildings behind it, when viewed from left side. For 4, maximum two building will be visible 4 and 5 from **3.** Accordingly, we can remove them from the cell domain.

3	{1,2,3}	{1,2,3,4}	{1,2,3,4,5}	{1,2,3,4,5}	{1,2,3,4,5}

For Clue C , C >1 and C<N(5). We can remove the entries from a cell that is lies in the range [(N+d+1-C),N], where d is the distance of the cell from the clue.

For Clue C=1. The cell at distance 1, has N(5) has only possibility. So we can directly remove all other entries from the domain.

Distance	**1**	**2**	**3**	**4**	**5**
1	{ 5 }	{1,2,3,4,5}	{1,2,3,4,5}	{1,2,3,4,5}	{1,2,3,4,5}

For Clue C=5. All the five building are visible, so only order possible will be 1,2,3,4,5 for that row.

This will resolve the whole row at once and ease the task.

5	{1}	{2}	{3}	{4}	{5}

We perform this clue elimination for every clue. At the end, we will have cells which are resolved or domain size is reduced.

2. **CONSTRAINT PROPAGATION -** In the first step we got some cells resolved with domain size of 1. So, these cells can only have one possible value, so for other cells in its row and column cannot have this value. This can reduce the domain for other cells.

Distance	**1**	**2**	**3**	**4**	**5**
1	{5}	{1,2,3,4,5}	{1,2,3,4,5}	{1,2,3,4,5}	{1,2,3,4,5}

Cell 1 has only 5 in its domain, so 5 can be removed from the domain of the other cells in the row and column.

For the row, after constraint propagation, it will look like:

Distance	**1**	**2**	**3**	**4**	**5**
1	{5}	{1,2,3,4}	{1,2,3,4}	{1,2,3,4}	{1,2,3,4}

.

Constraint propagation will be performed for every cell that has domain size 1. If while propagation, a cell domain size reduced to one, we call this method of constraint propagation for this cell also.

3. **ELIMINATION ASSIGNMENT –** In this step we check for each row and column, for a situation where a value is only present in the domain of **one** Cell. So, this cell(unique), must have this value. We assigned it or say remove all other entries from the domain of unique cell and again run the step 2 for this cell.

Example-

{1, 2, 4}	**{1,2,3}**	**{2,5}**	**{1,2,3}**	**{1,2,3}**

Here, Entry 4 and 5 are present only in the domain of a single cell. So, we can omit other entries from the domain.

{4}	**{1,2,3}**	**{5}**	**{1,2,3}**	**{1,2,3}**

After this, we run step 2(Constraint propagation) for the column of cell-1.

4. **SEQUENCE ELIMINATION –** As the same suggest, here omit the sequence that doesn't satisfying the clues. It is most important step of the algorithm and whole complexity depends on it. Also, it is very powerful for getting close to solution.

2	{4}	{1,2,3}	{5}	{1,2,3}	{1,2,3}	3

Possible sequence -	4, 1, 5, 2, 3
	4, 1, 5, 3, 2
	4, 2, 5, 1, 3
	4, 2, 5, 3, 1
	4, 3, 5, 1, 2
	4, 3, 5, 2, 1

Red sequence are the ones that doesn't satisfy the clue 3, means when we look from the right side, I will not able to se exactly 3 buildings.
4, 1, 5, 2, 3 - 3 and 5 buildings(only 2) will be visible and clue is 3.
So sequence domain shrinked to

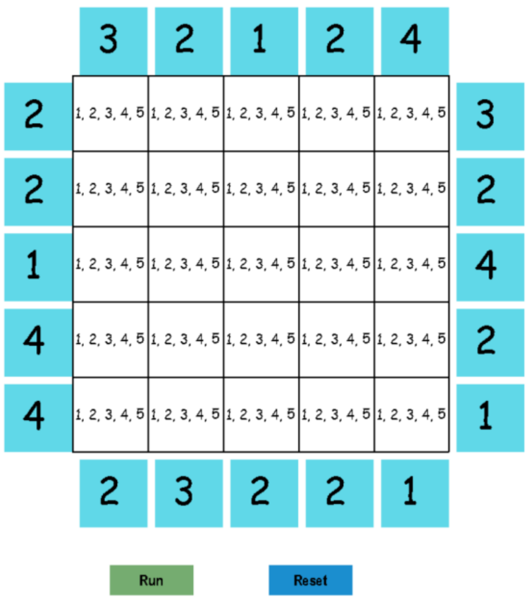
4, 1, 5, 3, 2
4, 2, 5, 3, 1
4, 3, 5, 2, 1

This will reduce the domain of cells

2	{4}	{1,2,3}	{5}	{2,3}	{1,2}	3

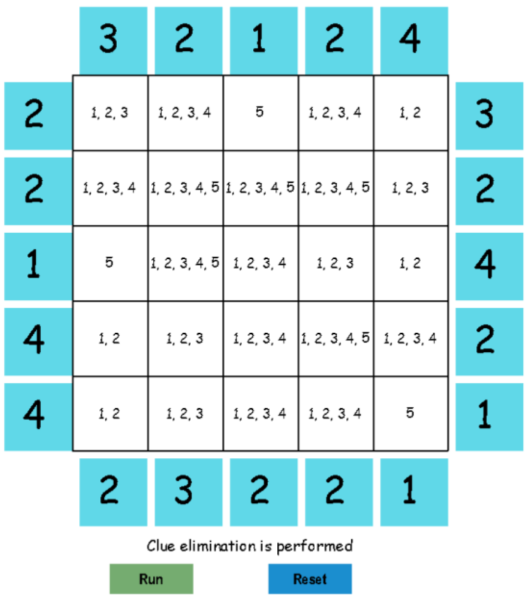
Here, we show only for row, but it will be perform for column also(according to the clues at top and bottom) and domain size will reduce further, may be to 1.
We repeat step 2,3 and 4 until domain size of each cells reduce to one and that will be our answer.
Optimization: - We are generating the sequence for all rows and columns. We can speedup, if we generate the sequence for the rows and columns which has least possible sequence and it is not resolved till yet. After that,we apply constraint propagation(if some cell is resolved) and elimination assignment. Also we can maintain the count of eliminated sequence in the previous iteration, if count doesn't changed for two iteration we can skip the row/column.

Worked Example



	3	2	1	2	4	
2	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	3
2	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	2
1	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	4
4	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	2
4	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1
	2	3	2	2	1	

FIGURE - 1



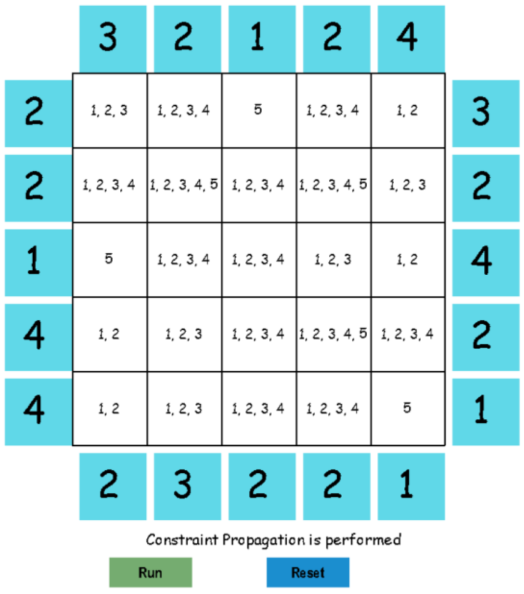
	3	2	1	2	4	
2	1, 2, 3	1, 2, 3, 4	5	1, 2, 3, 4	1, 2	3
2	1, 2, 3, 4	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3	2
1	5	1, 2, 3, 4, 5	1, 2, 3, 4	1, 2, 3	1, 2	4
4	1, 2	1, 2, 3	1, 2, 3, 4	1, 2, 3, 4, 5	1, 2, 3, 4	2
4	1, 2	1, 2, 3	1, 2, 3, 4	1, 2, 3, 4	5	1
	2	3	2	2	1	

Clue elimination is performed

FIGURE - 2

We start with a 5x5 puzzle(Figure-1) where each cell domain is a set {1,2,3,4,5}.
We perform Clue elimination(step -1 of algorithm) over it and we get figure-2 with shrinked domain for some cells according to the clues. Let us chck the last row. Cell-1 domain is shrinked by left 4, to {1,2} as 3,4,5 cannot be placed while satisfying the clue(4). Same reason for cell 2 and 3.
Right side clue(1) assigned cell-5 with only possible entry 5.
5 is removed from the cell 4 by the bottom clue(2). Like this every clue eliminate some values from somes cells. Clue 1 and 5 are much helpful as they assigned the values by reducing the domain size to 1. Clue 5 assigned the whole row. Other clues give some reductions.
Now in step-2, check the figure-3 and compare it to figure-2.

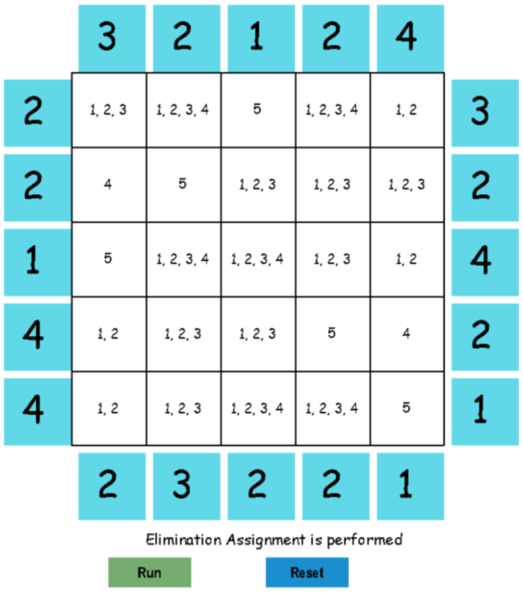
In the column -3 of the grid, 5 is removed from the cell(2,3), because 5 is already assigned to cell(1,3). Other things remained same here cause in the first step we didn’t get much assignments. But if we have, this step we definitely shows a better results in terms of reduced domains.



	3	2	1	2	4	
2	1, 2, 3	1, 2, 3, 4	5	1, 2, 3, 4	1, 2	3
2	1, 2, 3, 4	1, 2, 3, 4, 5	1, 2, 3, 4	1, 2, 3, 4, 5	1, 2, 3	2
1	5	1, 2, 3, 4	1, 2, 3, 4	1, 2, 3	1, 2	4
4	1, 2	1, 2, 3	1, 2, 3, 4	1, 2, 3, 4, 5	1, 2, 3, 4	2
4	1, 2	1, 2, 3	1, 2, 3, 4	1, 2, 3, 4	5	1
	2	3	2	2	1	

Constraint Propagation is performed

FIGURE - 3

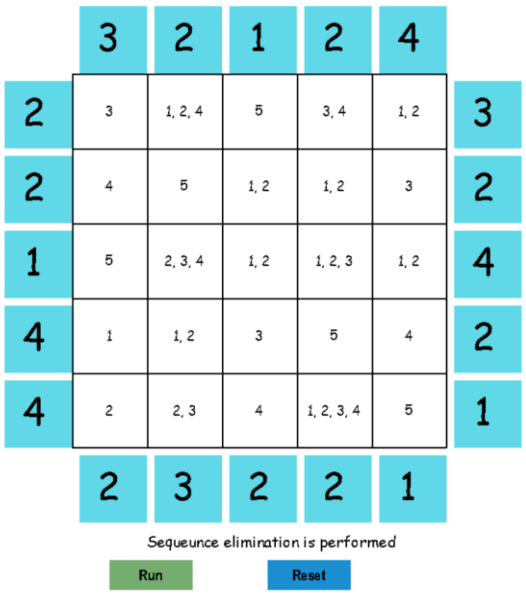


	3	2	1	2	4	
2	1, 2, 3	1, 2, 3, 4	5	1, 2, 3, 4	1, 2	3
2	4	5	1, 2, 3	1, 2, 3	1, 2, 3	2
1	5	1, 2, 3, 4	1, 2, 3, 4	1, 2, 3	1, 2	4
4	1, 2	1, 2, 3	1, 2, 3	5	4	2
4	1, 2	1, 2, 3	1, 2, 3, 4	1, 2, 3, 4	5	1
	2	3	2	2	1	

Elimination Assignment is performed

FIGURE - 4

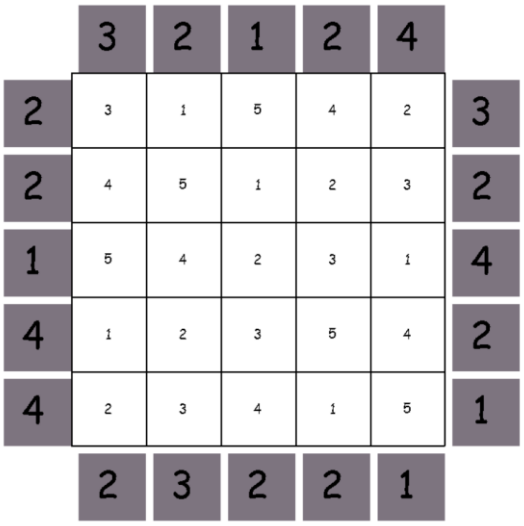
See the figure- 3. In the first column, 4 is possible only for cell (2,1). So it assigned to the cell and in its row contraint, and 4 in the domain of the cells in its row is removed.
In fifth column also, same situation for cell(5,4) is there. So 4 is assigned to cell(5,4) and is removed from the domains of the cell from the fourth row.
Similar situations were checked for each cells and according operation were performed and we get Figure 4 as the result.
Seem quite interesting. We already get 7 assignments out of 25 and very well shrinked domains for other cells without performing sequence elimination step.



	3	2	1	2	4	
2	3	1, 2, 4	5	3, 4	1, 2	3
2	4	5	1, 2	1, 2	3	2
1	5	2, 3, 4	1, 2	1, 2, 3	1, 2	4
4	1	1, 2	3	5	4	2
4	2	2, 3	4	1, 2, 3, 4	5	1
	2	3	2	2	1	

Sequeunce elimination is performed

FIGURE - 5



	3	2	1	2	4	
2	3	1	5	4	2	3
2	4	5	1	2	3	2
1	5	4	2	3	1	4
4	1	2	3	5	4	2
4	2	3	4	1	5	1
	2	3	2	2	1	

Sequeunce elimination is performed

FIGURE – 6 (Solution)

Figure- 5 shows the result after step-4, which is sequence elimination step on the input of figure- 4 at one time only. Lot of domains changed here and finding the reason is not easy, but it obey the rules/constraints and correctness can be checked easily by comparing the domains value with the clues.
We again perform the sequence elimination step on figure- 5, until we get the solution (each cell domain is reduced to one) in the figure- 6.

Algorithm Complexity

For the problem, we take a domain of size N for each cell. So space Complexity for storing the problem is O(N^3).
Step by step complexity analysis.
STEP -1: CLUE ELIMINATION – We run a loop for each clue (Total 4*N) and reduce the domains of the cells in the row or column corresponding to it. Reduction of domain takes at max N computation. So, the complexity for this step is O (4* N^3) or O(N^3) in the worst case. In worst case all the cells will be resolved, that would not be a case in a normal puzzle.
On average, it will be of order n-square.
STEP -2: CONSTRAINT PROPAGATION – Each cell (which is resolved and domain size is reduced to one) will check in its row and column for constraint defilement. Row/Column size is N, count of cells is N^2. So, complexity will be O(N^3).
STEP -3: ELIMINATION ASSIGNMENT – Same as of constraint propagation step, each cell check in its row and column for each of the element in its domain. So, complexity will be O(N^4).
STEP -4: SEQUENCE ELIMINATION - In the worst case, when none of the cell domain is reduced, complexity will be of order N! (n- factorial). But as we already performed the step 1 to 3, worst case will never arise. As this algorithm runs on a 5x5 puzzle, we built 8 and 22 sequence in examples. 240 to 8(or 22) is very well reduction. On 6x6 puzzle 2880 to 139 and 202. So overall complexity is depending on the hardness and size of the puzzle. So, we can solve puzzle of any size and hardness by this algorithm effectively in computational time.

Conclusion

With the help of AI, we successfully mimic the human style of solving a puzzle of Skyscraper, efficiently and effectively. Use of clue elimination, constraint propagation and elimination assignment methods reduce the number of sequences needed to check for solving the puzzle. Ordering the row/column in such a way that solving that row/column will helps to determine other cells values also, optimise the sequence elimination step by about a factor of N. This idea can be extended on types of puzzles like sudoku or N-queen and real life problems of transportation scheduling or other CSPs.

Reference

For the UI, pygame library is used and Idea is taken from the youtube video: *Lets make Sudoku with Python:* https://www.youtube.com/watch?v=r_cmJBgrq5k&ab_channel=APlusCoding
A valid(solvable) puzzle is taken from : https://www.puzzle-skyscrapers.com/
Idea for Algorithm taken from :
https://www.conceptispuzzles.com/index.aspx?uri=puzzle/skyscrapers/techniques