A1Q3

May 23, 2022 5:03 PM

A. Suppose we have a $N \times N$ grids where the top left cell is indexed as $(1,1)$ and the bottom right cell is indexed as (N,N)
Variables: (x_i, y_i) to represent the position of the i-th queen. i is an integer from 1 to N. x_i is the row number and y_i is the column number of the position of the i-th queen
Domain: Each x_i and y_i are an integer from 1 to N. $x_i, y_i \in [1, N], x_i, y_i \in Z, \forall i \in [1, N], i \in Z$
Constraints:
1) No two queens are on the same row(binary):
$$\forall \mathrm{i}, \mathrm{j}, \mathrm{i} \neq \mathrm{j}, \mathrm{i}, \mathrm{j} \in [1, N], i, j \in Z \Rightarrow x_i \neq x_j$$
2) No two queens are on the same column(binary):
$$\forall i, j, i \neq j, i, j \in [1, N], i, j \in Z \Rightarrow y_i \neq y_j$$
3) No two queens are on the same diagonal(binary):
$$\forall i, j, i \neq j, i, j \in [1, N], i, j \in Z \Rightarrow x_i + y_i \neq x_j + y_j \text{ and } x_i - y_i \neq x_j - y_j$$

B. Suppose we have a $N \times N$ grids where the top left cell is indexed as $(1,1)$ and the bottom right cell is indexed as (N,N)
Variables: A_{ij} to represent the status (whether there is a samurai)of the position (i, j).
Domain: $A_{ij} \in \{0,1\}\ \forall i, j, i, j \in [1, N],\ i, j \in Z$, where 1 indicates there is a samurai and 0 indicates there is not.
Constraints:
1) No two samurais are on the same row(binary):
Any two cells A_{im}, A_{in}, only one of them can be one
2) No two samurais are on the same column(binary):
Any two cells A_{mj}, A_{nj}, only one of them can be one
3) All samurais have a position(N^2-nary):
$$\sum_{i=1, j=1}^{N,N} A_{ij} = N$$
4) No two samurais are on the same grid(binary):
This might be hard to represent mathmatically.
Any two cells A_{ij}, A_{mn}, if they are on the same $M \times M$ grid, then only one of them can be one

C. Suppose we have a $N \times N$ grids where the top left cell is indexed as $(1,1)$ and the bottom right cell is indexed as (N,N)
Variables: A_{ij} to represent the number in the cell (i, j).
Domain: $A_{ij} \in [1, N], A_{ij} \in Z,\ \forall i, j, i, j \in [1, N],\ i, j \in Z$.
Constraints:
1) No two cells on the same row contain the same number(binary):
$$\forall i \in [1, N], i \in Z, \forall m, n \in [1, N], m, n \in Z, m < n, A_{im} \neq A_{in}$$

2) No two cells on the same column contain the same number (binary):

$$\forall j \in [1, N], j \in Z, \forall m, n \in [1, N], m, n \in Z, m < n, A_{mj} \neq A_{nj}$$

3) No two cells on the same grid contain the same number (binary):
$\forall k, k \in [0, M-1], k \in Z \Rightarrow \forall m, n, i, j \in [1, M], m, n, i, j \in Z, m \neq i, n \neq j, A_{(kM+i, kM+j)} \neq A_{(kM+m, kM+n)}$

4) Initial digits (domain constraints):
Some cells have a pre-decided value, so it cannot be any other values in the domain.

D. We can add domain constraints to the given CSP solver for the N-samurai problem N times based on the domain constraints of the Sudoku problem. There are N-numbers. For those numbers that some initial cells contain it, we choose a number each time, and pick all those initial cells with them out, set them to fixed number one in the N-samurai problem, and solve the samurai problem, so we can have a set of results that is valid for the number we chose. We run CSP for N-samurai without additional constraints one more time for all numbers that do not have an initial cell.

We can do a dfs search on those sets. Start by an empty N x N grid, we choose from the solution sets obtained by initial cells and add it to the grid as a new state. For example, we can choose any solution from set 1, add it to the current grid which forms the first level of the tree, then choose from set 2. The selection is only valid if no cell has a value is greater than one (which defines the neighbour). Finally, we do similar selection from the set without initial cells after all other sets are selected, and we select k times where k is the number of numbers that do not have an cells to contain it initially, following the same rule.