

Sudoku Pair Solver and Generator:

Sudoku Pair Solver:

Implementation:

- Sudoku Pair solver was implemented by using the pySAT module.
- Used NumPy and pandas to read the input CSV file, extract the given sudoku pair, store it as a NumPy array and provide the solved sudoku pair as a CSV file.
- Used CNF to encode the sudoku pair constraints. Each sudoku pair would have k^4 cells, and each cell has k^2 options. Hence, we use k^6 variables.
- The encoding that we have used resembles the Efficient one (Weber 2005) commonly used to solve the single sudoku puzzles.
- We extracted the filled values from the sudoku and store them as a list of assumptions.
- Now, we use minisat22 from pysat.solvers to solve this problem. Then we fetch the solved model and provide the solved sudoku pair as the output CSV file.

Assumptions:

- Only the sudoku is being given as input, and k is not provided separately.
- All packages used in the code are installed already in the system in which the code is to be tested.

Limitations:

- The code is not efficiently implemented and thus it is slow in python.(especially for $k > 3$) The main delay is due to the checkuniq function which checks if the module is unique or not.

How to run:

- Run the code using the command “python3 SAT_Sudoku_Solver.py”
- Then enter the name of the CSV file without “.csv” extension, for eg. “testinput”.
- Now, the solved sudoku pair will be in a CSV file named according to the name of input file, in our example as “testinput_output.csv” in the same folder.
- Other test cases are given in the “Test Cases” folder along with their corresponding solutions with the name format for the test case being “input4.csv” and its expected output being “input4_solution.csv”. Now, input1 does not have any possible solutions, so instead of CSV files, we have a txt file corresponding to the test case stating the expected output “None” in it.

Sudoku Generator 1

Implementation:

- We try to generate a filled sudoku pair using the constraints as described in question 1 and along with that, a blank list is passed as assumption to the minisat22 solver.
- From that sudoku, we make the assumption list.
- Now we make a set of integers from 1 to the length of the assumption list and take all subsets of this set and iterate through them.
- In each iteration, we randomly remove one number from the sudoku and check if it has a unique solution. If it has only one solution, then we accept that change and move on to the next number. Else we restore the previous sudoku and move on to the next number.
- Now, the uniqueness is tested as follows:-
 - First, the sudoku is solved. So, we have one solution.
 - Now, we add the not of this solution to the constraints and check for solutions.
 - If we find another solution, then we say that multiple solutions exist. Else, we say that a unique solution exists.

Assumptions:

- All packages used in the code are installed already in the system in which the code is to be tested.
- We can remove $n^2/2$ values from the fully filled generated sudoku after shuffling the list containing them

Limitations:

- The code is not efficient enough, and thus it takes a lot of time to generate sudoku. (especially for $k > 3$)
- The main delay is due to the checkuniq function which checks if the module is unique or not

How to run:

- Run the code in the “Main Code” folder by running the command “python3 SAT_Sudoku_Generator.py”.
- Enter the value of k.
- The corresponding sudoku pair will be generated and shown in the file “output.csv” in the same folder.
- Other test cases are given in the “Test Cases” folder along with their corresponding solutions with the name format for the test case being “test1.txt” and its expected output is only an example of the many possible sudokus generated as “test1_sol.csv”.

Sudoku Generator 2:

We have also made a file named Sudoku Generator 2 which has the code for generating sudoku with the globally minimal clue-set. The files 'input3.csv' and 'input6.csv' in the test case for question 1 have been generated using this code.

- Joint Project with Ujwal Jyot Panda