Use Deterministic Optimization formation to solve Sudoku problems

Zhiyu Cheng, Solar Shao, Chen Xu, Zihao Zhang

August 30, 2021

Abstract

1 Step-by-step Description

1. Set fixed constraints The first thing we do is install the "cvxopt"package. Then we use "check-by-submatrix" to check which number is missing from 1-9 by sub-3*3 matrix and plug in the missing value if the value can be determined. Next, we use "plugged-grids" to plug predetermined grids for the the board list and "index-to-plugged-grids" to get a predetermined grids plugged board-list from the index. Finally, we use the "fixed-constraints" function. There are several constraints in the function: each number appear once in each row, each column, and each grid; and each grid has one number.

2. Set given constraints In the part, we use the function named "given-constraint." Firstly, we use "convert-to-recursive-list" to convert quiz string to 9*9 recursive list and use "np. array" to convert list to array. Then we write a for loop to find out the location of nonzero elements in board-array. Next, we the nonzero elements and use lil-matrix to set an empty len(location-in-string)*729 matrix. Finally, we use "given-constraints" to change back to csr-matrix.

3. Plug in pre-determined grids to lower the calculation In this part, we use the "cvxopt" package and use "solvers.options" to import solvers. Then we import the data we wanted to analyse separately.

4. Use two constraints to set up the linear optimization problem For the part, we set the problem at first. And then we combine "fixed-constraints" with "given-constraint". Next, we employ "np.linalg.svd" and "Np.block" to convert the problem to a full-rank problem. In the end, we set the Linear optimization problem.

5. Solve the problem and judge the success rate The python program helped us to do this.

2 Success Rate

Public small data set A: small 1 24/24; small 2 325/1011
Public large data set B: large 1 828/1000; large 2 986/1000

The result we get from the project does not match the expectation of our group. In fact, we use the Genetic Algorithm (GA) rather than linear programming at first. We spent a large amount of time figuring out the result of the project by using GA before. Unfortunately, we cannot get any results by using GA method. Then we try to use the alternative method which is linear programming and we find that the method is easier and more convenient. Even though the result is not consistent with our group's expectation, we feel satisfied that we can get the result within such a short time.

3 Conclusion and Future Work

As one of the most effective and convenient optimization methods, linear programming can help us not only solve some Sudoku problems but also get precise solutions of Sudoku puzzles. Even though linear

programming optimization plays a good role in solving some easy or challenging problems, there are still some drawbacks and instability when linear programming optimization is used to solve difficult Sudoku puzzles. So, we still need a deeper study of the linear programming method later. Besides, as we mentioned above, we use the Genetic Algorithm (GA) instead of the linear programming method at first. From the project, we found that there are some differences between GA and the linear programming method. Different from the linear programing method, GA is more complicated and complex. Though we cannot figure out the results by using GA, we hope that we can use GA to solve the Sudoku puzzles soon. Therefore, we need to get a deeper understanding of optimization methods and find a method which is more efficient in solving Sudoku puzzles in the future.