Title: Solving Sudoku game by using Linear Programming **Date**:5/17/19

Group members: Yanxin Wang, Yaoqi Li, Qingcheng Zhang, Zhaoyue Wang

- State your method, mathematically, indicate clearly the notations and symbols and their meanings. Also try to explain why do you think your method could work better.

To solve the sudoku problem, we choose to use the Linear Programming method which treat the problem as min $f(X)$ subject to the linear constraints AX=B. A is the matrix construct by the constraints of sudoku which consists of row, column, box and cell. In the sudoku game, there are 9*9 cells. In each cell, we treated it as a 9*1 vector, which is o's except 1 on the position of the digit. So the shape of CELL is (81,729). For each row in this matrix, we defined a number(from 1 to 9). For example, 100000000 means 1; 010000000 means 2, and so on. As for the ROW in array A, it is a array which has 9 rows. Each row has 9 different choices of numbers(from 1 to 9) which represented by 9*9 matrix, therefore, the shape of ROW is (81, 729). (???) Similarly, the COL uses the same way to contribute an array and the shape of COL is same as the ROW, which is (81,729). For the BOX part, we consider one box as a 3*3 box. In one cell, there are three choices:100,010,001. One row has three cells, and there a three rows in the box. CLUE is a constraint which constructed by the given clues and the size of CLUE is L*729, the value of L will base on the given clues. Finally, A=[ROW COL BOX CELL CLUE] and X is our solution which has 81*9=729 (81 numbers, one number represents by 9 bits)rows. So B is a (324+L)*1 matrix and all entries are equal to 0.

Output:

Small 1:

```
Aver Time:   0.38 secs. Success rate: 20 / 20
Aver Time:   0.37 secs. Success rate: 24 / 24
```

Large 1:

```
Aver Time:   0.40 secs. Success rate: 16 / 20
Aver Time:   0.41 secs. Success rate: 32 / 40
Aver Time:   0.41 secs. Success rate: 47 / 60
Aver Time:   0.41 secs. Success rate: 64 / 80
Aver Time:   0.41 secs. Success rate: 78 / 100
Aver Time:   0.42 secs. Success rate: 92 / 120
Aver Time:   0.42 secs. Success rate: 107 / 140
Aver Time:   0.42 secs. Success rate: 124 / 160
Aver Time:   0.42 secs. Success rate: 138 / 180
Aver Time:   0.42 secs. Success rate: 155 / 200
Aver Time:   0.42 secs. Success rate: 173 / 220
Aver Time:   0.42 secs. Success rate: 191 / 240
Aver Time:   0.43 secs. Success rate: 208 / 260
Aver Time:   0.43 secs. Success rate: 224 / 280
Aver Time:   0.44 secs. Success rate: 242 / 300
Aver Time:   0.44 secs. Success rate: 257 / 320
Aver Time:   0.44 secs. Success rate: 275 / 340
Aver Time:   0.44 secs. Success rate: 291 / 360
Aver Time:   0.45 secs. Success rate: 308 / 380
Aver Time:   0.45 secs. Success rate: 328 / 400
Aver Time:   0.45 secs. Success rate: 344 / 420
Aver Time:   0.45 secs. Success rate: 358 / 440
Aver Time:   0.45 secs. Success rate: 375 / 460
Aver Time:   0.45 secs. Success rate: 392 / 480
Aver Time:   0.45 secs. Success rate: 410 / 500
Aver Time:   0.45 secs. Success rate: 425 / 520
Aver Time:   0.45 secs. Success rate: 441 / 540
Aver Time:   0.44 secs. Success rate: 456 / 560
Aver Time:   0.44 secs. Success rate: 474 / 580
Aver Time:   0.44 secs. Success rate: 492 / 600
Aver Time:   0.44 secs. Success rate: 507 / 620
Aver Time:   0.44 secs. Success rate: 524 / 640
Aver Time:   0.44 secs. Success rate: 540 / 660
Aver Time:   0.43 secs. Success rate: 556 / 680
Aver Time:   0.44 secs. Success rate: 573 / 700
Aver Time:   0.44 secs. Success rate: 591 / 720
Aver Time:   0.44 secs. Success rate: 606 / 740
Aver Time:   0.43 secs. Success rate: 623 / 760
Aver Time:   0.43 secs. Success rate: 636 / 780
Aver Time:   0.43 secs. Success rate: 653 / 800
Aver Time:   0.43 secs. Success rate: 667 / 820
Aver Time:   0.43 secs. Success rate: 683 / 840
Aver Time:   0.43 secs. Success rate: 701 / 860
Aver Time:   0.43 secs. Success rate: 719 / 880
Aver Time:   0.43 secs. Success rate: 736 / 900
Aver Time:   0.43 secs. Success rate: 752 / 920
Aver Time:   0.43 secs. Success rate: 767 / 940
Aver Time:   0.43 secs. Success rate: 786 / 960
Aver Time:   0.42 secs. Success rate: 803 / 980
Aver Time:   0.43 secs. Success rate: 818 / 1000
Aver Time:   0.43 secs. Success rate: 818 / 1000
```

Small2:

```
Aver Time:   0.37 secs. Success rate: 5 / 20
Aver Time:   0.37 secs. Success rate: 11 / 40
Aver Time:   0.36 secs. Success rate: 15 / 60
Aver Time:   0.36 secs. Success rate: 25 / 80
Aver Time:   0.36 secs. Success rate: 35 / 100
Aver Time:   0.36 secs. Success rate: 44 / 120
Aver Time:   0.36 secs. Success rate: 54 / 140
Aver Time:   0.36 secs. Success rate: 61 / 160
Aver Time:   0.36 secs. Success rate: 70 / 180
Aver Time:   0.36 secs. Success rate: 80 / 200
Aver Time:   0.36 secs. Success rate: 89 / 220
Aver Time:   0.36 secs. Success rate: 92 / 240
Aver Time:   0.37 secs. Success rate: 97 / 260
Aver Time:   0.40 secs. Success rate: 101 / 280
Aver Time:   0.39 secs. Success rate: 108 / 300
Aver Time:   0.39 secs. Success rate: 114 / 320
Aver Time:   0.39 secs. Success rate: 122 / 340
Aver Time:   0.39 secs. Success rate: 127 / 360
Aver Time:   0.39 secs. Success rate: 136 / 380
Aver Time:   0.39 secs. Success rate: 144 / 400
Aver Time:   0.38 secs. Success rate: 150 / 420
Aver Time:   0.38 secs. Success rate: 155 / 440
Aver Time:   0.38 secs. Success rate: 161 / 460
Aver Time:   0.38 secs. Success rate: 168 / 480
Aver Time:   0.38 secs. Success rate: 176 / 500
Aver Time:   0.38 secs. Success rate: 184 / 520
Aver Time:   0.38 secs. Success rate: 188 / 540
Aver Time:   0.38 secs. Success rate: 199 / 560
Aver Time:   0.38 secs. Success rate: 204 / 580
Aver Time:   0.38 secs. Success rate: 210 / 600
Aver Time:   0.38 secs. Success rate: 217 / 620
Aver Time:   0.38 secs. Success rate: 224 / 640
Aver Time:   0.37 secs. Success rate: 228 / 660
Aver Time:   0.37 secs. Success rate: 236 / 680
Aver Time:   0.37 secs. Success rate: 243 / 700
Aver Time:   0.37 secs. Success rate: 247 / 720
Aver Time:   0.37 secs. Success rate: 254 / 740
Aver Time:   0.37 secs. Success rate: 260 / 760
Aver Time:   0.37 secs. Success rate: 265 / 780
Aver Time:   0.37 secs. Success rate: 270 / 800
Aver Time:   0.37 secs. Success rate: 279 / 820
Aver Time:   0.37 secs. Success rate: 287 / 840
Aver Time:   0.37 secs. Success rate: 293 / 860
Aver Time:   0.37 secs. Success rate: 299 / 880
Aver Time:   0.37 secs. Success rate: 303 / 900
Aver Time:   0.37 secs. Success rate: 303 / 920
Aver Time:   0.37 secs. Success rate: 310 / 940
Aver Time:   0.37 secs. Success rate: 316 / 960
Aver Time:   0.37 secs. Success rate: 320 / 980
Aver Time:   0.37 secs. Success rate: 324 / 1000
Aver Time:   0.37 secs. Success rate: 324 / 1000
```

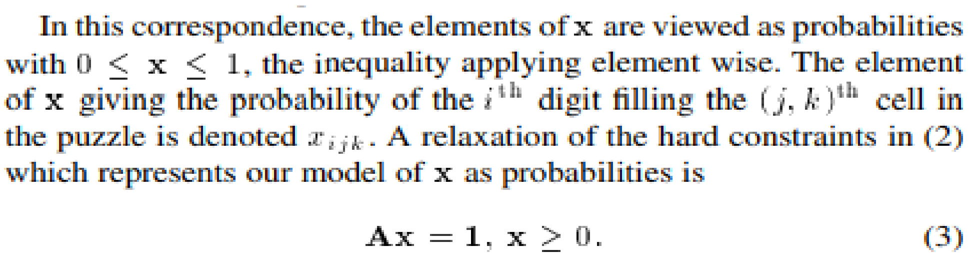
Large 2:

- State the success rate on data sets A, B with a table. Explain why or why not the result match your group's expectation.

When solving sudoku problem, we use the constraint to find the feasible solution at first, but there may have different feasible solutions for one cell which satisfies one constraint. Also, the feasible set may be large enough to contain an infinite number of points, but the sudoku problem will have a unique solution which we prefer. In this case our method can get the feasible solution and we need to consider the possibility or desirability of each feasible solution to find the unique solution.

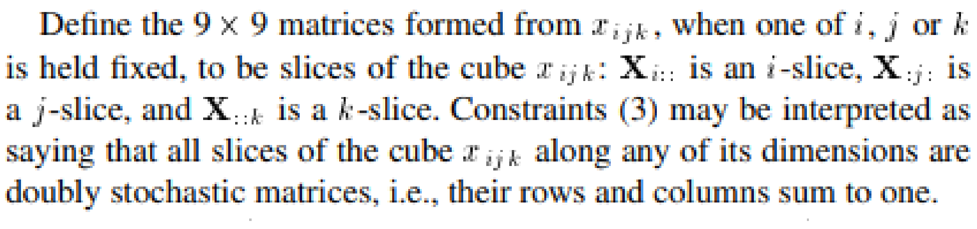
- If there are issues or additional future work, please state that as well.

To solve the issue which we described in the previous part, we need to consider a new constraint. The new constraint makes sure we can get the higher probability solution in the whole feasible solution set. And by using the new constraint, to let the feasible set and the solution set we get in the new constraint equal, we can improve the rate of correction in finding the unique solution in sudoku problem.



In this correspondence, the elements of $\mathbf{x}$ are viewed as probabilities with $0 \le \mathbf{x} \le 1$, the inequality applying element wise. The element of $\mathbf{x}$ giving the probability of the i^{th} digit filling the $(j, k)^{th}$ cell in the puzzle is denoted x_{ijk}. A relaxation of the hard constraints in (2) which represents our model of $\mathbf{x}$ as probabilities is

$$\mathbf{Ax} = \mathbf{1},\ \mathbf{x} \ge 0. \qquad (3)$$



Define the 9×9 matrices formed from x_{ijk}, when one of i, j or k is held fixed, to be slices of the cube x_{ijk}: $\mathbf{X}_{i::}$ is an i-slice, $\mathbf{X}_{:j:}$ is a j-slice, and $\mathbf{X}_{::k}$ is a k-slice. Constraints (3) may be interpreted as saying that all slices of the cube x_{ijk} along any of its dimensions are doubly stochastic matrices, i.e., their rows and columns sum to one.

References:

https://www.math.uci.edu/~brusso/entropyminim2012.pdf