Project 1: Sudoku Challenges

05/17/19 Ziru Yan, Zukang Yang, Ruozhu Wang, Mengdi Gao

The linear programming method in the starter code can achieve an accuracy score of approximately 0.2 - 0.3. We combine it with genetics algorithm to increase the accuracy.

Genetics algorithm is an algorithm that assembles evolution theory where only powerful genes from humans can be passed to the next generation. Mathematically, it at first initializes a population consisting of candidates and then it goes through a series of mutations to form new candidates which will be compared with the previous candidates before they become the new candidates. GA can be used to solve unconstrained optimization problems, like the one we are solving.

Here is the comparison between linear programming and genetics algorithm:

	Linear Programming	Genetics Algorithm
Small 2 accuracy	0.35	0.51
Advantages	Takes 4-5 seconds to solve each small puzzle successfully.	Can achieve high accuracy with enough clues.
Disadvantages	Can't solve hard puzzles or can solve some of them with overall low accuracy	Takes 6 min to run 10K iterations

Since each algorithm has strengths and weaknesses, in order to preserve the strengths from both methods, i.e. speed and accuracy, we decided to incorporate both of them altogether as our ultimate Sudoku solver. Specifically, we allow each puzzle to go through LP, and use the unsolved output as the input for GA. Aside from combining both methods, we changed the default solver in solvers.lp to glpk, which, after experiments, increases the accuracy of solving the puzzle with LP from 0.32 to 0.35 for small2.

In spite of the fact that our model is able to achieve an accuracy of 0.843 for small2, an accuracy of 0.927 for large1, an accuracy of 0.978 for large2, there are some improvements we could have done: according to the paper *Solving The Sudoku With The Differential Evolution*, DE can achieve the same performance as can GA. Hence, if we can implement DE in place of GA, the accuracy could be better.