Monte Carlo simulations 2006. Exercise X (Bonus)

To be handed in Fri May 5, solution not given

1. (40 p) Brute force MC solution of sudoku.

The aim of the standard version of the sudoku puzzle is to enter a numerical digit from 1 through 9 in each cell of a 9×9 grid made up of 3×3 subgrids (called "regions"), starting with various digits given in some cells (the "givens"). Each row, column, and region must contain only one instance of each numeral. A proper sudoku puzzle has exactly one correct solution.

Write an MC code which solves sudoku puzzles by brute force. It should first read in the puzzle from a file as a set of empty squares and given numbers. Then the code should generate a completely random initial guess of the solution on all the empty squares. After this it should perform some sort of simulated annealing optimization of the solution until the correct solution is found.

The solution algorithm *should not* use any logical algorithm in searching the solution. But you do of course need to device some sort of quality factor for the solution candidates to be used as the quantity to be minimized.

Demonstrate that the code works by running it on the initial states 1-4 given on the course web page. Return the solutions to these, the code, a short description of your solution algorithm, and tell how many iterations you need to always find a correct solution to any sudoku puzzle.

Any program which fulfills the criteria given and almost always manages to solve most sudoku puzzles in the number of iterations you tell, will get full points.

If you want to learn more about sudoku, wikipedia has an excellent article about it.

To give a point of reference, it took the lecturer about 4 hours to write a solver which solves most sudoku puzzle under 10 million iterations. Try to do better than this!. If you manage to make a code which consistently solves the example "Really hard" you are doing better than the lecturer!