ABSTRACT:

A Sudoku puzzle solver was implemented to demonstrate recursive backtracking with constraint satisfaction propagation, a common method for “Intelligent” machine decision making.

The solver was tested on puzzles rated by <https://qqwing.com/solve.html> to have difficulty levels of simple, easy, intermediate, and expert. These puzzles exit in the root director (ex: sud\_hard.txt) and may be entered at the program’s prompt for evaluation by the application. The algorithm and test results are described below.

ALGORITHM:

A Most-Constrained-Value heuristic was used and the algorithm operates by traversing the search space (as a tree) in a depth-first and post-order manner. Generating child nodes by giving precedence to the variable (i.e. Sudoku cell) with the greatest Most-Constrained-Value and backtracking when any variable is found to have zero remaining unconstrained values, the algorithm continues until a goal state (i.e. a correctly solved Sudoku puzzle) is reached or the search space is exhausted (i.e. no solution exists.)

The processing of a node in this tree represents the assignment of a valid value to a previously unpopulated variable. When this occurs, that value is propagated as a constraint to every adjacent variable (adjacency is defined here as occupying the same row, column, or box of a Sudoku grid). If, during this process, an unpopulated variable is found to have 8 constraints on it, it is populated with it’s last (i.e. it’s 9th) remaining value if doing so would not violate any existing constraints. If it would, the algorithm backtracks.

RESULTS:

|  |  |  |
| --- | --- | --- |
| **Rating** | **Puzzle** | **Solve Time(s)** |
| sud\_simple.txt | 0 5 0 0 1 0 8 0 0  2 0 7 6 9 0 0 0 3  0 0 0 0 3 0 0 0 9  8 0 2 1 0 5 0 3 0  6 0 0 0 0 7 0 0 8  0 0 9 0 0 0 0 0 6  3 0 8 0 0 6 0 0 0  1 7 0 0 0 0 0 0 0  0 0 6 0 0 0 0 0 0 | 0.006 |
| sud\_east.txt | 0 3 0 9 0 7 0 1 4  0 0 6 5 0 0 0 0 0  4 0 0 0 0 0 0 3 0  0 0 7 6 0 0 0 4 0  1 4 0 0 8 3 0 0 7  0 0 0 0 0 0 0 0 5  0 0 0 0 0 0 7 0 0  0 0 0 3 7 0 8 0 6  0 0 0 0 0 0 0 0 0 | 0.007 |
| sud\_interm.txt | 0 7 4 1 0 3 6 0 0  0 6 5 0 0 7 0 2 0  0 9 0 0 0 2 1 0 0  0 0 0 5 0 0 9 1 0  0 0 0 0 0 0 0 0 7  0 0 9 4 2 0 0 0 0  0 5 2 0 0 0 0 9 0  0 0 6 0 0 0 0 0 0  0 0 0 0 0 0 0 5 0 | 0.008 |
| sud\_expert.txt | 0 0 2 0 9 3 0 0 0  0 0 0 0 0 0 0 0 0  5 0 0 0 0 0 1 0 0  4 0 0 1 0 0 0 0 0  0 0 0 0 0 0 0 3 0  0 0 7 0 0 0 0 2 9  0 0 0 6 7 0 0 0 0  0 9 3 0 0 0 0 0 0  0 0 0 0 0 4 6 0 0 | 0.082 |

ADDITIONAL TESTING:

Additionally, 21 puzzles rated Extremely Hard are provided in /puzzles/more\_puzz.zip. The algorithm successfully solved each of these puzzles with an average time of less than 0.1 seconds.