AAA Project

Solving Sudoku using the backtracking algorithm

800107 - Daniel Ng See Cheong
839521 - Uzair Moolla

September 2, 2016

1 Introduction

Sudoku is a numerical based logic puzzle game. The idea is to solve an $n \times n$ grid filled with numbers, this grid usually consists of square blocks with 3 rows and 3 columns. These individual blocks are then arranged in a similar manner again, with 3 blocks along rows and 3 along columns (rank 3), producing a 9×9 matrix. The number of digits in a rank 3 matrix is given by $3^2 - 1$ ie. We need to fill the blocks with one of these numbers $1, 2, 3, 4, 5, 6, 7, 8, 9$. The rules for filling these blocks are as follows:

- Within each block we can only have single occurrence of a number.
- Along a each row we can only have single occurrence of a number.
- Along a each column we can only have a single occurrence of a number.

A well formed sudoku puzzle is one which has a unique solution. If a Sudoku of rank n is well-formed, then it must have $n^2 - 1$ distinct digits in it. The problem of solving an arbitrary sudoku puzzle of rank n is quite difficult. The reason for this is that as the rank of a Sudoku puzzle increases from n to n+1, the extra computational time needed to find a solution increases extremely fast. This places the solving of a rank n sudoku puzzle in the group of problems which are NP-complete.

#Fill in facts#

2 Aims

Since the problem of solving an $n \times n$ sudoku puzzle is NP-complete we aim to solve it using a heuristic backtracking algorithm. We will apply this algorithm to sudoku puzzles of rank 3 (9×9 puzzle). This will allow us to attempt to solve these sudoku puzzles in as little time as possible. We aim to find the best, average and worst case complexities of the algorithm through empirical analysis.

3 Summary of Theory

4 Experimental Methodology

5 Presentation of results

6 Interpretation of results

7 Conclusion

8 References

9 Acknowledgments