A study on the backtracking-based Sudoku solving algorithms

Steve Wen­­­

bSc of Computer Science

mcgill university

shuo.wen@mail.mcgill.ca

A picture containing text, clipart

Description automatically generated

2021

Abstract

This study is mainly focused on the efficiency of backtracking-based algorithms solving 3x3 Sudoku puzzles. In study, 3 backtracking algorithm and an additional dancing-links algorithm are implemented to carry out the study. Backtracking algorithm has 3 types: intuitive, block preprocessing, row preprocessing. performance is rated on the preprocessing problem generation time and the efficiency to cover the puzzle board. Instead of counting the elapsed time, the diverging branches of each set are counted. The testing data are all text file like Sudoku puzzles with difficulty ranging from very easy to very hard. After data collection, we then conclude that However, There are still some limitations on when it comes to higher level solving cases. None of these algorithms could work effectively in 4x4 hard mode. Further research will cast on finding productive algorithms for handling higher levels.

# Acknowledgement

# Special thanks to Professor Giulia Alberni for supported me and authorized me to use the template file processing code from the final project of COMP 250 at McGill

# **Table of contents**

# Introduction

## The sudoku puzzle

## Solutions of a puzzle

# Algorithm explained

## Backtracking algorithms

### Intuitive implementation

### Block based implementation

### Row based implementation

## Dancing Links algorithm

# Method

## Setup of the test

## Comparison

### Puzzle difficulty comparison

### Solving efficiency comparison

### Preprocessing generation time comparison

## Data collection

### Elapsed time recording

### Branches counting

## Find multiple solutions

## Reliability of the puzzles

## constraints of algorithms

# Analysis

## Preprocessing time distribution

## Diverging branches counting

1. Conclusion
2. Bibliography

1.Introduction

Sudoku is a puzzle game originating from France in the 19th century and got popularized in Japan in the nineteen eighties. In the twenty years recently, the game has gained vast interests among folks. Competitions are held and new variants are created to meet the expectation on it. Meanwhile, scientists started to step in and talk about the mathematical and logical mechanisms behind the simple grid filling process. More than that, with the rapid development of computational ability, computers started to play a significant row in find a solution towards an intricate puzzle. This study will follow the footsteps of former researchers and take a step forward into the backtracking-based algorithm analysis.

* 1. The Sudoku puzzles

Sudoku is a logical-based, combinatorial number-placement puzzle game. The classical Sudoku game is played on a big 9x9 grid made of 9 smaller 3x3 blocks.

The major objective of such game is to fill the board without breaking basic rules.

Three major rules to follow:

**Row:** Every row of the 9x9 gird should consist of 9 numbers exactly once from 1-9

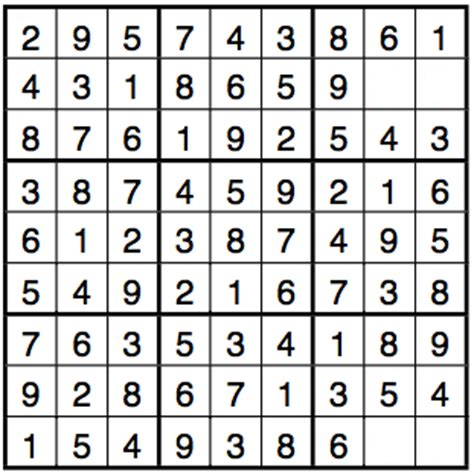
**Column:** Every column of the 9x9 gird should consist of 9 numbers exactly once from 1-9

**Block:** Every block of the 9x9 gird should consist of 9 numbers exactly once from 1-9

* 1. Solutions of the puzzle

A solution of the Sudoku puzzle is called when the 9x9 grid has no empty cells and every cell of it does not break the three basic rules.

However, under certain conditions, a puzzle may have multiple solutions.



1.A A typical puzzle with two solutions

In 1.A above, it is clear to see that within the same block, either 2 or 7 can be put in the left first cell and the other in the second cell. More often than not, an interesting puzzle would have a dozen solution or more solutions ready to be dug. This will drain the human player to death if he dares to try to find all the solutions to the puzzle. Under the power of the machine computation, the same process can be done in a few milliseconds. The huge leap of computational capability will stun me from time to time.

https://puzzling.stackexchange.com/questions/67789/examples-of-sudokus-with-two-solutions

2. Algorithm explained

Various algorithms have reached its tentacles into the Sudoku field. But the most intuitive and the highly efficient method still gives its honor to the backtracking algorithm. This algorithm is closer to the strategy we adopt when solving a Sudoku.

Recall when a Sudoku puzzle is given to you, you will tentatively try a few numbers in certain areas. If you cannot maneuver, you erase the cells where rules are broke and back to the original one. After constant trial and error, and if the puzzle is manageable, you will luckily find a solution. Backtracking algorithms follow this idea and operate thousands of times to find a solution. Since the machine cannot generate an overview of the whole board, it might not get the right decision at certain points. A step wrong will lead to a deep branch of backtracking which seems to fall into an infinite loop while the program is still trying hard to fill the grid in the totally misleading way. Therefore, it then becomes us programmers work to pave the way for the machine. Backtracking algorithms indeed can be optimized to accelerate the process. This chapter will cover the methods that are already optimized to give the reader a straight view on the exploited power of backtracking.