Backtracking Algorithm Implementation for Sudoku Solver

Jonathan Dervin, Roman Higginson, Zaryaab Qasmi

20 April 2024

1. Abstract
   1. Summary: A concise overview of the report, summarizing the purpose, methodology, main findings, and conclusions of your work on the backtracking Sudoku solver.
2. Introduction
   1. Background: Provide context for the Sudoku puzzle and the significance of solving it algorithmically.

Sudoku is a puzzle game that involves the player entering certain numbers from 1 to 9 into a 9x9 grid while being mindful of not violating the rules of the game. The rules are that when a player inputs a number, the number must not be the same as another number in the same row, the same column, and the same 3x3 unit. There are many strategies to this game that an experienced player can use to solve the puzzle with minimum errors. Some solving strategies

* 1. Problem Statement: Define the specific problem that the backtracking algorithm addresses in the context of Sudoku.

The problem that the backtracking algorithm addresses are the rules that this game abides by. The fact that each number in each row, column, and unit must be unique.

* 1. Objective: Outline the main objectives of implementing the backtracking algorithm, such as understanding its efficiency, effectiveness, and limitations in solving Sudoku puzzles.

For this project, the goal was to use several solving strategies to solving 5 boards. But when none of the solving strategies can be used, the program will resort to using the backtracking algorithm to solve them.

1. Methodology
   1. Algorithm Overview: Describe the backtracking algorithm, explaining how it attempts to fill the Sudoku grid one cell at a time and backtracks when it encounters a cell where no legal values can be placed.

A Sudoku Board is considered to be a Constraint Satisfaction Problem (CSP), which is a problem where we need to assign a set of variables that satisfy all of the restrictions given. The backtracking algorithm is a searching algorithm for CSPs where we use a search tree to assign a single variable per level. Since Sudoku is a CSP, the backtracking algorithm will also utilize its key components:

Variables: These represent every cell inside the Sudoku board

Domains: These represent a list of numbers that each cell can have.

Constraints: These represent the limitations each value in the cell must follow.

The backtracking algorithm is meant to assign a number from a domain to an empty cell while being consistent to the constraints of the Sudoku board. If the cell is not consistent, unassign the cell and go to the next number in the domain. If the cell is consistent, go to the next empty cell. If no number in the cell’s domain is consistent to the constraint, backtrack to the last cell that was assigned.

* 1. Implementation Details: Discuss the approach taken, data structures, and any particular aspects of the implementation like recursion, the choice of starting point, and how conflicts are detected and resolved. Here is where I’d like a statement and description of the creative/innovative approach taken.

1. Results
   1. Performance Analysis: Present the results of the algorithm’s performance, including metrics such as execution time, the number of puzzles solved, the complexity of puzzles handled, and the number of backtracks required for various puzzles.
2. Discussion
   1. Efficiency Evaluation: Analyze the efficiency of the backtracking algorithm in terms of time complexity and space complexity.
   2. Limitations: Discuss any limitations encountered with the backtracking algorithm, such as difficulties with certain types of puzzles or inefficiency in particular cases. Make recommendations for improvement.
3. Conclusion
   1. Summary of Findings: Summarize the key findings from your experimentation with the backtracking Sudoku solver.
   2. Recommendations: Offer recommendations based on your findings, such as improvements to the algorithm, strategies for optimization, or areas for further research.
4. References
   1. Citations: Include all the references for the materials, articles, or resources cited in your report, formatted according to the APA style.

<https://www.citationmachine.net/apa>

1. Appendices (if necessary)
2. Additional Material: Any supplementary material that supports the report, such as algorithm implementation, additional data, or extended analysis that is too lengthy to include in the main sections of the report. The use of Generative AI such as specific prompts put here as well\*.

\*I strongly encourage the use of Generative AI tools like ChatGPT, Gemini, Bing, etc to assist you but please remember to use these tools responsibly, which entails critically and ethically. As discussed in class, sometimes these tools produce nonsense but the authoritative manner presented often conveys competence. A useful prompt could be to ask: My team has been tasked with the implementation of Sudoku and we would like guidance on how to proceed using the backtracking algorithm.

I have neither given nor received unauthorized aid in completing this work, nor have I presented someone else’s work as my own.