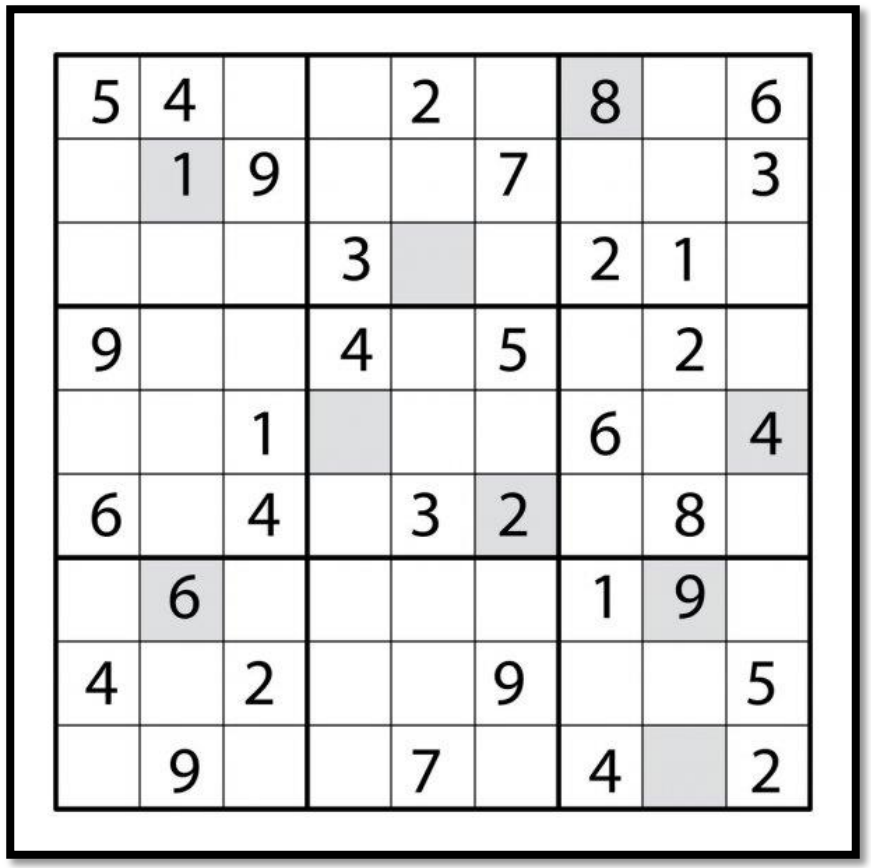
SUDOKU SOLVER USING PARALLEL COMPUTING



5	4			2		8		6
	1	9			7			3
			3			2	1	
9			4		5		2	
		1				6		4
6		4		3	2		8	
	6					1	9	
4		2			9			5
	9			7		4		2

Under the Guidance of
Dr. Hiteshwar Kumar Azad
Assistant Professor, SCOPE, VIT, Vellore

Harshit Mishra (19BCE0799)
Kartik Goel (19BCE2002)
Alokam Nikhitha (19BCE2555)

TABLE OF CONTENTS

01

ABSTRACT

ABSTARCT

1. Sudoku is a very commonly known game of puzzles. A classical sudoku contains a 9X9 gird and aims at filling range of numbers from 1 to 9 such that every row, every column and every 3X3 mini-grid has unique set of numbers without any repetitions.

2. The solution of a sudoku falls under the category of NP complete and thus the algorithm that have been proposed for solving it are computationally complex, that is, they have very high time complexity.

3. In our project we will be implementing a parallel based approach in order to solve sudoku using C and OpenMP.

4. We also aim to document a comparative study of both serial and parallel approaches to solve a sudoku.

02

INTRODUCTION

INTRODUCTION

1. Our project aims at implementing sudoku solver serially and in parallel by using a combination of existing algorithm.
2. In the parallel approach, several threads would be spawned to compute parallelly.
3. C programming and OpenMP will be used to perform elimination, lone ranger and twins in the given sequence to get an optimized and faster performance.
4. When the grid size reach a higher number like 25, the existing serial algorithm fails to find an optimal solution in given time and we will be overcoming this issue by implementing the algorithm parallelly in this project.

03

PROBLEM DISCUSSION

PROBLEM DISCUSSION

1. Sudoku is a combinatorial, logic based number placement game which originated in Switzerland and was popularized in Japan.

2. For an NxN grid, its rules are as followed:
   - Only numbers from 1 to N can be used to fill the grid.
   - Every mini-grid should contain all numbers from 1 to N with a frequency of 1 each.
   - Every row should contain all numbers from 1 to N with a frequency of 1 each.
   - Every column should contain all numbers from 1 to N with a frequency of 1 each.

3. As discussed earlier, the sudoku solver has been commonly implemented serially, but it proves to be slow and inefficient. Our project aims at speeding up this algorithm through means of parallel computing.

04

EXISTING SOLUTIONS

EXISTING SOLUTIONS

Backtracking

Naked Singles

Hidden Singles

Naked Pair

Hidden Pairs

Naked Triple

05

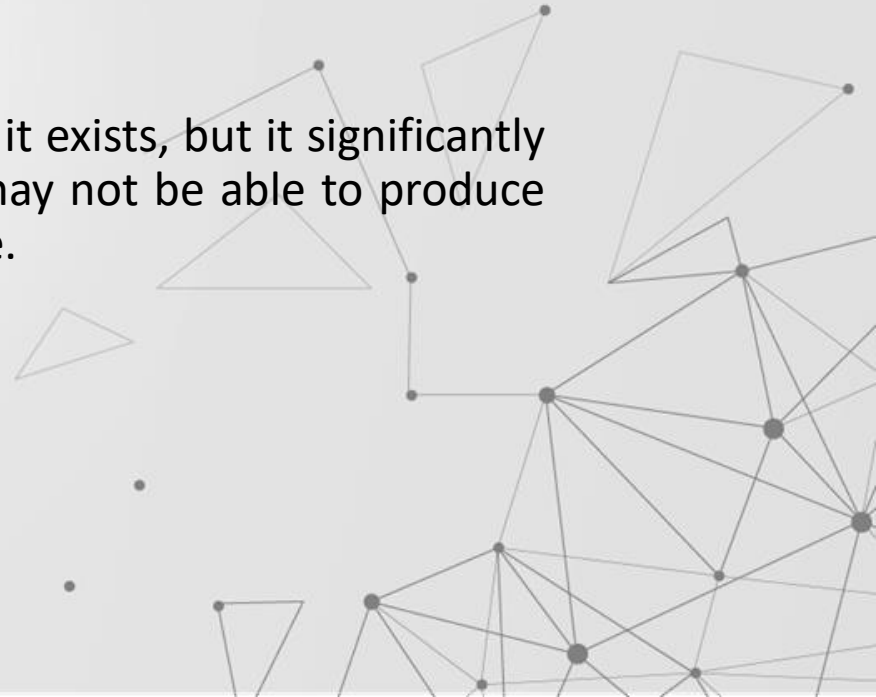
ISSUES IN THESE SOLUTION

ISSUES IN THESE SOLUTIONS

1. The brute force algorithm has its own limitations. The main one being increased execution time with increase in input size. In this algorithm, the time needed to find the solution depends upon the number of empty cells in the given sudoku board.

2. In various other algorithms, filling the naked singles can eliminate the choices in another empty cell because the number is already in either the same row, column or mini-grid.

3. The serial code guarantees to find the solution to sudoku, if it exists, but it significantly slow down as the size of the sudoku board increases and may not be able to produce the solution due to computational limitations of the machine.



06

REFERENCES

REFERENCES

[1] Saxena, Rahul & Jain, Monika & Yaqub, Syed. (2018). Sudoku Game Solving Approach Through Parallel Processing: ICCII 2017. 10.1007/978-981-10-8228-3_41.

[2] Cazenave, Tristan. (2021). A search based Sudoku solver.

[3] Yue, Tai-Wen & Lee, Zou-Chung. (2006). Sudoku Solver by Q'tron Neural Networks. 4113. 943-952. 10.1007/11816157_115.

[4] Weber, Tjark. (2021). A SAT-based Sudoku solver.

THANK YOU

Harshit Mishra
Karthik Goel
Alokam Nikhitha