

AI Assignment 3 Report

CSP to solve Sudoko

Team Members

- | | |
|------------------|----------|
| ● Ahmed Ehab | 22010494 |
| ● Mahmoud Hesham | 22011201 |

OVERVIEW

Sudoku is a logic-based number-placement game that challenges players to fill a **9x9** grid with digits from **1 to 9**. The objective is to complete the grid in such a way that each row, each column, and each of the nine 3x3 subgrids **contains all of the digits from 1 to 9 without repetition**.

The game has 3 difficulties for generating a random board

Easy generates a board with 60 cells filled

Medium generates a board with 40 cells filled

Hard generates a board with 30 cells filled

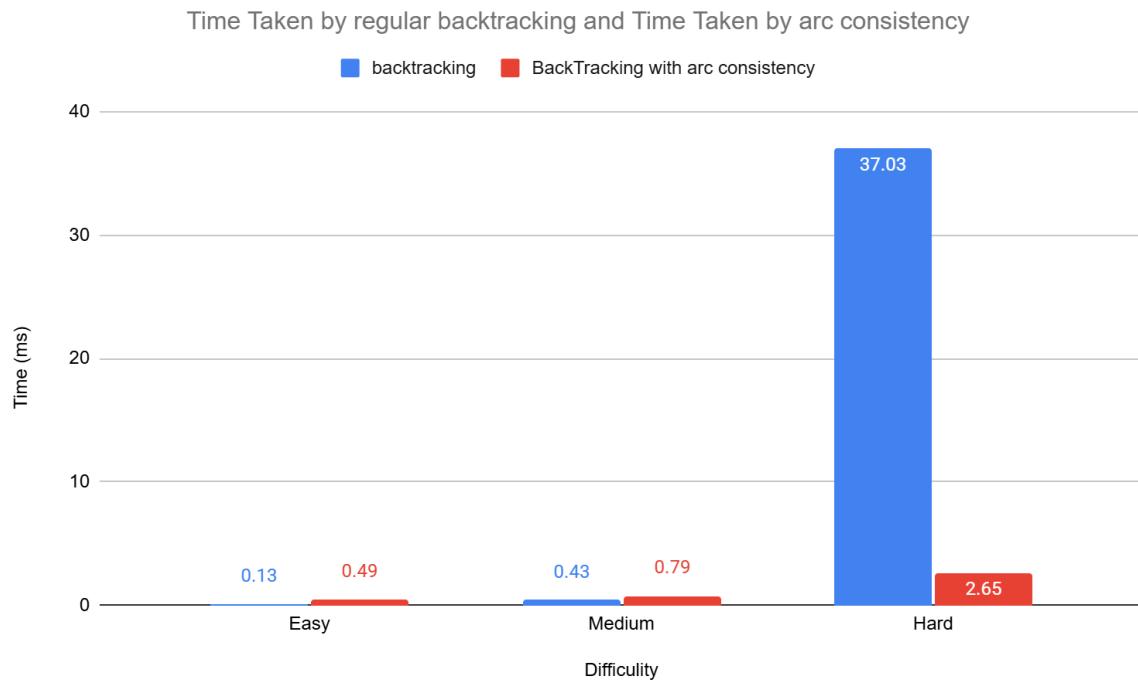
Generated cells are guaranteed to have a unique solution

But the game will allow the user to enter boards that has more than 1 solution

Algorithms Implemented

1. Backtracking
 - To solve the board
 - Check solvability
2. Backtracking with ARC consistency
 - To solve the board

Comparison



Arc consistency grows slower than backtracking with increasing number of empty cells

Data Structures and Implementation Details

Arc Consistency (AC-3 Algorithm)

Purpose: Constraint satisfaction algorithm that reduces the domain of variables (cell values) by removing values that violate Sudoku constraints.

Key Data Structures:

- Domains: List[List[Set]] - 9×9 grid of sets, each containing possible values (1-9) for that cell
- Queue: deque - stores cells with assigned values that need constraint propagation
- Arced Set: tracks which cells have been processed to avoid cycles

Algorithm Steps:

1. Initialize domains: assigned cells get singleton sets, empty cells get {1-9}
2. Queue all assigned cells
3. For each queued cell with value v:
 - a. Remove v from all cells in the same row
 - b. Remove v from all cells in the same column
 - c. Remove v from all cells in the same 3×3 box
4. If a cell's domain becomes singleton (one value), queue it
5. If any cell's domain becomes empty, return unsolvable
6. Repeat until queue is empty

Time Complexity:

$O(n^3)$ per constraint check, typically $O(n^4)$ overall for 9×9 grid Space Complexity: $O(n^2)$ for domains storage

Backtracking with Arc Consistency

Purpose: Combines arc consistency with backtracking search to solve puzzles that AC alone cannot complete.

Algorithm:

BacktrackWithSteps(board, steps):

1. Apply arc consistency to current board
2. If inconsistent (empty domain found), return FAIL
3. Find empty cell with smallest domain (Most Constrained Variable heuristic)
4. If no empty cells, return SUCCESS (solved)
5. For each value in cell's domain:
 - a. Save domain state snapshot
 - b. Assign value to cell
 - c. Record "backtrack_assign" step with domain snapshot
 - d. Recursively call BacktrackWithSteps
 - e. If success, return SOLUTION
 - f. If failure, record "backtrack_revert" step with saved domains
6. Return FAIL (no valid assignment found)

Key Data Structures:

- **Domain Snapshots:** List[List[Set]] - complete backup of domains before each assignment
- **Steps Log:** List[Dict] - chronological record of all algorithm decisions

Heuristics:

- **Minimum Remaining Values (MRV):** Choose cell with smallest domain
- Speeds up search by creating most constrained decision first

Puzzle Generation

Algorithm:

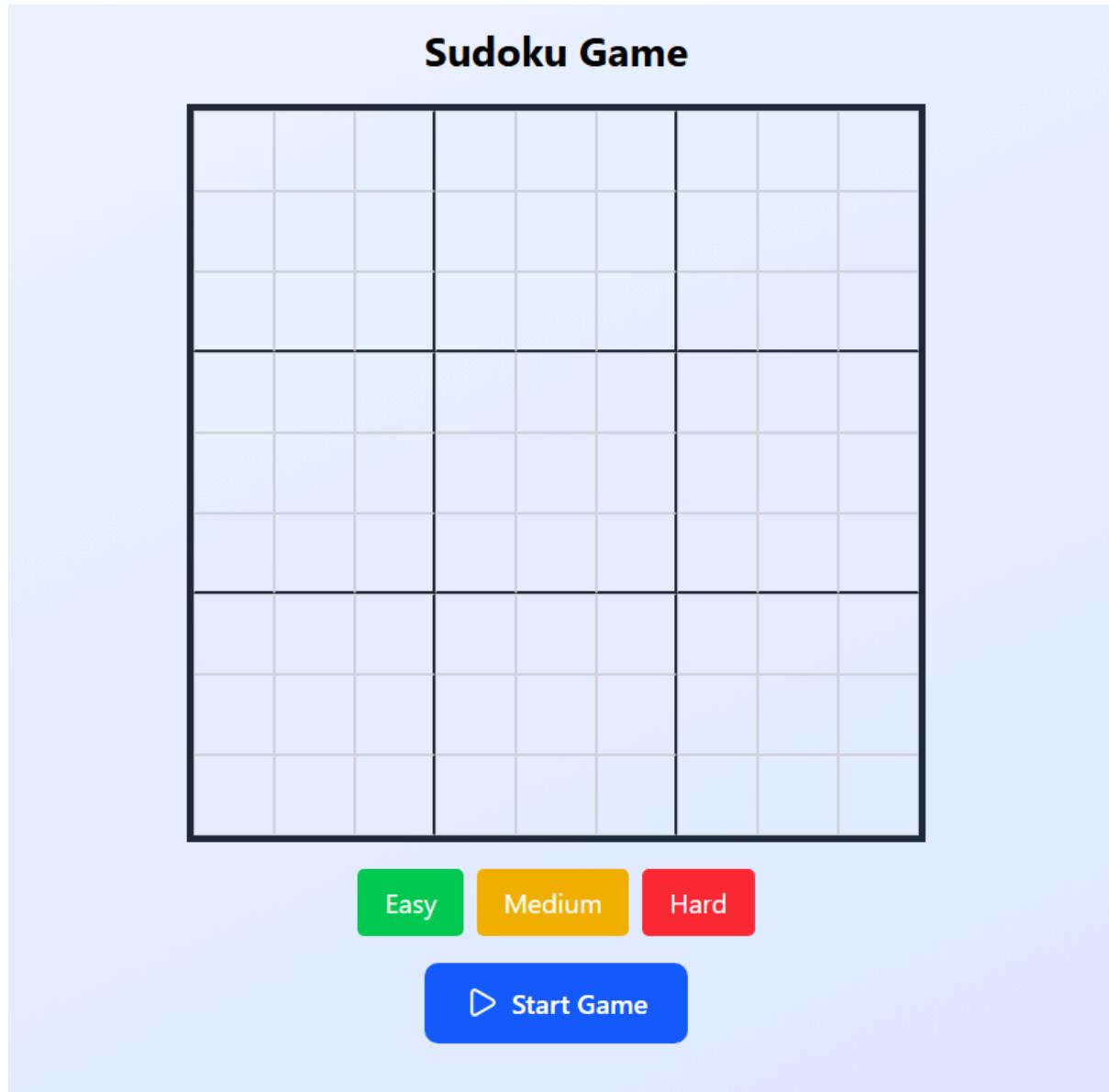
1. Generate complete valid board using randomized backtracking with `fill_board()`
2. Iteratively remove cells while maintaining unique solution
3. For each cell removal:
 - a. Check if puzzle still has exactly 1 solution using `get_number_of_solutions()`
 - b. Keep removal only if unique solution maintained
 - c. Restore cell if removal creates multiple solutions

Difficulty Levels:

- **Easy:** 60 filled cells (30 empty)
- **Medium:** 40 filled cells (40 empty)
- **Hard:** 20 filled cells (60 empty)

Sample runs

First page



The page after generating a medium difficulty board

Sudoku Game

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

Easy Medium Hard

▷ Start Game

Generated medium puzzle in 8.79ms

After pressing start game

Playing Sudoku

New Game

Undo Redo Check Solvability Solve AI Mode

Board is solvable and has only 1 solution

After pressing solve (this uses regular backtracking to solve the board)

Playing Sudoku

New Game

Undo Redo Check Solvability Solve AI Mode

Solved in 0.25ms

Placing an invalid number and press check solvability

The screenshot shows a 9x9 Sudoku grid. Some cells contain black numbers (2, 4, 7, 3, 5, 6, 9, 8, 1) and some are empty or have gray backgrounds. A message at the bottom states: "Invalid board: duplicate values found in row, column, or box."

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

Invalid board: duplicate values found in row, column, or box.

Pressed undo then AI mode

After 5 steps

AI Solver Mode Solved in 0.76ms [New Game](#)

Solver Steps

- Step 1: Arc Consistency**
Arc from (0, 1) to (0, 0)
Remove value 2 from domain
- Step 2: Arc Consistency**
Arc from (0, 1) to (0, 2)
Remove value 2 from domain
- Step 3: Arc Consistency**
Arc from (0, 1) to (0, 3)
Remove value 2 from domain
- Step 4: Arc Consistency**
Arc from (0, 1) to (0, 6)
Remove value 2 from domain
- Step 5: Arc Consistency**
Arc from (0, 1) to (1, 1)
Remove value 2 from domain
- Step 6: Arc Consistency**
Arc from (0, 1) to (3, 1)
Remove value 2 from domain
- Step 7: Arc Consistency**

< Previous Step 5 / 328 Next >

After more steps (the star means that this numbers where inferred)

AI Solver Mode Solved in 0.96ms [New Game](#)

Remove value 5 from domain

- Step 238: Arc Consistency**
Arc from (6, 4) to (8, 4)
Remove value 5 from domain
- Step 239: Arc Consistency**
Arc from (6, 4) to (7, 5)
Remove value 5 from domain
- Step 240: Arc Consistency**
Arc from (6, 4) to (8, 3)
Remove value 5 from domain
- Step 241: Arc Consistency**
Arc from (6, 4) to (8, 5)
Remove value 5 from domain
- Step 242: Arc Consistency**
Arc from (7, 0) to (7, 5)
Remove value 8 from domain
- Step 243: Arc Consistency**
Arc from (7, 0) to (7, 6)
Remove value 8 from domain
- Step 244: Arc Consistency**
Arc from (7, 3) to (7, 2)

< Previous Step 242 / 369 Next >

In the end

AI Solver Mode

Solved in 0.62ms [New Game](#)

Inferred **6** at cell **(0, 8)** (via arc consistency)

Step 363: Arc Consistency
Arc from **(4, 8)** to **(5, 8)**
Remove value **9** from domain

Step 364: Arc Consistency
Arc from **(5, 7)** to **(5, 8)**
Remove value **4** from domain

Step 365: Arc Inferred
Inferred **3** at cell **(5, 8)** (via arc consistency)

Step 366: Arc Consistency
Arc from **(1, 1)** to **(1, 8)**
Remove value **6** from domain

Step 367: Arc Inferred
Inferred **5** at cell **(1, 8)** (via arc consistency)

Step 368: Arc Consistency
Arc from **(1, 1)** to **(0, 1)**
Remove value **6** from domain

Step 369: Arc Inferred
Inferred **3** at cell **(0, 1)** (via arc consistency)

< Previous Step 369 / 369 Next >