Introduction

1. **Artificial Intelligence (AI)**
   * AI mimics human intelligence to solve problems.
   * AI uses a "state space," which includes:
     + **Start State:** Initial condition.
     + **Goal State:** Desired outcome.
     + **Rules:** Define how states change.
2. **Sudoku Game Basics**
   * A 9x9 grid divided into nine 3x3 smaller grids.
   * Each row, column, and 3x3 box must have numbers 1-9 without repetition.
   * Sudoku trains logical, not arithmetic skills.
3. **Problem in Sudoku**
   * How to find solutions using AI techniques.
   * **Two algorithms are studied:**
     + Breadth First Search (BFS).
     + Depth Limited Search (DLS).

**Algorithms Overview**

1. **Breadth First Search (BFS):**
   * **How it works:**
     + Starts at the root node (empty grid).
     + Explores all possible numbers for the first box, then the next box, and so on.
   * **Advantages:**
     + Always finds a solution if one exists.
     + Can find multiple solutions.
   * **Disadvantages:**
     + Uses lots of memory (stores all nodes).
     + Slower because it explores every possibility at every level.
2. **Depth Limited Search (DLS):**

* **How it works:**
  + Searches deeply, exploring one branch of possibilities until a solution is found or a limit is reached.
  + Backtracks if no solution is found at the current depth.
* **Advantages:**
  + Faster because it doesn't explore all shallow levels.
  + Uses less memory (only stores the current active path).
* **Disadvantages:**
  + Requires a pre-set depth limit.
  + If the depth limit is too small, it might miss solutions.

**How BFS Solves Sudoku**

1. Start at the root node (empty grid).
2. Explore all possible numbers for the first box (Box-0).
3. Move to the next box (Box-1):
   * If it contains a fixed number, move on.
   * Otherwise, explore all valid numbers.
4. Repeat until all 81 boxes are filled or no solution is found.
5. If a solution exists, BFS will find it by systematically exploring every possibility.

**How DLS Solves Sudoku**

1. Start at the root node (empty grid).
2. Explore one possible number for the first box (Box-0).
3. Move to the next box and explore one valid number:
   * If no valid numbers exist, backtrack to the previous box.
4. Continue searching until a solution is found or all options are exhausted.
5. If the depth limit is reached without a solution, backtrack or increase the limit.

Key Differences Between BFS and DLS

|  |  |  |
| --- | --- | --- |
| **Aspect** | **BFS** | **DLS** |
| **Search Style** | Explores all nodes level by level. | Explores one branch deeply first. |
| **Speed** | Slower due to broad exploration. | Faster for deeper solutions. |
| **Memory Use** | Requires more memory (stores all nodes). | Requires less memory. |
| **Guarantee** | Always finds all solutions. | Can miss solutions if depth is wrong. |

**Conclusions from Study**

1. DLS is more efficient and faster for Sudoku.
2. BFS is better for finding all possible solutions.
3. Neither algorithm guarantees a solution for every Sudoku puzzle.
4. Research is limited to 3x3 Sudoku grids; future work could explore larger grids or other algorithms (e.g., A\*).