RIDA BATOOL
BSDSF23M007
ARTIFICIAL INTELLIGENCE (LAB REPORT)

Comparative Analysis of Search Algorithms

1. Introduction

This report presents a comparative analysis of various **uninformed and informed search algorithms** for solving a given problem. Uninformed search algorithms explore the state space **without any additional domain knowledge**, whereas informed search algorithms **use heuristics** to improve efficiency.

We analyze and compare the performance of:

- **Uninformed Search**: Breadth-First Search (BFS), Depth-First Search (DFS), Depth-First Iterative Deepening (DFID)
- **Informed Search**: A* Search (Manhattan Heuristic), A* Search (Out-of-Place Tiles Heuristic), Best-First Search (Manhattan Heuristic), Best-First-Search (Out-of-Place Tiles Heuristic)

Use Cases of Search Algorithms

- **BFS**: Guarantees the shortest path but is memory-intensive (stores all explored nodes).Explores nodes level by level.Uses a queue (FIFO) to track visited nodes.
- **DFS**: Uses less memory but may not find the optimal path.Efficient for problems with deep solution. Uses a stack (LIFO) to track visited nodes.
- **DFID**: Balances between BFS (when BFS consumes too much memory) and DFS (when DFS is memory efficient but has a depth-limit issues) , finding the shortest path with better memory efficiency. Slower as compare to DFS due to repetitive depth-limited searches.
- **A*** (Manhattan & Out-of-Place Tiles): Uses heuristics to efficiently find the optimal solution.Uses a cost function f(n) = g(n) + h(n), where g(n) is the actual cost and h(n) is the heuristic estimate to the goal.High memory consumption as it stores many states.
- **Best-First Search**: Fastest among informed searches but does not guarantee the shortest path.

2. Experimental Setup

1. **Test Problem:** Solving an **8-Puzzle Problem**
2. **Initial State:**

```
4 7 8
3 6 5
1 2 _
```

3. **Goal State:**

```
1 2 3
4 5 6
7 8 _
```

4. **Performance Metrics:**

   - Execution Time (using time/timeit module)
   - Memory Consumption (using memory_ profiler module)
   - Number of Moves Suggested
   - Number of Nodes Visited

3. Results and Observations

3.1 Execution Time Comparison

Algorithm	**Execution Time (s)**
BFS	9.837321 seconds
DFS	3.737193 seconds
DFID	5.121319 seconds
A* (Manhattan)	0.083439 seconds
A* (Out-of-Place Tiles)	1.699603 seconds
Best-First Search (Manhattan)	0.055634 seconds
Best-First Search (Out-of-Place Tiles)	0.143751 seconds

3.2 Memory Consumption Comparison

Algorithm	**Memory Used (MB)**
BFS	160.18 MB
DFS	156.59 MB
DFID	158.92 MB
A* (Manhattan)	42.88 MB
A* (Out-of-Place Tiles)	59.63 MB
Best-First Search (Manhattan)	42.25 MB
Best-First Search(Out-of-Place Tiles)	43.65 MB

3.3 Number of Moves & Nodes Visited

Algorithm	**Moves Taken**	**Nodes Visited**
BFS	25	292559
DFS	107767	269175
DFID	33	13361
A* (Manhattan)	25	1008
A* (Out-of-Place Tiles)	25	19302
Best-First Search (Manhattan)	69	272
Best-First Search(Out-of-Place Tiles)	217	1397

4. Comparative Analysis

Execution Time

- **Best-First Search** was the **fastest (0.055s)** due to its greedy nature but produced a sub optimal solution.
- **A*** with **Manhattan heuristic (0.083s)** was the best performer among optimal algorithms.
- **Uninformed searches** (BFS, DFS, DFID) took significantly longer.

Memory Usage

- **Best-First Search (42.25 MB)** was the most memory-efficient.
- **DFS (156.59 MB)** used less memory than BFS but was unreliable for finding the shortest path.
- **BFS (160.18 MB)** and **DFID (158.92 MB)** required more memory.

Solution Optimality (Moves Taken)

- *BFS,A* (Out-of-Place Tiles) , and A* (Manhattan)* found the shortest path (25 moves).
- **Greedy Best-First Search (Out of Place Tiles)** found sub optimal path (217 moves).
- **DFS had a longer and less sub-optimal (107767 moves)** due to its depth-first nature.

Nodes Visited

- **Uninformed algorithms explored more nodes** than **informed ones**, with **A*** (Manhattan) visiting only 1008 and Best First Search (Manhattan) visiting only 272 nodes respectively compared to **BFS's** 292559.

5. Uninformed vs. Informed Search

Uninformed Search

- **BFS**: Guaranteed **shortest path** but slow and memory-intensive.
- **DFS**: Fast and memory-efficient but found **a long and sub optimal path**.
- **DFID**: Found **shortest path** like BFS but **used less memory**.

Informed Search

- **A*** (Manhattan) was the most **balanced** algorithm—efficient, optimal, and fast.
- **A*** (Out-of-Place Tiles) was slightly worse than Manhattan but still effective.
- **Best-First Search** was the fastest but produced **a longer path**.

Heuristic Comparison

- **Manhattan heuristic was better than Out-of-Place Tiles**:
  - Explored **fewer nodes (1008 vs. 19302 for A* and 272 vs. 1397 for Best First Search respectively)**
  - Found **a shorter path (69 vs. 217 moves for Best First Search)**

6. Conclusion

- **A*** (Manhattan) was the best overall, balancing **execution time, memory, and optimal**.
- **Best-First Search was the fastest** but provided a longer and sub optimal path.
- **BFS and DFID were reliable** but slow and memory-intensive.
- **DFS was unpredictable**, sometimes finding long paths.
- **Using heuristics significantly improves efficiency**, with **Manhattan heuristic performing the best**.