CSAI 301 - Project Phase 1

Vacuum Cleaner Robot Problem

Ahmed Wael 202201415

Problem Description

The Vacuum Cleaner Robot Problem involves a robot operating in a 5x5 grid environment where certain cells are dirty and need to be cleaned. The robot must navigate the grid and clean all dirty cells while minimizing the total cost of operations.

With a floor size of 5*5 (25 cells total) - 10 cells dirty and the robot can take actions like Move Up, Down, Left, Right, and Clean and each action has a cost of 1

Algorithms Implemented

Breadth-First Search (BFS)

- Strategy: Explores level by level (FIFO queue)
- Complete: Yes
- Optimal: Yes (for uniform cost)
- Time Complexity: O(b^d)
- Space Complexity: O(b^d)

Depth-First Search (DFS)

- Strategy: Explores deep paths first (LIFO stack)
- Complete: Yes (with depth limit)
- Optimal: Yes
- Time Complexity: O(b^m)
- Space Complexity: O(bm)

Uniform Cost Search (UCS)

- Strategy: Expands lowest cost node first
- Complete: Yes

- Optimal: Yes
- Time Complexity: O(b^(C*/ε))
- Space Complexity: O(b^(C*/ε))

Iterative Deepening Search (IDS)

- Strategy: Combines BFS and DFS benefits
- Complete: Yes
- Optimal: Yes (for uniform cost)
- Time Complexity: O(b^d)
- Space Complexity: O(bd)

Greedy Best-First Search

- Strategy: Expands node with lowest h(n)
- Complete: Yes
- Optimal: Yes
- Fast but potentially suboptimal

A* Search

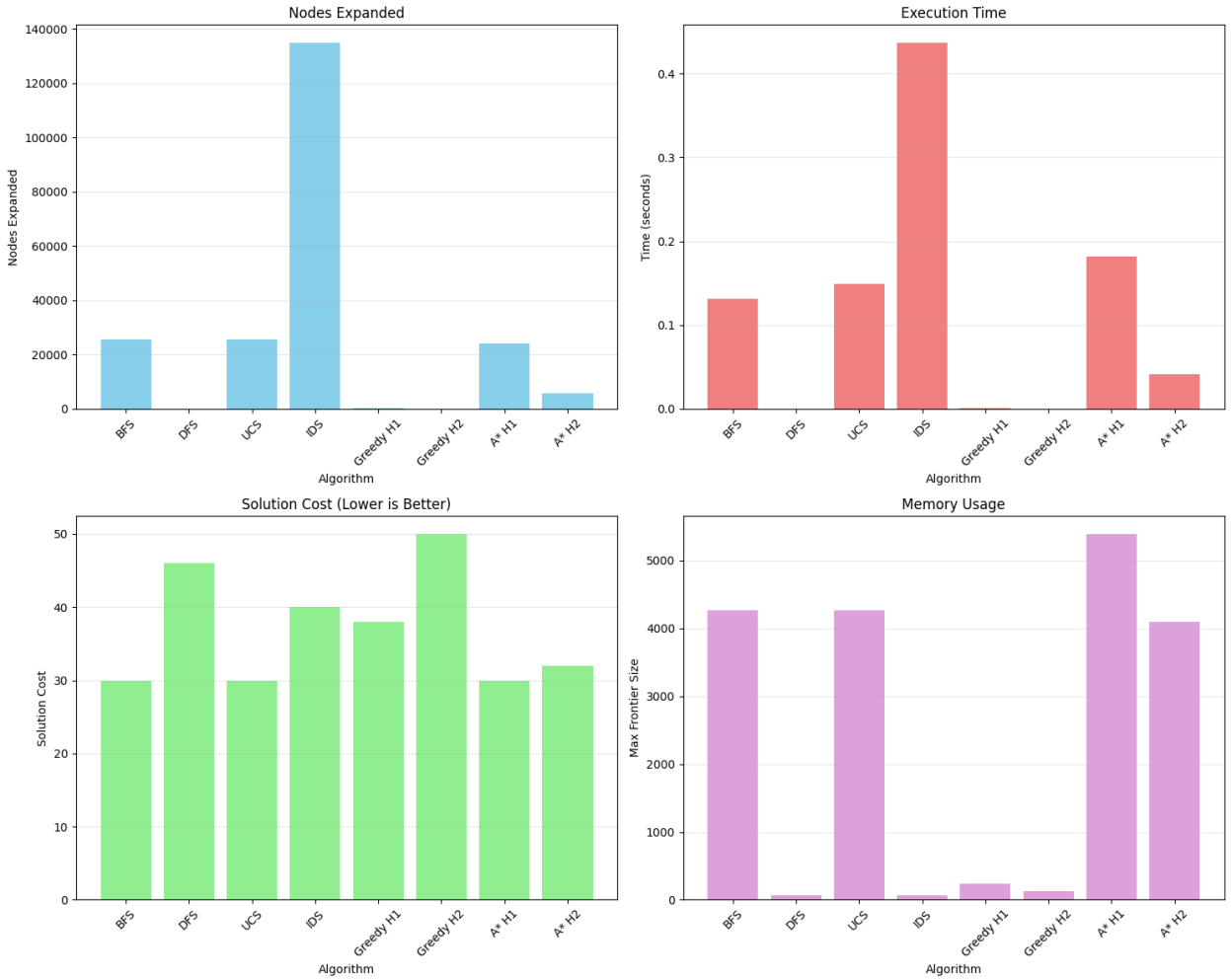
- Strategy: Expands node with lowest f(n) = g(n) + h(n)
- Complete: Yes
- Optimal: Yes (with admissible heuristic)
- Time Complexity: O(b^d)
- Space Complexity: O(b^d)

Performance Results

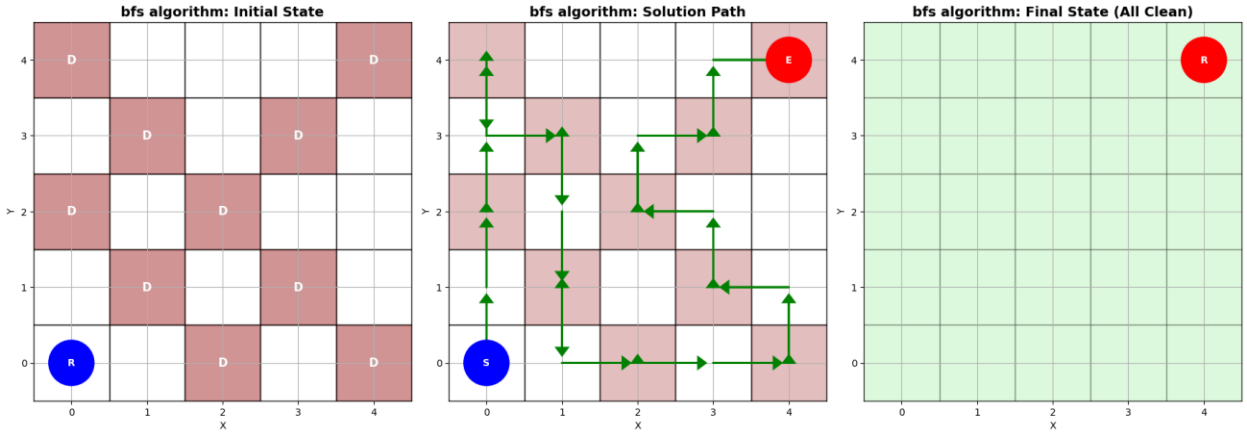
Performance Table:

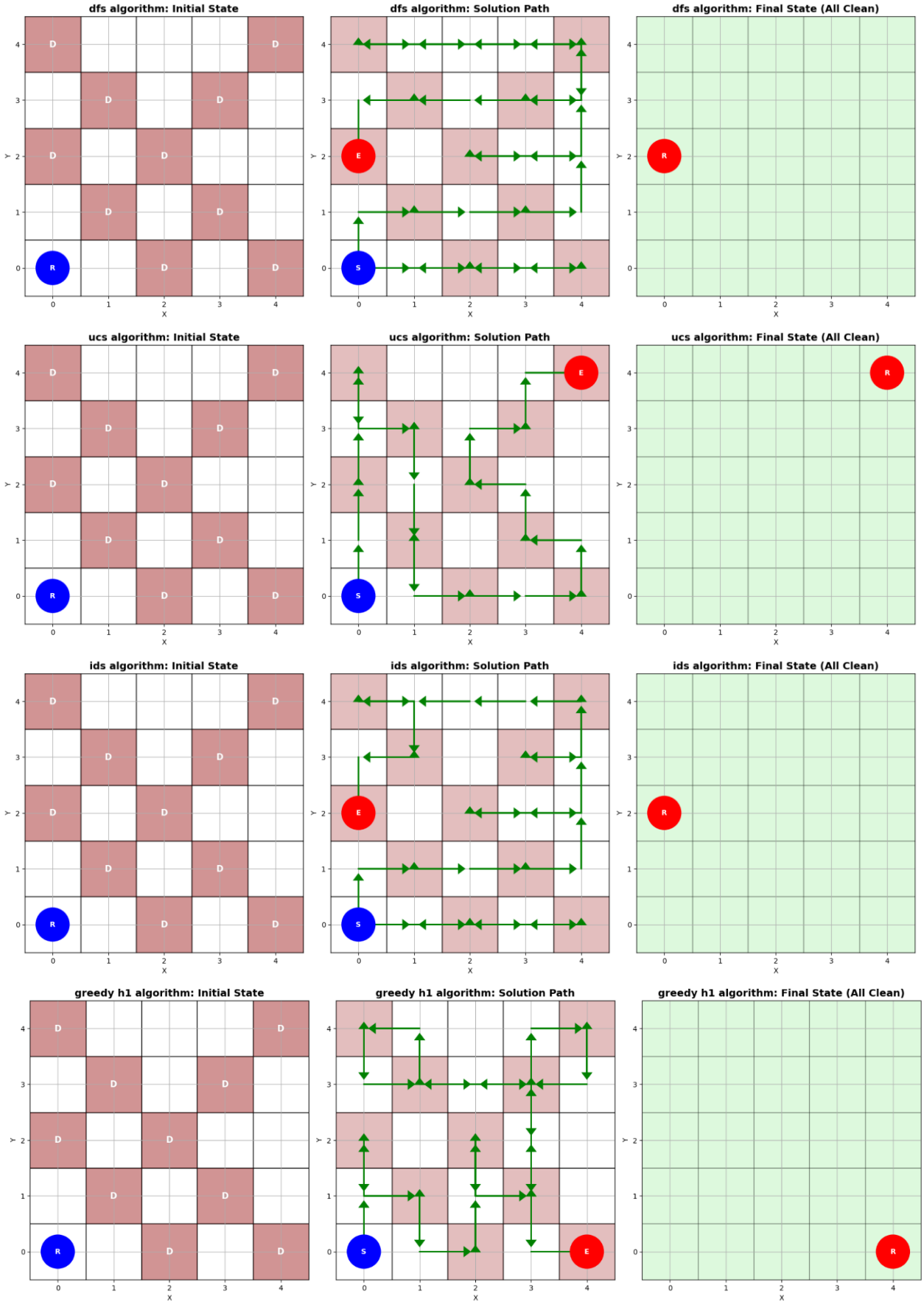
```
================================================================================================
PERFORMANCE COMPARISON TABLE
================================================================================================
Algorithm            Success    Cost       Nodes Expanded       Time (s)             Memory
------------------------------------------------------------------------------------------------
BFS                  1          30         25483                0.1311               4260
DFS                  1          46         60                   0.0002               71
UCS                  1          30         25483                0.1497               4260
IDS                  1          40         134741               0.4364               65
Greedy H1            1          38         222                  0.0012               234
Greedy H2            1          50         78                   0.0004               125
A* H1                1          30         24130                0.1817               5387
A* H2                1          32         5758                 0.0414               4092
================================================================================================
```

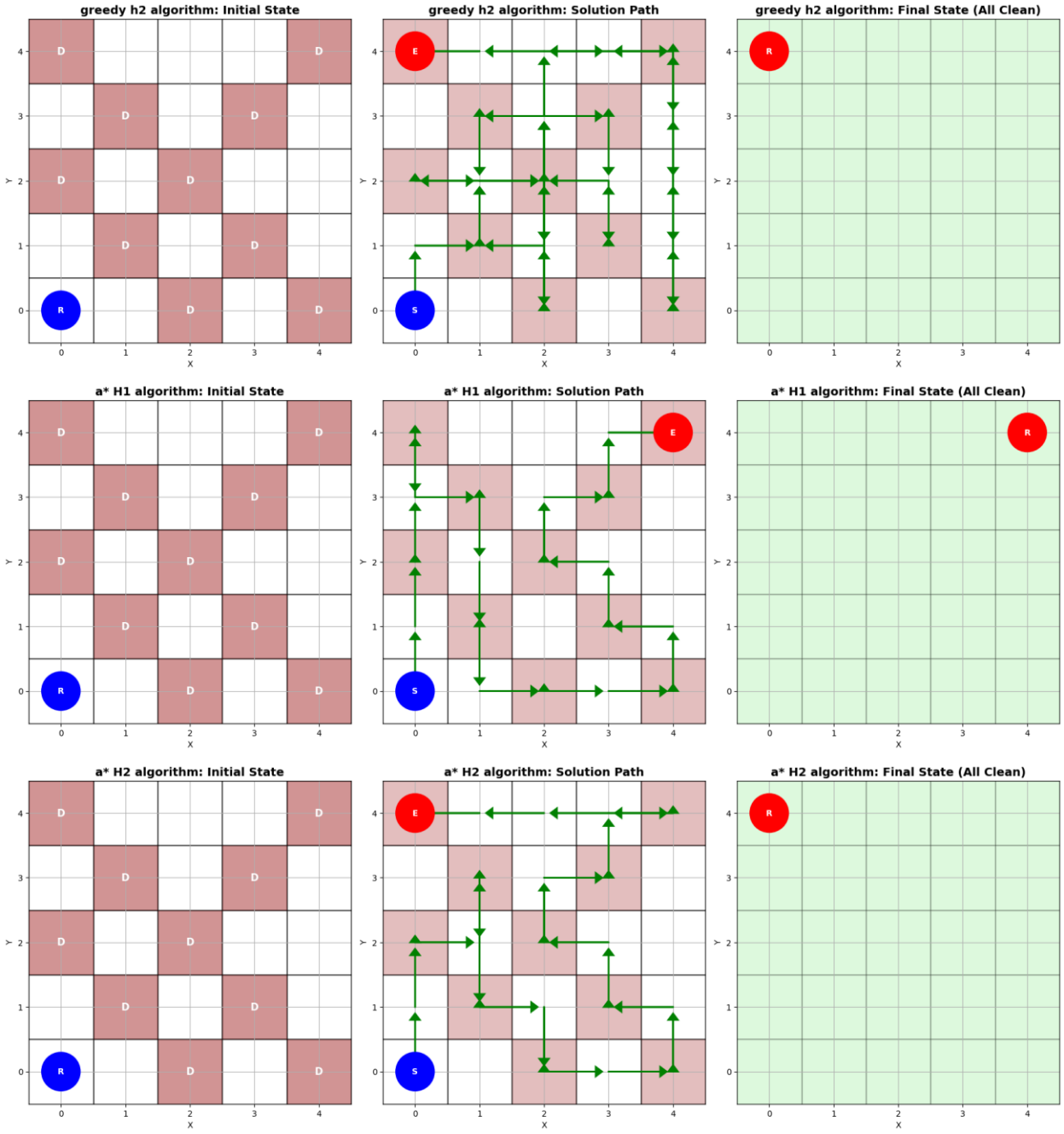
Performance Charts:



Solution Path for every algorithm




dfs algorithm: Initial State
dfs algorithm: Solution Path
dfs algorithm: Final State (All Clean)
ucs algorithm: Initial State
ucs algorithm: Solution Path
ucs algorithm: Final State (All Clean)
ids algorithm: Initial State
ids algorithm: Solution Path
ids algorithm: Final State (All Clean)
greedy h1 algorithm: Initial State
greedy h1 algorithm: Solution Path
greedy h1 algorithm: Final State (All Clean)
D
R
S
E
X
Y


greedy h2 algorithm: Initial State
greedy h2 algorithm: Solution Path
greedy h2 algorithm: Final State (All Clean)
a* H1 algorithm: Initial State
a* H1 algorithm: Solution Path
a* H1 algorithm: Final State (All Clean)
a* H2 algorithm: Initial State
a* H2 algorithm: Solution Path
a* H2 algorithm: Final State (All Clean)
D
R
S
E
X
Y