Warehouse Robot Pathfinding System – Phase I Report

==========

1. Abstract

This project presents the design and implementation of a warehouse robot navigation system using classical AI search algorithms on a dynamically generated weighted grid. The robot must navigate from a start point to a pickup location, then continue to a drop-off point while avoiding obstacles. Six algorithms—BFS, DFS, IDDFS, UCS, Greedy, and A*—were implemented and evaluated. Metrics such as optimality, completeness, memory usage, visited nodes, and execution time were recorded.

2. Problem Modeling

The warehouse is represented as a 10×10 grid with obstacles (|), normal terrain (*), medium-cost terrain (~), and high-cost terrain (^). A valid map ensures reachability between S $\rightarrow$ O and O $\rightarrow$ D using BFS.

A state is defined as (row, column, has_item), enabling a two-phase pathfinding structure.

3. Algorithms

- BFS: Uninformed, complete, optimal only in unweighted grids. High memory usage.
- DFS: Uninformed, not complete in infinite spaces, not optimal. Low memory usage.
- IDDFS: Complete, optimal in unweighted grids, but slow; moderate memory.
- UCS: Complete and optimal with weighted costs; high memory usage due to priority queue.
- Greedy Best-First: Not complete, not optimal; very low memory.
- A*: Complete and optimal with admissible heuristics; better memory use than UCS.

4. Experimental Results

SUMMARY TABLE

Algorithm | Path Length | Visited | Expansions | Time (s) | Complete | Optimal | Memory

---

BFS | 14 | 118 | 103 | 0.00032 | Yes | Only unweighted | High

DFS | 20 | 49 | 30 | 0.00027 | No | No | Very Low

IDDFS | 28 | 64 | 27 | 0.00608 | Yes | Yes (unweighted)| Low

UCS | 14 | 100 | 100 | 0.00107 | Yes | Yes (weighted) | High

Greedy | 30 | 51 | 64 | 0.00052 | No | No | Very Low

A* | 14 | 24 | 24 | 0.00038 | Yes | Yes | Moderate

5. Discussion

Uninformed algorithms expand significantly more nodes and fail to consider terrain cost. DFS performs worst in terms of path quality. IDDFS is complete but slow.

UCS and A* produce optimal paths in weighted environments. A* is the most efficient, requiring the fewest expansions while maintaining optimality.

Greedy is fast but unreliable and produces long paths.

6. Conclusion

A* is the best method for intelligent warehouse navigation under weighted grid constraints. BFS is useful for uniform grids, while UCS provides optimality without heuristic guidance at the cost of memory consumption.

This concludes Phase I. Phase II will extend this work into reinforcement learning.