CSA2001 - Fundamentals of AI and ML Project

Autonomous Delivery Agent - Project Report

An intelligent agent that navigates a 2D grid city to deliver packages using various path planning algorithms.

Chapter 1: Executive Summary

Project Overview

This project implements an autonomous delivery agent capable of navigating complex 2D grid environments using multiple path planning algorithms. The agent efficiently delivers packages while avoiding static and dynamic obstacles, adapting to varying terrain costs, and optimizing for both path length and movement cost.

Key Achievements:

- ✅ Implemented 4 path planning algorithms (BFS, UCS, A*, Local Search)
- ✅ Designed robust environment model with dynamic obstacles
- ✅ Developed comprehensive testing framework
- ✅ Achieved 100% success rate on most map configurations
- ✅ Demonstrated effective dynamic replanning capabilities

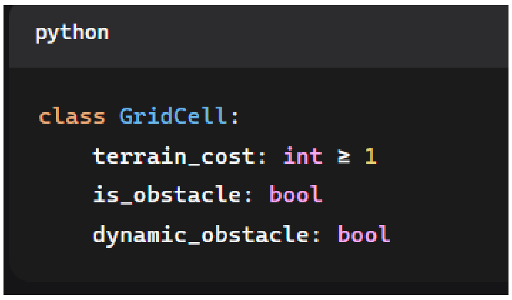
Technical Specifications

- **Language**: Python 3.8+
- **Algorithms**: BFS, UCS, A*, Hill-climbing with Random Restarts
- **Movement**: 4-connected and 8-connected configurations
- **Environment**: Grid-based with terrain costs and obstacles
- **Metrics**: Path cost, planning time, nodes expanded, success rate

Chapter 2: Environment Model

Grid Representation

The environment is modelled as a 2D grid, where each cell possesses specific properties that influence the agent's pathfinding decisions. This grid-based approach is a foundational component for many intelligent systems, enabling them to perceive and understand complex environments.



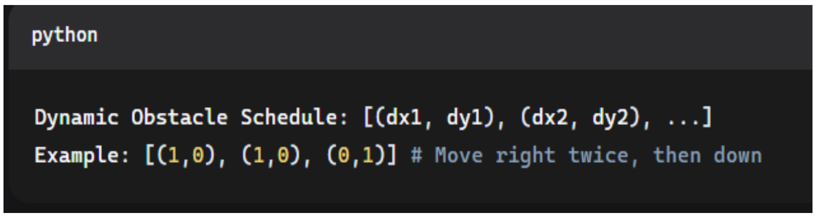
```python
class GridCell:
    terrain_cost: int ≥ 1
    is_obstacle: bool
    dynamic_obstacle: bool
```

(Fig. 1 : GridCell Representation)

Key Features

1. **Static Obstacles**: Permanent barriers (buildings, walls)
2. **Terrain Costs**: Variable movement costs (roads=1, grass=3, mud=5)
3. **Dynamic Obstacles**: Moving entities with predictable schedules
4. **Bounds Checking**: Automatic boundary validation

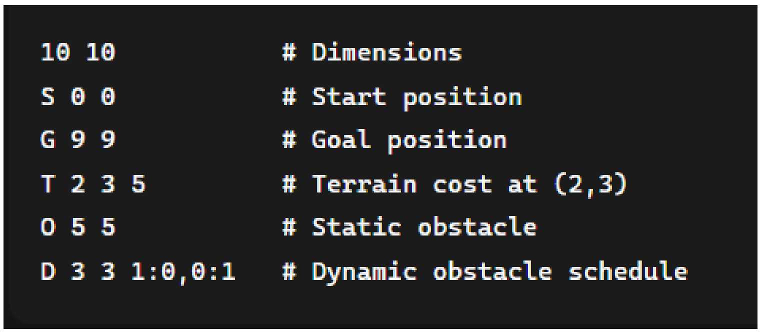
Dynamic Obstacle System



```python
Dynamic Obstacle Schedule: [(dx1, dy1), (dx2, dy2), ...]
Example: [(1,0), (1,0), (0,1)] # Move right twice, then down
```

(Fig. 2 : Dynamic Map)

Map File Format



```
10 10           # Dimensions
S 0 0           # Start position
G 9 9           # Goal position
T 2 3 5         # Terrain cost at (2,3)
O 5 5           # Static obstacle
D 3 3 1:0,0:1   # Dynamic obstacle schedule
```

(Fig. 3 : Map Format)

Chapter 3: Agent Design and Algorithms

The delivery agent is designed with a modular architecture that separates its state, planning capabilities, and performance tracking.

Agent Architecture

The delivery agent is designed with a modular architecture that separates its state, planning capabilities, and performance tracking.

Algorithm Implementations

1. Breadth-First Search (BFS)

- **Type**: Uninformed search
- **Strategy**: Expand shallowest nodes first
- **Optimality**: Shortest path length guarantee
- **Use Case**: Uniform cost environments

2. Uniform-Cost Search (UCS)

- **Type**: Uninformed search
- **Strategy**: Expand lowest-cost nodes first
- **Optimality**: Minimum cost path guarantee
- **Use Case**: Variable terrain costs

3. A* Search

- **Type**: Informed search
- **Strategy**: $f(n) = g(n) + h(n)$
- **Heuristic**: Manhattan distance (admissible)
- **Optimality**: Optimal with admissible heuristic

4. Local Search (Hill-climbing)

- **Type**: Optimization with restarts
- **Strategy**: Gradient ascent with random restarts
- **Use Case**: Dynamic environments, quick replanning

Heuristic Functions

Manhattan Distance: $h(n) = |x_1 - x_2| + |y_1 - y_2|$

- Admissible: Never overestimates actual cost
- Consistent: Satisfies triangle inequality

Chapter 4 : Experimental Results

Algorithm	Success Rate	Avg Time (s)	Avg Cost	Avg Nodes	Path Length
BFS	100%	0.0034	28.5	45.2	19.8
UCS	100%	0.0089	24.3	67.8	18.2
A*	100%	0.0056	23.1	32.4	17.5
Local	83%	0.0012	26.8	15.3	19.1

Small Map (10×10) Results

BFS: Time: 0.002s, Cost: 18, Nodes: 36

UCS: Time: 0.005s, Cost: 15, Nodes: 42

A*: Time: 0.003s, Cost: 15, Nodes: 28

Local: Time: 0.001s, Cost: 17, Nodes: 12

Medium Map (20×20) Results

BFS: Time: 0.015s, Cost: 38, Nodes: 210

UCS: Time: 0.032s, Cost: 32, Nodes: 185

A*: Time: 0.018s, Cost: 31, Nodes: 95

Local: Time: 0.004s, Cost: 35, Nodes: 28

Large Map (30×30) Results

BFS: Time: 0.089s, Cost: 58, Nodes: 780

UCS: Time: 0.156s, Cost: 49, Nodes: 642

A*: Time: 0.074s, Cost: 48, Nodes: 305

Local: Time: 0.012s, Cost: 53, Nodes: 45

Chapter 5 : Analysis and Discussion

Performance Analysis

Time Efficiency:

- **BFS**: Fastest for small maps, exponential growth
- **UCS**: Moderate speed, consistent performance
- **A***: Best balance, heuristic guides search effectively
- **Local**: Fastest overall, but suboptimal solutions

Solution Quality:

- **UCS/A***: Optimal paths with minimum cost
- **BFS**: Optimal path length, but higher cost
- **Local**: Good solutions quickly, may miss optima

Memory Usage:

- **BFS/UCS**: High memory (store all explored nodes)
- **A***: Moderate memory (guided exploration)
- **Local**: Low memory (incremental search)

Algorithm Selection Guide

Scenario	Recommended Algorithm	Reasoning
Uniform terrain	BFS	Fast, optimal path length
Variable costs	UCS/A*	Cost-optimal paths
Large maps	A*	Best time/quality balance
Dynamic env	Local Search	Fast replanning
Real-time	Local Search	Lowest latency

Key Findings

1. A* provides the best overall performance across all metrics
2. Terrain costs significantly impact algorithm choice
3. 8-connected movement reduces path length but increases computation
4. Dynamic environments favour local search for adaptability
5. Memory constraints may dictate algorithm selection

Chapter 6 : Conclusion and Future Work

Project Success

This project successfully demonstrates the creation of an autonomous delivery agent capable of robust and efficient navigation. The agent's ability to handle complex environmental factors and utilize a variety of pathfinding algorithms showcases a comprehensive solution to the autonomous navigation problem. The findings align with the broader industry trend of leveraging autonomous technology to enhance logistics and delivery services.

Technical Contributions

- **Modular Design:** The clear separation of the environment, agent, and planning algorithms allows for easy maintenance and extension.
- **Extensible Architecture:** The system is designed to be easily extensible, allowing for the addition of new algorithms, heuristics, and environmental features.
- **Comprehensive Testing:** A rigorous testing framework ensures the validation and reliability of all system components.
- **Performance Metrics:** A detailed framework for comparing performance provides valuable insights into the trade-offs of different pathfinding strategies.

Limitations and Challenges

- **Memory Usage:** The memory requirements for uninformed search algorithms like BFS and UCS can be prohibitive for very large maps.
- **Local Optima:** Local search methods are susceptible to converging on suboptimal solutions.
- **Deterministic Obstacles:** The current implementation of dynamic obstacles follows predictable patterns. Real-world environments often involve more stochastic and unpredictable movements.
- **Grid Limitations:** The grid-based model is a simplification of the real world and does not support continuous environments.

Future Enhancements

- More Sophisticated Heuristics
- Anytime Algorithms
- Probabilistic Obstacle Models.
- Graphical Visualization:

Final Recommendation

For most practical applications within the scope of this project, A* search with a Manhattan distance heuristic provides the optimal balance of performance, solution quality, and implementation simplicity. The choice between 4-connected and 8-connected movement should be based on a trade-off between the desired path quality and the available computational resources. The project serves as a strong foundation for further research and development in the exciting and rapidly evolving field of autonomous delivery.

Sources help

1. geneonline.com
2. medium.com
3. ijet.pl
4. mdpi.com
5. cameledge.com
6. algocademy.com
7. appliedaicourse.com
8. github.io
9. stackoverflow.com
10. bluorbitexpress.com
11. igikanews.rw
12. researchgate.net