**CSA2001**

**1. Introduction** This project covers the implementation of an autonomous agent for package delivery in a simulated 2D city. The key challenge is efficient pathfinding in a complex environment containing static, variable-cost, and dynamic obstacles, all under a strict fuel constraint. The implemented agent is rational, using an informed search algorithm to minimize cost and a local search strategy to react to unforeseen environmental changes.

**2. Environment & Agent Model** The environment is a 2D grid parsed from .txt files. Cells are categorized as start, goal, wall, or standard terrain. Terrain cells have associated integer movement costs. Dynamic obstacles are defined in a .json file, which specifies their coordinates at each time step. The agent's state is defined by its position (r, c), current fuel, and the current time step, which is critical for avoiding dynamic obstacles.

**3. Algorithm Implementation** *3.1 A Search:*\* The primary planner is a time-aware A\* search. The state representation in the search is a (position, time) tuple. This prevents collisions by ensuring the agent and a dynamic obstacle do not occupy the same cell at the same time. The heuristic function, h(n), is the Manhattan distance, which is admissible for a 4-connected grid with a minimum cost of 1. This guarantees A\* finds a cost-optimal path. Fuel constraints are checked before adding any new state to the frontier.

**3.2 Local Search Replanning:** To handle unexpected obstacles, a fast local search is used. When the agent's path is blocked, it attempts a constrained random walk from its current position to a viable "rejoin point" on its original path. This provides a rapid, "good enough" patch without the overhead of a full replan. If this fails, a full A\* replan is triggered as a fallback.

**4. Experimental Results** The A\* planner was evaluated on several static maps. Performance was measured by path cost and the number of nodes expanded during the search.

| Map | Path Cost | Nodes Expanded |
| --- | --- | --- |
| small | 5 | 8 |
| medium | 12 | 86 |
| large | 92 | 512 |

The data shows that nodes expanded, a measure of search complexity, scales reasonably with map size. The heuristic effectively prunes the search space.

**5. Analysis** A\* provides a significant performance gain over uninformed methods like UCS. The heuristic guides the search, drastically reducing the number of nodes expanded compared to the brute-force expansion of UCS. This makes A\* suitable for larger, more complex maps where an uninformed search would be computationally infeasible.

The local search replanner proved effective for simple blockages, but the full A\* fallback is essential for robustness, as seen in the demo where the local search failed and a full replan was required to find a new optimal path.

**6. Conclusion** The implemented agent successfully navigates complex and dynamic environments. The combination of a time-aware A\* for optimal initial planning and a robust replanning strategy creates an efficient navigation system. This project demonstrates the practical application of informed search algorithms in solving AI pathfinding problems.