Performance Analysis of Pathfinding Algorithms in a Dynamic Urban Delivery Simulation

1. Introduction
This report details the design and outcomes of a simulation project created to evaluate the efficacy of various pathfinding algorithms for autonomous delivery agents in complex, dynamic urban environments. The simulator models realistic constraints such as static obstacles, variable terrain costs, and time-dependent traffic blockages.

2. Simulation Environment
The environment consists of four distinct grid-based city maps (Bhopal, Indore, Shivpuri, Jabalpur), each with unique layouts and challenges:

- **Static Obstacles:** Impassable walls that define routes.
- **Variable Terrain:** Certain cells incur higher traversal costs, simulating difficult road conditions.
- **Dynamic Traffic:** Predefined cells become temporarily blocked at specific time intervals, introducing temporal challenges.

3. Agent Architecture
The delivery agent is an autonomous entity with the following key attributes and behaviors:

- **State Tracking:** Maintains current position, time step, fuel level, and delivery status.
- **Path Planning:** Utilizes one of four core algorithms (BFS, UCS, A*, Randomized Local Search) for initial route generation.
- **Dynamic Replanning:** Upon encountering an unexpected traffic blockage, the agent triggers a replanning routine using a randomized local search to find an alternative path.
- **Logging:** Records all significant events, including replanning triggers and delivery completion.

4. Algorithmic Approaches and Heuristics

- **Breadth-First Search (BFS):** An uninformed search strategy that expands all nodes at the present depth prior to moving deeper. It uses no heuristic.
- **Uniform-Cost Search (UCS):** Prioritizes the path with the lowest cumulative cost from the start node, effectively handling variable terrain costs.
- *A Search:** Combines the actual path cost (like UCS) with a heuristic estimate of the cost to the goal (Manhattan Distance), guiding the search efficiently.
- **Randomized Local Search:** A stochastic method used for replanning; it probabilistically explores neighboring cells to quickly find a viable detour around unexpected obstacles.

5. Experimental Findings
The algorithms were evaluated across all four cities based on average path cost, number of nodes explored, and success rate.

City	Algorithm	Path Cost	Nodes Explored	Success

Bhopal	A*	13	29	Yes
	UCS	13	41	Yes
	BFS	13	45	Yes
	Randomized Local	17	58	Yes
Indore	A*	17	37	Yes
...	...	...	...	...

Table: Summary of algorithm performance metrics across different urban grids.

Key Observations:

- **A*** consistently demonstrated superior efficiency, finding optimal or near-optimal paths while exploring the fewest nodes.
- **UCS** found optimal paths but required exploring a larger search space due to the lack of a goal-directed heuristic.
- **BFS**, while reliable, was the least efficient, often expanding a significantly higher number of nodes.
- **Randomized Local Search**, though less optimal in cost, proved crucial for successful real-time replanning.

6. Dynamic Replanning Demonstration

The system's robustness was tested by introducing dynamic traffic blockages. For instance, in Bhopal, a traffic jam at cell (2,0) during time step 3 forced the agent to recalculate its route. The agent successfully halted, executed the randomized local search algorithm to find a detour, and completed the delivery, validating the dynamic response mechanism.

7. Conclusion and Future Directions

The simulation successfully models key challenges in urban delivery logistics. A* is identified as the most efficient algorithm for static planning, while stochastic local search methods provide necessary adaptability in dynamic conditions. Future enhancements could include more sophisticated traffic models, probabilistic reasoning for replanning, multi-agent coordination, and integration with real-world geographic data.