Delivery Simulator: Detailed Report

Environment Model

The simulator models four urban grid cities: Bhopal (8x6), Indore (7x7), Shivpuri (6x8), and Jabalpur (9x5). Each city map includes:

- **Static obstacles (Walls):** Impassable cells defined by coordinates that constrain routes.
- **Variable terrain costs:** Some cells impose higher traversal costs, simulating difficult terrain or road conditions.
- **Dynamic traffic obstacles:** Time-dependent traffic blocks appear at predetermined cells and specific time steps, adding temporal complexity.
- **Delivery tasks:** Predefined start and goal positions specify delivery routes in each city.

For example, Bhopal includes walls blocking a corridor, a terrain cell with a cost of 3 at (5,2), and traffic blocked at (2,0) during time steps 3 to 5. Other cities similarly combine walls, terrains, and traffic locations with different layouts and costs. The model simulates realistic urban delivery environments requiring spatial and temporal navigation decisions.

Agent Design

The delivery agent is a grid-based robot with attributes:

- Positional coordinates and an internal clock tracking time steps.
- Fuel resource consumed is proportional to terrain cost for movement.
- A delivery completion counter.
- A history log tracking events (blockages, replanning, completion).

Operation involves:

- Generating an initial route using one of four algorithms (BFS, UCS, A*, or randomized local search).
- Executing the route stepwise, updating position, time, and fuel.
- Detecting dynamic obstacles (traffic blocks) appearing unexpectedly.
- Triggering dynamic replanning via randomized local search when blockage occurs.
- Logging replanning events and delivery success or failure.

This design ensures real-time responsiveness to dynamic urban traffic obstacles within fuel constraints.

Heuristics Used

Algorithm	Description	Heuristic/Strategy				
BFS	Explores nodes in uniform breadth-first layers; no cost weighting.	None; treats all moves as equal.				
UCS	Prioritizes paths by cumulative minimal cost considering terrain.	Cost-so-far prioritization.				
A*	Best-first search combining UCS cost and Manhattan distance heuristic.	$h =	x_1 - x_2	+	y_1 - y_2	$.
Random Local	Stochastic local search sampling neighbor moves weighted by the heuristic sum of cost and distance.	Probabilistic choice balancing exploration and exploitation.				

A* leverages the Manhattan distance heuristic to efficiently guide exploration towards goals, reducing the number of nodes expanded versus uninformed methods. Random Local excels at replanning to dynamically circumvent unexpected obstacles.

Experimental Results

Results shown as average over deliveries simulated in each city for all four algorithms:

City	Algorithm	Cost	Nodes Explored	Success
Bhopal	BFS	13	45	True
	UCS	13	41	True
	A*	13	29	True
	Local	17	58	True
Indore	BFS	18	68	True
	UCS	17	55	True
	A*	17	37	True
	Local	21	64	True
Shivpuri	BFS	13	50	True
	UCS	12	43	True
	A*	12	31	True

	Local	15	54	True
Jabalpur	BFS	22	82	True
	UCS	20	71	True
	A*	20	48	True
	Local	24	76	True

A* consistently explores fewer nodes and finds minimal or near-minimal cost paths due to heuristic efficiency. UCS also finds minimal cost paths but with broader exploration. BFS expands the most nodes, lacking cost or heuristic guidance. Random Local search has higher exploration and cost but is essential for replanning.

Dynamic Replanning Proof-of-Concept

During execution, if a traffic obstacle suddenly blocks the courier's next planned move, replanning is triggered. For example, in Bhopal at time 3, traffic occurs at (2,0):

```
Blocked at 2,0 at t=3
New route found
Delivered in Bhopal! Cost=13
```

The agent halts, executes randomized local search from the current position to goal to identify a feasible reroute circumventing the blockage, then resumes the delivery successfully. This implementation demonstrates robustness to dynamic obstacles in real time.

Analysis and Conclusion

The delivery simulator combines environment complexity with agent adaptiveness:

- Urban settings modeled with static and dynamic obstacles plus varied traversal costs.
- The agent balances fuel and path cost, adapting dynamically through heuristic search and stochastic replanning.
- A* search emerges as the most parameter-efficient planner in static conditions.
- Randomized local search effectively handles replanning under changing environments despite overhead.

This system models realistic delivery scenarios with time-variant obstacles and terrain variation; its replanning capability supports delivery robustness critical in dynamic urban traffic. Future work could explore richer traffic modeling, probabilistic replanning, multi-agent coordination, and real-world map integration.