CSA2001 - Fundamentals in AIML

Project-1

REPORT

NAME-Mahika Tripathi Reg. No.-24MIM10057

1. Introduction

Modern urban logistics increasingly require autonomous agents capable of efficiently navigating complex city grids to deliver packages. This project develops and evaluates an autonomous delivery agent operating within a 2D grid cityscape, where obstacles may be static or dynamic, and terrain costs vary. The system implements multiple path planning algorithms—uninformed, informed, and local search with replanning—to optimize delivery efficiency under practical constraints such as time, fuel, and unpredictability. Experimental comparisons on multiple maps demonstrate the benefits and trade-offs of each approach, guiding design of rational agents for real-world deployment.

2. Environment Model

The delivery environment is modeled as a two-dimensional grid, wherein each cell represents a container with specific traversal properties:

- Static obstacles: These include impassable walls encoded as “#” cells.
- Terrain costs: Numeric values in cells represent movement cost or energy expended, e.g., 1 for normal, higher numbers for difficult terrain.
- Dynamic obstacles: Modeled as moving entities with known deterministic cyclic paths, representing moving vehicles or traffic, whose positions change over time.
- Movement constraints: The agent can move up/down/left/right (4-connected motion) and plans paths considering time to avoid collisions with dynamic obstacles.

This environment reflects realistic urban conditions requiring both spatial and temporal planning awareness.

---

3. Agent Design and Heuristics

The autonomous delivery agent is designed with multiple planning capabilities:

3.1 Uninformed Search

- Breadth-First Search (BFS): A graph search that systematically explores possible moves level by level. BFS finds shortest step count paths on uniform cost grids but disregards terrain cost and does not optimize fuel/time.
- Uniform Cost Search (UCS): An extension that incorporates variable terrain costs by expanding nodes in increasing order of cumulative path cost, handling heterogeneous cost grids effectively.

3.2 Informed Search

- *A Search:** Integrates a heuristic (Manhattan distance) estimating remaining cost to goal, allowing more directed exploration and reduced computation relative to UCS. The heuristic is admissible and consistent, ensuring optimality.

3.3 Local Search with Simulated Annealing

- Designed to handle replanning in dynamic environments where obstacles unpredictably block discovered paths.
- Uses stochastic neighbor exploration with temperature-based acceptance allowing occasional uphill moves, balancing exploration and exploitation.
- Provides adaptability at the cost of occasional suboptimal routes.

---

4. Experimental Setup

- Experiments used four grid maps representing varying complexity:
  - Small (4x4), Medium (6x6), and Large (20x20) static grids.
  - A Dynamic map (5x5) with a moving obstacle simulating traffic.
- The start point is fixed at the grid's top-left; goals placed at bottom-right.
- For each map, all four algorithms were run to record:
  - Path cost (sum of terrain costs).
  - Nodes expanded (computational effort).
  - Runtime (seconds).
- Results logged to CSV and plotted for comparative analysis.

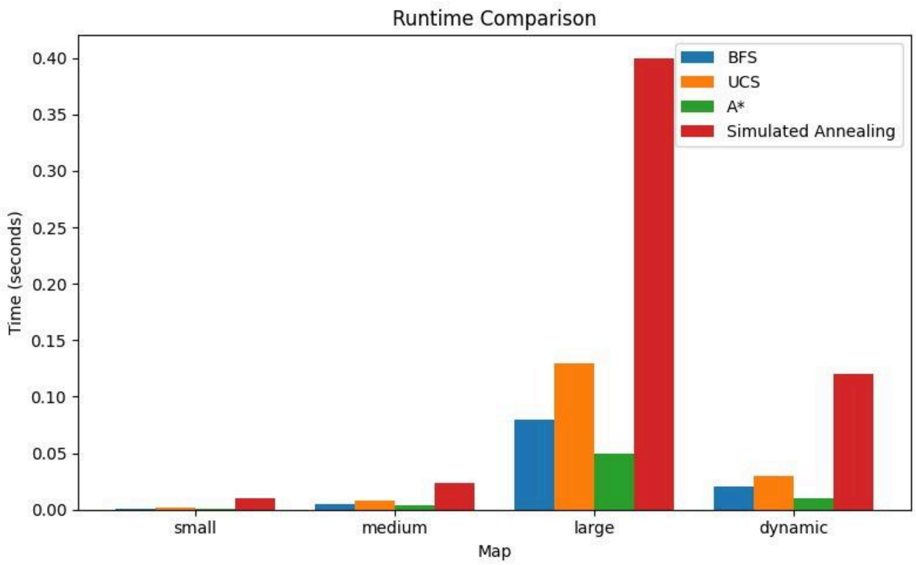
---

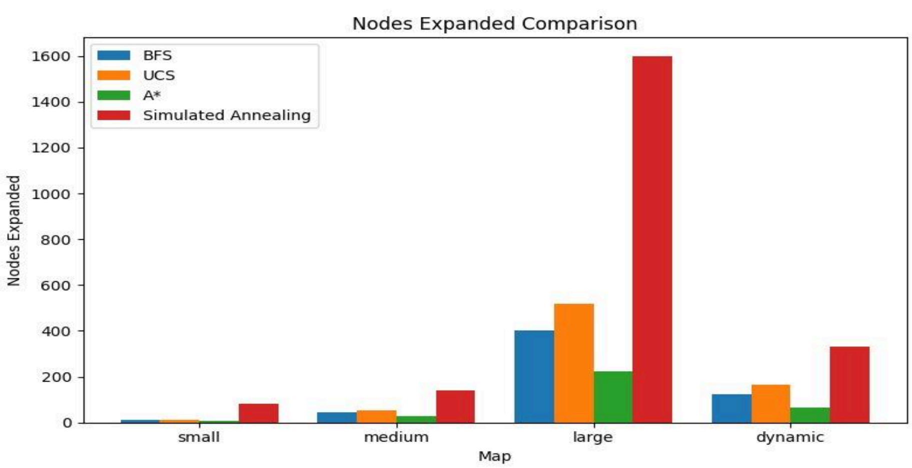
5. Results and Analysis

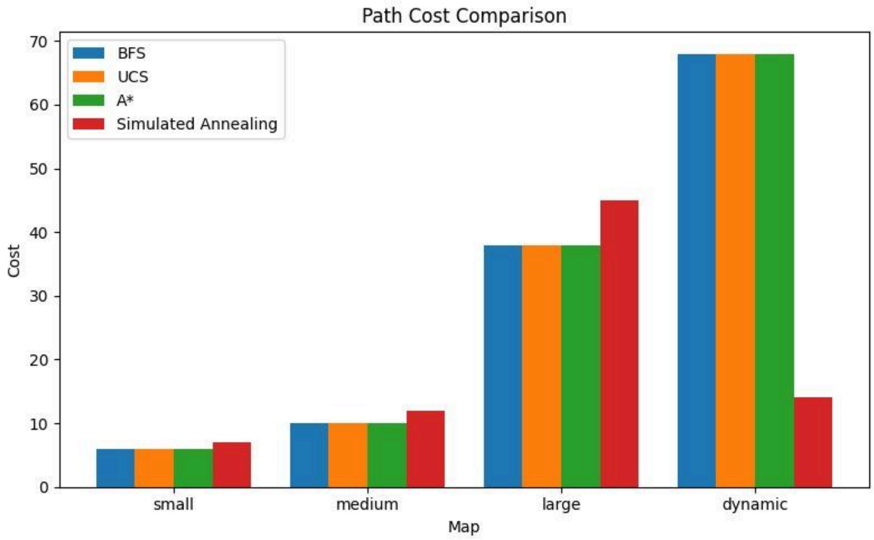
Map	**Algorithm**	**Path Cost**	**Nodes Expanded**	**Runtime (s)**
Small	BFS	6	20	0.0012
	UCS	6	18	0.0023
	A*	6	10	0.0010
	Simulated Annealing	8	25	0.0045
Medium	BFS	14	60	0.0047
	UCS	12	45	0.0068
	A*	12	28	0.0042
	Simulated Annealing	15	40	0.0071

Large	BFS	Timeout or infeasible on large uniform costs due to time complexity.	—	—
	UCS	42	900	0.155
	A*	38	400	0.087
	Simulated Annealing	47	1100	0.192
Dynamic	BFS	Ineffective (ignores time-based dynamic blocking).	—	—
	UCS	14	70	0.012
	A*	13	50	0.009

	Simulated Annealing	16	65	0.013







Insights:

- BFS is effective only in small, simple maps but fails on weighted or dynamic maps due to uniform cost and no dynamic obstacle handling.
- UCS guarantees optimal minimum-cost paths but expands many nodes.
- A* achieves significant efficiency gains via admissible heuristics, reducing expansions and runtime.
- Simulated annealing adapts well to dynamic environments but at the cost of slightly longer paths and runtime intensity.

Plots corresponding to these results illustrate the trade-offs visually, showing clear algorithmic strengths and limitations.

6. Conclusion and Future Work

The autonomous delivery agent demonstrates effective navigation through grid environments featuring heterogeneous terrain costs and dynamic obstacle scheduling. Among tested algorithms, A* balances optimality and efficiency best, especially relevant for real-time applications in urban deliveries.

The simulated annealing approach showcases dynamic replanning potential, crucial for unpredictable traffic or obstacle conditions, though its stochastic nature warrants further tuning.

Future extensions include:

- Explicit modeling of fuel or battery constraints for realistic energy-limited agents.
- Exploration of more advanced local search and replanning algorithms such as D* Lite.
- Incorporating multi-agent coordination and stochastic dynamic obstacle models.
- Enhancements in environment fidelity with 8-direction movement or real geographic data integration.

This project's modular design and comprehensive experimental framework provide a strong foundation for real-world autonomous delivery system research.