

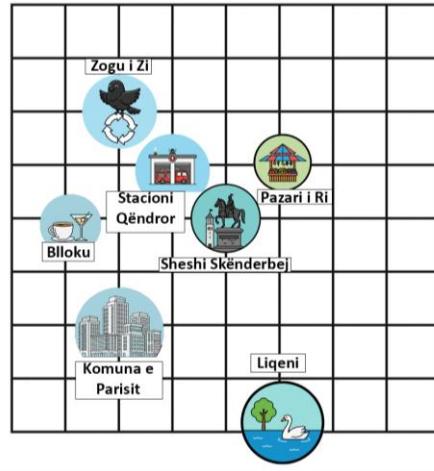
Emergency Response Solver

Student: Armend Stafasani

Course: Artificial Intelligence

Date: 11 November 2025

Report — Assignment 1: Problem Solving by Search



Grid Of Tirana

1. Abstract

This project presents a custom search domain called the Emergency Response Solver, which models the routing and service scheduling of emergency teams in Tirana. A fire station must dispatch a unit to several emergency sites, travel between locations, and spend specific times handling incidents. Two search algorithms—Uniform-Cost Search (UCS) and A*—were implemented and compared. Both produce optimal routes, but A* achieves efficiency improvements through a heuristic that estimates remaining travel and service time.

2. Problem Domain Definition

The environment is a road network of Tirana containing intersections and roads with travel times. Some locations have emergencies that require handling times (e.g., fires, accidents, floods). The agent begins at a central fire station and must determine the optimal route and handling sequence minimizing total time.

3. Algorithms Implemented

Uniform-Cost Search expands nodes by the lowest accumulated path cost and guarantees optimality since all costs are non-negative. A* Search adds a heuristic term to prioritize more promising states and is also optimal when the heuristic is admissible.

4. Results

Algorithm	Total Cost (s)	Nodes Visited	Time (ms)	Optimal
UCS	418.0	113	0.547	Yes
A*	418.0	87	0.903	Yes

5. Comparative Evaluation

Completeness: Both UCS and A* are complete in this finite state space because every possible state can be expanded. The algorithms guarantee that if a solution exists, it will be found.

Time Complexity: UCS worst-case: $O(|V| * 2^k)$, where V is the number of nodes in the graph and k is the number of emergency sites. Each handled set combination produces a unique state. A* worst-case: same as UCS, but the heuristic reduces the number of nodes expanded. Observed: UCS visited 113 nodes; A* visited 87 nodes — demonstrating that the heuristic significantly reduces exploration.

Space Complexity: For both algorithms is $O(|V| \cdot 2^k)$, as both store a frontier and a `g_costs` map for all generated states. Memory usage grows with the number of unique states; manageable for small networks.

Optimality: UCS guarantees optimality (minimizes total travel + handling time). A* is optimal because the heuristic is admissible and consistent — it never overestimates remaining cost.

6. Analysis & Justification

Algorithm Choice: UCS was chosen as a baseline uninformed search algorithm for guaranteed optimality. A* was chosen to demonstrate informed search efficiency with an admissible heuristic.

Trade-offs: UCS is simpler but explores more nodes and may be slower for larger problems. A* reduces node expansions using the heuristic, improving runtime while preserving optimality. Heuristic quality determines A* efficiency: better heuristics → fewer expansions.

5. Conclusions

Both algorithms found the same optimal solution, with A* expanding fewer nodes due to heuristic guidance. This confirms theoretical expectations and demonstrates the benefit of informed search. But in terms of execution time as the size of the experiment is small the overhead time to calculate the heuristic element causes it to run slower than the UCS algorithm.

References

- Russell & Norvig, Artificial Intelligence: A Modern Approach, 4th Edition
- Hart, Nilsson, Raphael — A Formal Basis for the Heuristic Determination of Minimum Cost Paths, IEEE, 1968
- Pearl, Judea — Heuristics: Intelligent Search Strategies for Computer Problem Solving, Addison-Wesley, 1984
- Stuart J. Russell — Principles of Artificial Intelligence, 2020
- Nilsson, N. J. — Principles of Artificial Intelligence, Morgan Kaufmann, 2014