Artificial Intelligence Laboratory 2: A* Search Algorithm
DT8042 HT22, Halmstad University

Mohammad Hammoud, {mohham19}@hh.se, {civilingenjör i datateknik}

November 2022

Introduction

This lab aims to introduce different search algorithms (random, exhaustive, greedy search algorithm, and A* search algorithm with customized heuristics). Those algorithms will be implemented and applied in two domains:
- Path planning: find the shortest path from the agent's current position to the goal.
- Poker game: find a sequence of bids for the agent to win more than 100 coins from the opponent within four hands (given a known, deterministic strategy for an opponent).

Q1: What types of search algorithms you have learned from the lecture? Please briefly introduce them here.

BFS (Breadth First search) is an uninformed search that begins at the robot's starting position and begins looking for the goal by expanding all of the successors of the root node.

DFS (Depth First Search) starts at the top node and goes as far as it can down a given branch (path), then backtracks until it finds an unexplored path and explores it.

The greedy algorithm solves the problem by choosing the best available option. It does not bother whether the current best result will bring the optimal result.

A* algorithm calculates the shortest distance between the initial and final states.

Euclidean distance is the shortest path between the source and destination, while Manhattan distance is the sum of all the actual distances between the source and destination.

Task 1: Path Planning

For tasks 1...

Q2: Implement and apply uninformed search algorithms, e.g. random search, BFS, and DFS. Do they find the optimal path on both maps? How many nodes were expanded? Please include a few example plots of the grid maps with the evaluation values of each cell and the path found.

Yes, it was successful in finding out the optimal path. For: Random: nodes were expanded are 448 and the path was 38. BFS: nodes were expanded are 5398 and the path was 134. DFS: nodes were expanded are 1359 and the path was 800.

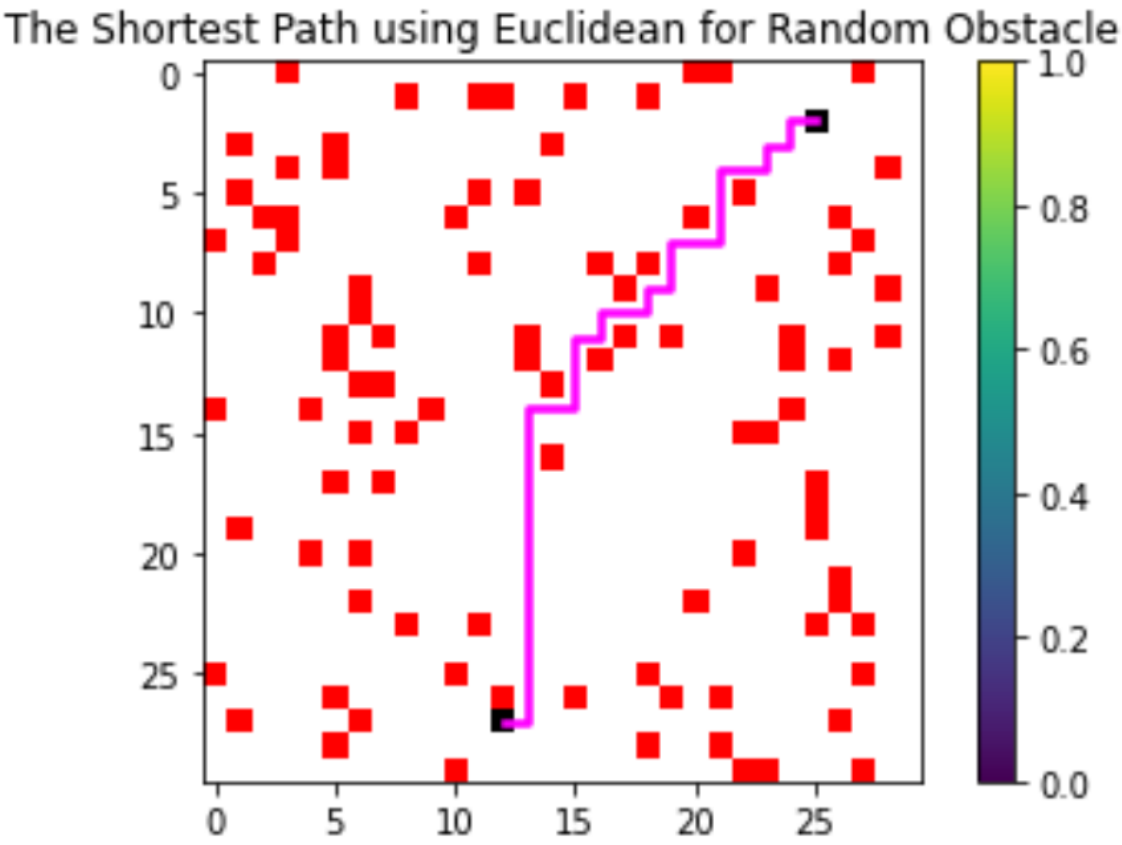

Figure 1: Random using Euclidean.

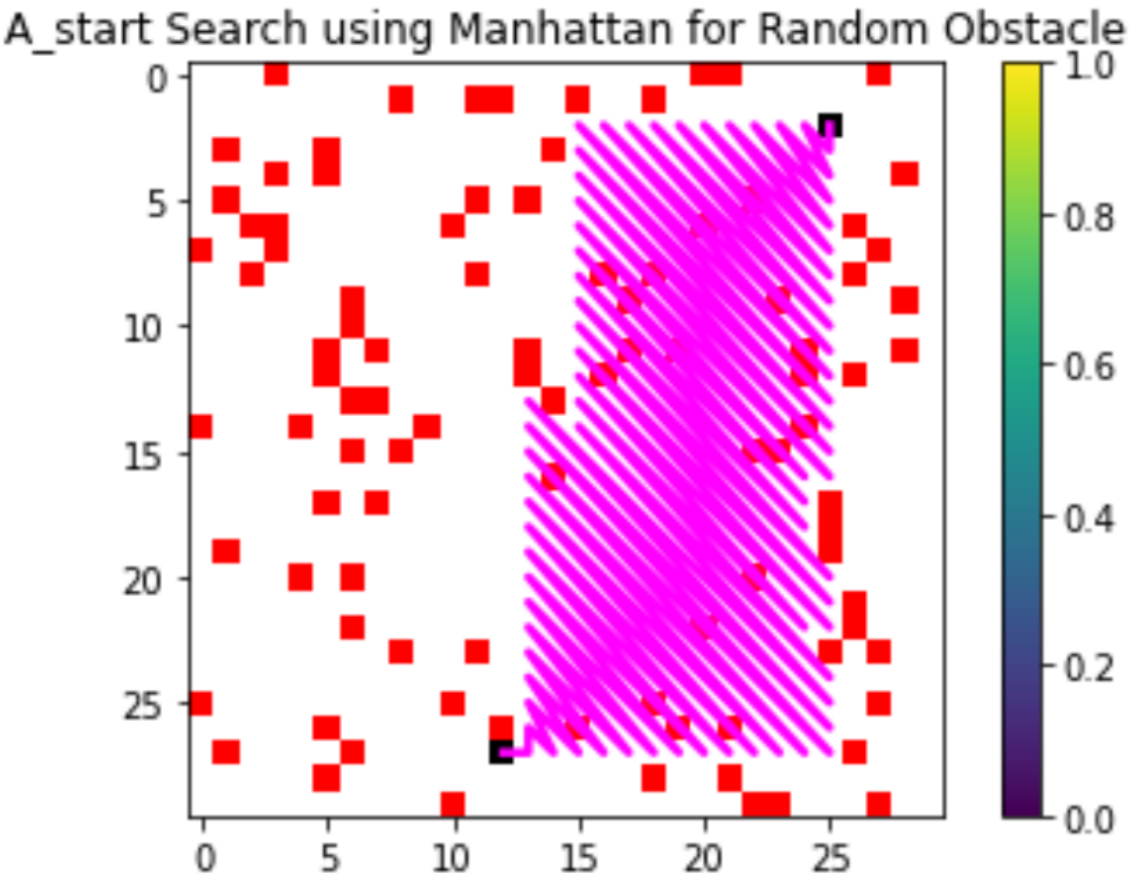

Figure 2: Random using Manhattan.

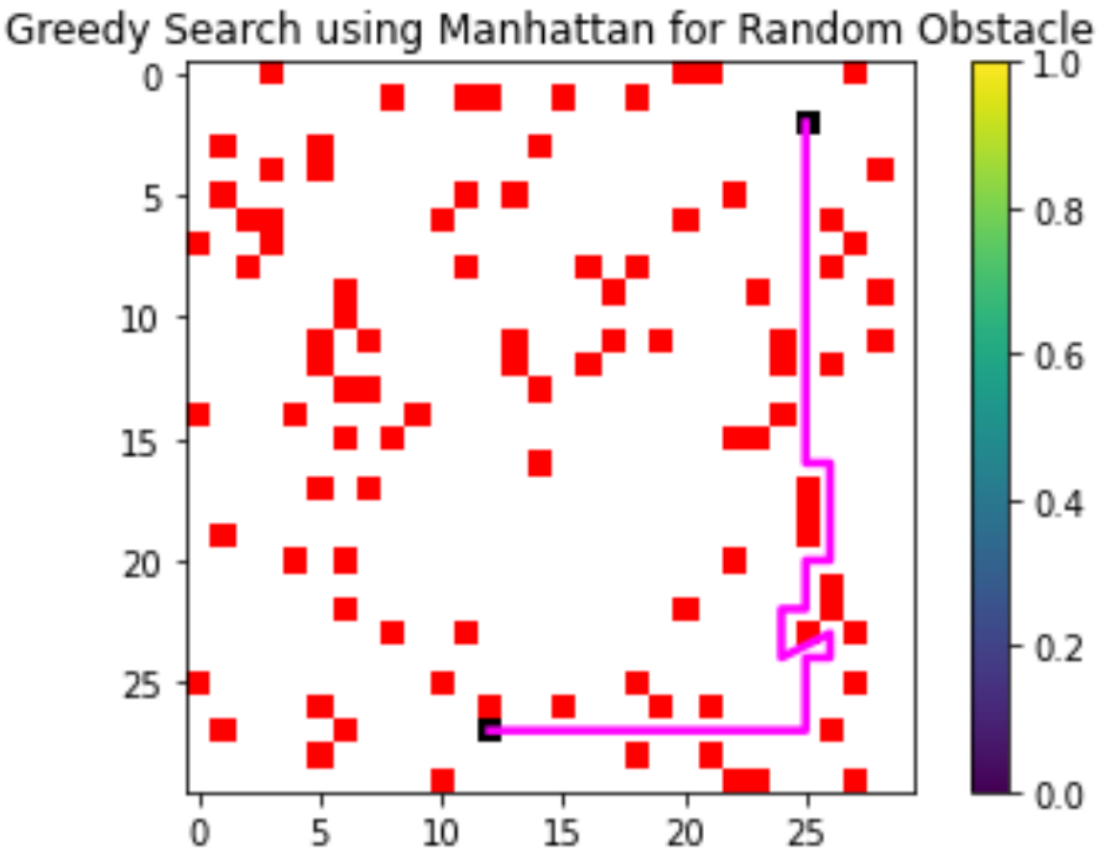

Figure 3: Greedy Search using Manhattan.

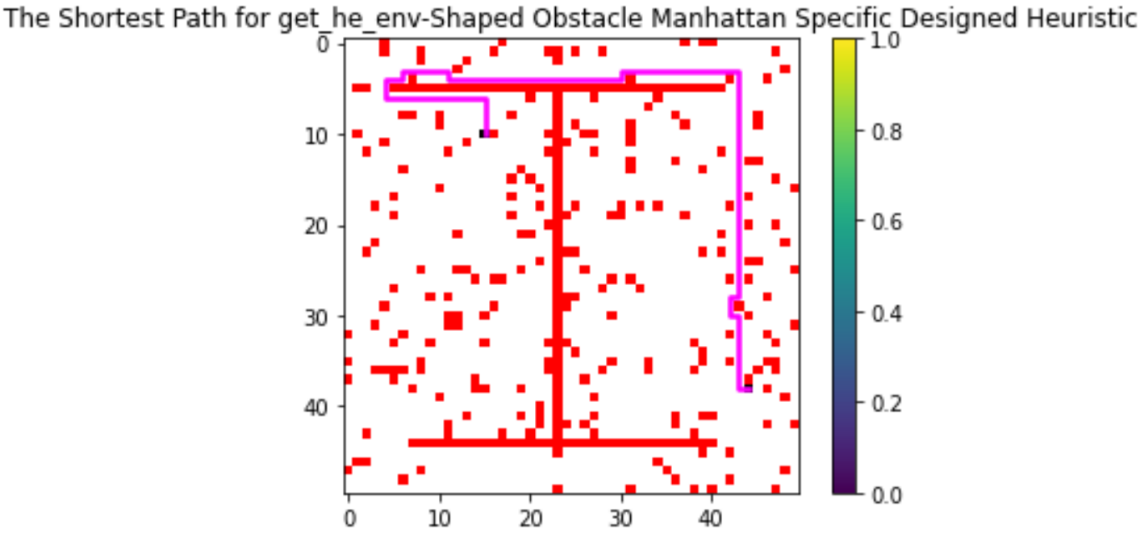

Figure 4: Greedy search using Manhattan.

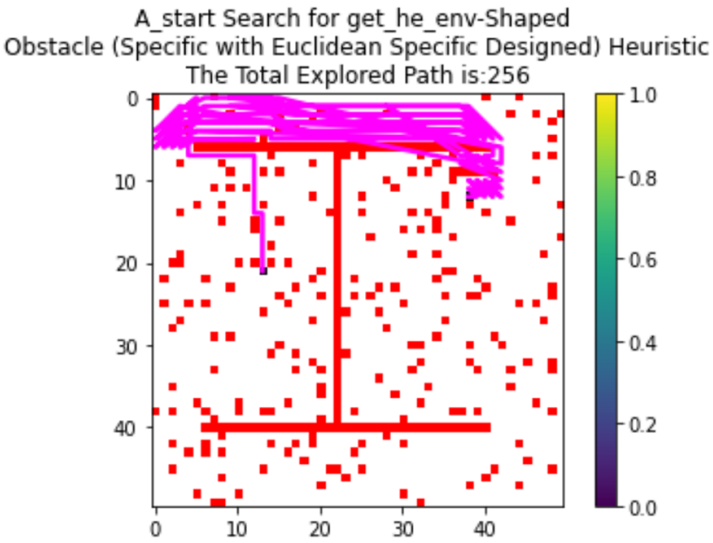

Figure 5: A* search using Euclidean.

Q3: Implement and apply greedy and A* search algorithm. What heuristic function have you implemented? Do both algorithm found the optimal path on both maps? Please provide examples.

Manhattan and Euclidean functions are implemented. Both algorithms did find the optimal path.

Q4: Compare different search algorithms. For each search algorithm, run the experiment multiple times (e.g. 20) and fill in the following table.

From this experience, we can see that the last algorithm (A*) did better than the others.

Search algorithms	Number of nodes expanded	Time consumed	Optimum count
Random	201	0.0202	15.133152
BFS	1592	719.208	41,648
DFS	14016.5	343.9133	322.2491
Greedy	113.12	228.5651	219.259
A*	105.95	108.76	118.869

Table 1: Performance comparison (Path Planning)

Task 2: Poker Bidding

Q5: Implement and apply random, BFS, DFS, and greedy search algorithms. What have you observed? Do all of them found a solution? How many nodes were expanded?

Q7: Briefly describe the heuristic function implemented. How many nodes were expanded with the proposed heuristic function? Is the solution optimal?

Q8: Compare different search algorithms.

Search algorithms	Number of nodes expanded	Number of hands	Number of biddings
Random	...	...	...
BFS	...	...	...
DFS	...	...	...
Greedy	...	...	...
A*	...	...	...
...	...	...	...

Table 2: Performance comparison (Poker Bidding)

Conclusion

I found this lab exciting to learn, especially understanding these algorithms, how they work and how to implement them in a real environment.