UMEÅ UNIVERSITET
Computer Science
Assignment rapport

September 14, 2022

5DV124 - OU1, V1

Fundamentals of Artificial Intelligence

Search

Name	Gustaf Söderlund	Per Sondell
CS Email	et14gsd@cs.umu.se	c19psl@cs.umu.se
Umu Email	gusa0038@student.umu.se	peso0131@student.umu.se

Teacher
Ola Ringdahl

Contents

1 Introduction

This rapport will discusses our solution for Assignment 1 - Search. We will be comparing the different algorithms that we implemented and give a short discussion about how the work went and how we delegated the work.

2 Comparison between algorithms

As expected BFS finds the shortest path through the mazes while DFS does not. This is apparent in the medium maze, see Table 1, where BFS finds the shortest path, because we reach a vertex with a minimum number of edges from a source vertex[1], while DFS does not. We also see that the A* algorithm also finds the shortest path, in addition A* is more efficient than bfs and expands fewer nodes to find the path. The reason for expanding fewer nodes is thanks to the heuristic, so the algorithm knows what direction it should search in.

Depending on the problem the performance of the algorithms will change. For instance if the problem were to solve a maze with weighted vertices the DFS algorithm would find the shortest path and minimum spanning tree. We would see that DFS is superior compared to BFS in weighted graphs.

On the other hand if the problem were to solve a maze with un-weighted vertices the BFS algorithm would find the shortest path and minimum spanning tree. This shows that the performance difference between the two algorithms is depending if the maze is weighted or un-weighted.

Another interesting use case for DFS is its power to solve decision-making trees used in games or puzzles. DFS makes a decision then it explores all paths through this decision, it stops if the decisions leads to a win. Since BFS considers all neighbors first it is not optimal for puzzle problems.

We can see in Table 1 that A* seems to be superior to BFS and DFS when a good heuristic is used. However a limitation to A* is that if a poor heuristic is used it can become slower than BFS. Another limiting factor for A* is if the graph is small or very sparse [2]. BFS exceeds when the goal vertex is close to the source, while DFS is better when the solution(s) is far away from the source [1].

Maze, Algorithm	Final Cost	Nodes Expanded
Medium maze, DFS	130	146
Medium maze, BFS	68	269
Medium maze, A*	68	221
Large Maze, DFS	210	390
Large Maze, BFS	210	620
Large Maze, A*	210	549

Table 1: Table to represent the performance for each algorithm used on the medium and large maze

3 Discussion

This is the first assignment we worked on together as a team and we both think it went very well and organised. We didn't face any big issues that delayed the workflow except that we were overthinking in the beginning. The reason of why we were overthinking was because we didn't understand the python code that was given in the assignment fully. We solved this problem by reading the documentation of the given code. This helped us making the code more easy to understand and much less code needed to solve the actual problem.

When we started this project we didn't have that much experience with coding in Python. We felt that the assignment was very friendly and easy to understand which made it easier to work on the assignment.

While doing the assignment we used the "pair-programming" technique by taking turns on who wrote the code to ensure that both of us knew what was going on in the code. We made sure to always do the work on campus to easier have a discussion face to face when working.

References

[1] Difference between BFS and DFS, <https://www.geeksforgeeks.org/difference-between-bfs-and-dfs/> [Retrieved 14 Sep 2022].

[2] In what cases would BFS and DFS be more efficient than A* search algorithm?, <https://stackoverflow.com/questions/49912983/in-what-cases-would-bfs-and-dfs-be-more-efficient-than-a-search-algorithm/> [Retrieved 14 Sep 2022]