
Übung 1: Search Space Design

Rafael Ruz Gómez
Miguel Robles Urquiza

5 November 2017



Universität Hamburg

Exercise 4.3

Provide example mazes that show the differences between and properties of the search strategies. Describe these properties

Breadth first is useful when you want a solution containing the fewest arcs. It's obvious that bread first search have few arcs than depth first search.

The order in which the neighbors of a node are added to the stack can be tuned so that solutions are found on the first try.

Exercise 4.4

Are there cases in which your program is unable to find a solution? Provide example mazes.

We have found just one case in wich our program is unable to find a solution, when there is no real solution. We meant, when we can not arrive to the goal because the goal is totally surrounded by 'x'.

Exercise 5.2

Write a second heuristic function that works correctly with portals. What do you have to change?

We check which distance is the minimum distance between our actual position and each portal. If taking this portal we arrive to the goal with minimum cost, we choose this path.

Exercise 5.3

The maze above is a slightly modified version of the environment (Not possible path between start and goal). How does your search react in this case? Can you ensure termination?

We send a message to the user telling that the program has been unable to find a path between the start node the goal node.

Yes, we can ensure termination

Exercise 5.4

For each of the search strategies used so far document the time and memory resources used by the algorithm in terms of expansion operations performed on the frontier of the search and the maximal number of nodes in the frontier.