# COS30019 – Introduction to Artificial Intelligence

# Assignment 1 report

# Option B – The Robot Navigation Problem

# Student name: Do Tuan Dat

# Student ID: 103804603

Table of Contents

[Cover page 1](#_Toc160531984)

[I. Instructions 3](#_Toc160531985)

[II. Introduction 5](#_Toc160531986)

[III. Search algorithms 5](#_Toc160531987)

[IV. Implementation 7](#_Toc160531988)

[V. References 10](#_Toc160531989)

## Instructions

To use the program, first head to the root folder, and on the top bar containing the path to the folder, type “cmd”, then press Enter.

A screenshot of a computer

Description automatically generated

A command line interface will open, type in “search.bat”, followed by the file containing information about the map (rows and columns, initial state, goal states, wall positions), followed by the method used for searching, each separated by whitespace. An example would be: “search.bat RobotNav-test.txt BFS”. Available methods for searching are: BFS (breadth-first search), DFS (depth-first search), GBFS (greedy best-first search), AS (A star), and CUS1 (depth limited search).

A screenshot of a computer

Description automatically generated

The output would contain:

On the first line: the file name used to provide map information, the method used, and the number of nodes in search tree, each separated by whitespace.

On the second line: the initial position and the goal state position.

On the third line: the path to reach one of the goal states from the initial position.

If the depth-limited search method (CUS1) is used, however, the command line interface would prompt the user, asking them to set a maximum depth for traversing, before outputing the number of nodes and the path to goal, like so:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Introduction

The robot navigation problem presents users with a map of predefined number of rows and columns, the initial position of the robot, the goal states that the robot needs to reach, and the wall positions, which prevents the robot from entering. The problem requires users to implement different tree-based searching algorithms to find a path for the robot to arrive at one of the goal states.

For better comprehension of the problem, a tree is a data structure with a collection of nodes, which are directly, and undirectly connected as parent and child nodes, with the root node being the first node and has no parents. Given a node that acts as the goal state, many different searching algorithms are proposed to create paths from the root node to the goal node, taking on consideration of time, space, and the advantages and disadvantages of other algorithms.

## Search algorithms

The search algorithms used in the program are:

1. Breadth-first search

**Advantages:**

* Since breadth-first search will always expand all of the nodes from 1 layer before it goes to the next layer, if there exists a goal state, it is guaranteed that breadth first search is going to reach it.
* This property also guarantees that breadth-first search is going to return the shortest path, if there exists multiple goal states, since the nodes in the higher levels will always be visited before the ones in lower levels.

**Disadvantages:**

* Breadth-first search has to remember all the nodes it has visited in order to expand each and every one of them, so the algorithm will take up a lot of memory.
* If the goal state position is far away from the root node, then the algorithm will take lots of time to visit all nodes from previous layers before it reaches the layer that contains the goal node.

1. Depth-first search

**Advantages:**

* Depth-first search will take up less memory during search than breadth-first search, because the algorithm uses stack structure, which only memorize the path from the root node to the current nodes.
* Depth-first search will also take up less space than breadth-first search, because it will traverse only on the left most branch, rather than layers to layers.

**Disadvantages:**

* Depth-first search does not guarantee a path to the goal node in case the search space is infinite or contain loops.

1. Greedy best-first search:

**Advantages:**

* Greedy best-first search is easy to implement, since it only needs to find the node with the lowest heuristic score to traverse.
* If the goal state is close to the root node, then greedy best-first search would be very fast and efficient.

**Disadvantages:**

* Greedy best-first search requires a heuristic function to work, which increases the algorithms, complexity.
* The path chosen by greedy best-first search may not be optimal, since it only chooses the most promising nodes to traverse.

1. A star:

**Advantages:**

* A star counters one of the disadvantages of greedy best-first search by taking the cost taken to reach current node into account, which makes its path to the goal node more optimal.

**Disadvantages:**

* A star requires more merory in order to remember the cost to each state that it may visit.

1. Depth limited search (CUS1):

**Advantages:**

* Depth limited search eliminate the possibility of depth-first search traversing in an infinite depth space by predefining a maximum search depth, this also decrease required memory compared to depth-first search.

**Disadvantages:**

* If the predefined maximum search depth does not contain the goal node, then the algorithm cannot return a solution.
* The algorithm requires more speculation and calculation to define an accurate number for maximum search depth.

## Implementation

1. State

A state is an object that holds the position of a node, a pointer of that node to its parent node, its depth on the search tree, and has methods to get and set its data.

A screenshot of a computer program

Description automatically generated

1. Map Structure

The Map Structure class encapsulates data about the map: number of rows and columns, the robot initial state, its goal states, and wall positions. In addition, the class also has a temporary parent pointer to support linking nodes with their parents. A representation of the Map is also created as a 2D vector to remember discovered (visited nodes) and wall positions.

1. Breadth-first search pseudo code

A computer screen shot of a black screen

Description automatically generated

1. Depth-first search pseudo code

A screenshot of a computer program

Description automatically generated

1. Greedy best-first search pseudo code

A computer screen shot of text

Description automatically generated

1. A Star pseudo code

A screen shot of a computer

Description automatically generated

1. Depth limited pseudo code (CUS1)

A screenshot of a computer program

Description automatically generated

## References

1. Pedamkar, P. (2021, March 3). *Depth limited search: Learn the example of depth limited search*. EDUCBA. https://www.educba.com/depth-limited-search/
2. Tree Data Structure. (n.d.). https://www.cs.cmu.edu/~clo/www/CMU/DataStructures/Lessons/lesson4\_1.htm
3. Unadkat, V. (2020, September 8). *Advantages and disadvantages of AI algorithms*. Medium. https://medium.com/@vatsalunadkat/advantages-and-disadvantages-of-ai-algorithms-d8fb137f4df2