Lab Exercise 7 Sample Work

Module 2: Design of Problem-Solving Agent

Introduction

In various real-world scenarios, navigating through complex environments efficiently is a challenging task. One such problem is finding a path from a starting point (point A) to a destination (point B) in a maze-like structure [1]. This problem has wide applications in various domains, ranging from robotics and autonomous vehicles to video games and logistics.

To tackle this problem, artificial intelligence agents can be employed to intelligently explore and navigate through the maze, optimizing for factors such as the shortest path, avoiding obstacles, and making efficient decisions.

Once useful real-world applications for this task are for the emergency evacuation planning. During emergency situations in crowded areas, such as buildings, stadiums, or amusement parks, finding the fastest and safest evacuation route is of utmost importance [2]. AI agents can assist in analyzing the layout of the premises, identifying potential obstacles or bottlenecks, and devising efficient evacuation plans. By considering factors like crowd density, exit capacity, and obstacles, these agents can optimize the evacuation process and help save lives.

In this exercise, our team wish to design and formulate an agent which simulating a person escaping from one point within the building to the exit using the shortest path.

Type of Agent

The agent to be modelled is a **Non-Autonomous Utility-Based Agents**:

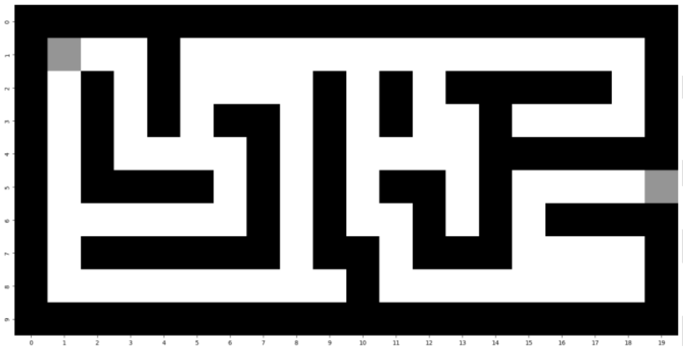
1. It is non-autonomous agent as all the information are provided and made available to agent (built-in / pre-defined). Agent will just need to respond based on these information (maze information, initial and goal states, as well as cost) along with set of rules (search algorithm) provided. The agent will then figure out the best action to be taken (utility / algorithms).
2. There may be many action sequences that can achieve the same goal. Hence, some utility functions need to be defined allowing the agent to reason and determine the best solution.

Types of Environments

Properties	**Elaborations**
Accessible	All maze information i.e. the nodes and its adjacent nodes / weights are accessible by the agent. Initial & goal states will also being given to aid the agent to infer and make decision. No other information are unknown to the agent.
Deterministic	The next state of the environment is completely determined by the current state and the actions selected by the agent. The state of the maze is completely determine by the current action or movement of the agent. Hence it is deterministic.
Sequential	The environment for this problem is sequential as the current action taken will influence the next actions. For example, when a position is visited, the agent should not revisit the same location as this will cause an endless loop.
Static	Environment does not change while an agent is deliberating

Discrete	A limited number of distinct precepts and actions. Agent can move north, south, east, or west one step at a time. These steps are considered finite and there is not options for half step or fraction steps.

Problem Formulation

Properties	**Description**
Definition of Problem	A maze with the problem of finding an escape path from the **start** position (grey on the left) to the **goal** position (grey on the right) on a two-dimensional grid. The wall is represented by black blocks while the agent traverse / walkway is represented by white area in the maze below:  Total number of game states is equals to total number of white areas in the maze. The game state change when the agent moves from one white area to another white area.
Initial State	Any coordinate given within the maze. In this case the grey block on the left in the maze.
Action Sets	Move to unblocked [north, east, west, south] position, explore neighbouring (adjacent) position.
Goal Test Predicate	To successfully achieve the goal, the agent should: 1. Reach the goal position (grey block on the right in the maze). 2. Take the least steps to reach goal position (shortest path). Less steps == low energy consumption Time taken to calculate and travel does not matters in this scenario
Cost Function	Each step / movement of the agent will cost/consume 1 energy point from the agent
Solution	A path from initial position (initial state) to goal position (goal state) derived from search algorithms

References:

1. https://qiao.github.io/PathFinding.js/visual/
2. Zhang, F., Qiao, Q., Wang, J. and Liu, P., 2022. Data-driven AI emergency planning in process industry. *Journal of loss prevention in the process industries*, *76*, p.104740. https://www.sciencedirect.com/science/article/abs/pii/S0950423022000171