**ICT203 Assignment 1**

Table of contents

## **Requirements/Specification**

**Problem Statement**

Develop an intelligent agent to navigate a taxi in a given environment to pick up and drop off passengers using the shortest path. The environment is a 5x5 grid with specific locations for picking up and dropping off passengers. The agent will use search algorithms to determine the optimal route for each task while maximising the reward.

**Assumptions and Conditions**

1. **Environment Grid:** The environment is a 5x5 grid where each cell represents a possible location for the taxi.
2. **Passenger Locations:** The taxi can pick up passengers from four possible locations: Red (0), Yellow (1), Green (2), Blue (3). If the passenger is inside the taxi, the state is represented by 4.
3. **Drop-off Locations:** The taxi can drop off passengers at four possible locations: Red (0), Yellow (1), Green (2), Blue (3).
4. **Actions:**

* 0 = move south
* 1 = move north
* 2 = move east
* 3 = move west
* 4 = pick up passenger
* 5 = drop off passenger

1. **Rewards:**

* Each step incurs a penalty of -1 point
* Proper pick-up and drop-off actions earn a reward of 10 points.

1. **Objective:** Maximise the final reward by picking up and dropping off passengers using the shortest possible path.

**Formalisation of the Problem**

1. **State Space:** Represented by the tuple (taxi\_position, passenger\_location, destination). The taxi\_position is represented by coordinates (x,y), and the passenger\_location and destination are integers from 0 and 3.
2. **Action Space:** Action is defined as (move south, move north, move east, move west, pick up, drop off).
3. **Transition Model:** Defines how the state responds to actions. For instance, moving north decreases the y-coordinate by 1, while picking up a passenger changes the passenger\_location to 4 (inside the taxi).
4. **Reward Function:** Each move action returns a reward of -1. Successful pick-up and drop-off actions return a reward of 10.
5. **Goal State:** Achieved when the passenger has been picked up and dropped off at the correct location.

**Search Algorithms**

The following search algorithms will be implemented:

1. **Breadth-First Search (BFS):** Explores the state spaces level by level
2. **Uniform Cost Search (UCS):** Explores the state space by expanding the least costly node.
3. **A\* Search (AFS):** Uses a heuristic to guide the search, aiming to find the shortest path more efficiently.

**Solution Design**

* Reading the environment
* Move to pick-up location
* Move to drop off location
* Calculate reward

**Assumptions**

* Taxi moves freely within grid boundaries
* Environment and initial state are set up correctly
* Actions are sufficient for navigation

**Expected Outputs**

* Path for each algorithm
* Total reward
* Total steps

## **Design/Algorithm**

**Solution Design Overview**

The solution revolves around navigating a taxi in a grid environment to efficiently pick up and drop off passengers using Breadth-First Search (BFS), A\* Search, and Uniform Cost Search algorithms. Each algorithm’s effectiveness will be compared to demonstrate their optimisation levels in pathfinding.