CS 312: Artificial Intelligence Laboratory
Lab 5 Report

S V Praveen - 170010025 Deepak H R - 170010026

March 1, 2020

1 Introduction

The objective of this task is to solve the N City Shortest Path Problem. Given a set of N cities (coordinates) and distances between them, find the shortest path from source city to destination. For the given problem, the inputs have following property such that the distance between any two cities - u, v is equal to $(1+rand(0,1))^2 * ||u-v||_2$. This helps us to model the problem in a more realistic sense by having distances more than the length of the straight line joining the two cities.

2 Description

2.1 State Space

Given N cities, each city represents a state.

2.2 Start Node and Goal node

The start node is s_0 = 0 and the goal node is s_{N-1} = N-1

2.3 MoveGen(u)

The function takes a state u as input and returns a set of states that are reachable from the input state u in one step.

Algorithm 1 moveGen(u)

```
procedure MOVEGEN(u)
    $neighbours \leftarrow ()$                    ▷ initialize neighbours to empty set
    for v, w of $u$ do
        $neighbours.append(v, w)$              ▷ $v$ is vertex and $w$ is distance(v, u)
    return $neighbours$                 ▷ neighbours are required moves generated
```

2.4 GoalTest(state)

Returns true if the input state is goal and false otherwise.

Algorithm 2 goalTest(state)

```
procedure GOALTEST(state)
    if $state == N - 1$ then
        return $true$
    return $false$                                    ▷ state is not goal
```

3 Heuristic Functions Considered

The following functions are appropriately designed to satisfy the constraints as shown below. Here v is the goal state.

3.1 Over-Estimate

The heuristic function $h_1(x)$ is attained by multiplying an arbitrarily large constant k to make the heuristic over estimate. Here we choose $k = 10$.

$$h_1(u) = k * ||u - v||_2 \tag{1}$$

3.2 Under-Estimate

The heuristic function $h_2(x)$ is attained by taking the exponential of the negative euclidean distance between u and goal v. We wanted h such that h is less than euclidean distance and at the same time not monotone in nature.

$$h_2(u) = e^{-||u-v||_2} \tag{2}$$

3.3 Monotone

The heuristic function $h_3(x)$ is attained using euclidean distance between u and goal v. This is monotone by the nature of inputs as discussed in the introduction.

$$h_3(u) = ||u - v||_2 \tag{3}$$

4 Observations and Analysis

The results obtained using A* algorithm with various heuristics are summarized in the below table -

Input	Length of Shortest Path Found		
	$h_1(u)$	$h_2(u)$	$h_3(u)$
input$_1$	798	723	723
input$_2$	306	306	306
input$_3$	2605	2582	2582
input$_4$	4490	4468	4468

We see shortest path is always found when the monotone property($h_1(u)$) is enforced in the heuristic or when we underestimate in the heuristic($h_2(u)$). As expected, overestimating($h_3(u)$) does not always guarantee optimal path as A* isn't guaranteed to be admissible in this case.