Aasish Tammana

CSE 571 – Artificial Intelligence

Homework 2

Exercise 1.1

a) The path costs will be directly proportional to the depth when the individual step costs are same. In such scenario, the Breadth first search is nothing but the Uniform Cost Search.
b) Let us say the evaluation function is represented by f(n) which is given as a sum of the shortest path from start node to goal path and the cost function. If the cost function is a function -depth(n) then the Depth First Search is the same as Best first search.
c) We know that A* function can be represented by f(n) = h(n) + g(n) where h(n) is the heuristic function and g(n) is the function for cost from start to present. If the heuristic function h(n)=0 then we get f(n) = g(n) which is nothing but uniform cost search.

Exercise 1.2

a) As per the problem statement, the agent can move in North, South, West and East directions or stay, hence maximum possible transitions are 5.
b) A search tree has 5 possible states at depth 1 but at depth 2 number of states would be 5^2=25. However, we need to

consider only distinct states. Therefore, number of distinct states is given by 2k(k+1) + 1

c) Maximum number of nodes in breadth first search is given by branching factor to the power of depth. Since branching factor is 5, max nodes are given by $5^{\{|x|+|y|\}}$ where x and y are goal state coordinates.

d) Breadth first search = depth^2 which is $\{|x| + |y|\}^2$ because depth is given by $|x| + |y|$

e) In breadth first Graph search, the maximum number of nodes is given by

$$2(x + y)^2 + 2(x + y) + 1$$

f) Yes, h= |u-x|+|v-y|can be regarded as an admissible heuristic because the agent is restricted to the 2D grid and can only move within it.

g) As mentioned previously, the function can be represented by f(n) = h(n) + g(n). Heuristic function h(n)=0, we get f(n) = g(n) which is the shortest path. Hence f(n) is a function x+y.

h) If we remove a few links, the shortest distance between the start node and goal node will increase. Therefore, the function will be an underestimate but still be admissible.

i) When we add more links between nodes, there is always a chance that the path cost will reduce i.e., the function will no longer be an underestimate. Hence it is not admissible.

Exercise 1.3

a) There is a total of n^2 locations and the total number of vehicles is n. The total number of states is given by $(n^2)^n$.

b) At any given point, each vehicle can choose among 5 states. Hence, the maximum branching factor is 5^n

c) In the scenario where other vehicles are present on the grid, there is an option for the vehicle to hop over other vehicles. This means lesser number of steps to reach the goal. Since there are no other cars on the grid, the admissible function can be written as $H(n)=|(n-i+1)-x_i|+|n-y_i|$ which is the distance.

d)

i. Not admissible because the total work done for moving all vehicles is greater than $\sum h_i$ and since hoping of vehicles is allowed, the function will overestimate.

ii. Not Admissible. The total work done per step can never exceed the total number of vehicles n. The max $\{h_1 ... h_n\}$ heuristic overestimates the cost in this scenario as well and hence is not admissible.

iii. Admissible. The min $\{h_1 ... h_n\}$ is an admissible heuristic because the number of steps is given by total work over work per step. This value is greater than or equal to the minimum of $\{h_1 ... h_n\}$ which is the work per step.

Exercise 1.4

a)

- Initial State – Farmer, wolf, goat, and cabbage on one side/bank before crossing the river.
- Actions – Carry one of the three and cross the river to reach the goal side, or move alone but ensure goat is not alone with cabbage or wolf is not alone with goat.
- Transitions – Let us denote Goat as G, Wolf as W, Cabbage as C and Farmer as F. If A is starting point and B is goal state (The other bank) represented as (A, B)
  One of the Transition Model can be represented as
  (FCGW, --) -> (CW, FG) -> (FCW, G) -> (C, FWG) -> (FGC, W) -> (G, FCW) -> (FG, CW) -> (--, FCGW)
- Goal State – Farmer crosses the river to the other bank with all his purchases.
- Path Cost – Assume uniform cost for all movements.

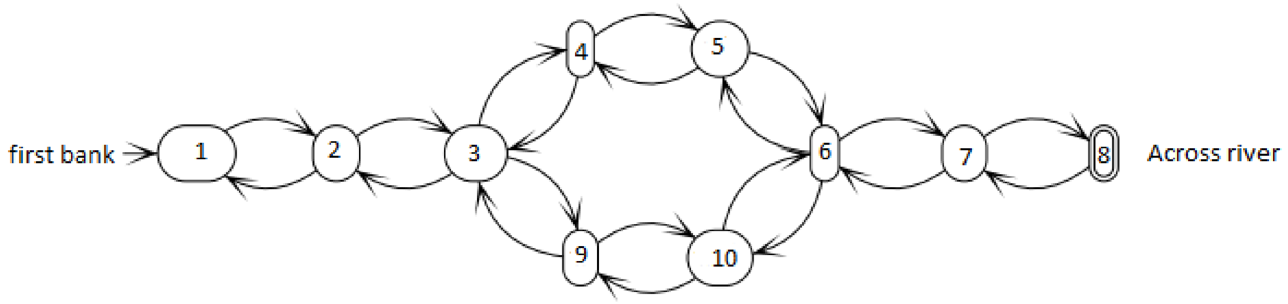
Since there are 4 elements, total possible states are 4^2=16 possible states.

b) Let us label each of the states from 1 to 8 and the cost per step is uniform.

(FCGW, --) - 1
(CW, FG) - 2
(FCW, G) - 3
(C, FWG) - 4
(FGC, W) - 5
(G, FCW) -6
(FG, CW) - 7
(--, FCGW) – 8
(W, FCG) – 9
(FWG, C) - 10

As stated above, the transition model can be represented as

Solution 1: (FCGW, --) -> (CW, FG) -> (FCW, G) -> (C, FWG) -> (FGC, W) -> (G, FCW) -> (FG, CW) -> (--, FCGW)



As seen in the diagram another solution is to just swap the cabbage and wolf while transitioning in step 4 and 5 which gives us the second solution and both are admissible.

Exercise 1.5

The pseudo code for Uniform cost search can be written as follows:

```
Function Uniform_Cost_Search(problem) returns a path
        While frontier is not empty
                Parent<-Pop(frontier)
                For child in parents do
```

```
                    If child.state is not in table of states or if it is cheaper path
                        Table of states<-child
                        Add child to frontier
                        If child is goal:
                            Cheaper path = child
        Return cheaper path
```

The pseudo code for Uniform cost search can be written as follow

```
Function Iterative-Deepening(problem) returns path
For range(depth)
    If root == destination
        Return true
    If depth == 0
        Return False
    For child in root:
        If Iterative-Deepening (child, goal, depth-1)
            Return True
    Return False
```

Combining the two algorithms we can say:

```
Function Cost-Iterative-Deepening(problem) returns path
    While frontier is not empty
        Parent<-Pop(frontier)
        For child in parents do
            If child.State is not in table of states or if it is cheaper path
                Table of states<-child
                Add child to frontier
                If Cost-Iterative-Deepening (child, frontier-1)
                    Return True
    Return False
```

Completeness: Yes, when the depth is finite
Optimality: Yes, since it is uniform path costs it is optimal
Computational: $O(b^m)$
Space Complexity: O(b*m)

Where depth is m and branching factor is b