Intro to AI, Assignment-1

Jaskaran Singh Luthra - 110090236

October 2022

1 Answer 4

In the given problem we had a 10x10 grid in which an agent is placed which can move to 8 adjacent squares. Considering there is no obstacle then the agent can reach any cell in the grid in 9 moves or less.

In the given figure below.
a) X is the starting state in the grid which has 8 possible options to go ahead
b) Y is the goal state
c) Possible paths to go 8 (adjacent Squares)
As there is no redundant path and obstacle in the path as shown in the fig 1.

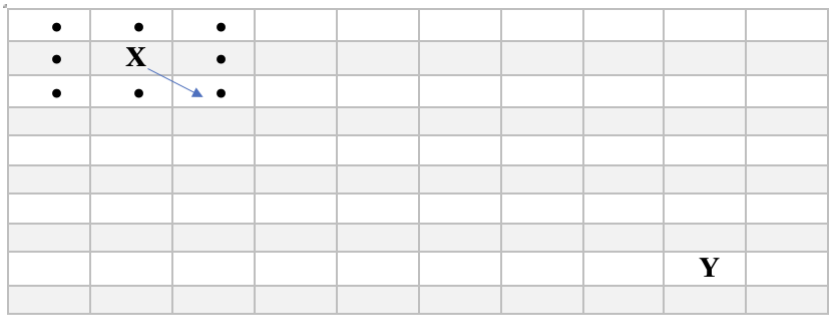


Fig 1. Initial State figure. A 10x10 grid of100 squares with start and end denoted as X and Y respectively.

The agent can start moving diagonally to reach the goal state.

No. of cells Diagonally between Start (X) and End (Y) = 10-1 = 9 moves, where 1 is deducted because we agent is at Start so that cell is not counted. Now, after moving 1 cell, again the agent has 8 options as redundant path are not allowed therefore the agent is now left with only 7 options. As the agent can not traceback.

o Denotes the state already visited
• Denotes the possible state to move.

Case 1 – If redundancy exists

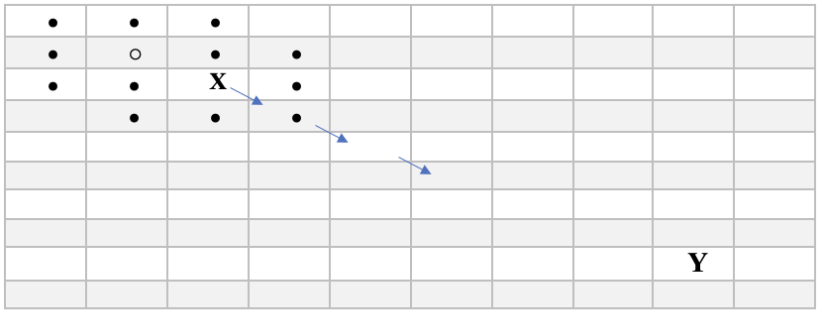


Fig 2. Agent moves to the next stage and previous state becomes visited.

X can move in 8 directions, at every cell. Therefore to reach the goal we need 9 moves and at each move we have 8 options. So the no. of possibilities / no. of paths of length 9 are :

8 options, 9 times, $8 * 8 * 8 * \ldots\ldots = 8^9$ paths.

It is not exactly 89 paths because at the edges and corners the options to move are less than 8. So, the total no. of paths are near to 100 million i.e. 1 million paths per state as there are 10 states

Case 2 – If Redundancy does not exist

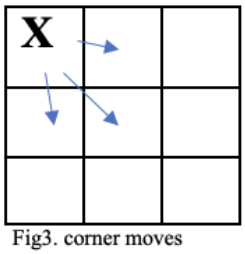


Fig3. corner moves

Refer to Fig 2. In which the previous state is represented by a circle. Now the agent cannot move back to the previous state. Therefore, the options were reduced to 7 and there will be no loopy paths. Once the agent stucks in a loopy path it can go up o infinite iterations. So this situation can be prevented by removing the redundant paths.
When no. of cells in the grid is huge then removing the redundant paths can exponentially decrease the no. of paths to reach the goal state. So, if we eliminate redundant paths, we can complete a search roughly a million times faster.

2 Answer 5

In this question, we explore the difference between uniform-cost search and A* search Given below is the roadmap of Romania

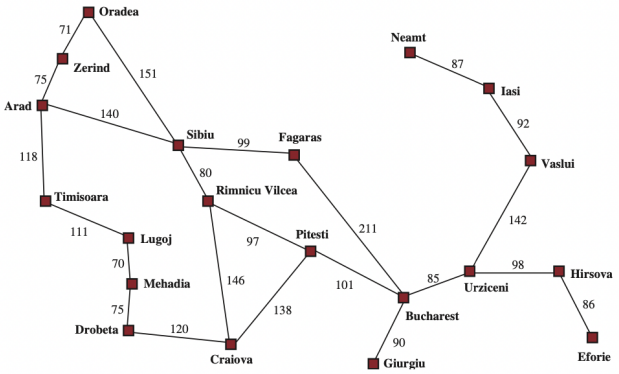


Figure 1: Roadmap of part of Romania.

a. Uniform Cost Search
Intial state is Oradea and we have to reach Bucharest
a) First we **detect nodes, at Oradea** we have two options Zerind(71) and Sibiu(151).Out of which Zerind has less distance.
b) **Zerind** is expanded and again we detect nodes, here we have only one option Arad(75). Now we **compare the total distance from Oradea** to Arad and Sibiu which is 71+75 = **146** and **151** respectively.Now, Arad has less distance
c) **Arad** is expanded, here we **detect Sibiu(140) and Timisoara(118)**. Now we have to compare the distances, for Sibiu 146+140=286, Timisoara 146+118=264 and we compare these values with Oradea from Sibiu(151) also.
d) **Sibiu** is expanded and noded detected at this point is Rimnicu Vilcea (80) and Fagaras(99). Again we have to compare the distances **150+80=231 (Rimnicu)** and **151+99=250(Fagaras)**.
e) **Rimnicu Vilcea** is expanded, detected nodes are Pitesti(97) and Craiova(146).
f) **Fagaras** is expanded. Here node detected is Bucharest(211). Distance at this point is 250+211=461. Finally, we reached the Goal State.
g) After reaching Goal State, we check for more paths available to us.
h) **Timisoara** is expanded and node detected at this point is Lugo(111), distance here is 286+111=397.
i) Now we expand **Pitesti** and detect Bucharesti(101), Distance till here is 231+97+101 = 429.
Finally, from Oradea to Bucharest the shortest Path is :
Oradea - Sibiu - Rimnicu Vilcea - Pitesti - Bucharest

b. A* Search

The most common informed search algorithm is A* search (pronounced "A-star search"), a best-first search that uses the evaluation function

f(n)=g(n)+h(n)

where g(n) is the path cost from the initial state to node n, and
h(n) is the estimated cost of the shortest path from n to a goal state, so we have
f (n) = estimated cost of the best path that continues from n to a goal.

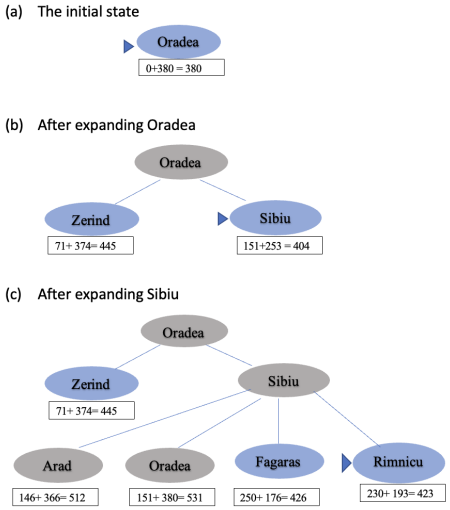
Intial State: Oradea
Goal State: Bucharest

Given below is the stages of A* search for Bucharest.

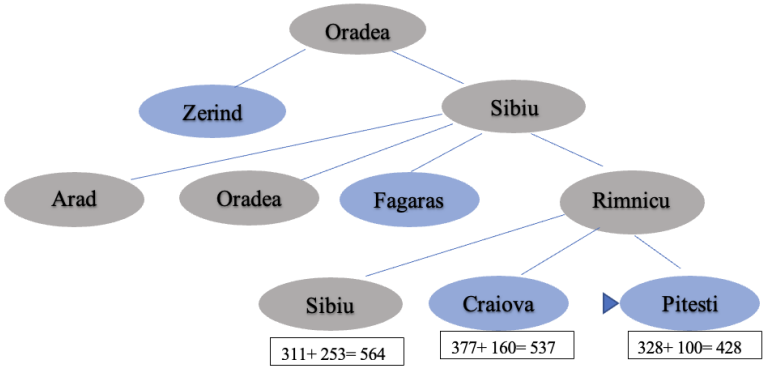
Arad	366	**Mehadia**	241
Bucharest	0	**Neamt**	234
Craiova	160	**Oradea**	380
Drobeta	242	**Pitesti**	100
Eforie	161	**Rimnicu Vilcea**	193
Fagaras	176	**Sibiu**	253
Giurgiu	77	**Timisoara**	329
Hirsova	151	**Urziceni**	80
Iasi	226	**Vaslui**	199
Lugoj	244	**Zerind**	374

Figure 2: Values of heuristic function h(n) - straight line distance to Bucharest.

Nodes are labeled with $f = g + h$
The h values are the straight-line distances to Bucharest taken from Figure 1.



(d) After expanding Rimnicu Vilcea



(e) After expanding Pitesti