Introduction to Artificial Intelligence
Assignment 1
Report

Rizvan Iskaliev, Innopolis University, BS19-04
r.iskaliev@innopolis.university

9 March 2021

1 Backtracking path searching algorithm

The search starts at the actor's starting position. It explores the neighboring positions and checks which of them are safe to go (either they are not infected, or the actor has the mask or has visited the doctor). Then, recursively, the search continues from each of these positions.
At each time the search algorithm starts from a particular cell, it first checks whether the current cell is the home. If this is true, then the path to home is the path to the current cell. Its length is compared with the length of the shortest path for this moment. If the length is smaller, then this path is considered and the global value of the minimal path length is updated to be the length of the current path.
If the considered cell is not the home then an algorithm calculates the estimated home path's length by adding the current path length (path to the position from which the search is started) and the Chebyshev's distance [1] between the considered position and home. If the length of the path is not smaller than the minimal length achieved by previous searches then the search started from the current position terminates and the algorithm tries to find a path from other positions. It is worth mentioning that from the beginning when no path has been found yet, the minimal length is set to be $2N$. This heuristic gives a huge performance gain for the Backtracking algorithm. It works well when the number of covids is no more than 2 (**the number of covids should be 2 according to the assignment statement**). Otherwise, this value should be changed to adapt to the new number of covids.
It is remarkable that if the current path traveled by the actor contains the mask or a visit to the doctor, the algorithm drives the actor towards the home with the shortest path from the current cell without concern about covid infected cells and then performs length comparisons as described earlier.
Once all positions are considered, the algorithm returns all paths with their respective lengths in the list from which by taking the minimal length of the path the shortest path to home is extracted and is given as an output of this algorithm given a particular map. If the algorithm failed to find the path to home, the corresponding message is printed.

The second variant of the perception of covid by an actor also has been implemented for the Backtracking algorithm. However, it does not influence the algorithm's performance since the backtracking path searching algorithm described above at each cell considers all (and only) adjacent (neighboring) cells and does not pay attention to what is happening behind them. Furthermore, there can be such maps that the mask or doctor is the neighbor of the covid infected cell, and the shortest path to home is to visit this cell and go straight to home with no fear of covid. An example of such map is shown below (we have a 9x9 map (A - actor; C, c - covid source and its infected cells, M - mask, D - doctor, H - home):

D								H
				c	c	c		
				c	C	c		
				c	c	c		
			M					
		A				c	c	c
						c	C	c
						c	c	c

Even if the actor perceives the covid through one cell, in any case, the adjacent cell with the mask (M) is needed to be checked since if we take the mask and go straight to the home, we get the shortest path to the home.

The statement that for two variants of the perception of the covid there is no statistically significant difference between performances of Backtracking algorithm is statistically justified in Section 3 (Statistical Analysis).

2 A* path searching algorithm

For this algorithm to work it is needed to maintain the following lists: *Open* that contains the cells that are not explored yet and *Closed* that includes already checked ones. At the beginning, *Open* contains only the starting node (Actor's initial position), *Closed* is empty. The search is started from the starting node. The node from which the search is launched is said to be the current node. The algorithm firstly checks whether the current node is the home. If it is, then the path leading to this node is the shortest path to home. If the current node is not the home, then this node is added to the *Closed* list. Then, for the current node, the adjacent (neighboring) nodes that are safe to go (either they are not infected, or the actor has the mask or has visited the doctor) are found and the algorithm chooses which of them to add in the *Open* list. First of all, the adjacent nodes should not be in *Closed* list. Then, for each of them, the G (the number of steps from the initial node to the adjacent node that is considered), H (Chebyshev's distance [1] between the adjacent node and the home), and F (sum of G and H) costs are calculated. If some of the adjacent nodes are already in the Open list, then they are added to it only if their new G costs are less or equal than previous ones. If an adjacent node is not in the Open list then this node is added into the list. The paths to the appended adjacent nodes were set to be the path to the current node connected with the coordinates of adjacent nodes. At this moment, the *Open* list should not be empty, otherwise, the path to home does not exist and the algorithm terminates printing the corresponding message. If the *Open* list is not empty, then the new current node is selected from this list choosing by the minimal F cost, and if there is more than one node with the same minimal F cost, then from those nodes the node with minimal H cost is taken. The new current node is then deleted from the *Open* list and the search starts from this node, repeating the process described above. It is worth to mention that if on the path to current node the actor has picked the mask or has visited the doctor, the algorithm drives the actor towards the home with the shortest path from the current node without concern about covid infected cells.

The second variant of the perception of covid by an actor also has been implemented for A* algorithm. However, it does not influence the algorithm's performance since the A* path searching algorithm described above each time selects the node with minimal F cost from only adjacent (neighboring) cells and what is happening through the neighboring cells does not affect the algorithm's choice.

The statement that for two variants of the perception of the covid there is no statistically significant difference between performances of A* algorithm is statistically justified in Section 3 (Statistical Analysis).

3 Statistical analysis of the algorithms

Twenty five times the map was randomly generated and each of the algorithm with two variants were run on the same map. The map size was set to 9x9, the number of covids was set to 2, the number of doctors was 1, the number of masks was 1 (as it was in the Assignment statement). **Note, that during the running the program, the opportunity to enter the map size and the number of each type of agents is provided. Agents are generated randomly on the map.**

The table with runtime measurements

Measurement №	Backtracking (variant-1) runtime (ms)	A* (variant-1) runtime (ms)	Backtracking (variant-2) runtime (ms)	A* (variant-2) runtime (ms)
1	24	4	24	4
2	19	8	21	8
3	9	6	10	7
4	9	2	9	2
5	18	3	18	3
6	4	2	5	2
7	6	2	5	2
8	5	1	5	1
9	7	2	7	2
10	12	4	12	5
11	17	1	17	1
12	19	3	20	3
13	1	3	1	3
14	3	2	3	2
15	2	2	3	2
16	23	6	23	7
17	10	2	11	3
18	20	3	20	3
19	1	4	1	4
20	18	2	19	3
21	9	2	10	2
22	13	6	12	7
23	7	1	7	2
24	5	1	6	1
25	7	3	7	4

For the algorithms comparisons, consider the null hypothesis H_0 stating that there is no statistically significant difference between performances of algorithms) and the alternative hypothesis H_1 stating the opposite. If we reject the H_0 then we automatically accept the H_1 hypothesis. Also, for comparisons, it is needed to calculate the t-value. It can be calculated as $\frac{|\overline{x_1}-\overline{x_2}|}{\sqrt{\frac{S_1^2}{n_1}+\frac{S_2^2}{n_2}}}$, where $\overline{x_1}$ and $\overline{x_2}$ are the mean values, S_1^2 and S_2^2 are the sample variances, n_1 and n_2 are the sample sizes corresponding to the first and the second algorithms to be considered respectively.

- Comparison of the Backtracking (first variant) and the A* (first variant)

	Min runtime (ms)	Max runtime (ms)	Mean runtime (ms)	Sample variance	Standard deviation	Sample size
Backtracking (variant-1)	1	24	10.72	50.46	7.1035	25
A* (variant-1)	1	8	3	3.3333	1.8257	25

In this case, the t-value $\simeq 5.2629$. There are $25 + 25 - 2 = 48$ degrees of freedom. Selecting in the t-table the 48 degrees of freedom and the confidence level to be the 99.9%, the critical value is 3.505 which is much smaller than the calculated t-value. Hence, by the two-tailed t-test the null hypothesis H_0 is rejected and the alternative hypothesis H_1 is accepted. That is, with the confidence of 99.9%, the performance of the algorithms is different. According to the mean values of runtime of the both algorithms, on average the A* algorithm (variant 1) is faster than the Backtracking algorithm (variant 1).

- Comparison of the Backtracking (second variant) and the A* (second variant)

	Min runtime (ms)	Max runtime (ms)	Mean runtime (ms)	Sample variance	Standard deviation	Sample size
Backtracking (variant-2)	1	24	11.04	51.7067	7.1907	25
A* (variant-2)	1	8	3.32	4.06	2.0149	25

Here, the t-value $\simeq 5.1689$. There are $25 + 25 - 2 = 48$ degrees of freedom. Selecting in the t-table the 48 degrees of freedom and the confidence level to be the 99.9%, the critical value is 3.505 which is much smaller than the calculated t-value. Hence, by the two-tailed t-test the null hypothesis H_0 is rejected and the alternative hypothesis H_1 is accepted. That is, with the confidence of 99.9%, the performance of the algorithms is different. According to the mean values of runtime of the both algorithms, on average the A* algorithm (variant 2) is faster than the Backtracking algorithm (variant 2).

- Comparison of two variants of the Backtracking algorithm

	Min runtime (ms)	Max runtime (ms)	Mean runtime (ms)	Sample variance	Standard deviation	Sample size
Backtracking (variant-1)	1	24	10.72	50.46	7.10352025	25
Backtracking (variant-2)	1	24	11.04	51.7067	7.1907	25

In this comparison, the t-value $\simeq 0.1583$. There are $25 + 25 - 2 = 48$ degrees of freedom. Selecting in the t-table the 48 degrees of freedom and the confidence level to be the 99.9%, the critical value is 3.505 which is much bigger than the calculated t-value. Hence, by the two-tailed t-test the null hypothesis H_0 is accepted. That is, the performance of the algorithms is similar with 99.9% confidence. However, according to the mean values of runtime of two variants of this algorithm, first variant is a little bit faster. This can be explained by the fact that in the second variant the search algorithm performs extra checks for the covid through one cell.

- Comparison of two variants of the A* algorithm

	Min runtime (ms)	Max runtime (ms)	Mean runtime (ms)	Sample variance	Standard deviation	Sample size
A* (variant-1)	1	8	3	3.3333	1.8257	25
A* (variant-2)	1	8	3.32	4.06	2.0149	25

In this case, the t-value $\simeq 0.5884$. There are $25 + 25 - 2 = 48$ degrees of freedom. Selecting in the t-table the 48 degrees of freedom and the confidence level to be the 99.9%, the critical value is 3.505 which is much bigger than the calculated t-value. Hence, by the two-tailed t-test the null hypothesis H_0 is accepted. That is, the performance of the algorithms is similar with 99.9% confidence. However, according to the mean values of runtime of two variants of this algorithm, first variant is a little bit faster. This can be explained by the fact that in the second variant the search algorithm performs extra checks for the covid through one cell.

4 PEAS description with respect to the Actor agent

- Performance measure: an agent reaches home with minimum number of steps
- Environment
  An environment is the map where the Actor agent moves and tries to reach the home.
  Properties of the environment:
  1. **Partially observable** because:
     a. The Actor agent does not know the location of covid agents, it can perceive covid only when standing next to the infected cells (variant 1 of perception) or when standing 1 cell away from the infected cells (variant 2 of perception).
     b. The Actor agent does not know the doctor's and mask's locations until he goes to their cells.
  2. **Single-agent** since there is only one agent (Actor) who interacts with the environment.
  3. **Deterministic** because the next state of the environment can be determined by the previous state and the action applied into the environment. That is, given the current path traveled by the Actor agent and the its current move, the Actor's position on the map can be determined.
  4. **Sequential** since the current move has consequences on future moves.
  5. **Static** because the environment does not change as the Actor agent is thinking about its moves.
  6. **Discrete** since there states in the environment after each of the Actor's move.
  7. **Known** because the designer of the agent has full knowledge of the rules of the environment (for example, rules how to make moves or the rule that covid cells are dangerous for the Actor, etc.).
- Actuators:
  a. Visiting the neighboring (adjacent) cells
  b. Wearing the mask
  c. Visiting the doctor
- Sensors:
  a. Agent's ability to perceive the covid
  b. Information about the home location

5 Maps that were impossible to solve

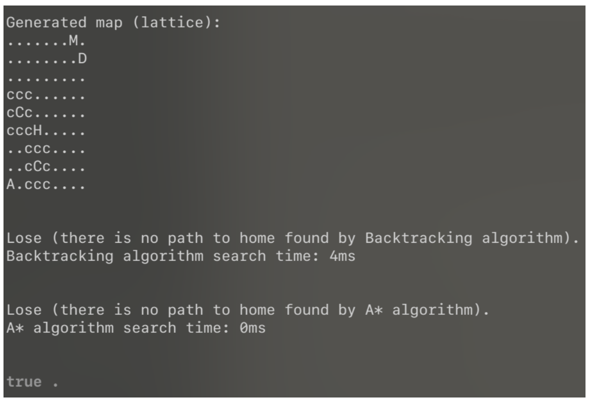
							M	
								D
c	c	c						
c	C	c						
c	c	c	H					
		c	c	c				
		c	C	c				
A		c	c	c				

D			c	c	c			
	H	M	c	C	c			
c	c	c	c	c	c			
c	C	c						
c	c	c						
A								

						c	c	c
						c	C	c
						c	c	c
				c	c	c	D	
				c	c	c	H	
A				c	c	c		M

These are three examples of the maps that were impossible to solve. A - the Actor agent's position
C - Covid agents' positions, c - covid infected cells
M - mask's position, D - doctor's position
H - home.

For example, for the first map, the output of the program is the following (**.** denotes an empty cell):



```
Generated map (lattice):
.......M.
........D
.........
ccc......
cCc......
cccH.....
..ccc....
..cCc....
A.ccc....


Lose (there is no path to home found by Backtracking algorithm).
Backtracking algorithm search time: 4ms


Lose (there is no path to home found by A* algorithm).
A* algorithm search time: 0ms


true .
```

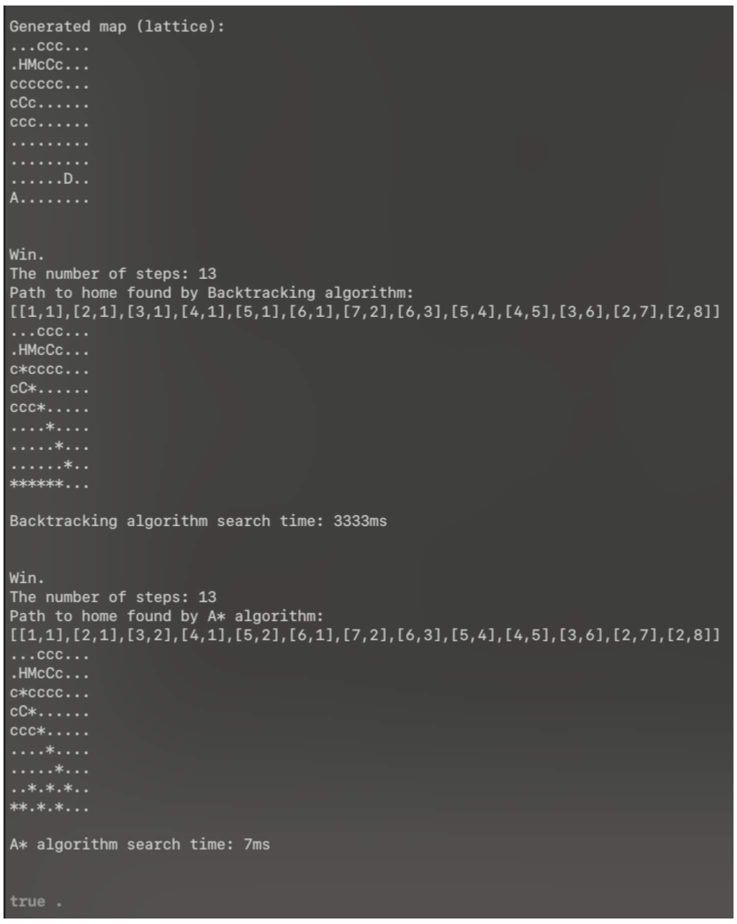
In general, the map is impossible to solve if the home, mask, and doctor (all of them) are separated from the actor by the map boundaries or covid infected cells. By "separated" it is meant that the actor cannot go to these cells unless he dies.

6 Interesting outcome / map

In this example, the home and mask are separated from the Actor agent by the map boundaries and covid infected cells. However, the actor can reach the doctor first and then go straight to the home with immunity to covid.

			c	c	c			
	H	M	c	C	c			
c	c	c	c	c	c			
c	C	c						
c	c	c						
						D		
A								

The output of the program is the following (`.` denotes an empty cell and `*` indicates an actor's steps to the home):



```
Generated map (lattice):
...ccc...
.HMcCc...
cccccc...
cCc......
ccc......
.........
.........
......D..
A........


Win.
The number of steps: 13
Path to home found by Backtracking algorithm:
[[1,1],[2,1],[3,1],[4,1],[5,1],[6,1],[7,2],[6,3],[5,4],[4,5],[3,6],[2,7],[2,8]]
...ccc...
.HMcCc...
c*cccc...
cC*......
ccc*.....
....*....
.....*...
......*..
******...

Backtracking algorithm search time: 3333ms


Win.
The number of steps: 13
Path to home found by A* algorithm:
[[1,1],[2,1],[3,2],[4,1],[5,2],[6,1],[7,2],[6,3],[5,4],[4,5],[3,6],[2,7],[2,8]]
...ccc...
.HMcCc...
c*cccc...
cC*......
ccc*.....
....*....
.....*...
..*.*.*..
**.*.*...

A* algorithm search time: 7ms


true .
```

7 References

[1] Wikipedia contributors, "Chebyshev distance," Wikipedia, The Free Encyclopedia, 02-Feb-2021. [Online]. Available: `https://en.wikipedia.org/w/index.php?title=Chebyshev_distance`. [Accessed: 09-Mar-2021].