IAI Assignment №1

Alexander Kurmazov - 19BS574

March 2021

1 Backtracking search

The first algorithm I have chosen to implement is the Backtracking search. It starts with the goal node and "backtracks" to the starting one, and the nodes between these two form a path. Then, all the possible paths are examined on its length, and if a new derived length is less then the current best length, the last one is updated. Also, it has an optimization: if a path that is currently being explored already has the length more or equal to the best one, then the further backtracking is stopped.

2 A* search

The second algorithm is the A* search. It uses the concept of opened and closed lists, where nodes are stored in the following form: a pair of coordinates, G, H, and F values. G is the cost from the starting node to the current node, H is the heuristic function, that is the cost from the current node to the goal node, and F is the sum of G and H. Opened list stores the nodes that have not been expanded yet, while closed list stores those that were already visited. The goal condition achieved when the goal node enters the closed list.

In my implementation, there are actually five different A* search calls. The first, second and third are to find a path from the starting node (0, 0) to the home, mask, and doctor nodes correspondingly; the fourth and the fifth are to find a path from the mask node to the home node, and from the doctor node to the home node correspondingly. Then, the second and the fourth paths are merged (since it is in fact one path from the starting node to the home node, visiting the mask node), and the third and the fifth paths are merged (similarly to the previous justification). Finally, the two merged paths and the original path from the starting node to the home node are compared, and the shortest of them is the actual result of the A* search start point. This extra four A* calls are needed because there might be a case, when it is crucial to pick up a mask or visit a doctor to make it to the home node in the shortest possible way, because the original A* search from start to home may just miss the mask and doctor nodes.

3 Scenarios

In the submitted code only the first scenario (where an actor can perceive a covid only from an adjacent node) was implemented. In fact, there are no difference in these two scenarios in terms of code, because both algorithms, Backtracking search and A* search, do not use this fact when choosing next moves - they only care about the presence of a covid in its Moore neighbourhood. However, one could write some predicates to take this fact into account, but they would not (and should not) affect the resulting path in its length. Speaking of the execution time, writing such predicates would only increase (or at least would not change) the time, because there will be more predicates to process and satisfy by a program.

4 PEAS description

The **performance measure** is whether the correct path was found (or it was concluded that there were no best path) or not.

The **environment** is the field, two covids, a mask, a doctor, and the home.

Properties of the environment:

- Partially observable (because the actor cannot see mask, doctor, and perceive covid from every position within the field)
- Single agent (because there is only one agent - the actor)
- Deterministic (because we determine the next state of the environment given the previous state and the action we are applying into it [1])
- Sequential (because every move the actor makes affects the future actions)
- Static (because the environment can change its state only when the actor makes a move)
- Known (because the design is aware about the rules of the "game")

The **actuator** is the actor's ability to move within its Moore neighbourhood.

The **sensor** is the actor's ability to perceive a covid from one or two cells away (depending on the scenario)

5 Statistical comparison

I have run 20 different tests on the same input for both Backtracking and A* searches, and have obtained the results that can be found here.

Then, I have calculated variances for both sets of outputs:

$$\mathrm{Var}_{A^*}(x) = 1.411$$
$$\mathrm{Var}_{BT}(x) = 618787987.2$$

Since the variance "is the average of the squared differences from the mean" [2], the interesting results can be noticed here. The variance of the A* search outputs sample is pretty small, which gives us a clue that the algorithm works uniformly in terms of execution time for different generated maps. In contrast, the variance of the Backtracking search outputs sample is enormous comparing to the A* variance, which means that the execution time is strongly dependent on the generated map. Of course, the sample used is relatively small, and such results could be just a coincidence, but it is interesting to see knowing how these algorithms work, and such results could be not completely useless.

Next, I have applied a two-sample t-test for this sets of data using Python **scipy** module and got the following result:

$$statistics = -3.686$$
$$pvalue = 0.00156$$

Which can be interpreted as: since the pvalue is not significant, i.e, less then 0.05, we can conclude that the two samples of data are significantly different [3] (which can be also seen with naked eye).

6 Graphical representation of the impossible maps

Notation:

- S - starting cell
- C - covid cell
- M - mask cell
- D - doctor cell
- # - covid infected cell

	0	1	2	3	4	5	6	7	8
0	S								
1									
2									
3									
4								#	#
5								#	C
6					#	#	#	#	#
7					#	C	#		M
8					#	#	#	D	H

Program output:

Homes: [(8,8)]
Covids: [(5,7), (8,5)]
Masks: [(8,7)]
Doctors: [(7,8)]

*** A* result ***
Path not found

*** Backtracking result ***
Path not found

	0	1	2	3	4	5	6	7	8
0	S		#	C	#				
1			#	#	#				
2	#	#	#				H		
3	#	C	#	M					
4	#	#	#						
5									
6									
7									
8								D	

Program output:

Homes: [(6,2)]
Covids: [(3,0), (1,3)]
Masks: [(3,3)]
Doctors: [(7,8)]

*** A* result ***
Path not found

*** Backtracking result ***
Path not found

7 References

[1] J. Brown, ”Introduction to Artificial Intelligence Week 3”, 2021.
[Online] Available: https://moodle.innopolis.university/pluginfile.php/104037/mod_resource/content/2/week3%20lecture.pdf

[2] ”Learn How Standard Deviation Is Determined by Using Variance”, Investopedia, 2020. [Online] Available: https://www.investopedia.com/ask/answers/021215/what-difference-between-standard-deviation-and-variance.asp [Accessed: 14 March 2021].

[3] R. Bedre, ”Student’s t-test in Python”, 2021. [Online] Available: https://www.reneshbedre.com/blog/ttest.html [Accessed: 14 March 2021].