

# AI assignment

*Rafik Hachana - BS19-04*

## Introduction

In this report, we will be analyzing solutions to help an actor agent reach home without catching Covid in a 9\*9 square lattice grid, the map also contains a doctor and a mask, both can help the actor to gain immunity to covid. We will discuss 2 algorithms to help the actor reach the home cell in the minimum number of steps: Backtracking, and the A\* algorithm. We will explain each algorithm as well as the PEAS model of our agent, and we will do a statistical analysis of the 2 algorithms with 2 different variants. Extra files are also attached to this report, they are mentioned in detail at the end.

## Algorithm description: Backtracking

Backtracking is the most naive algorithm, since it is an uninformed search. It will try every possible path until it reaches the home cell. The implementation is explained below.

Backtracking is recursive. It has a state consisting of the current cell, the safety bit (which describes the immunity of the actor in the current cell), and the list of cells that were visited on the way.

In each recursive call, we check all the possible next moves (maximum of 8), a move is possible if the cell is within the bounds of the map, and the cell is safe or we are immune to covid. Once the next cell is picked, we add it to the list of visited cells, increase the depth of the backtracking (number of steps taken so far) and recursively do the same for the next cell. The algorithm succeeds if it finds itself in the home cell, and it fails if it runs out of remaining steps to do.

In this way, we can use the backtracking algorithm to find the shortest path: We pick a depth  $k$  for the backtracking and try to find a path from the actor's position to the home cell, if we succeed, we try to find a shorter path. If we fail, it means that it is impossible to find a path shorter than  $k$ , so we try again with a bigger depth. This can be done using binary search, it wraps the backtracking algorithm, and it can be considered as a pruning strategy since we don't have to check all the possible paths to find the shortest one. A reasonable starting range for the depth is  $[0, 25]$ .

**Difference between variant 1 and variant 2:** In variant 2, the backtracking has a slight modification: we prioritize visiting cells that are far from covid. This modification doesn't make the algorithm more efficient, since in many cases the cells adjacent to covid are part of the path, this will be discussed further in the statistical analysis.

## Algorithm description: A\* algorithm

The A\* star algorithm is an informed search algorithm, it uses a heuristic to choose the next cell to visit. The heuristic ensures that the next cell is the most likely to get the actor closer to the home cell. In the implementation, the heuristic of each cell was chosen to be the sum of the diagonal distance between the cell and the home cell, and the shortest path (so far) to the cell from the home cell. In the second variant, cells that are adjacent to covid had a penalty on their heuristic value, so that we prioritize visiting other cells.

The algorithm keeps 2 lists while running: the open list and the closed list. The open list is the queue of cells to visit and it is kept sorted in ascending order by the heuristic value, and the closed list is the list of cells that are visited already. We start by having an empty closed list, and an open list with the starting cell. We check all the neighbors of the cell. If it is legal to move to the neighboring cell and the cell is in neither the open list or the closed list, we calculate the heuristic of the cell and push it to the open list. Otherwise, if the cell is in the closed list or the open list, we compare the length of the shortest path to it from the starting cell: If the value of the shortest path in the list is bigger than the current one, then we update the information of the cell in the closed/open list, otherwise we ignore it.

The next iteration of the algorithm starts by picking the first cell in the queue. We keep doing the same to each cell until we encounter the home cell and we succeed. The algorithm fails if the open list gets empty and we didn't encounter the home cell.

Unlike backtracking, A\* doesn't visit the cells in the order of the path. Therefore, we need to keep track of the parent of each cell, so that later we can construct the path by starting at the destination and following the parent of each node.

## **PEAS model of the agent**

**Performance measure:** The agent is expected to determine if there is a way to reach home from the start cell without catching covid, and to find the path of minimum steps. The shorter the path the better.

**Environment:** The environment of the agent is the 9\*9 lattice grid, with its content: the doctor, the mask, the covid agents, and the home cell.

The properties of the environment are:

- Partially observable: The agent can only see the content of its current cell, and of the neighboring cells in case of covid (depending on the variant). It cannot observe the whole environment at once.
- Multi-agent: We can consider the doctor, mask and covid as agents. Since the doctor and mask are there to help the agent, and the covid agent is capable of beating the agent and making it lose.
- Deterministic: We can know the outcome of every step the agent takes. There is no stochastic element in the environment.
- Static: The environment doesn't change while the agent is deciding its next step. The home, covid, mask and doctor cells are all stationary.
- Known: We know in advance the rules of the environment and the behaviour of the other agents.
- Sequential: The next step depends on the steps previously taken in the map
- Discrete: The actions of the agent are discrete moves from a cell to another cell in its Moore neighborhood.

**Actuators:** The actuators of the agents are the mechanism used to move from one cell to the next.

**Sensors:** The sensors are the actor's ability to perceive its current cell for home, doctor or mask. And its ability to perceive covid cells that are one step away in variant 1, or covid cells that are 2 steps away in the second variant.

## Statistical analysis

The data was obtained by generating random maps with 2 Covid cells and running 4 algorithms on them (Backtracking variant 1, Backtracking variant 2, A\* variant 1, A\* variant 2), and then recording the execution time of each algorithm as well as the number of steps of the shortest path. This test was done 50 times. The dataset with all 50 entries can be found in the .csv file attached with the assignment.

The following was found:

Value	Backtracking V1	Backtracking V2	A* V1	A* V2
Average	4.068	4.072	0.004	0.005
Variance	211.1142398	213.3238315	0.00000915959 1837	0.00001531469 388
Standard deviation	14.52977081	14.60560959	$3.01 \cdot 10^{-3}$	$3.91 \cdot 10^{-3}$

We can see how the extended perception in the 2nd variant doesn't help the performance of the algorithm, the performance is actually slightly worse because of the extra computation performed.

From this data we can perform the Student T-test:

The null hypothesis is:

$$H_0 = \{a \text{ pair of algorithms is the same (in terms of performance)}\}$$

Each algorithm has 50 samples, therefore, the degree of freedom is:  $50+50-2 = 98$ .

Let's pick the probability of rejecting the null hypothesis to be 0.05 (95% confidence), in this case the critical t-value is 1.6606.

The next step is to compare the algorithms by calculating the T-value for each pair using the following formula:

$$T = \frac{|\mu_1 - \mu_2|}{\sqrt{\frac{V_1 + V_2}{98}}}$$

### Comparing Backtracking variant 1 and backtracking variant 2:

The T-value is 0.0268, which is less than the critical value. Therefore the null hypothesis is preserved: Backtracking has the same time performance in both variants.

### Comparing A\* variant 1 and A\* variant 2:

The T-value is 4.096, which is larger than the critical value. The null hypothesis is rejected here. Therefore A\* has a different time performance depending on the variant. However, the extra information in the 2nd variant doesn't help the performance: The performance is worse because of the extra calculations, the extra Prolog inferences, and in some cases the extra number of cells visited to find the home cell (in the case where the path home contains cells adjacent to covid).

### **Comparing Backtracking variant 1 and A\* variant 1:**

The T-value is 2.769, which is larger than the critical value. The null hypothesis is rejected, and we conclude that Backtracking and A\* don't have the same performance in variant 1. This is explained by the efficiency of the A\* informed search compared to the exponential time complexity of backtracking.

### **Comparing Backtracking variant 2 and A\* variant 2:**

The T-value is 2.756, which is larger than the critical T-value. The null hypothesis is rejected. A\* and Backtracking don't have the same performance in variant 2. This is due to the same reasons explained in the previous comparison.

### **Notes on the dataset:**

Contrasted with A\*, we can see how backtracking takes exponentially more time when looking for a longer path. It takes at most 2.5 seconds to find a path of 8 steps, but runs for more than a minute on average if the required path length is 10. In the latter, A\* doesn't exceed 0.014 seconds.

### **Impossible maps**

In the following maps, the actor cannot find a path to reach home from the start position (1,1).

								HOME
	DOC							
X	X	X					MASK	
X	COVID	X	X					
X	X	COVID	X					
ACTOR	X	X	X					

				X	X	X	MASK	HOME
				X	COVID	X		DOC
				X	X	X	X	X
						X	COVID	X
						X	X	X
ACTOR								

The following map was practically unsolvable with backtracking, because of the large length of the minimum path required. The algorithm would run for an indefinite amount of time.

MASK					X	X	X	HOME
					X	COVID	X	X
					X	X	COVID	X
						X	X	X
ACTOR								DOC

### Interesting map outcome

In the following map, the A\* Star algorithm gives a longer path than the one given by backtracking, simply because of the nature of its greedy heuristic: It will ignore the doctor cell and take a longer path around the covid infected cells instead.

						X	COVID	X
						X	X	X
				X	X	X	HOME	
				X	COVID	X	P	
			P	X	X	X	P	
		P		P	P	P	P	
	P	MASK						
ACTOR							DOC	

In the above map, A\* would skip the mask while it is following the heuristic. Which is not the shortest possible path here. (The path is denoted by the letter P).

### Conclusion

For this task, we can see that A\* is a better algorithm in terms of performance, however it can give a longer path in some cases. This might be solved by improving the heuristic in some way, or adding a random element to the search where the algorithm would take a random step from time to time. Backtracking is much slower but it is guaranteed to find the shortest path.

### Attachments

With this report, I have included the full Prolog code (file: knowledge.pl), with the option to execute it with a randomly generated map of 1 or 2 covid agents, or with a user-input map where the user can specify the number of covid agents. For instructions on how to run it check the readme.txt file.

The whole dataset for the statistical analysis is included in the file data.csv .