Artificial Intelligence

Final Assignment Report
AI model

Members: Miguel Castuera García / Marcos Caballero Cortés
NIAs: 100451285 / 100451047
Group: 89

Table of contents

Final Assignment Report **1**

Table of contents **2**

1. Executive Summary. **3**

Problem statement. 3

Proposed solution 3

2. Objectives. **3**

3. Formal Description of the AI model. **3**

4. Methodology. **3**

5. Resources Used. **4**

6. Development. **4**

7. Results. **4**

8. Conclusions. **4**

9. Budget. **4**

1. Executive Summary.

Problem statement.

In the public roads of this country we have a potential problem to be solved. According to several studies, people lose on average more than half an hour due to traffic jams each day. If an average person works an average of 250 days a year they are losing a little bit more than 5 days in traffic jams each year. At a crossroad, the flow of vehicles in each direction encounters a traffic light before reaching the intersection, which can be either green or red. An operator must select every 20 seconds which of the three traffic lights to turn green, this seems to be something that would be done in the 1950's, so let us introduce our solution.

Proposed solution

Our solution to this is to implement an artificial intelligence using the sensors that the DGT already has to measure the traffic levels in each street. This AI will solve all our country's trafic problems.
Based on our calculations, this will create more than 2000 high qualified jobs, it will prevent people from suffering from accidents, saving more than two hundred million euros each year in car repairs, and last but not least, it will save the lives of at least 30 people every year in our country.

2. Objectives.

Our main objective is to develop an artificial intelligence that is capable of solving the traffic problem in our cities. This will make the citizens to be more happy and to have more time for themselves, also solving some of the traffic accidents produced by bottlenecks and saving the citizens a lot of money due to the accidents.

Based on some studies, the lack of roadblocks reduces stress of the people and also reduces noise levels in the streets, allowing people to rest and sleep better. This, in people's lives is some great news, as they can be more productive in the day, having more time after finishing work, homework or whatever they finished doing. Furthermore, traffic jams are worse in big cities, as we can see in Madrid, everyone is in a rush to get straight to where they have to be, even if they have the time to not rush, as far as we know, this is caused because people don't know if they are expecting a traffic jam or an accident that will cause them to get late to work. This can also be solved by the implementation of our product, as, our AI will solve that problem, this will make them be sure about the time they have to leave home to get wherever they want.

3. Formal Description of the AI model.

The automata is defined as follows: <S,A,P,R>

- S: States = $S_T \epsilon$ {'HHH', 'HHL','HLH','HLL','LHH','LHL','LLH', 'LLL']
- A: Actions = {'N', 'E', 'W'}
- R: Cost
  - c('N') = 1
  - c('E') = 1
  - c('W') = 1
- P: Transition function

Action E

$P_E(S_T + 1/S_T)$

	HHH	HHL	HLH	HLL	LHH	LHL	LLH	LLL
HHH	0.616438	0.0	0.383562	0.0	0.0	0.0	0.0	0.0
HHL	0.246305	0.41133	0.118227	0.224138	0.0	0.0	0.0	0.0
HLH	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
HLL	0.0	0.0	0.380952	0.619048	0.0	0.0	0.0	0.0
LHH	0.22973	0.0	0.119369	0.0	0.448198	0.0	0.202703	0.0
LHL	0.088636	0.15	0.027273	0.077273	0.154545	0.302273	0.079545	0.120455
LLH	0.0	0.0	0.302139	0.0	0.0	0.0	0.697861	0.0
LLL	0	0	0	0	0	0	0	0

Action N

$P_N(S_T + 1/S_T)$

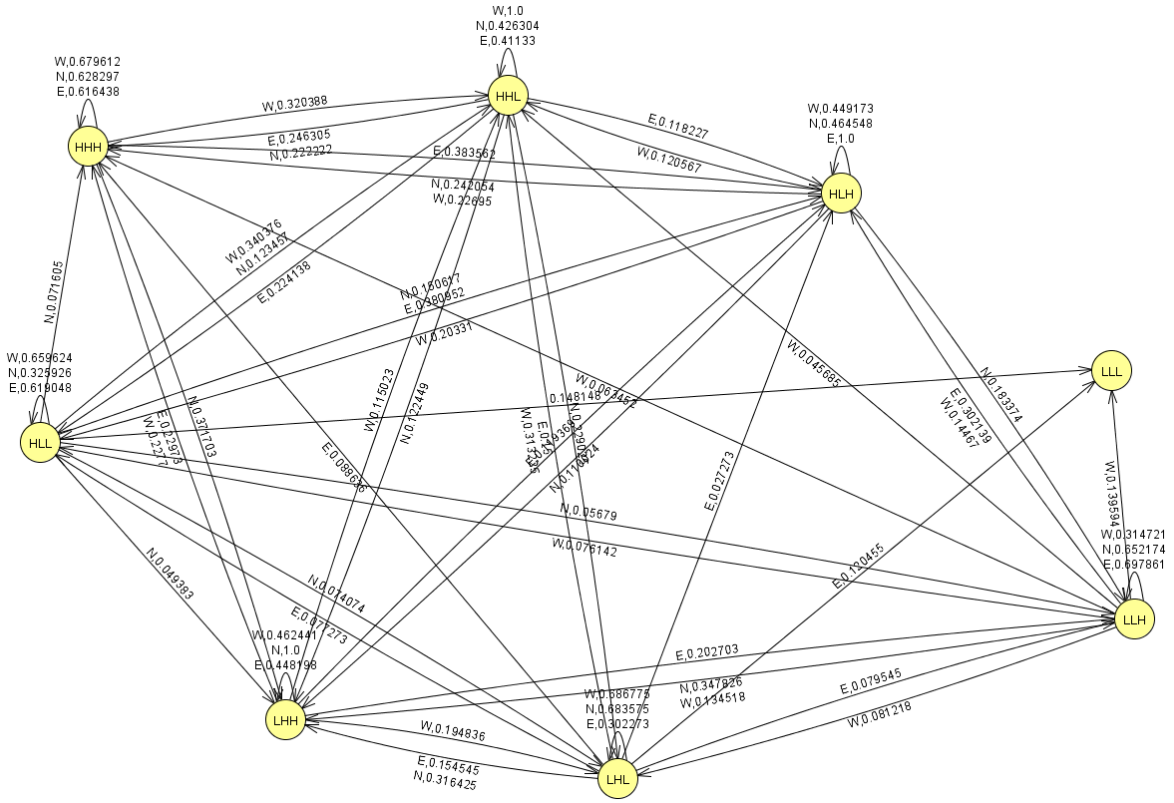
	HHH	HHL	HLH	HLL	LHH	LHL	LLH	LLL
HHH	0.628297	0.0	0.0	0.0	0.371703	0.0	0.0	0.0
HHL	0.222222	0.426304	0.0	0.0	0.122449	0.229025	0.0	0.0
HLH	0.242054	0.0	0.464548	0.0	0.110024	0.0	0.183374	0.0
HLL	0.071605	0.123457	0.150617	0.325926	0.049383	0.074074	0.05679	0.148148
LHH	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
LHL	0.0	0.0	0.0	0.0	0.316425	0.683575	0.0	0.0
LLH	0.0	0.0	0.0	0.0	0.347826	0.0	0.652174	0.0
LLL	0	0	0	0	0	0	0	0

Action W

$P_W\,(S_T + \ 1/S_T)$

	HHH	HHL	HLH	HLL	LHH	LHL	LLH	LLL
HHH	0.679612	0.320388	0.0	0.0	0.0	0.0	0.0	0.0
HHL	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
HLH	0.22695	0.120567	0.449173	0.20331	0.0	0.0	0.0	0.0
HLL	0.0	0.340376	0.0	0.659624	0.0	0.0	0.0	0.0
LHH	0.2277	0.115023	0.0	0.0	0.462441	0.194836	0.0	0.0
LHL	0.0	0.313225	0.0	0.0	0.0	0.686775	0.0	0.0
LLH	0.063452	0.045685	0.14467	0.076142	0.134518	0.081218	0.314721	0.139594
LLL	0	0	0	0	0	0	0	0

Below, we find the graphical representation of the probabilities. We won't include the ones with probability 0, because they will never happen. The state LLL is the final state, so our automaton shouldn't have any other possible connection with other states. We won't include cost on our representation as the cost of all the transitions is 1.



4. Methodology.

To make it easier for us to perform well in this project, we divided the problem into smaller pieces. Then we used some dictionaries to access the data more easily, such as the one used to save the probabilities; we also used two dictionaries called V1 and V2, the first one is for getting the minimal cost, and another one that is being updated all the time, this last one it is updated depending if it finds or not a better action to perform, depending on the cost. Inside those dictionaries we can find several lists, depending on which of the lists you focus, you can see the probabilities, separated by actions and states, the cost or the traffic light that will be turned on.

We divided the code in three parts; in the first one tried to get the data given by the excel file that we were provided with. In the next part we got all the data from the previous part and we performed a function to get all the probabilities from each state and each action to the next state. This second part makes it easier to get the best performance and the optimal policy.

5. Resources Used.

For creating this AI we used python to develop the software.
Apart from that we contacted some universities in the US so they could share their opinion about what they would change or what they would implement. This way we ended up with some fresh and good ideas for the implementation, making sure we were doing it right.

For the implementation in the last part of the code, we used the formula provided in the theory slides to get the optimal policy.

For the implementation of the automata we used a software called Jflap, which we used in the subject TALF in the last semester.

6. Development.

For the development of this AI we first had to take into account the file of data available for developing this project. First of all our program has to read the content of that file. For that we implemented the python library csv, whis makes this process more efficient and easy. Then we create a for loop, which takes all that information of the file and writes it into a python list called “ListOfWords”, this will make the rest of the process more easy to perform.

Then, we had to compute all the probabilities, so we created a new method called “funcDictionaryOfProbabilities”, which computes the probability of all the states in a list. Inside the dictionary there are 3 different entries, inside those entries, there are 8 different lists, referencing each one of the states. In those lists, we have the probabilities of each state to go to the next one. For example, in the first entry “E”, we have 8 different lists, in the

first list, which is referencing the state “HIghHighHigh” we have this values: [0.6164383561643836, 0.0, 0.3835616438356164, 0.0, 0.0, 0.0, 0.0, 0.0], there is a probability of 0.6164 aprox to stay in that same state, also, there is a possibility of 0.38356 aprox that it goes to “HighLowHigh” state.

Then, we created a for loop to calculate the optimal policy. There are two dictionaries at first, one for getting the minimal cost, and another one that is being updated all the time, this last one it is updated depending if it finds or not a better action to perform, depending on the cost. Inside the loop, for finding the best policy, we implemented the Bellman equation, which searches the way to get to the final state with the minimal cost.

7. Results.

Question 1

When the starting data of all the directions is Low, we have to take into account that this one is the final state, then, we have reached our goal. In the case that we are already in that state, we would finish at that moment, as we reached our goal.

Question 2

It is true that the statement didn’t say anything about the costs of the actions, but in this exercise it makes sense. As the only thing that is changing is the lights of the traffic lights, we do not think that's something that has a big cost. Apart from that we decided to use a cost 1 in each of the costs, as we tested many times and this approach was the one with less iterations, making it the most efficient way to outlook the solution.

Question 3

HighHighHigh [40.678849913693995, 'E']
HighHighLow [37.9182874850963, 'E']
HighLowHigh [38.07418881042905, 'W']
HighLowLow [30.34485670274234, 'N']
LowHighHigh [38.235556110558676, 'E']
LowHighLow [31.734643757478374, 'E']
LowLowHigh [30.632565632018125, 'W']
LowLowLow [1.0, 'W']

Question 4

For computing the optimal policy we used bellman’s equation, the results are shown in question 3.

To understand this, we have to think about a representation of all the states, each one of the first elements of the lists are the costs for going from the state we are in to the next state, which is the following one. The second value of the lists is the traffic light that is turning green so the cars can go through.

Question 5

According to the statement we have been given, we only can have traffic in three directions, so in this case, we can't compute a result, as our program has been designed to have traffic in north, east or west directions.

8. Conclusions.

technical comments related to the Project development and personal comments: difficulties, challenges, benefits, etc.)
As we already commented in the first section, the implementation of our AI model will have a lot of benefits, it will reduce the number of accidents in crossroads, saving lives and saving a lot of money to the owners of the cars, as well as to the insurance companies. It also may help the people living in the buildings near the intersections as the absence of traffic jams will reduce the levels of noise near the crossroads, reducing the stress levels and helping them to rest. The implementation will also make the people living in the cities it will be implemented in, to not be in a rush, as they know they won't be waiting too much on the road due to traffic congestions, also reducing the risk of accidents.

The difficulties we have encountered are because of the implementation of the AI model in code, as we were not used to do that. Because of that, we spent the most of our time coding, trying to make it work. After a lot of suffering, we were able to create functional, cutting edge, and ready to use AI. We had several problems trying to read the excel file we were given, but the most challenging thing we found was to implement the bellman's equation, but after some research, we were able to make it.

9. Budget.

After a big study and weeks of work we have determined that our work would have cost at least 1.250.000 euros each, as in the development of this AI we spent a lot of time and this AI can be implemented in all crossroads of the world, in this case, the governments will get the money of the investment in less than a year.
We are a team of two people, and we can be contacted by several companies as we have a great reputation in the development of artificial intelligence. We usually get paid for results and not for hours. Usually we agree to get paid 25 % of the total cost before the development of the project, as numerous projects we have done before were made successfully and we satisfied our clients in terms of what they asked us to get. After presenting the project we should get the rest of the money before giving them full access to it.
In previous projects we also had a deadline date, if this was the case, we usually agree to not get paid a 1% of the total amount per day after the deadline is reached.