AAMAS Project Report - Autonomous Driving Agents

Group 34 - Alameda

Diogo Eusébio - 87650 Rodrigo Lima - 83559 Simão Nunes - 86512

1 INTRODUCTION

We plan to tackle one of the problems that was proposed in the assignment document: Movement, Traffic and Routes with autonomous vehicles.

Driving automation will change the lives of everyone who uses a vehicle. It will improve the quality of life of the elderly who can't drive, it will increase roadway safety, and it will overall improve a significant portion of the transportation sector. We chose this topic because we believe that driverless cars could redefine transportation, cities, and countless tangential industries but the success of these relies upon its understanding of the route and its external surroundings.

For this project our agents will be cars with sensors and communication capabilities, as such they will follow a reflection; communication; action cycle.

This project was created and tested with Unity Developer Tool Version 2019.3.11.

2 ENVIRONMENT

We started this project by creating the environment where the *Agents* will act upon. This environment can be described as a unidirectional *Graph*, where *Nodes* are described by their position in a plane and their connections to other nodes. Some *Nodes* have *Traffic Lights* that are perceived and respected by the *Car* agents. Those *Traffic Lights* are controlled by a *Traffic Light Manager*, that groups them into their intersections and controls the timing at which they interchange.

3 AGENTS

3.1 Properties

Following the definition of the Environment, we created the *Car* agent. We aim to create a simplified version of reality. As such, *Cars* have a set of properties we deemed essential to tackle this problem: position; speed; gas consumed on trip; distance traveled on trip; carrying capacity; gas consumption multiplier and static and dynamic traffic flow costs. These traffic flow costs represent how much traffic each car generates on the graph. The static cost is added to each edge belonging to the path when the car calculates the path it's going to take and is never deducted until the end of the simulation, meant to represent the wish a car has to use that path, and can be interpreted by other cars as previous knowledge of traffic, i.e.: an agents know that a street is propitious to traffic, independently of time.

, *2020*

https://doi.org/

The dynamic flow cost is added to the edge the car is currently traversing and removed once it reaches another edge, giving a real time representation of the car's traffic generation on the graph.

We defined these properties with values we believed roughly represented real world parameter relations when comparing the different types of cars.

Table 1: Properties per Car type

Car type	Capacity	Speed	Gas Consumption	Dynamic traffic cost
Individual	1	45	1.1	9%
Mini Cooper	2	40	1.0	10%
Truck	5	35	1.4	15%
Bus	10	30	1.35	20%

3.2 Sensors & Behavior

Our *Cars* are reactive and their sensors can be categorized into two groups: **Local sensors** and **Global sensors** being vertically layered, where **Local sensors** have hierarchical priority over **Global sensors**, as we want to ensure that everything on a cars proximity is fine (i.e.: not crashing into another car) before worrying about traffic on a global scale and how it will affect path-finding.

3.2.1 ***Local Sensors.*** Responsible for avoiding collisions, detecting and respecting *Traffic Lights*. In our simplified simulation the car's speed is binary, either being 0 or a constant value defined prior to execution and there is no notion of acceleration, from one frame to the next the car speed immediately changes.

Our agents anticipate collisions with ray cast whiskers, meaning that if the whisker detects another object it triggers the car to stop until the object it detected is no longer touching the whisker. The configuration we used consisted of two whiskers oriented forward at the width of the car, as shown in the next figure. Configurations that would detect a wider range might cause cars to stop when detecting oncoming traffic, that is actually causing no problems.

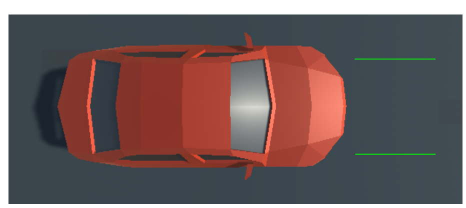

Figure 1: Representation of local sensors.

We believe this configuration is the most appropriate because cars can't move sideways, and it's okay to have cars pass next to each other as long as they don't touch.

3.2.2 ***Global Sensors.*** Responsible for reading traffic flow and updating the car's path according to a distance and flow cost multiplication:

$$NextNode = min\{Connections(distance * traffic\ flow\ cost)\}$$

where the distance is euclidean and the traffic flow cost is initialized at **1** and incremented by the agents. For example, imagine 2 cars with the same path, one behind the other, if the first car has a **10%** traffic flow cost it will mean that second car will rather find another route at the next intersection that's no more than 10% longer than it's current path rather than keep on their initial path following the first car.

3.3 Communication

On the Local scale cars do not communicate with each other. We feel there is no need nor reason for this, as: firstly, in a more realistic setting not all cars will be autonomous, and their integration will be gradual. As such, they need to be able to deal with proximity conflicts only trough sensory stimulus because non-autonomous cars cant be expected to know how to communicate. Dealing with such problems must be handled in a way that passes a Turing test, meaning that human drivers can't distinguish them from others.

On a Global Scale communication was implemented in an indirect way. The information is stored in a centralized structure, our *Graph*. All cars read and update their traffic flow costs on it, to avoid dealing with hundreds of individual messages being sent by all cars to all other cars. We believe this, in a way, simulates an online GPS service that measures traffic providing results similarly to *Google Maps*. As explained previously in section 2.2, the cars update a value that represents a traffic flow cost they are causing, that is available for all other cars to read when they update their paths.

4 TESTING AND RESULTS

4.1 Local sensors testing and efficiency of traffic lights

First we wanted to test the car's local sensors and compare the viability of an environment with *Traffic Lights* with an environment without them.

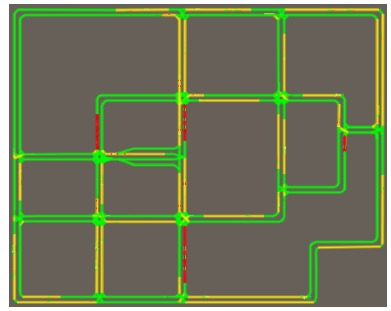

Figure 2: Connection's flow with 90 random agents, with traffic lights.

The figure 2 is a snapshot of a simulation where 90 cars (of random types) are randomly wandering through the graph, that is, when they reach a node they go to a random connection. The colors represent the flow of each connection, green being no traffic, yellow being light traffic and red meaning a lot of traffic. We observed that there is some congestion, as cars accumulate at red lights. This is a problem of finding the optimal ratio between the car's speed and the rate at which *Traffic Lights* change color.

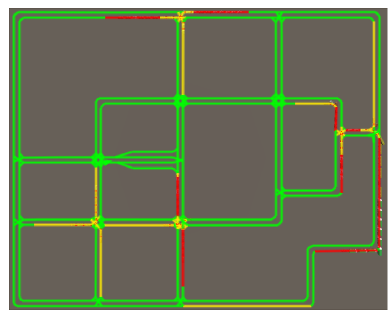

Figure 3: Connection's flow with 90 random agents, without traffic lights.

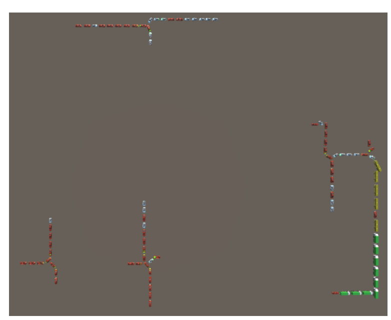

Figure 4: Clearer representation of figure 3.

The figure 3 is a snapshot of a simulation with the same parametrization but without *Traffic Lights*. We expected this to yield better results than it's counterpart, but that wasn't the case, even thought the results were not to different. As such, we will say that these tests are inconclusive as we suspect the problem lies within the mechanisms used for the Local sensors, since we observed traffic jams would occur because cars would stop in the middle of the intersections, as demonstrated in figure 4.

We believe that in a realistic setting with more robust sensors and decision making systems, that include communication of some sort, intersections with no *Traffic Lights* will be more efficient, having less cars stop and for shorter periods of time, as they would coordinate with precision, human drivers can't. Unfortunately we were unable to create such system. As a result the following tests were only performed with *Traffic Lights* on.

On the other hand, one result we did manage to get was synchronization of movement when *Traffic Lights* turn green, so it is also plausible that the existence of a law enforcing mechanism such as *Traffic Lights* can be efficient, especially if the problem mentioned before, of finding the best ratio between car's speed and color change rate is solved. This also lessens the work load that cars have, placing the responsibility of traffic coordination on a traffic light management system instead of the cars themselves.

4.2 Global sensors testing and coordination efficiency

Secondly we tried testing the car's communication capabilities and the efficiency of the heuristic explained in section 3.2.2 by comparing it's results to a greedy approach, where all cars just tried to minimize the distance traveled. We tested for each type of car separately and fixed the capacity (to 100 units) and route (equal start and end node).

Table 2: Quantity of each car type to reach fixed capacity

Car type	Capacity	Quantity
Individual	1	100
Mini Cooper	2	50
Truck	5	20
Bus	10	10

Regarding time performance, the results yielded the following:

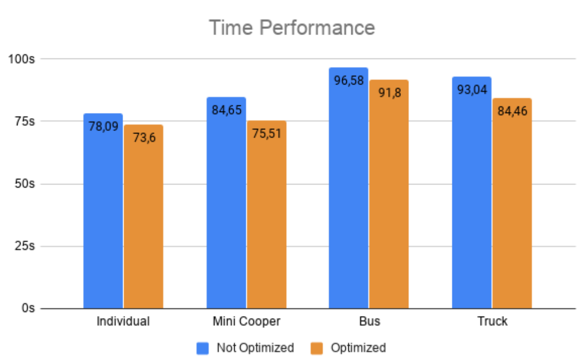

Figure 5: Time Performance Histogram.

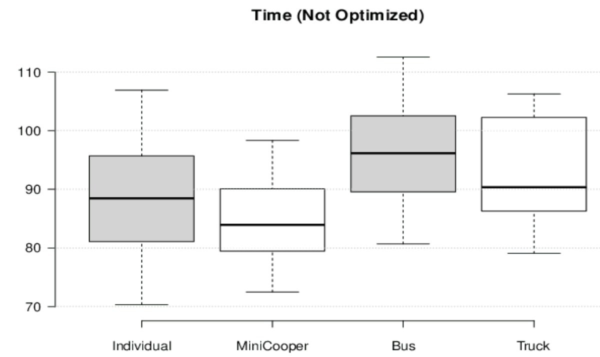

Figure 6: Time analysis for non optimized routing.

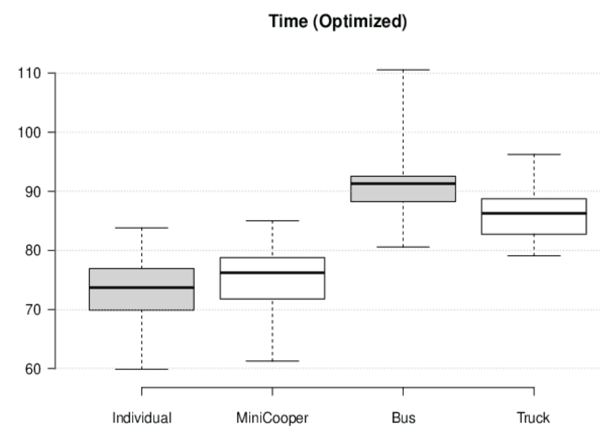

Figure 7: Time analysis for optimized routing.

Leading to a quick conclusion that the path optimization heuristic based on traffic flow really reduces the time spent in a route.

Regarding distance performance, the results yielded the following:

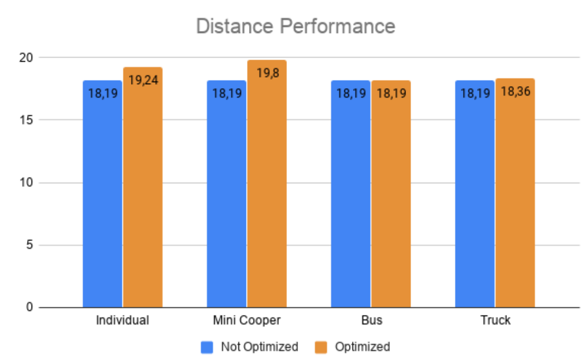

Figure 8: Distance Performance Histogram.

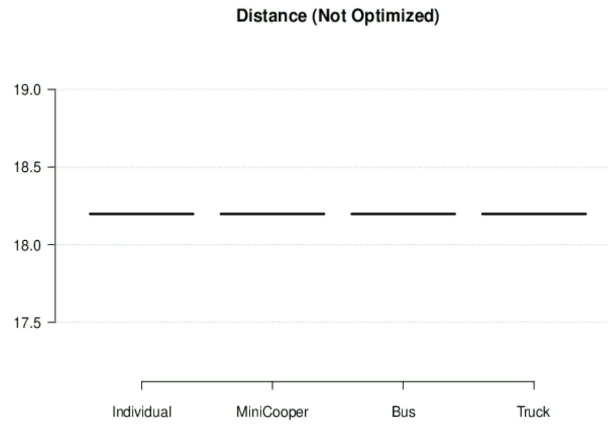

Figure 9: Distance analysis for non optimized routing.

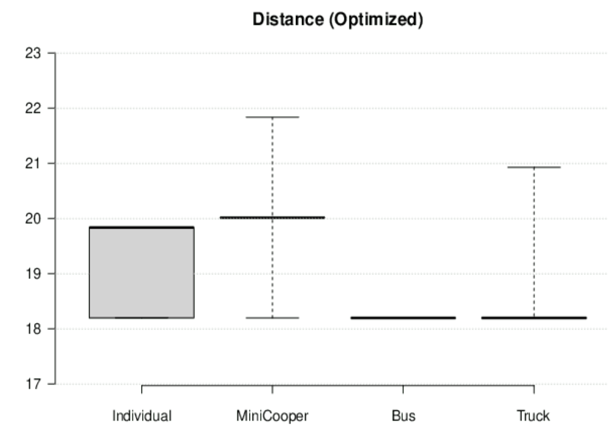

Figure 10: Distance analysis for optimized routing.

Therefore, from figure 8 we can understand that the cutback in time comes with a downside, greater distance traveled. That is true to all agents except the bus, which reduced it's time and maintained the distance. That can be odd at first to understand, but we ensured (by watching the simulation) that all buses did not follow the same path. So it might only be due to our graph implementation, where some different routes can have the same distance.

Furthermore, figure 9 allows the understanding that not having a coordination mechanism leads to all agents following the same path, which yields that the distance traveled is always the same. This in an overloaded environment is detrimental, as the agents will act in a greedy way, wanting to travel smaller distances, affecting everyone else's time and consumption results in a negative way.

Regarding gas consumption, the results yielded the following:

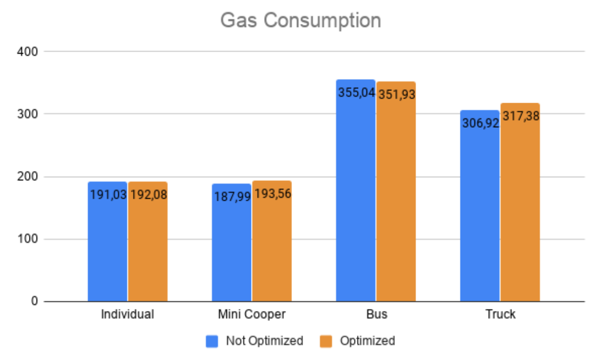

Figure 11: Gas Consumption Histogram.

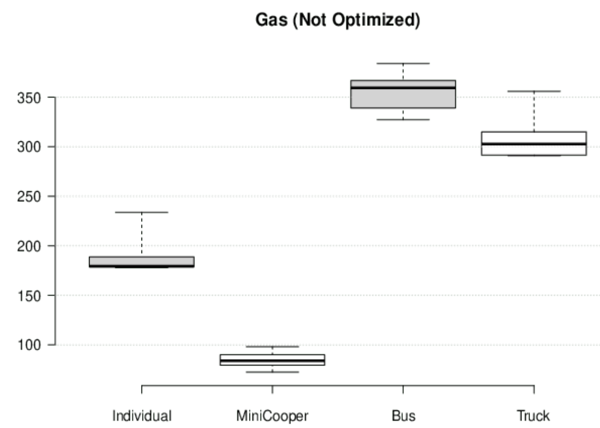

Figure 12: Gas Consumption analysis for non optimized routing.

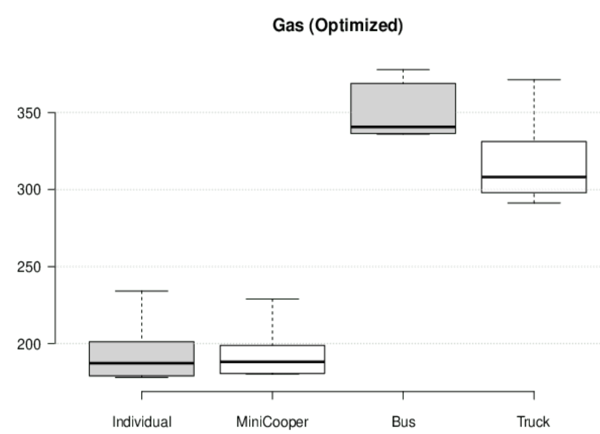

Figure 13: Gas Consumption analysis for optimized routing.

The analysis of figures 11, 12 and 13 leads to a conclusion which we could already expect, since the distance traveled is greater, the gas consumption is greater for most cases. Although that happens for most cases, in the case of buses that is not true, but if we remember the previous results, the distance traveled for the bus was the same for both cases, which leads to a plausible conclusion that our coordination mechanisms avoids traffic congestion and therefore avoiding stop starting, leading to a lower gas consumption.

4.3 Traffic Flow: Individual cars and Community transports

Lastly we wanted to hypothesize if the coordination of individual cars is good enough to make the need for community transports obsolete. Meaning that we wanted to compare a world where everyone as immediate access to an individual car that is, for example, able to park next to it's home and job place, and immediately accessed whenever each individual wants; versus a world where everyone relies on public transporting with hopes that the reduced traffic due to vehicle sharing, makes traffic so much better that travel times are reduced to the point that having to wait for public transports it's on average worth it.

Due to time constraints we did not implement the system where individual persons wait for community transports.

5 CONCLUSION

In conclusion our results leave us optimistic for the future of autonomous driving, as the results, although far from optimized, suggest that it is possible to create a system of artificial agents that can perform at least on par with humans and most probably be more effective.

5.1 Future Work

In this section we'll discuss what we would do if we kept working in this project.

The first step would be to fix the problems our project currently has. Making the Local sensors more robust. As already discussed, we believe the agents should be able to deal with environmental changes merely trough reactive behavior, in a way that it allows multiple agents to coordinate themselves only by reacting to the world and following traffic rules and good practices without the need for communication.

Secondly we would fine tune the agents' properties to make the simulation more realistic, as the values we used are only speculative guesses. To do this we'd need to gather data from real world examples: on how much more gas does a truck spent compared to a car; how much faster are individual cars compared to busses; how much more traffic does a bus cause compared to car, and find out if the way our system works is a good representation of reality or should the system be changed completely.

And finally we would study one other variation of the heuristic that prioritizes the minimization of gas consumption. And try to observe how agents with different wishes, integrate themselves. Some agents might want to minimize consumption while others might want to minimize travel time, trying to define how considerate an agent should be to others. Ultimately deciding if traffic is a coordination or competition problem over resources.