AAMAS Project Report - Traffic Junction

Group #024

João Simões
nº 92499

André Oliveira
nº 93686

Maria Beatriz Venceslau
nº 93734

ABSTRACT

Self-driving cars are gaining popularity and the most important thing when programming such cars is to ensure the safety of the drivers as well as the pedestrians. This is especially true in instances where the cars have to manage traffic on their own, for example in a traffic junction with no road signs.

This project intends to simulate the behaviour of traffic in a 4-way junction as a multi-agent system, given that it is a common real-world problem. In these cases, cars can take one of two approaches: to follow the conventional rules or to be self-interested.

In the development of the project, we will assume the problem to consist of cars as autonomous agents with predetermined paths and evaluate their behaviour according to evaluation metrics such as number of steps taken by an agent from start to finish, number of collisions that occur on a run and finally the number of dead ends.

KEYWORDS

AAMAS; Coordination; Cooperation; Multi-Agents; Traffic; Greedy

1 INTRODUCTION

Our motivation to build this system originates from wanting to simulate an environment that represents a current and common problem. This system can be used to understand how congestion and the strategies each driver takes, affect the circulation of cars at a junction.

This type of problem has been approached by Anurag Koul[1] by using a collection of multi-agent environments based on OpenAI gym. Our environment will be taken from his GitHub repository. Sukhbaatar, Sainbayar, and Rob Fergus[3] wrote a paper in which they explored the effects of communication in the environment we are using. They explored the failure rates for different types of models, concluding that communication consistently improves performance. Based on their conclusions we decided to use an agent that also uses communication.

Our project aims to simulate a traffic junction, where cars have to coordinate and cooperate to avoid collisions and dead ends while following their designated path. Agents will spawn at the beginning of any of the four lanes and will be assigned a predefined route to navigate the junction and reach the edge of the grid. Each agent will follow one of four different approaches to manage the traffic jam.

Our system relates to a real-life problem as every driver will have encountered this scenario in everyday life. There are many strategies one may take to reach the other side of the junction. The common one, in the absence of traffic signs, is to give priority to the cars on the right. However, one may choose to give priority to reaching their destination as fast as possible over respecting every traffic law, such is the case of an ambulance for example.

The objective of this project is to analyse what happens when there is an increase in traffic versus a reduction in traffic, as well as to compare the behaviour of the different types of agents developed. We will implement a greedy agent, a cautious agent, and one that respects the convention laws of yielding passage to the cars on their right, which we will refer to as the decent agent. We can also compare the behavior of these agents with a baseline (a random approach). Each agent can communicate their intentions however, only the decent one will consider all the received intentions when making a decision. The goal of this project will be to identify the best way to manage traffic, analysing the number of collisions and dead ends reached, as well as the time taken, as evaluation metrics.

2 APPROACH

2.1 Environment

Our environment consists of a four-way junction. At each step, there is a probability for a new car to appear from each of the four directions.

Each of these cars is assigned one of the three path directions (right, forward, or left). Once the cars reach their destination, they are removed from the environment.

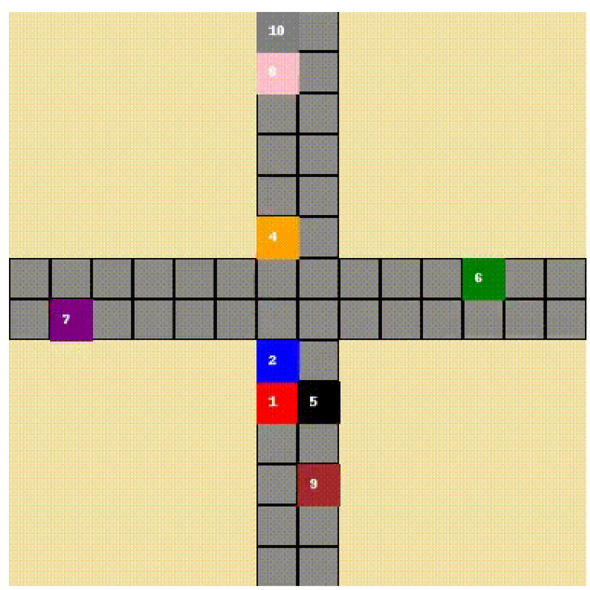

Figure 1: Environment Example

AASMA Project, IST, 2022, Lisbon, Portugal

https://doi.org/

2.1.1 *Environment Properties[2].*

- Inaccessible - The agent can not obtain complete, accurate, up-to-date data about the environment's state, as it can only see what is in its range of vision.
- Deterministic - Each action an agent can make has a single guaranteed effect (stay in place or continue in its path).
- Static - The world does not change while the agent is deliberating (it deliberates at each time step and nothing happens in between time steps).
- Discrete - The environment has a fixed, finite number of possible actions and precepts.
- Episodic - The world can be divided into a series of intervals (episodes) independent of each other, with each episode representing the behaviour of the selected number of cars managing the junction.

2.2 System Architecture

Our environment consists of a 4-way junction on a 14x14 grid. New cars are added to the environment with a probability p from all directions until the maximum number of cars is reached. When a car is spawned, it is randomly assigned one of the three possible routes (left, forward, or right).

At every time step, a car has two possible actions: gas, which advances it by one cell on its route, or brake, to stay at its current location. A car will be removed once it reaches its destination at the edge of the grid.

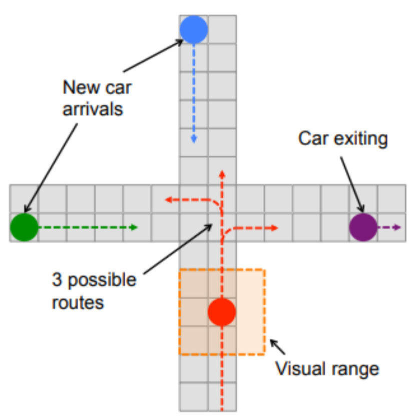

Figure 2: Environment Architecture [3]

If two cars share the same position, it is considered a collision.

The simulation is terminated after a maximum number of steps and is classified as a failure if one or more collisions have occurred.

Each car is represented by a one-hot binary vector set *(n, l, r)*, that encodes its unique ID, current location, and assigned route number respectively. Each agent controlling a car can only observe other cars in its vision range. Each car can inform the others about their intentions.

We have an agent that follows the conventional traffic rules and believes that the car to its left will give him the priority in the intersection and intends to wait and give priority to the car at its right.

We also have another agent that follows a greedy approach. This agent will always move forward. It does not pay attention to the other agents, since its only goal is to reach the end of its path.

We then have an agent that follows a cautious approach. An intermediate between the two previous approaches, since it is self-interested, it intends to finish as fast as possible without crashing however, it will pay attention to other cars, but only if they are directly in front of him.

All of the agents will try to reach their destinations without crashing.

2.3 Multi-Agent System

Our project consists of deliberative agents since the environment does not change while they make their decision, and each agent has the intention of getting to their destination as fast as possible without incurring in collisions.

2.3.1 *Agent Properties[2].*

- Deliberative - The agent will present a deliberative behavior, with beliefs and intentions.
- Coordination - The agents need to work together to reach the other side of the junction without collisions.
- Cooperation - The agents will achieve a better result if all cooperate.
- Autonomous - The agent can act independently and achieve a goal.
- Mobility - The agent can change its location in the environment, by moving forward in its path.

2.3.2 *Agent Sensors.*

- Agent Data
  - Car ID.
  - X and Y coordinates of the agent's current position.
  - Direction in which it will turn at the junction.
- Surrounding Environment
  - Cars in the coordinates surrounding the agent in a specified neighbourhood.

2.3.3 *Agent Actuators.*

- Gas - The agent will advance a square in the direction of its destination.
- Break - The agent will stay in the same square and wait.

2.3.4 *Communication Between Agents.*

- At each time step, each agent can inform the rest of the agents about its intention of moving or waiting for its turn to move. The information the agents share is their current coordinates in the grid and the coordinates of the position they intend to occupy in the next time step.

3 EMPIRICAL EVALUATION

To evaluate our system we will execute runs with different teams of agents. First, we will consider runs with only one type of agent, four teams one with each type of agent. And then runs with a mixture of types of agents.

In these combinations, we will not take into consideration the random agent, as it will only be used as a baseline for the obtained results. We will group the agents into three teams consisting of agents of two different types, where the proportion will be half and half.

Our system will be evaluated by the following metrics:

- Time steps one agent takes from start to finish.
- Number of collisions.
- Number of dead-ends.

from which we will get the agent/team of agents that has the best results in these metrics, i.e. the one that reaches the destination first, with the least collisions, and without getting into a dead end. When evaluating time steps and collisions we will ignore the runs that reach a dead-end to obtain more accurate results.

In a first general estimation, we believe that the Greedy agent will take the least amount of time steps to reach its destination and that the Decent agent will have the least amount of collisions and reach fewer dead ends.

3.1 Number of time steps in a given run

When evaluating the number of time steps needed for all agents to reach their destination we took into consideration the number and type of agents involved in a given run. To evaluate the effect the number of agents had on their behavior we first started with 1 agent and increased the amount until we had a total of 10. To evaluate the effect of the types of agents, we collected results from runs with teams consisting only of one type of agent and runs with a mixture of types of agents.

Since our environment consists of a 4-way-junction, the number of time steps spent yielding passage and thus not advancing towards the agents' destination will increase once we cross the four-agent mark, as all directions of the junction will have a car. In the case of the baseline, the random agent won't exhibit this behaviour since its actions are random, which can cause one or more agents to not advance enough that 4 cars will be at the intersection.

We also believe that the number of time steps will increase with the number of cars in the environment. As more cars are added, more will spend more time yielding passage and thus not moving forward towards their destination. We think this is true for both sets of teams, both with one type of agent and two types of agents.

This is confirmed by our results. Both in figure 3 and 4, we can see that the number of time steps presents an increase that accompanies the rise in the number of agents. And we can verify that that increase is smaller until we pass the four-agent mark.

Due to the nature of our environment and the effect it has on the results until we hit the four-car mark, the results that are useful to evaluate this metric are only the ones that involve four agents or more.

Having analysed the results in terms of the effect of the number of agents in the environment, we can now analyse them considering the effect the type of agents and the type of teams have on the behavior of the agent.

We can believe that all the agents developed with a decision process that takes into consideration the intentions, beliefs, and observations of the agents will have a significantly lower amount of time steps when compared to our baseline, the Random agent,

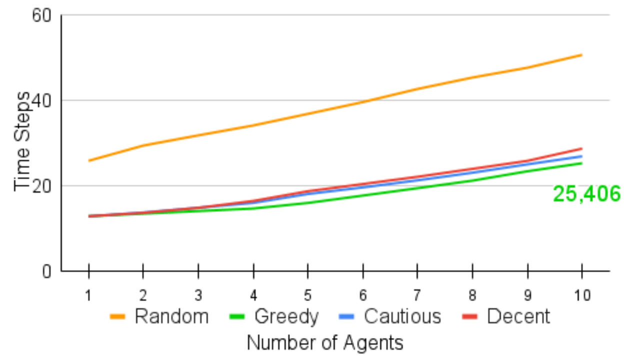

Figure 3: Number of time steps needed for the teams with one type of agent

which takes none of this information into account. Keeping this in mind we can then remove the Random agent from this analysis.

When comparing the three agents left, Greedy, Cautious and Decent, we estimate that the Decent agents will have the highest time step count of all three, since their decision process includes yielding passage whenever necessary to avoid collisions. This agent will then be the one that stops more often thus increasing its time step count. We then estimate that the Greedy agent will take the lowest number of time steps to finish a run since its only intention is to reach its destination as fast as it can. This agent will only stop if it has to, and will never yield passage to another. This will grant him the least amount of stops and thus the lowest number of time steps.

When we take into account the runs with teams that have a mixture of agents, we can then argue that the teams that include a greedy agent, also have the lowest time steps count when compared to the team that doesn't. If we then analyse which other agent in combination with Greedy will obtain the least number of times steps taken we can take the conclusion that the Decent agent will have the highest number of stops into consideration. We then conclude the combination of Greedy and Cautious has the best result for this metric in this group of teams.

These conclusions are supported by the results in figures 3 and 4, where we can see that the Greedy agent, represented in red in figure 3 and the Greedy/Cautious team, represented in blue in figure 4 have the lowest time step count in their respective teams.

When comparing the two obtained teams we can get the maximum time step count they obtained from figures 3 and 4. We can see that the Greedy agent has a lower maximum time step count, 25 406, than the Greedy/Cautious team, which has a 26 227 time step count. From that, we can conclude that the Greedy single-agent type team is the best approach to the traffic junction.

3.2 Number of collisions

To analyze the number of collisions we decided to exclude runs where dead-ends occur, since most of the time when such an event happens, collisions will arise at every time step as a result of agents being stuck in the middle of the junction.

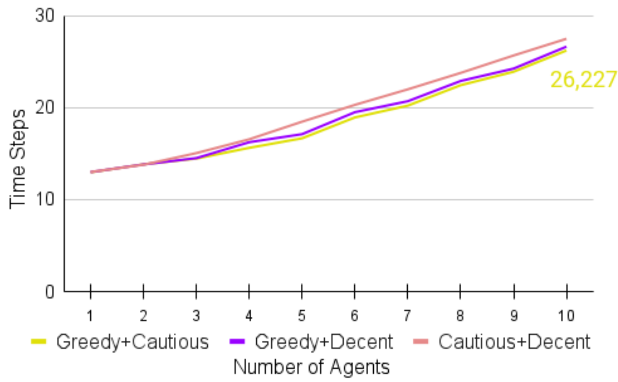

Figure 4: Number of time steps needed for the teams with more than one type of agent

This metric shows us how reckless or how safe an agent acts and is a very good indicator of what a real-world application should follow to achieve good results since, in a real-world scenario, where cars are autonomous and self-driving, the most important aspect should be the safety of its occupants and itself in this order. With this in mind, car manufacturers should look for a zero collision agent instead of one that arrives faster but with a higher risk of impact.

The results we obtained were the outcome of the average collisions per run over five hundred runs and as said previously, not counting dead-ends. What we can immediately observe in figure 5 is that all 3 agents that derive from the baseline (the random agent) improve upon it significantly with the decent agent showing a brilliant result of 0 collisions.

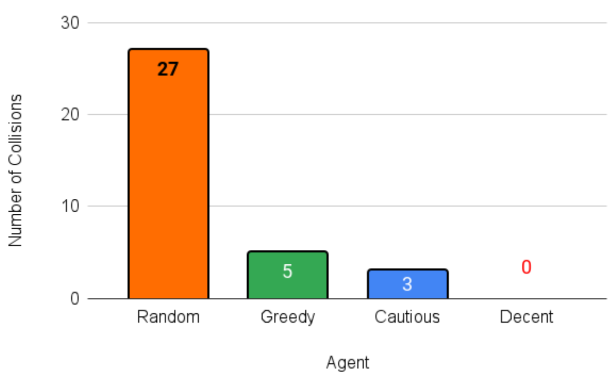

Figure 5: Average number of collisions per run for the teams with only one type of agent

A high number of collisions on the baseline was to be expected since it does not follow any specific rule that guides its behavior and therefore does not try to avoid collisions, which makes them occur very often.

For the greedy agent, collision numbers are definitely lower even though this agent also does not try to avoid them, instead of focusing on reaching its destination as fast as possible which indeed he can do from what we have observed on the number of time steps metric.

There's also a satisfying result for the cautious agent since it improves upon greedy as it tries to reach its destination as soon as possible but avoids collisions that would happen right in front of him although it does not reach the zero collision point that happens with the decent agent. This last result shows that abiding by traffic rules proves to be the best way to avoid collisions and therefore have a safer agent that could be used to circulate in a real-world environment since it's the only option where no incidents occurred.

When different agents were combined, what we hoped to achieve was an environment where different ways of acting could co-exist and use each other's strengths where for example having decent agents would make it so that collisions would not occur and greedy agents could reach their destination as fast as possible. This theory proved to be wrong and from figure 6 we can observe that having a greedy agent in the environment will always make the number of collisions be at least the same as in the environment with only greedy agents (5 collisions) while having the decent agent can improve this behavior but it's not able to stop collisions from happening and when both were combined, they achieved a total of four collisions. The environment with decent and cautious agents is also interesting and might be an indicator that with more tweaks may be different agents can co-exist since they achieved only one average collision per run.

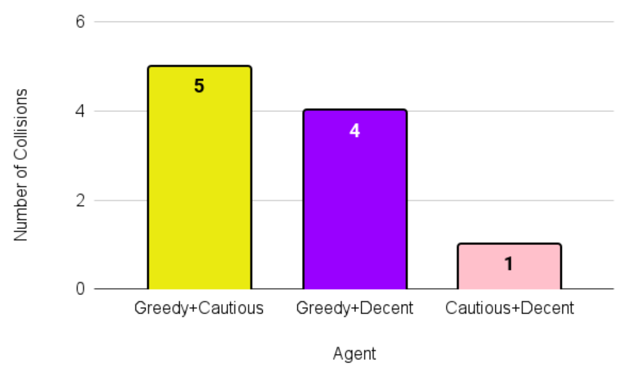

Figure 6: Average number of collisions per run for the teams with more than one type of agent

From this metric, we can conclude that the decent agent has the best results and it can improve upon the others when combined. The greedy agent on the other hand still gets a high number of collisions when combined with the others.

3.3 Number of dead-ends

Since most of the agents do not take into account the intentions of the rest of them, we expected several situations where we reach a dead end.

A dead-end is reached if the simulation takes 100 steps. If this is the case, the simulation is ended before all the cars can reach their final destination.

For most agents, a dead-end is reached if four agents enter the junction at the same time and none of them wants to take a right turn, since all of them need to take a step in front which is blocked at the time by another agent in the same situation. The only agent that does not necessarily follow this rule is the Random Agent which may not reach its final destination only because the set of actions it chose was mainly to stop.

We expected to reach more dead-ends with the Greedy and the Cautious Agents since they don't rely on the intentions of the rest of the agents in the environment and to have zero dead-ends with the Decent Agent.

The results of the experiment, regarding the teams composed of a single type of agent, can be found in figure 7. As expected, the Greedy and the Cautious Agents had several simulations that ended in dead-ends.

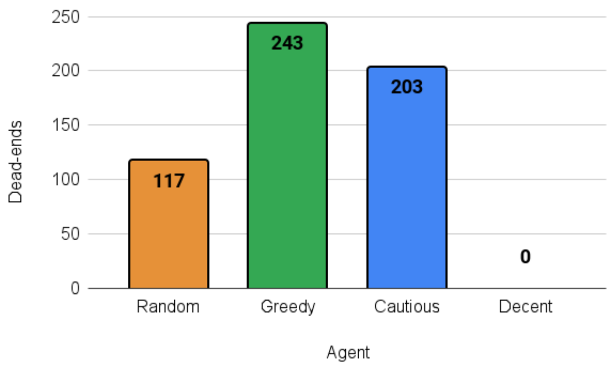

Figure 7: Number of dead ends reached for the teams with one type of agent

On the other hand, the Decent Agent didn't have a single dead-end. This happens because of his ability to communicate and use the other agents' communication to make a better decision. As mentioned before, the agent only moves if no one is in front and, in the case of the junction's entrance, if no one is in its top-left corner of the view. Although the agent can't see the squares in the rest of the entrances, he can receive through communication where everyone is, where they intend to go, and their direction. This way, if a dead-end is about to happen, the agent with the bigger ID will wait instead of moving to a situation that is impossible to solve.

When dealing with teams composed of different agents, the results can be found in figure 8 as expected, the simulations with the Greedy and Cautious Agents reach several situations of dead-ends because none of them take into account the communication of each other.

On the other hand, every team with Decent agents has better results since this agent uses communication to avoid these situations. The dead-ends found in these games can be caused by a situation as described before and the agent that is supposed to wait for the others to solve the situation is not a Decent Agent.

4 CONCLUSIONS

In the development of our project, we evaluated it according to three metrics. The first accessed the speed of an agent and the

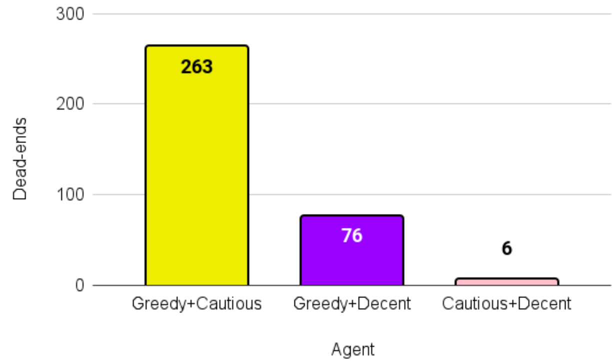

Figure 8: Number of dead ends reached for the teams with more than one type of agent

second and third the safety of the agents. From the first metric, the time steps count, we conclude that the Greedy approach is the best option when speed is the ultimate goal. The other two metrics, the number of collisions and dead ends, evaluate the safety of the driver and pedestrians. These two metrics lead us to conclude that the Decent approach is the best option for safety.

We can deduce from this that following traffic rules and abiding by road conventions is the safest way to have these types of agents circulate. When we look at a real-world application where cars can finally be self-driving, we want an approach that is considered safe or at least safer than what we currently have installed, which a Decent approach can accomplish.

In real-life, different manufacturers will have different software installed on their vehicles, and different types of vehicles will require different approaches. An example of the latter is emergency vehicles, which will have approaches more similar to cautious. When developing future work, we should take into consideration that self-driving cars should definitely integrate some way of avoiding collisions if they want to be viable for real-life use, such as incorporating the collision-avoiding behaviours that the Decent agent exhibit, as well as developing new ways for cars with this approach to be able to interact with cars that follow different procedures, since both types will occupy the same roads. If we were to improve our work, we could make experiments with an agent with the same decision process as the Decent agent but without having access to communication to see how this would affect their performance.

REFERENCES

[1] Anurag Koul. 2019. ma-gym: Collection of multi-agent environments based on OpenAI gym. https://github.com/koulanurag/ma-gym. (2019).
[2] Alberto Sardinha. 2022. https://fenix.tecnico.ulisboa.pt/disciplinas/AASMA/2021-2022/2-semestre/lectures--aulas-teoricas. University Lectures. (2022).
[3] Sainbayar Sukhbaatar, Rob Fergus, et al. 2016. Learning multiagent communication with backpropagation. *Advances in neural information processing systems* 29 (2016).