AAMAS Project Proposal - Traffic Junction

Group #024

João Simões
nº 92499

André Oliveira
nº 93686

Maria Beatriz Venceslau
nº 93734

ABSTRACT

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 which cars can take one of two approaches: to follow the conventional rules or to be self-interested.

In this project, we will assume the problem to consist of cars with predetermined paths as autonomous agents and evaluate their behaviour according to some evaluation metrics.

KEYWORDS

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

1 INTRODUCTION

1.1 Problem Definition

Our project aims to simulate a traffic junction, where cars have to coordinate and cooperate to follow their designated path while avoiding collisions.

Agents will spawn at the beginning of any of the lanes and will be assigned a predefined route to navigate the junction and reach the edge of the grid. Each agent will follow a different approach to manage the traffic jam.

1.2 Problem Relevance

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 to the right.

In our project, we will simulate the traffic at a junction, where we can increase or reduce the amount of traffic to find which approach works best in each scenario.

1.3 Motivation

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.

1.4 Objectives of the Project

The objective of this project is to simulate a traffic junction in which we can analyse what happens when there is an increase in traffic versus a reduction in traffic. In this system, we can also compare the behaviour of a greedy agent versus one that respects the convention laws of yielding passage to the cars on their right.

The goal of this project would be to identify the best way to manage traffic, analysing the number of collisions and time taken as an evaluation metric.

1.5 Related Work

This type of problem has been approached by Anurag Koul[1] by using a collection of multi-agent environments based on OpenAI gym. In this Github repository, it's possible to observe how similar problems can be solved with the same multi-agent development when inserted in different environments, one of them being a traffic junction from which our work is inspired.

Other tests like partial visibility and other models on a similar environment were made by Sukhbaatar, Sainbayar, and Rob Fergus[3]. The metrics used in their work will also be taken into consideration.

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 directions.

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

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).

AASMA Project, IST, 2022, Lisbon, Portugal

https://doi.org/

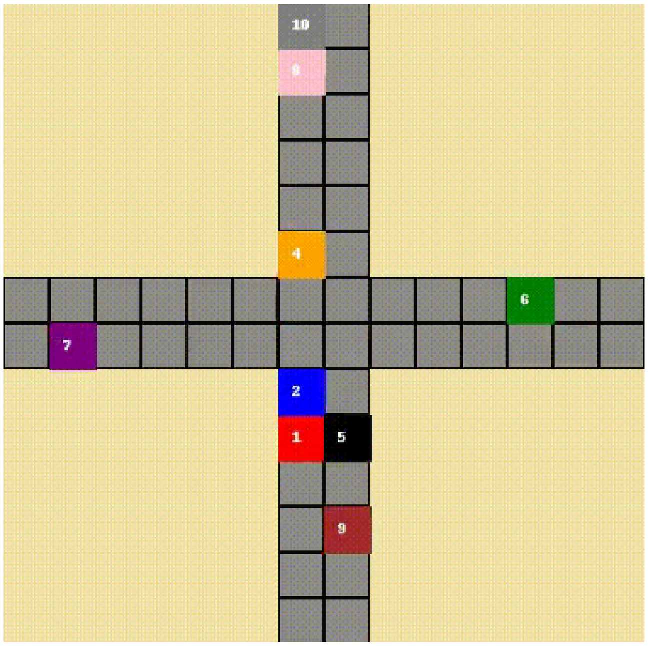

Figure 1: Environment Example

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.

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

The simulation is terminated after a number maximum 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 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. It is self-interested (intends to finish as fast as possible without crashing). This agent will only pay attention to other cars 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

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.

- Each agent can inform the rest of the agents about its intention of moving or waiting for its turn to move.

3 EMPIRICAL EVALUATION

Our system can be evaluated by the following metrics:

- Number of collisions in a given run.
- Time steps one agent takes from start to finish.

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).