Road Traffic Management: Control of Intersections

Chaymae Chouiekh[1], Ali Yahyaouy[1], My Abdelouahed Sabri[1], Abdellah Aarab[1] and Badraddine Aghoutane[2]
[1]*University of Sidi Mohamed BenAbdellah Faculty of Sciences Dhar El Mahraz, LISAC Laboratory, Fez,Morocco*
[2]*University Moulay Ismail, Computer Science and Applications Laboratory, Meknes,Morocco*

Keywords: Road traffic simulation, Reinforcement Learning, SUMO, Q-learning, Control of intersections.

Abstract: The road traffic problem is a social and economic enigma that contributes to a permanent increase in mortality rate so that traffic management in main road networks has become a significant challenge in many systems. To solve this problem, there are many solutions to manage intersections as they are leading causes of congestion generation, especially in the parts of space shared by vehicles. The most straightforward and most well-known solutions are the traffic lights and the "STOP" signs, which generally promote one flow over another. Such events cause delays for vehicles because they force them to stop frequently and for varying periods. If vehicles flows are significant, these local delays can lead to congestion. For this reason, this research is interested in studying the problem of vehicle intersections where we present a reinforcement learning model that is the development and implementation of one of the reinforcement learning algorithms (Q-learning) to simulate road traffic using the SUMO road traffic simulator.

1 INTRODUCTION

The development and evaluation of reinforcement learning techniques in real-world problems are far from trivial. One such task is to simulate the dynamics of an environment and the behavioral interactions of agents. Nevertheless, Reinforcement learning has been applied successfully and has given promising results in several areas (Future, 2020), such as traffic, networks. Intelligence robotics, games (Hassabis, 2016), among others. A particularly relevant field is that of trafficking. It is well known that traffic problems are encountered every day, even in small towns. As such, trafficking has attracted particular attention from the artificial intelligence community.

In particular, given its distributed and autonomous nature, the traffic has shown an exciting testbed for reinforcement learning algorithms.

However, an important aspect to consider here concerns how these scenarios are validated. In general, the deployment of new technologies in traffic areas is only possible after extensive experimentation. Thus, traffic simulation appears to be a safe and economically efficient way to validate such scenarios. Here, it is necessary to model the driver's behavior and simulate vehicles on the road network.

A representative tool here is the SUMO simulator, which models the system at a microscopic level, i.e., even the position and speed of vehicles are simulated. On the one hand, the most straightforward approach is to use existing libraries that facilitate this process by providing algorithms and validation scenarios (Liza Lunardi, Gabriel De Oliveira, & L. C. Bazzan, 2017). However, no such library model presents the required level of detail.

Moreover, extending these libraries to work with natural traffic environments (for example, using the SUMO simulator) is not a simple process.

In this context, we have taken the initiative to develop a road traffic management approach by analyzing intersections that aim to facilitate the development and validation of reinforcement learning techniques. Specifically, it includes implementations of well-known reinforcement learning algorithms (Q-learning). It also contains traffic environments(Simulated using SUMO), allowing validation of reinforcement learning algorithms in very detailed traffic simulations.

Chouiekh, C., Yahyaouy, A., Sabri, M., Aarab, A. and Aghoutane, B.
Road Traffic Management: Control of Intersections..
DOI: 10.5220/0010731800003101
In *Proceedings of the 2nd International Conference on Big Data, Modelling and Machine Learning (BML 2021)*, pages 239-243
ISBN: 978-989-758-559-3

2 ASSOCIATED APPROACHES

In this section, we rely on the book (S. Mammar, 2007) where we extract some of the most widely used systemic approaches to solving traffic problems that we are primarily interested in studying the intersections and discussing some of the disadvantages of these approaches:

TRANSYT (Traffic Network study Tool (G.E.Robinson, 1992) Is one of the first proposed systems that were based on off-line optimization that generates optimal coordination plans between signal lights of a network for a given time. TRANSYT requires many input parameters, such as the geometry of the intersections arteries, the flow of vehicles, the rate of vehicles on each exit track of each intersection, the minimum green light time, initial light plans, and initial values for cycle times and phase shifts.

TRANSYT is a centralized system and is not traffic-responsive (A.A. Guebert et G. Sparks, 1990).

The SCOOT (Split cycle and offset optimization Technique) (P. B. Hunt, D. I. Robertson, R. D. Bretherton et R. I. Winton, 1981) is an urban traffic control system that is operational in more than 30 cities in the United Kingdome and abroad, it is a fully adaptive system that collects data from vehicle detectors and then calculates parameters that reduce delays and stops. It is a decentralized system and wholly adapted to the traffic situation.

Nevertheless, SCOOT remains a weak adaptive system compared to others, given its slight gradual variations phases in each cycle.

3 AGENT ARCHITECTURE

Also called soft bot ('robot software') (J.Ferber, 1995) is a computer science software that performs various actions on a continuous and autonomous basis on behalf of an individual or an organization. An intelligent agent is most often classified according to the role he plays. The agent-based architecture corresponds to intelligent systems control and software agent layouts, representing the connections between the components as shown in Figure 1:

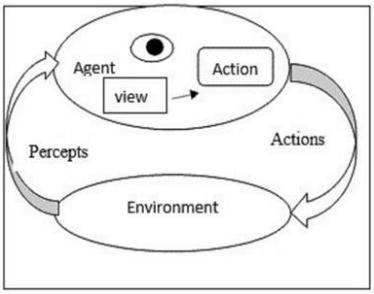

Figure 1: Agent architecture.

4 TRAFFIC SIMULATION

As we did not find a suitable data source for our target, we decided to simulate the data from a road network using the SUMO road simulator, where I performed simulations at intersections of the crossing type as shown in Figure 2:

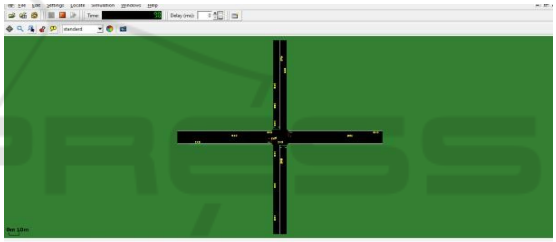

Figure 2: Simulation of vehicle intersections at crossroads.

5 LEARNING BY REINFORCEMENT

Learning by reinforcement is a learning method for Machine Learning models (Issam, 2012) (S.Sutton & Barto, 2020). In other words, it let the algorithm learn from its own mistakes. To learn how to make the right decisions, artificial intelligence is directly confronted with choices. If they are wrong, they will be penalized. On the opposite, if they make the right decision, they will be rewarded. To obtain even more rewards, Artificial intelligence will do its best to optimize its decision-making (M.Baland, D.Loenzien, P.Haond, 2006) as we show in Figure 3:

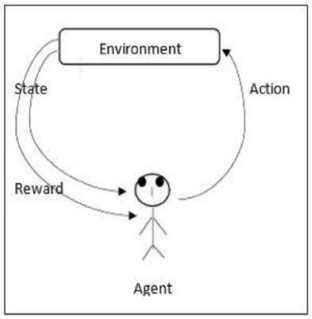

Figure 3: Learning by reinforcement scenario.

6 REINFORCEMENT LEARNING ALGORITHMS

Exploration: Exploration, where we gather more information that could lead us to better decisions in the future.
Exploitation: Exploitation, where we make the best decision based on current information.
Epsilon greedy algorithm: Epsilon Greedy (P.Review, 2017) is a simple method to balance exploration and exploitation by choosing between exploration and random exploitation. It is pretty simple, epsilon was used to evaluate the exploration rate, it is a percentage, and greedy says it means taking the optimal actions that we think optimal at the time t, and so epsilon greedy is the Tradeoff between exploration and exploitation (Baeldung, 2021).
Q-function: Q-value function is an exact method used to solve the learning by reinforcement problem. The purpose of the task is to find the expected usefulness from the states by acting a and (subsequently) by acting optimally. Let's say indeed what is the number of rewards that I can hope to have from that state one looks towards the future, In the other words, it defines the value of taking action A in state S under a policy, denoted by (S, A), as the expected return R starting from Staking action a, and after that following policy (IA, 2020). One of the critical properties of Q is that it must satisfy the Bellman optimality equation, according to which the optimal Q value for a state given action is equal to the maximum reward that the agent can get from an action in the current state and the maximum discount reward that he can get from any possible peer state-action that follows. The equation looks like this:

$$Q\,\pi\,(s, a) = E\,\pi\,[Rt\,|s_t = s, A_t = a] = E\pi\,[\sum_{j=0}^{T} \gamma^j\, r_{t+j+1}|\, s_t = s, A_t = a] \quad (1)$$

Bellman Equation (Zhu & Jia, 2020): The Bellman equation shows up everywhere in the Reinforcement Learning literature, being one of the central elements of many Reinforcement Learning algorithms. It allows a redefinition of the Q value function recursively. So, it gives us a way to determine the optimal policy:

$$Q\,\pi\,(s, a) = \sum_{s'} P_{ss'}^{a}(r(s, a) + \gamma.\sum_{a'}\pi(a'|s').\,Q\,\pi(s', a')) \quad (2)$$

So, we are looking to find the optimal state-action value function that indicates the maximum reward we are going to get if we are in state s and taking action A from there on-wards:

$$Q^*\,(s, a) = \max \pi\, Q\,\pi\,(s, a) \quad (3)$$

Bellman proved that the optimal state-action value function in state s and taking action a is

$$Q^*\,(s, a) = \sum_{s'} P_{ss'}^{a}(r(s, a) + \gamma.\max_{a'} Q_*\,(s', a')) \quad (4)$$

Q-learning algorithm: Q-Learning (K.S.Hwang, S.W. Tan, C.C.Chen, 2004) is a basic form of strengthening learning that uses Q values (also known as action values) to improve the behavior of the learning agent. This is the process of iterative updating of Q values for each state-action pair using the Bellman equation. Q-learning is arguably one of the most applied representative reinforcement learning approaches and one of the off-policy strategies. Since the emergence of Q-learning, many studies have described its uses in reinforcement learning and artificial intelligence problems (Beakche, Myeonghwi, Gaspard, & Jong Wook, 2019).

Q table: This table will allow us to know the expectation of reward for each case for each action. In the initial case, the number of rewards is zero. This q-table becomes a reference table for our agent to select the best action based on the q-value (Violante, 2019).

7 PROPOSED SYSTEM

To improve network traffic, we can play on the layout of the signaling where we control the state of the signaling light using the simulator of road traffic SUMO which we offer a sumo map that we can transform into a dataset that has visualized as we see in Table 1:

Table 1: Extraction result of the data concerning the connections between the nodes.

Node1	Node2
-gneE0	gneE3
-gneE0	gneE2
-gneE0	gneE1
-gneE1	gneE3
-gneE1	gneE0
-gneE1	gneE3
-gneE2	gneE2
-gneE2	gneE0
-gneE2	gneE3
-gneE3	gneE2

Q-learning Optimization:
Trained Model:
To create our training algorithm, first of all, when our agent explores the environment during ten episodes we initialize our matrix Q-table which has N columns and M rows such as N is the number of actions and M is the number of states, then we randomly select our action (left, right, low, high....) or we exploit the Q-values that are already calculated by using the epsilon value and compare it to the random-uniform function (0,1), which returns an arbitrary number between 0 and 1 and to get the next state and reward we perform the chosen action in the environment. And finally, we calculate the maximum Q-value of the actions corresponding to next-state, to be easily able to update our Q-value with the new q-value. This process is repeated over and over again until learning is stopped. In this way, Q-table is updated.
Model Evaluation: Let us evaluate the performance of our model. We do not need to explore actions further, so now the following action is always selected using the best Q-value.
It can be seen from the assessment that we show Figure 4, that the performance of the officer has improved considerably and the fact that he does not meet any penalty does not mean that he has chosen the right action.
With Q-learning during exploration, the agent makes mistakes at the beginning, but once he has sufficiently explored (given most states), he can act judiciously by maximizing rewards by performing intelligent movements.

```
Starting SUMO
Results after 10 episodes:
Average timesteps per episode: 716.0
Average penalties per episode: 0.0
```

Figure 4: Evaluation result of the model.

8 CONCLUSIONS

We developed a road traffic management model by analysing intersections in a road traffic simulation environment (SUMO). We had the opportunity to put into practice and implement the Q- learning model, which is based on reinforcement learning to manage intersections. This method is effective in overestimating action values under certain conditions to optimize intersections. However, it was not previously known whether, in practice, such overestimation is shared, whether it adversely affects performance and whether it can generally be avoided, which is why there are many ways to improve it.

REFERENCES

A.A. Guebert et G. Sparks. (1990). Timing plan sensitivity to changes in platoon dispersion. Santa Barbara: California.

Baeldung. (January 15, 2021). Epsilon-Greedy Q-learning. Beach, o., Myeonghwi, K., Gaspard, H., & Jong Wook , K. (September 2019). Q-Learning Algorithms: A Comprehensive Classification and Applications. IEEE Access PP(99).

Crites, R., & Barto, A. (1996). Improving Elevator Performance Using Reinforcement Learning. advances in Neural Information Processing Systems 8.

Future, H. (2020, février 3). Apprentissage par renforcement : une IA puissante dans toujours plus de domaines.

G.E.Robinson. (1992). Regulation of division of labor in insect societies. Annual review of entomology, 37(1):637-665.

Hassabis, D. S. (Wednesday, January 27, 2016). AlphaGo: Mastering the ancient game of Go with Machine Learning.

IA, J. T. (Jun 11, 2020). The Bellman Equation. V-function and Q-function Explained.

Issam, E. A. (05/04/2012). Apprentissage par renforcement – de la théorie à la pratique.

J.Ferber. (1995). Les systèmes multi-agents, vers une intelligence collective. InterEditions.

K.S.Hwang, S.W. Tan,C.C.Chen. (2004). Cooperative strategy based on adaptive Q-learning for robot soccer systems.

Liza Lunardi, L., Gabriel De Oliveira , R., & L. C. Bazzan, A. (May 2017). Developing a Python Reinforcement Learning Library for Traffic Simulation. Proceedings of

the 17th Adaptive Learning Agents Workshop. São Paulo.
M.Baland, D.Loenzien, P.Haond. (2006). Intelligence Artificielle. Pearson Education.
P. B. Hunt, D. I. Robertson, R. D. Bretherton et R. I. Winton. (1981). SCOOT a responsive traffic method of coordinating signals. TRRL: Rapport technique.
P.Review. (2017). The Authors. Elsevier B.V.
Ramos, G. D. (May 2017). Developing a Python Reinforcement Learning Library for Traffic Simulation. Proceedings of the 17th Adaptive Learning Agents Workshop. São Paulo.
S.MAMMAR. (2007). Systèmes de Transport Intelligents, modélisation, information et contrôle. Lavoisier.
S.Sutton, R., & Barto, A. G. (6 février 2020).
Reinforcement Learning. The MIT Press.
Violante, A. (Mars 13 2019). Simple Reinforcement Learning: Q-learning.
Zhu, Y., & Jia, G. (2020). Dynamic Programming and Hamilton–Jacobi–Bellman Equations on Time Scales.