Intelligent traffic light control using deep reinforcement learning

Khady Gaye, Mame Diarra Diouf, Honorine Gnonfin,
Mame Diarra Diop, Salomon Muhirwa

September 29, 2023

1 Introduction

Transport is an essential part of our lives. With so many people travelling by car, motorcycle or on foot in the same time interval, traffic jams are a major problem. At crossroads, collisions multiply, passengers lose more time than expected and all this delays the development of our communities. The purpose of traffic lights is to regulate traffic flow at intersections. These lights are programmed in a deterministic way, with a predefined duration associated with each color. Sometimes, this model is not optimal because the flow of traffic on each lane varies over time. In these cases, we need to make the traffic light system intelligent, so that it can make decisions based on the state of the intersection. This is the background to our topic on Intelligent Traffic Light Control Using reinforcement learning. We use Deep Q-Network (DQN) and Double Deep Q-Network (DDQN) to implement an intelligent agent. A lot of work has been done in this area. For example in [6], the authors propose a more effective deep reinforcement learning model for traffic light control and test our method on a large-scale real traffic data set obtained from surveillance cameras [5]. In this project, we consider the simplest configuration of one way and unidirectional traffic and we used deep Q-learning to train the intelligent agent. So in section 2, we present some works related to the topic, in section 3, we explain our representation of traffic light environment, the agent and its interaction with the congestion.

2 Related Works

Traffic light control is a crucial aspect of urban traffic management, and there have been numerous research works and methods developed in this field. In this section we present some conventional methods and Deep Reinforcement Learning (DRL) methods. Traffic light control is in fact a mixed-integer linear programming problem that is NP-hard [4]. Conventional methods mainly include classic optimization based methods, evolutionary algorithms, and fuzzy logic. Classic optimization based methods usually adjust the cycle time and the phase splits based on parameters and heuristic rules. The time for green light is usually longer than needed, which causes a large amount of wasted time for waiting cars or pedestrians. [2] Classic optimization based methods can be applied to single and multiple road intersections, providing effective tools for traffic engineers and planners to manage urban traffic and improve transportation efficiency. The choice of method depends on factors such as the complexity of the road network, available data, and computational resources. These methods have poor scalability. If the road network is large, the number of parameters is also large, leading to much time spent on parameter calculating and tuning. Moreover, heuristic rules and some parameters are highly based on expert knowledge without optimality guarantee. [1] Evolutionary algorithms especially genetic algorithms are usually used in multi-objective optimization: fuel consumption, number of stops, traffic flow throughput, and vehicles' waiting time. In theory, these algorithms are based on the concept of survival of the fittest through stochastic optimization and heuristics. Fuzzy logic is a computational approach used in traffic light control problems, that allows for reasoning in situations of uncertainty and imprecision. It has been applied effectively in traffic light control systems to improve traffic flow and responsiveness to changing conditions. In the recent years Deep Reinforcement Learning (DRL) has gained significant attention as a powerful approach for optimizing traffic light control in urban environments. DRL leverages neural networks and reinforcement learning algorithms to adaptively learn and optimize traffic signal timings based on real-time traffic conditions. The problem is well suited to be approached in this way as there can be a finite action space and the task of minimizing delay or some similar metric lends itself well to the reinforcement learning framework. Work in the traffic agent RL space typically takes the latter approach by incorporating the metric into a negative reward function such as proportional to the amount of delay a car experiences in two consecutive states. Some work has also been done around non-deterministic duration signals which expands the action space somewhat [3].

3 Methodology

We focus on a single intersection road and we present in this section our modeling. The environment we consider is made of two roads crossing at one intersection. To make it simple, we suppose unidirectional traffic flows as we can see on figure 1.

3.1 Environment

The state of the environment is made of the number of vehicles on the road North-South, the number of vehicles on the road West-East and the configuration of the light. The light configuration is either red or green. We represent the observation of the environment by (X_1, X_2, L). The action the control agent

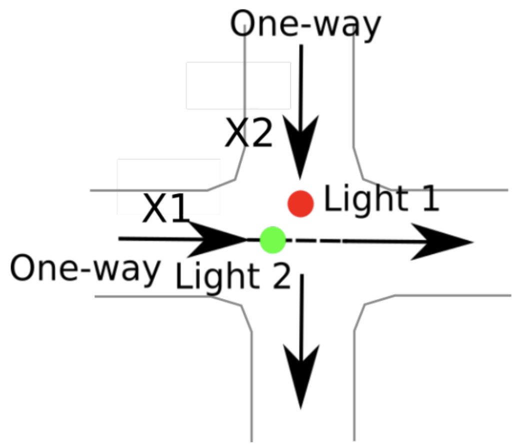

Figure 1: An intersection with two traffic flows

can do is either changing the state of the light or keeping the current color. To evaluate the agent intelligence, we give it a reward after each action. In our case, we look at the number of vehicles waiting on the roads to pass the intersection. Then the reward is given by $R = -X_1^2 - X_2^2$.

3.2 Deep Q-Network

In this section, we will explore the possibility of using Deep Q-Networks (DQN) to optimize traffic light control policies in real time in new intelligent transportation systems by an MDP approach in a single intersection scenario. Our optimization goal is to minimize the congestion cost in time slot t may be expressed as a function $Z(X(t))$ of the queue state, with $X(t) = (X1(t), X2(t))$ and $Z(X(t)) = X_1^2(t) + X_2^2(t))$. The goal is to find a dynamic control policy which selects actions over time slots 0,1,...,T so as to minimize the long-term expected discounted cost $\mathbf{E}[\sum_{t=0}^{T} \gamma Z(X(t))]$, with $\gamma \in (0, 1]$ representing a discount factor. We define $Q(s, a)$ to be the maximum achievable expected discounted reward (or minimum discounted congestion cost in our context) under the optimal policy starting from states $= (X; L)$ when action a is taken. $Q(s, a)$ satisfies the equation:

$$Q(s, a) = r(s, a) + \gamma \sum_{s \in S} p(s, s; a) \max_{a' \in A} Q(s, a) = r(s, a) + \gamma \mathbf{E}[\max_{a \in A} Q(s, a)] \quad (1)$$

with $r(s, a) = Z(X)$ denoting the reward (i.e., negative congestion cost) in queue state X, and $p(s, s; a)$ denoting the transition probability from states to stateswhen action a is taken. The neural network of DQN is used as a function approximator to estimate the action-value function. The neural network is trained by adjusting its parameters in each iteration to reduce the mean-squared

error in the Bellman equation. The DQN algorithm uses an experience replay buffer to update network parameters. It has two advantages:
1) enabling the stochastic gradient decent algorithm and
2) removing the correlations between consecutive transitions.

3.3 Double deep Q-Network

The goal of Double Deep Q network is to reduce overestimation by decomposing the max operation in the target into action selection and action evaluation. Compared with the deep Q Network architecture used to estimate the Q value of actions for a given DQN state, which tends to significantly overestimate action values.decomposing the max operation in the target into action selection and action evaluation. We evaluate the greedy policy according to the online network, but we use the target network to estimate its value. The update is the same as for DQN, but replacing the target with:

$Y = R_{t+1} + \gamma Q(S_{t+1}, argmax(S_t, aa, \theta_t, \Theta'_t)$

Compared to the original formulation of Double Q-Learning, in Double DQN the weights of the second network are replaced with the weights of the target network for the evaluation of the current greedy policy. `https://arxiv.org/pdf/1509.06461.pdf`

4 Results and discussion

To implement our deep Q-Network, we use two hidden layers for the neural network and ReLu as activation function. With 10000 episodes, we got the figure 2 for the training. When we evaluate the model on 1000 episodes, we get

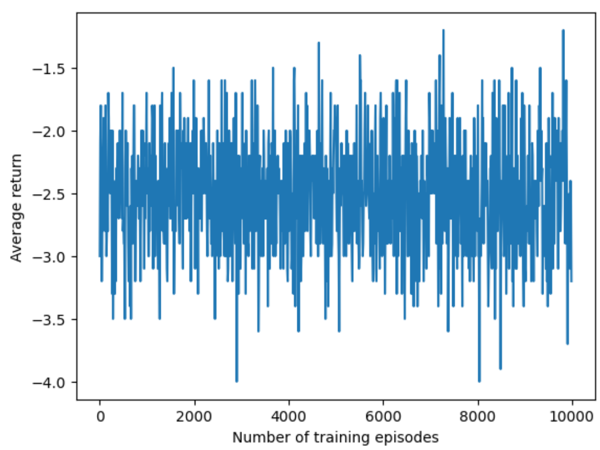

Figure 2: An intersection with two traffic flows

the figure 3 describing the behavior of the reward with respect to the number of episodes.

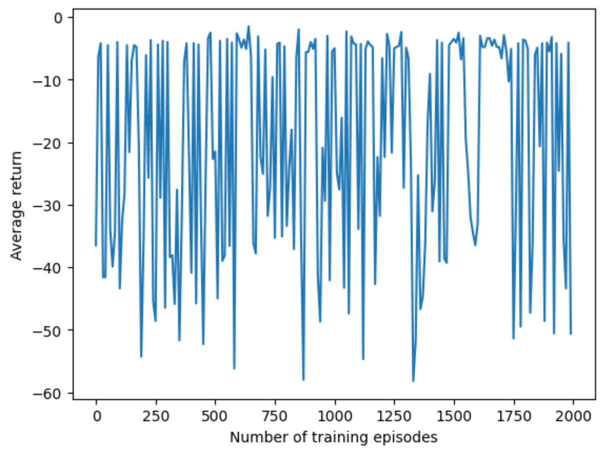

Figure 3: An intersection with two traffic flows

About the double deep Q-network, we've got the figure 4 for the reward evolution.

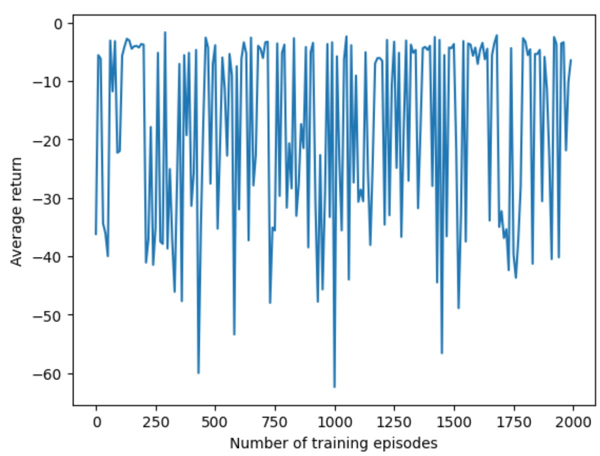

Figure 4: Reward of ddqn agent with respect to the number of episodes

We can see that the model doesn't learn a lot. That may be due to the simplicity of the environment. In fact, we assume that only one vehicle pass the intersection when the light is green for one way and also the vehicles are coming one by one. Or may be, we should have used another definition of the reward. All those are explorations paths that we couldn't cover because of the time.

Conclusion

We have explored the potential for deep Q-network (DQN) algorithms to optimize real-time traffic light control policies. Assuming a simple one-lane intersection with two phases: green and red. The goal is to minimize the number of waiting vehicle at the intersection. The biggest worry of this project was convergence of reward oscillating that oscillates and difficulty in achieving the desired performance.

References

[1] Micha Hofri and Keith W Ross. On the optimal control of two queues with server setup times and its analysis. *SIAM Journal on computing*, 16(2):399–420, 1987.

[2] Peter Koonce and Lee Rodegerdts. Traffic signal timing manual. Technical report, United States. Federal Highway Administration, 2008.

[3] Xiaoyuan Liang, Xunsheng Du, Guiling Wang, and Zhu Han. A deep reinforcement learning network for traffic light cycle control. *IEEE Transactions on Vehicular Technology*, 68(2):1243–1253, 2019.

[4] Wei-Hua Lin and Chenghong Wang. An enhanced 0-1 mixed-integer lp formulation for traffic signal control. *IEEE Transactions on Intelligent transportation systems*, 5(4):238–245, 2004.

[5] Xiao-Yang Liu, Ming Zhu, Sem Borst, and Anwar Walid. Deep reinforcement learning for traffic light control in intelligent transportation systems. *arXiv preprint arXiv:2302.03669*, 2023.

[6] Hua Wei, Guanjie Zheng, Huaxiu Yao, and Zhenhui Li. Intellilight: A reinforcement learning approach for intelligent traffic light control. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2496–2505, 2018.