Bol. Soc. Paran. Mat.
©SPM – E-ISSN-2175-1188
SPM: www.spm.uem.br/bspm

(3s.) **v. 2026 (44) 3** : 1–18.
ISSN-0037-8712
doi:10.5269/bspm.79801

Mathematical Model of Reinforcement Learning for Dynamic Traffic Light Control

Hala Khankhour*, Najat Rafalia, and Jaafar Abouchabaka

ABSTRACT: Nowadays, transport has become an essential element for the modern societies. So, the management of networks has also become important. Among the most used tools for the management of these networks, we find traffic lights. These lights do not adapt to the amount of traffic (fixed time for each traffic light). The evolution of new technologies has made it possible to solve this problem and to make traffic lights smart. The objective of this work is to propose a new dynamic control solution for intelligent traffic lights using agent concept and reinforcement learning combined with deep learning. The main advantage of our system is to provide adaptation between traffic lights and smooth traffic flow in different conditions.

Key Words: Smart traffic lights, traffic control, agent system, reinforcement learning, deep learning.

Contents

1. Introduction

The population growth and the increasing number of road users are a significant some of various problems. For example, a driver who frequently encounters traffic jams not only wastes time but also consistently experiences stress, which can directly lead to accidents and indirectly contribute to health issues. Additionally, the lives of road users are at risk due to gas emissions and pollutant emissions caused by traffic congestion. Therefore, the primary issue is congestion, which has become a major concern for transportation experts. In this context, artificial intelligence plays a significant role; we are focusing here on two aspects:

- (i) Regulating traffic patterns: This refers to the management and regulation of the way in which road traffic is organized, controlled, and directed.

* Corresponding author.
2020 *Mathematics Subject Classification:* 90B20, 68T05.
Submitted October 29, 2025. Published February 22, 2026

- (ii) Situation analysis using agent-based simulations: this involves employing computer simulations that replicate the behavior of drivers and other relevant entities, such as vehicles and pedestrians, within a simulated environment. Our paper contributes to this field of research, with a specific focus on an unexplored scenario, a four-lane intersection governed by traffic signals. We aim to implement an autonomous system responsible for managing this intersection by perceiving real-time traffic conditions and leveraging experience gained from simulated scenarios that emulate plausible traffic conditions. The simulations themselves are agent-based, conducted through the Simulation of Urban Mobility (SUMO) tool [1], providing a synthetic yet realistic environment for exploring potential regulatory actions' outcomes. An essential aspect is SUMO's provision of an Application Programming Interface (API) for interaction with external programs. This feature has facilitated the definition of a credible set of observable environmental aspects, enabling control over traffic lights based on the learning agent's decisions. Additionally, it allows us to leverage statistics gathered by SUMO to analyze overall traffic flow, consequently defining the rewards for the actions executed by the traffic light control agent.

The paper is divided as follows, initially, we offer a concise overview of the pertinent segment within the current state of the art regarding traffic signal management employing reinforcement-learning methodologies. Subsequently, we present the related work that has been selected for the purpose of this investigation. The implementation of the method we have defined and adopted will be presented in Section 4, followed by a description and discussion of the results obtained. Conclusions and future developments will close the paper.

2. State of the art

Every year, the number of road users increases and exceeds the capacity of certain roads, resulting in traffic jams and long queues. This phenomenon of congestion has led to the development of new traffic control techniques, including intelligent traffic lights. This section provides an overview of the work being done on intelligent traffic signals. In the field of dynamic programming, Wu and his colleagues introduced the "Dynamic programming" method in their work published in 2009 [2]. This approach offers the advantage of providing a globally optimal solution. However, it is worth noting that this method comes with certain drawbacks, particularly its high cost in terms of memory space utilization, making it a relatively expensive option. In the field of network optimization, the "Branch and Bound" method, as presented by W. Glankwamdee in 2008 [3], offers distinct advantages. One of its primary benefits lies in its ability to effectively reduce time complexity, making it an attractive choice for solving complex network problems. However, it's essential to note that the "Branch and Bound" approach is not without limitations. This method is best suited for small networks, and its effectiveness may diminish when applied to larger, more extensive network scenarios. Therefore, while it offers significant advantages in terms of time efficiency, its applicability is somewhat constrained by the size and complexity of the network under consideration. In the field of neural networks, Bingham and his colleagues introduced their "Neural network" method in 2001 [**?**]. This approach offers the benefit of flexibility and adaptability, making it suitable for various applications. However, a notable challenge associated with this method is the difficulty m accurately determining the parameters of the network used, which can affect its effectiveness in practice. In 2020, Y. Aibeche introduced the "Genetic algorithm" method [4], which has been notable for its benefits in terms of convergence to the optimal solution. However, it is important to note that one of the issues associated with this method is the extended calculation time required, making it a consideration in its application within various contexts in the field. In the field of reinforcement learning, Gender's 2018 [**?**] method has garnered significant attention. Dynamic environments, making it a strong contender in the domain, know this approach for its notable adaptability. However, one of the primary drawbacks associated with this method is the substantial time and computing resources, it demands for the training process. These computational demands can pose practical challenges for its widespread application and deployment. Among these methods, we are interested it reinforcement learning because of its dynamics. Q learning is one of the fundamental reinforcement learning algorithms. It stores all the possible state action Q values for an agent's optimal policy in an array. On the other hand, using reinforcement-learning algorithms, the state space is generally so large that it is impossible to discover

and save every state-action pair (Q-values). Consequently, we propose in this work to approach these values with a supervised learning model, in our case Deep Learning.

3. Related works

3.1. Reinforcement Leaning

One of the overarching objectives within the field of Artificial Intelligence (Al) is the development of machines capable of emulating the intelligent behaviors exhibited by human beings. Attaining this objective necessitates the ability of an Al system to engage with its surroundings and acquire the capacity to make informed decisions within that context reinforcement learning [7] stands out as an established domain in Al, demonstrating proficiency in autonomous learning through experiential knowledge. Successful applications of reinforcement learning span diverse domains, including gaming [8], robotics [9], and the optimization of traffic light control. In the framework of a reinforcement learning (RL) problem, an autonomous agent actively observes the environment, discerning a state (s_t) that signifies the environment's condition at time t. Subsequently, the agent selects an action (a_t) that triggers a transition in the environment, leading to a new state (s_{t+1}). Following this environmental transition, the agent receives a reward (r_{t+1}), indicative of its performance against a predefined measure. The agent's primary goal is to acquire a policy (π^*) aimed at maximizing the cumulative reward expectancy derived from its actions while adhering to π^*. Illustrated in Figure 1, the conventional reinforcement learning cycle encapsulates this iterative process.

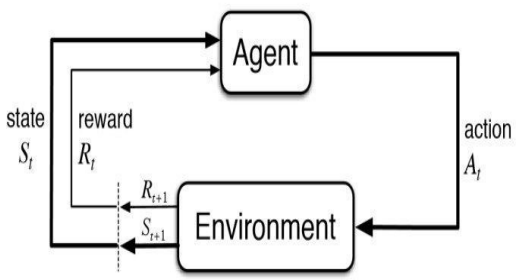

Figure 1: The reinforcement learning cycle.

Well known in the field of reinforcement learning, Q-learning is one of the fundamental algorithms. It allows all Q values for each state-action to be stored, which helps define an optimal strategy for the agent. This table is updated using a specific equation (1).

$$\mathbf{Q}(\mathbf{S},\mathbf{a}) = \mathbf{Q}(\mathbf{s},\mathbf{a}) - \alpha \cdot (\mathbf{Q}(\mathbf{s},\mathbf{a}) - \mathbf{r} - \gamma \cdot \mathbf{Q}'(\mathbf{s}',\mathbf{a}')) \tag{3.1}$$

To accelerate convergence, we replaced this equation with equation (2):

$$\mathbf{Q}(\mathbf{S}_t,\mathbf{a}_t) = \mathbf{r}_{t+1} + \gamma \cdot \max_{\mathbf{A}} \mathbf{Q}'(\mathbf{S}_{t+1},\mathbf{a}_{t+1}) \tag{3.2}$$

Where:

- $\mathbf{r}_{t+1}$ Represents the reward obtained after performing action a_t in state s_t.
- $\mathbf{Q}'(\mathbf{S}_{t+1},\mathbf{a}_{t+1})$ Corresponds to the Q value associated with action $\mathbf{a}_{t+1}$ in state $\mathbf{S}_{t+1}$, i.e., the state following the execution of action a_t in state s_t.
- γ is a discount factor that allows less importance to be given to future rewards compared to immediate rewards.

3.2. Learning in Traffic Signal Control

Traffic light control is a well-suited application context for reinforcement learning (RL) techniques: In this context, the objective is for one or more autonomous agents to optimize the efficiency of vehicular traffic traversing one or multiple intersections governed by traffic lights. The application of reinforcement learning for the control of traffic lights is justified by various factors [10]:

- When appropriately educated, agents utilizing reinforcement learning demonstrate the capability to adjust to diverse scenarios, such as road accidents or unfavorable weather conditions.
- Reinforcement learning agents possess the capability to autonomously acquire knowledge without the need for supervision or pre-existing familiarity with the environment.
- The agent requires a streamlined environmental model, primarily focused on state representation. The agent learns through the system's performance metric, namely, the reward.

Reinforcement learning methods employed in managing traffic lights tack.le the subsequent hurdles [10]:

- Improper sequencing of traffic light signals: Traffic signals generally select phases according to a static, predefined policy. This method could lead to the triggering of an incorrect traffic signal phase in a scenario that could lead to heightened travel durations.
- Inappropriate duration of traffic lights: Each traffic light phase has a predefined duration that does not depend on current traffic conditions. This could lead to unnecessary waiting during the green phase.

Although the potential benefits of the reinforcement learning (RL) approach to traffic signal control mentioned above exist, not all of those objectives have been accomplished, and (as we will show in the remainder of the article) the current approach represents only an initial stride within this broader scope of work. To implement a reinforcement learning algorithm, it is imperative to delineate the state representation, enumerate the permissible actions, and articulate the reward functions. Subsequently, we delineate the most embraced methodologies for formulating these components within the domain of traffic signal control.

- State representation: The state corresponds to the agent's interpretation of the environment at a given time step. Within the literature, representations of state space exhibit variations in information density. In instances of low information density representations, the lanes at intersections are typically discretized into cells along their length. Subsequently, these lane cells are translated into vector cells, with a binary assignment of 1 indicating the presence of a vehicle within the lane cell and O denoting its absence [6]. Some methodologies incorporate additional data, utilizing a vector indicating the presence of cars alongside the inclusion of a vector encoding the relative velocity of vehicles [12]. The addition of the current traffic light phase could be presented as an additional vector. Regarding state representations that contain a wealth of information, typically, the agent is provided with an image capturing the current scenario of the entire intersection, i.e., a snapshot from the employed simulator; multiple consecutive snapshots are then stacked to provide the agent with a comprehensive view of vehicle dynamics [13].
- Actions representation: In the domain of traffic signal control, the agent's actions are characterized by varying degrees of flexibility, as outlined below. Within the category of actions with limited flexibility, the agent can opt for predefined sets of light combinations. Once an action is chosen, a fixed duration ensues before the agent can make a new configuration selection [12]. Some studies have granted the agent increased flexibility by stipulating variable phase durations [14]. An agent endowed with greater flexibility selects at each simulation step from a predetermined set of light combinations. However, the chosen action remains inactive until the minimum required time to release at least one vehicle has elapsed [13]. Alternatively, a distinct approach involves a predefined cycle of activated light combinations within the intersection. The agent's action is manifested in the decision of when to transition to the next light combination, with this determination made at each simulation step [6].

- Reward representation: The agent utilizes the reward signal to assess the impact of its latest action within the current state. Typically, this reward is defined as a function of key performance indicators related to intersection efficiency, such as delays in vehicles, queue lengths, waiting times, or overall throughput. Many studies involve computing the change in cumulative vehicle delay resulting from different actions, where vehicle delay is quantified as the duration in seconds that a vehicle remains stationary [13]. Alternatively, cumulative vehicle stay time, representing the seconds a vehicle has been stationary since entering the environment, can be employed as a metric [12]. Additionally, certain studies adopt a composite approach by combining multiple indicators in a weighted sum [15].

3.3. Adopted models and learning algorithms:

The latest advancements in reinforcement learning research have presented various potential solutions for tackling the traffic signal control challenge. It is evident from the studies that diverse algorithms and neural network structures can be applied. However, it is crucial to note that while certain common techniques are essential, they alone are insufficient to guarantee optimal performance. Among the predominant algorithms employed for addressing this issue, Q-leaning stands out. The achievement of the agent's optimal behavior involves leveraging neural networks to approximate Q-values based on a given state. Often, this method involves the utilization of a Convolutional Neural Network (CNN) to analyze the environmental condition and derive features from either an image or a spatial depiction [13]. Genders, Ravi, and Gao et al. [12] utilize a Convolutional Neural Network (CNN) to extract features from the spatial representation of the environment. The network's output, combined with the current phase, is then fed into two fully connected layers, ultimately connecting to the output nodes representing Q-values. This approach demonstrated favorable outcomes in [12], showcasing its effectiveness across various traffic light policies, such as prioritizing longer queues and adhering to fixed schedulesln, a comparative analysis is conducted against a shallow neural network [13]. Mousavi et al. [13] conducted a comprehensive examination of a dual-pronged strategy to tackle the traffic signal control challenge. The first facet employs a value-based approach, while the second adopts a policy-based methodology. In the value-based approach, predictive modeling of action values is carried out through the minimization of means squared error of Q-values using the stochastic gradient-descent method. Conversely, the alternative approach focuses on teaming the policy by iteratively updating the policy parameters, thereby enhancing the likelihood of selecting optimal actions. Convolutional Neural Networks (CNN) serve as effective function approximates for feature extraction from intersection images. In the value-based approach, the output corresponds to action values, while in the policy based approach, it manifests as a probability distribution across potential actions. The outcomes reveal that both approaches exhibit commendable performance relative to a predefined baseline and demonstrate resilience against instability issues. In this paper [14], a neural network architecture employing deep stacked auto encoders (SAE) is employed for Q-value learning. This methodology leverages auto encoders to minimize the discrepancy between the predicted Q-values from the encoder neural network and the target Q-values, utilizing a designated loss function. The study demonstrates superior performance compared to conventional reinforcement learning (RL) methods.

4. Implementation of the method

The traffic microsimulation tool utilized in this study is Simulation of Urban Mobility (SUMO) [16]. SUMO provides a comprehensive software package encompassing an infrastructure editor, a simulator interface, and an application programming interface (API). These components empower users to devise and implement tailored configurations and functionalities for road infrastructure, as well as facilitate data exchange during traffic simulations. In this investigation, we aim to explore the potential enhancement of traffic flow through intersections controlled by traffic lights employing artificial intelligence techniques. The agent in this context is embodied by the traffic light system, which engages with the environment to optimize a specific metric of traffic efficiency. With this overarching framework, the problem addressed in this paper is succinctly defined: given the intersection's state, the objective is to determine the optimal traffic light phase from a predefined set of actions. This choice is geared towards enhancing rewards, ultimately optimizing the traffic flow within the intersection. The standard workflow of the agent is depicted in Figure 2.

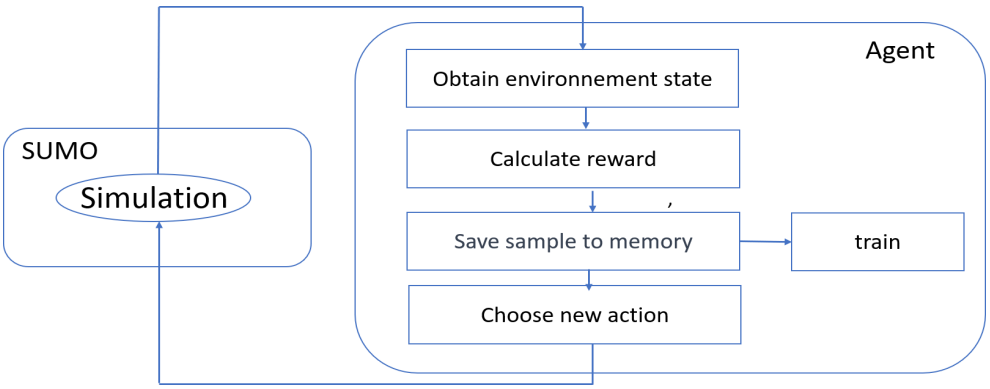

Figure 2: The agent's workflow.

It is crucial to emphasize that in the SUMO application, the temporal progression is delineated through simulation steps. The agent, however, engages only at specific steps, following adequate evolution of the environment. In this study, each step dedicated to the agent's operations is referred to as an 'agent step,' while steps dedicated to simulation are simply termed 'steps.' Consequently, after a designated number of simulation steps, the agent initiates its sequence of operations by acquiring the current state of the environment. Simultaneously, the agent computes the reward for the previously executed action, utilizing a metric reflecting the prevailing traffic conditions. The data set encompassing comprehensive information from recent simulation steps is stored in memory and subsequently retrieved for training purposes. Subsequently, the agent is poised to select a new action based on the current environmental state, thereby resuming the simulation until the next interaction with the agent occurs. The depiction of the agent's operational space is illustrated in Figure 3. This area constitutes a four way intersection, with four lanes per arm converging upon the intersection from the cardinal directions, leading to an equal number of lanes per arm extending from the intersection. Each arm spans a length of 750 meters. Within each lane on every arm, specific directional options for vehicles are defined: the rightmost lane facilitates right turns or proceeding straight, the two central lanes mandate straight-ahead movement, and the left-most lane exclusively permits left turns. Situated at the intersection's center, a traffic light system, under the control of the agent, regulates incoming traffic. Each lane has its own designated traffic light on the rightmost side, whereas the other three lanes share a single traffic light. These traffic lights adhere to European regulations, except for the absence of a pause between the conclusion of the yellow phase and the commencement of the green phase. It is important to note that this setting does not involve pedestrians, sidewalks, or pedestrian crossings.

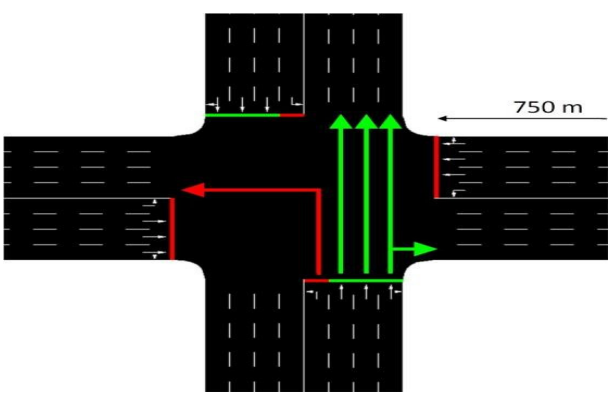

Figure 3: The environment.

4.1. Mathematical model of the reinforcement learning methodology

To develop a system using reinforcement learning, it is crucial to define the state representation, the available actions, the reward function, and the learning approaches used by the agent. It is crucial to underscore that the components of these agents, as presented in this paper, can be seamlessly substituted with those of a traffic monitoring system in practical applications. This distinguishes our study from others in the field, as it exhibits lower technical feasibility requirements, rendering it more adaptable to real-world implementations.

4.1.1. Representation of the state. The agent's current state encapsulates an overview of the environmental status at a particular step in the agent's sequence, commonly referred to as 's_t'. To ensure efficient traffic optimization through learning, it is crucial for authorities to provide detailed data on vehicle distribution along all roads. The chosen representation aims to provide the agent with insight into the spatial positioning of vehicles within the environment at agent step 't.' Drawing inspiration from DTSE [17], our approach encodes less information in this state. Specifically, our state design includes only spatial details about vehicles in the environment, and the discretization of the continuous environment involves irregular cells. Emphasizing realism, our chosen state representation diverges from recent works on traffic signal controllers that advocate information-rich states. These states, while theoretically comprehensive, pose implementation challenges due to the difficulty in gathering the required information. Therefore, this paper explores the viability of achieving favorable results with a simpler and more practical state representation.

4.1.2. Action set. The action set defines the possible actions accessible to the system. In this scenario, the system refers to the traffic light setup, where taking an action means initiating a green signal for a particular group of lanes for a predetermined period. The options available within the action space are selected from a predetermined set of green signal configurations. In this study, the duration of the green signal is fixed at 10 seconds, while the yellow signal lasts for 4 seconds compassing all conceivable actions within the agent's purview.

$$A = \{NSA, NSLA, EWA, EWLA\} \tag{4.1}$$

Each action within set (4.1) is defined as follows:

- North-South Advance (NSA): Triggers the green signal for vehicles traveling straight or making right turns from the north and south directions.
- North-South Left Advance (NSLA): Initiates the green signal for vehicles turning left from the north and south directions.
- East-West Advance (EWA): involves enabling the green signal for vehicles proceeding straight or making right turns from the east and west directions.
- East-West Left Advance (EWLA): Triggers the green signal for vehicles turning left from the east and west directions.

The Figure 4 provides a visual representation of these four actions.

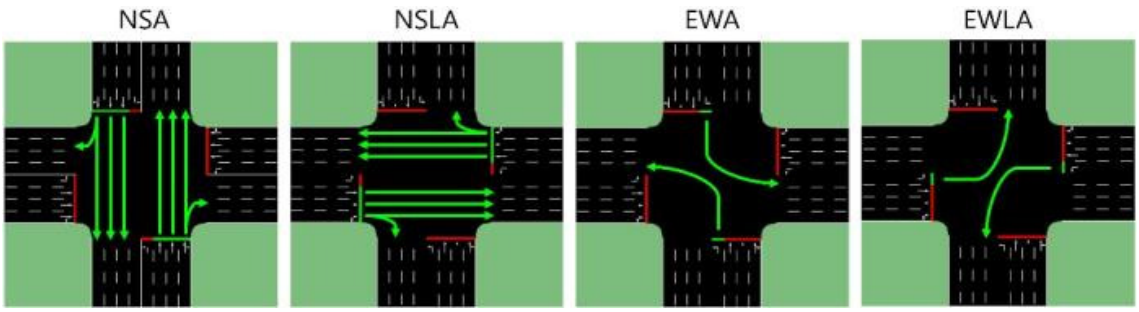

Figure 4: Graphical representation of the four possible actions.

4.1.3. Function for determining rewards. The reward in this context is the outcome that the environment provides to the agent following the execution of an action from a given state, specifically aimed at enhancing the smooth flow of traffic through a junction over time. The determination of this reward is contingent upon a performance metric reflecting traffic efficiency. Various metrics, such as throughput, average delay, waiting time, and journey time, have been explored in the literature. In our model, we opt to employ cumulative waiting time as a pivotal parameter influencing the fluidity of the road network. Its calculation is defined by Equation (4).

$$\mathbf{TCA}_t = \sum_n \mathbf{TA}(\mathbf{veh}, t) \tag{4.2}$$

- $\mathbf{TCA}_t$: the aggregate waiting time-by-time t.
- $TA(veh, t)$ the duration, in seconds, that a vehicle has been waiting at time 't' since its introduction into the environment.
- n the total number of vehicles present in the vicinity at a specific time, represented as 't'.

4.1.4. Mathematical model of Deep Q-Learning. The research methodology employs Deep Learning, which combines two extensively utilized components within reinforcement learning: deep neural networks and Q-Learning (QL). QL is a model-agnostic technique in reinforcement learning, assigns a value referred to as the Q-value to actions performed in specific states within an environment. In scholarly literature, the Q-value is formally defined by equation (5)

$$\mathbf{Q}(\mathbf{S}_t, \mathbf{a}_t) = \mathbf{Q}(\mathbf{S}_t, \mathbf{a}_t) + \alpha(\mathbf{r}_{t+1} + \gamma \cdot \max_{\mathbf{a}'} \mathbf{Q}(\mathbf{S}_{t+1}, \mathbf{a}') - \mathbf{Q}(\mathbf{S}_t, \mathbf{a}_t)) \tag{4.3}$$

The equation updates the current value, $\mathbf{Q}(\mathbf{S}_t, \mathbf{a}_t)$, by incorporating a term reduced by the learning rate α. Inside the brackets, $\mathbf{r}_{t+1}$ stands for the reward connected to acting $\mathbf{a}_t$ from state $\mathbf{S}$. The subscript $t+1$ denotes the temporal connection between taking an action and receiving its subsequent reward. In the $\mathbf{Q}$-function $(\mathbf{S}_{t+1}, \mathbf{a}_t)$, $\mathbf{S}_{t+1}$ represents the subsequent state following action $\mathbf{a}_t$ in state $\mathbf{S}_t$. The $\max_{\mathbf{a}'}$ term selects the action $\mathbf{a}'$ in state $\mathbf{S}_{t+1}$ with the highest value among all available actions. The discount factor, ranging between 0 and 1, diminishes the importance of future rewards when compared to immediate ones. In this manuscript, a modified rendition of equation (4.3) is employed and denoted as equation (6). Subsequently, it will be specifically referenced as the Q-learning function throughout the remainder of this document.

$$\mathbf{Q}(\mathbf{S}_t, \mathbf{a}_t) = \mathbf{r}_{t+1} + \gamma \cdot \max_{\mathbf{A}} \mathbf{Q}'(\mathbf{S}_{t+1}, \mathbf{a}_{t+1}) \tag{4.4}$$

In the given context, the term 'reward $\mathbf{r}_{t+1}$' refers to the reward obtained after the execution of action 'a_t' in state '$\mathbf{S}_t$'.The expression $\mathbf{Q}'(\mathbf{S}_{t+1}, \mathbf{a}_{t+1})$ denotes the Q-value associated with the action '$\mathbf{a}_{t+1}$' taken in the subsequent state, i.e., state '$\mathbf{S}_{t+1}$', following the execution of action 'a_t' in state '$\mathbf{S}_t$'.As delineated in equation (4.3), the discount factor γ introduces a modest discounting of future rewards in comparison to immediate rewards. Notably, post-training, the optimal action 'a_t' chosen from state 's_t' corresponds to maximizing the Q-function $\mathbf{Q}(\mathbf{S}_t, \mathbf{a}_t)$, In essence, optimizing the Q-learning function entails adhering to the most effective strategy learned by the agent. In the realm of applying reinforcement learning, the vastness of potential states often makes fully exploring and storing every state-action pair impractical. Therefore, to address this challenge, a neural network approximates the Q-learning function. This study employs a fully connected deep neural network design: it includes an input layer with 80 neurons, followed by five hidden layers, each with 400 neurons and using a rectified linear activation function (ReLU). An output layer comprising four neurons with a linear activation function caps the network. Each neuron in this layer represents the value of a specific action within a given state. Figure 5 visually illustrates the architecture of this deep neural network.

4.1.5. Mathematical model of the deep neural network. Experience replay [18] is a technique utilized during training to improve the agent's performance and the overall effectiveness of the learning process.

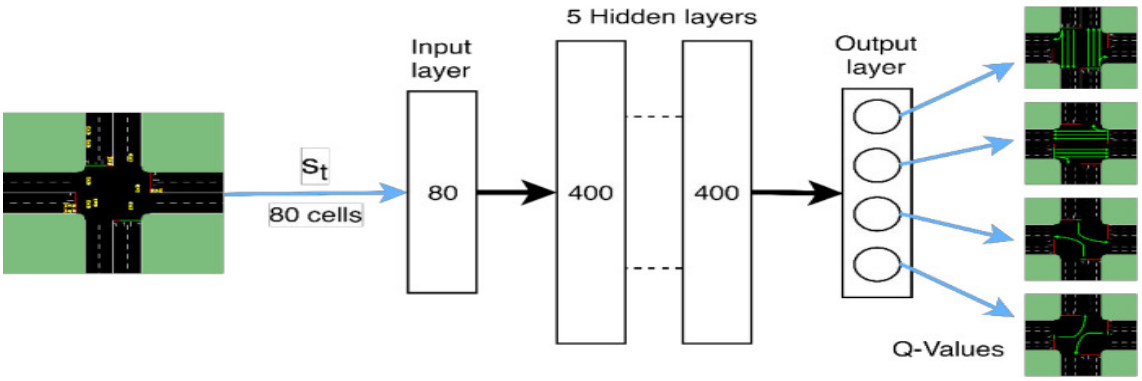

Figure 5: Scheme of the deep neural network.

The training method consists of supplying the agent with essential information packaged as a randomly arranged dataset known as a batch, rather than instantly exposing the data gathered during the simulation, as shown in Figure 5. This batch is extracted from a data structure appropriately named 'memory', where each piece of information collected during training is stored. Specifically, a piece of information, identified as m, is defined as a quadruplet (7) encompassing crucial details.

$$\mathbf{m} = \{\mathbf{S}_t, \mathbf{a}_t, \mathbf{r}_{t+1}, \mathbf{S}_{t+1}\} \tag{4.5}$$

Where $\mathbf{r}_{t+1}$ represents the reward received upon acting $\mathbf{a}_t$ in state $\mathbf{s}_t$, resulting in the transition to the next state $\mathbf{S}_{t+1}$ within the environment. This method is employed to mitigate correlations within the observation sequence, given that the environment's state at s_{t+1} directly stems from the preceding state s_t. High correlation can impede the training effectiveness of the agent. Figure 6 illustrates a depiction of the data collection task.

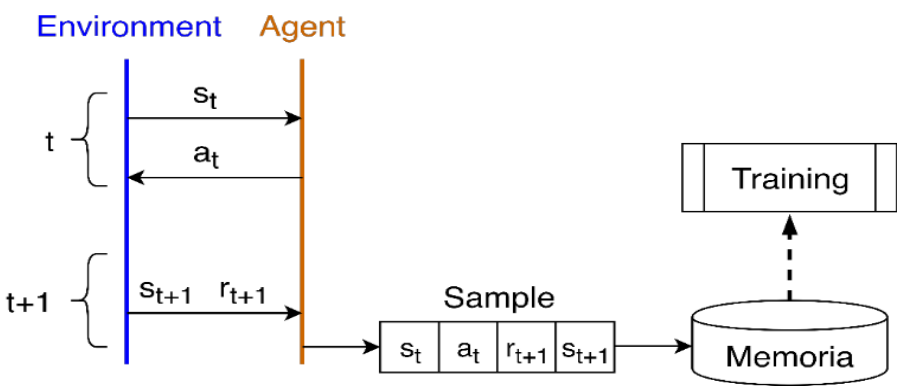

Figure 6: Scheme of the data collection.

As previously discussed, the experiment replay technique necessitates a memory system characterized by two parameters: The memory size, set at 50,000 samples, represents the storage capacity for storing samples. Meanwhile, the batch size, fixed at 20, refers to the number of samples retrieved from memory during a single training instance. If the memory reaches full capacity during a particular stage of the agent, the oldest sample is systematically removed to accommodate new incoming samples. A training

instance involves the iterative learning of the Q-value function by leveraging the information within a batch of extracted samples. Each sample within the batch is utilized for training purposes. From the perspective of an individual sample, which encompasses the elements $\{\mathbf{S}_t, \mathbf{a}_t, \mathbf{r}_{t+1}, \mathbf{S}_{t+1}\}$, the subsequent operations are executed:

- Anticipation of $\mathbf{Q}(\mathbf{S}_t)$ values, encapsulating the agent's present understanding of action values at the current state, is performed.

- Prediction of $\mathbf{Q}'(\mathbf{S}_{t+1})$ values. These represent the agent's knowledge of action values from state $\mathbf{S}_{t+1}$.

- The update of $\mathbf{Q}(\mathbf{S}_t, \mathbf{a}_t)$ is conducted, reflecting the value associated with a specific action, $\mathbf{a}_t$, chosen by the agent during the simulation. The element $\mathbf{r}_{t+1}$ denotes the reward linked to the action $\mathbf{a}_t$, and $\max_{\mathbf{A}} \mathbf{Q}'(\mathbf{S}_{t+1}, \mathbf{a}_{t+1})$ is derived from the prediction of $\mathbf{Q}'(\mathbf{S}_{t+1})$, signifying the highest anticipated future reward or the optimal action value expected by the agent from the state $\mathbf{S}_{t+1}$. This value is then scaled by a factor γ, emphasizing the significance of immediate rewards.

- Training the neural network involves utilizing the state $\mathbf{s}_t$ as input and deriving the desired output, which encompasses the updated $\mathbf{Q}(\mathbf{S}_t, \mathbf{a}_t)$ values incorporating the maximum anticipated future reward from the revised Q value.

After successfully approximating the Q-learning function through the deep neural network, optimal traffic efficiency is attained by selecting the action with the highest value corresponding to the current state. Addressing a significant challenge in reinforcement learning tasks involves establishing an action selection policy during the learning phase, where the dilemma lies between opting for an exploratory action to potentially acquire more knowledge or selecting an exploitative action to optimize existing knowledge within a changing environment. In this study, the ϵ-greedy exploration policy is adopted, as defined by equation (8). This policy assigns a probability ϵ for the current episode h to choose an exploratory action and, consequently, a probability of $1 - \epsilon$ to select an exploitative action, thereby navigating the delicate balance between exploration and exploitation.

$$\epsilon = 1 - h/H \tag{4.6}$$

In the given equation, 'h' denotes the current training episode, and 'H' represents the total number of episodes. Initially, the value of 'ϵ' is set to 1, indicating that the agent engages solely in exploration. As the training advances, the agent gradually shifts towards exploitation, leveraging the knowledge acquired, until it eventually engages exclusively in exploitation.

5. Result and discussion

The program is executed using the jupyter code editor. A training phase is launched, followed by a simulation phase presented in the following figures 7, 8, 9 and 10:

```
----- Episode 1 of 20
Simulating...
Total reward: -38576.0 - Epsilon: 1.0
Training...
Simulation time: 6.9 s - Training time: 0.0 s - Total: 6.9 s

----- Episode 2 of 20
Simulating...
1/1 [==============================] - 0s 251ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 37ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 41ms/step
1/1 [==============================] - 0s 40ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 36ms/step
1/1 [==============================] - 0s 27ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
Total reward: -28758.0 - Epsilon: 0.95
Training...
```

Figure 7: Model in training.

```
Total reward: -4943.0 - Epsilon: 0.1
Training...
Simulation time: 45.4 s - Training time: 9244.8 s - Total: 9290.2 s

----- Episode 20 of 20
Simulating...

    Total reward: -4840.0 - Epsilon: 0.05
    Training...
Simulation time: 100.9 s - Training time: 14807.0 s - Total: 14907.9 s

----- Start time: 2023-10-11 09:13:54.421596
----- End time: 2023-10-12 11:55:40.811926
----- Session info saved at: C:\Users\S2M\Agent_DRL_SUMO\models\model_1\
```

Figure 8: End of training

After this training phase, the simulation phase can now begin.

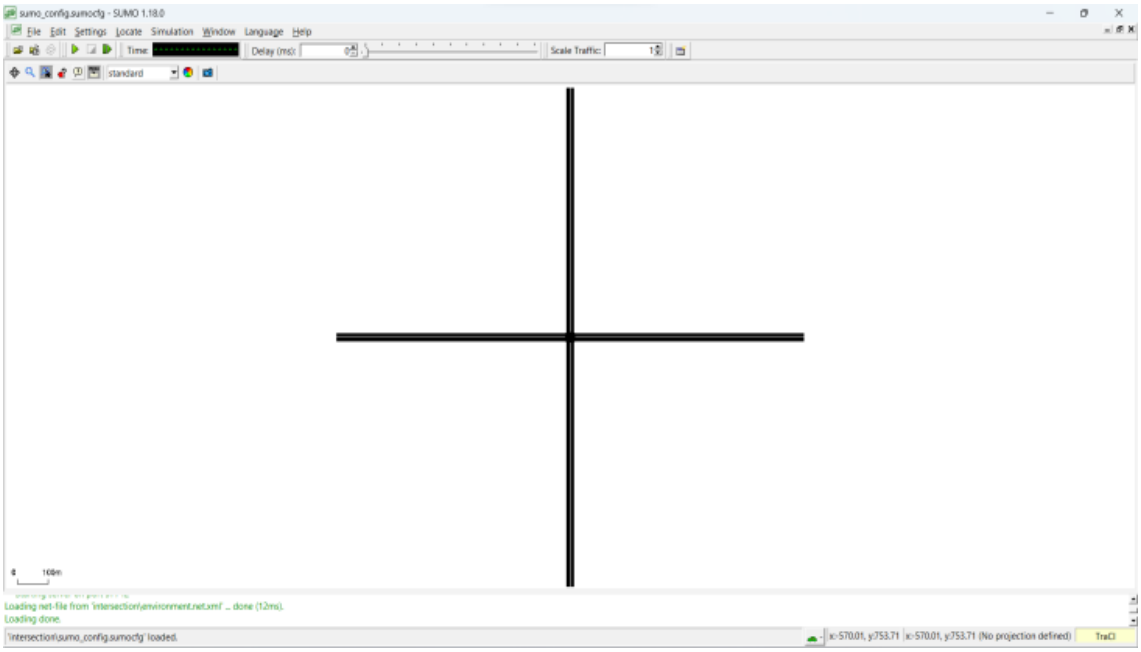

Figure 9: Presentation of the SUMO intersection.

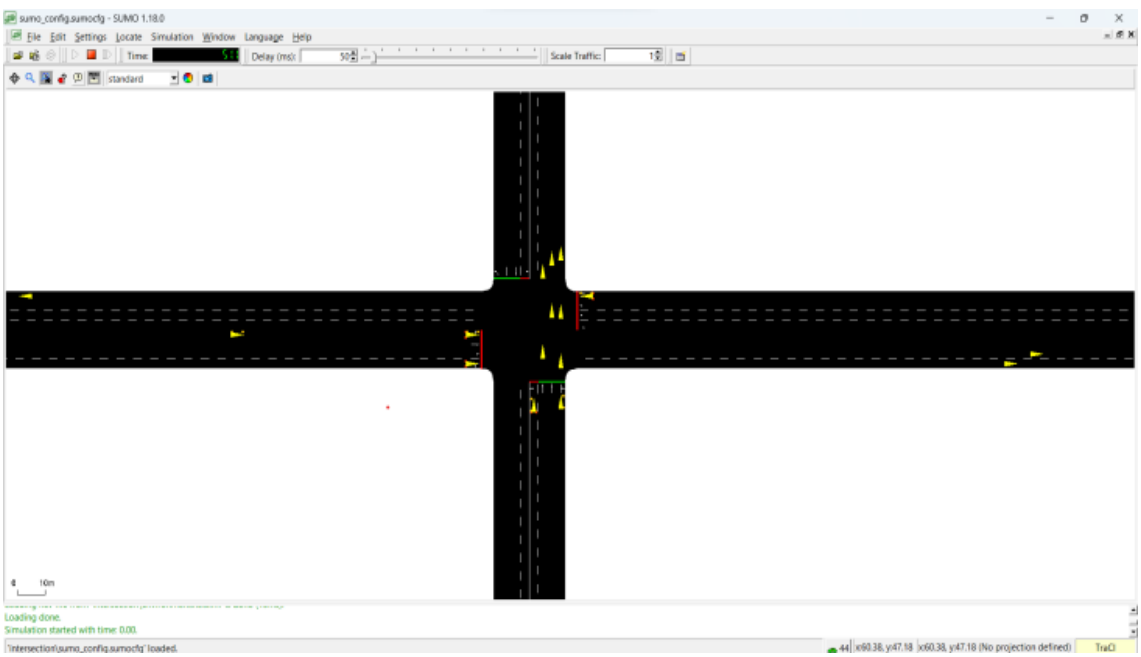

Figure 10: Simulation in progress.

The results obtained were presented in the form of graphs evaluating the performance of the proposed system for the training and test stages and are presented below in figure 11, 12 and 13.

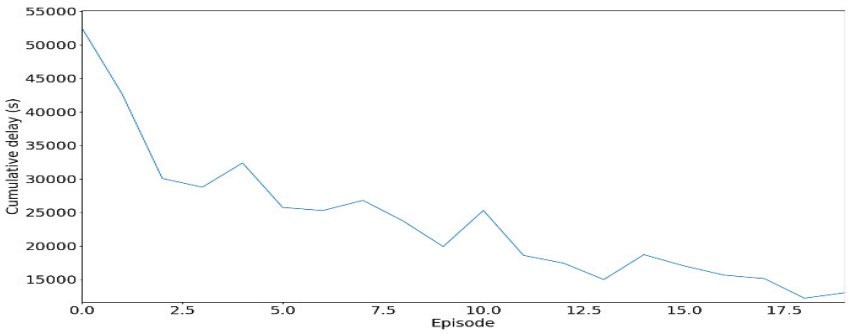

Figure 11: Cumulative waiting time for the training phase.

Figure 11 shows the training result of our model as a function of cumulative waiting time. It can be seen that the waiting time decreases with the number of episodes, which means that our model is well trained. The results discussed are the number of vehicles (Queue) as a function of the traffic flow.

For low throughput, the number of incoming vehicles is set at 100. For medium throughput, the number of vehicles varies between 200 and 500, whereas for high throughput, the number of vehicles exceeds 800.

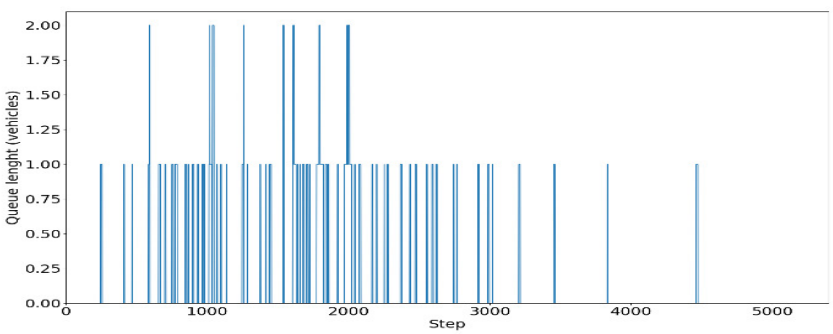

Figure 12: Tail for low throughput (100 vehicles).

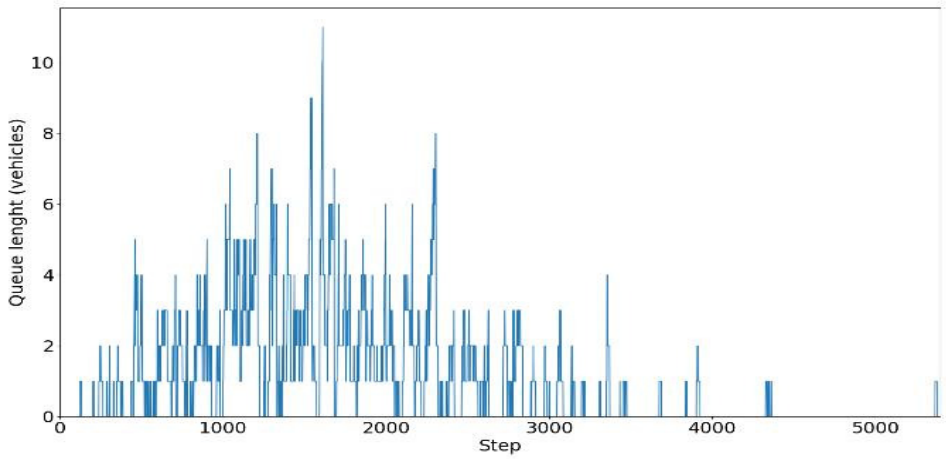

Figure 13: Tail for medium traffic (500 vehicles).

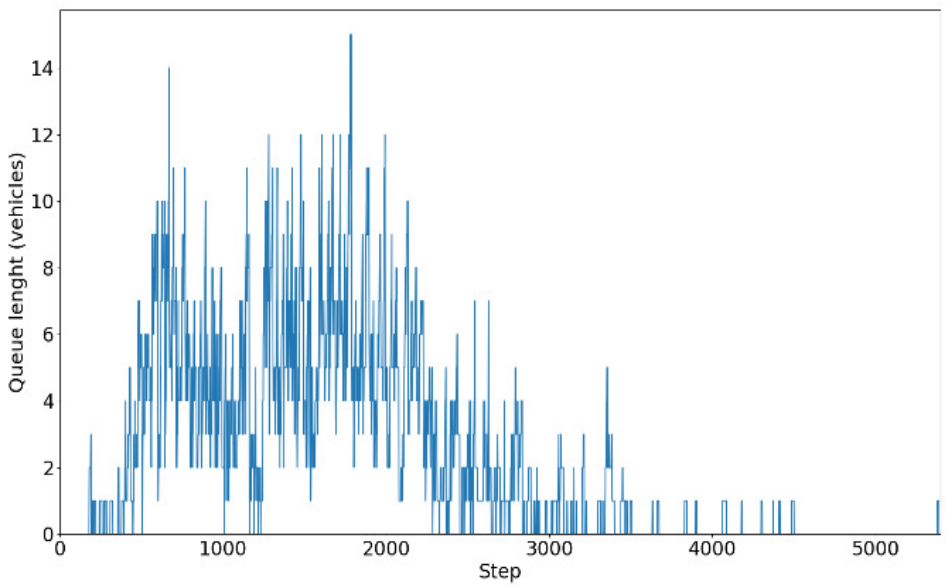

Figure 14: High-speed tail (1000 vehicles).

The previous figures show the queue resulting from the test of the trained model. In all three figures, we can see that there are peaks at certain times. These peaks differ from one figure to another depending on the type of flow. From the figure 12, there are 2 vehicles maximum, in the figure 13, there are 10 vehicles maximum and figure 14, there are 16 vehicles maximum. We can also see that the number of vehicles does not increase exponentially; on the contrary, after each peak the number of vehicles vehicles decreases. These results reflect the adaptation of the proposed model to different types of traffic and ensure:

- Adaptation of signaling

- Elimination of congestion.

- Smoother traffic flow.

5.1. Comparison of our proposed model Deep Q-Learning (DQL), RR and FCM

In this discussion section, we will compare our proposed model DQL with various techniques and algorithms (RR and FCM) used in Intelligent Traffic Signal Control (ITSC) [19]. In accordance with the methodology presented by Round Robin Scheduling in [19], each agent is assigned a pre-defined time interval during which its corresponding signal arm displays the green light. This set duration remains consistent across different traffic situations, even in cases of minimal traffic. Each interval consists of 30 time steps of green signal, succeeded by three time steps of yellow signal.

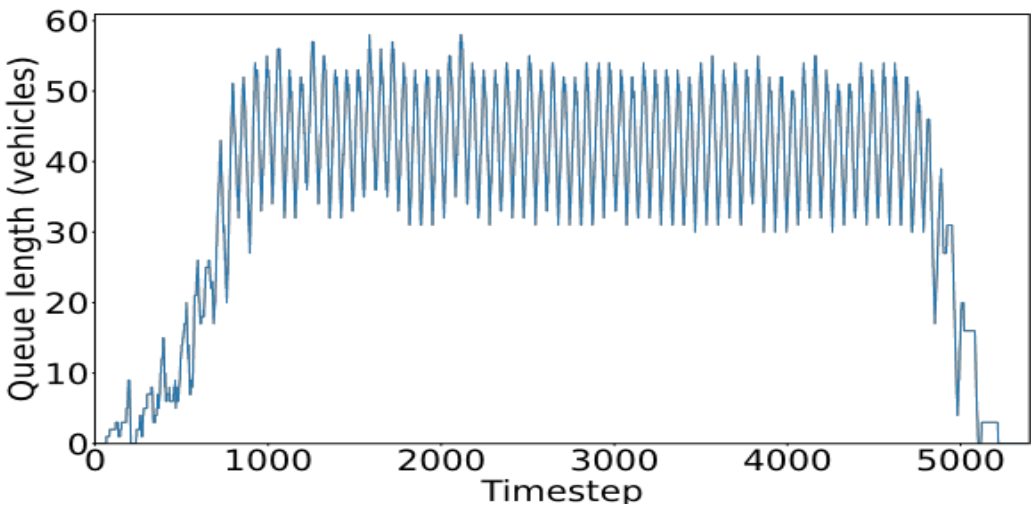

Figure 15: Total queue lengths vs timestep using Round Robin (RR) (Hrishit Chaudhuri, Vibha Masti, Vishruth Veerendranath, and S Natarajan1)

This adaptation represents a slight variation of the Round Robin (RR) method, as shown in Figure 15. In this paper [19] The core principle here is to enable the signal system to exhibit basic 'learning' capabilities by collecting data from the environment and adapting its operations accordingly. As a result, the action space is structured as a set of distinct time intervals, offering choices for the traffic management system. The action corresponds to the duration for which the agent signals green.

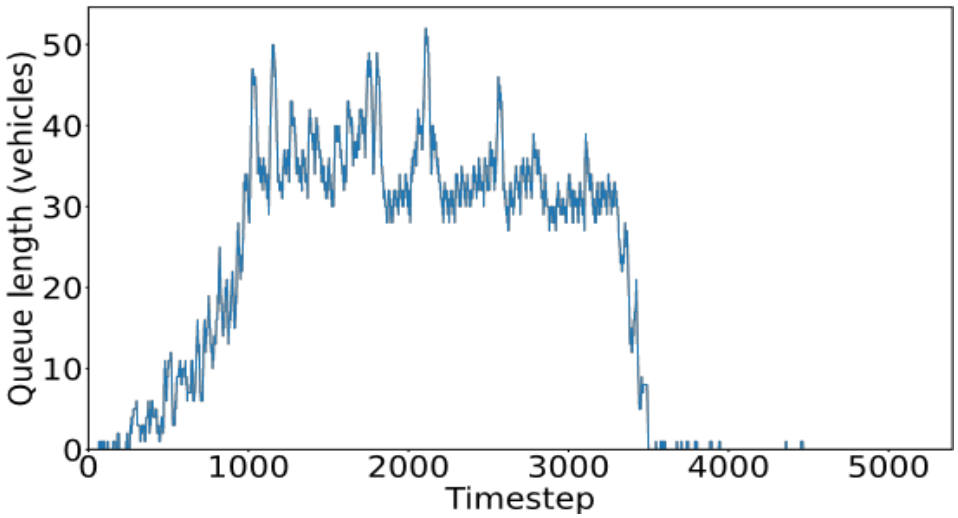

Figure 16: Total queue lengths vs timestep using Feedback Control (Hrishit Chaudhuri, Vibha Masti, Vishruth Veerendranath, and S Natarajan1)

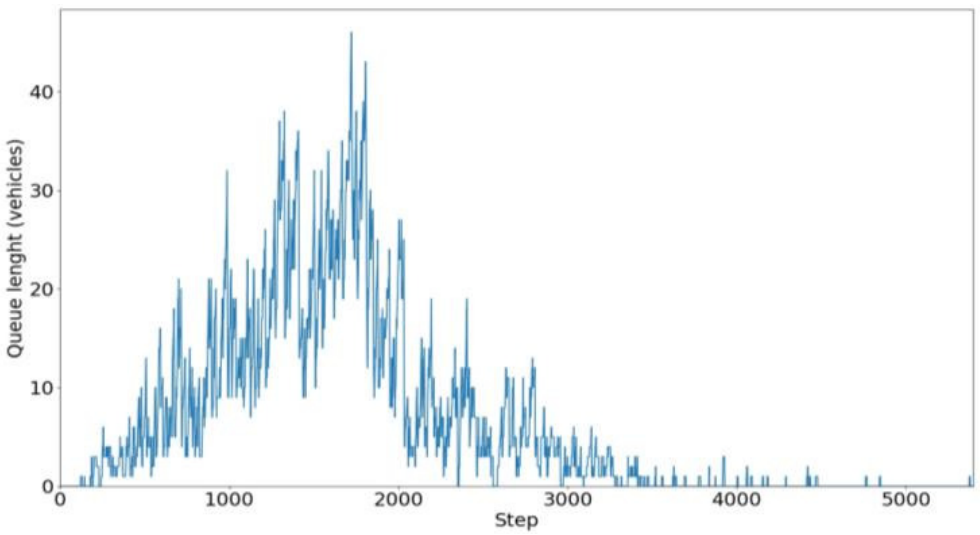

Figure 17: Total queue lengths vs time step using our proposed system Deep Q-Learning (DQL)

The previous figures 15, 16 and 17 show the queue resulting from the three models (RR, FCR, DQL). In all three figures, we can see that there are peaks at certain times. These peaks differ from one figure to another depending on the model. For figure 15: 55 vehicles maximum, figure 16, 50 vehicles maximum and figure 17: 45 vehicles maximum. We can also observe that the number of vehicles shown in the results of our proposed model in Figure 17 does not increase exponentially. On the contrary, after each peak, the number of vehicles decreases. Contrary to the results of the other two models (RR and FCM) presented in Figures 15 and 16, which show that the number of vehicles increase constantly. Therefore, our proposed model (DQL) sets itself apart with remarkable performance achievements, even if this means compromising on computational intensity. The value of DQL lies in its capacity to deliver impressive results, which can be especially advantageous when 12 optimizing traffic signal control under complex

and dynamic traffic patterns.

6. Conclusion

The issue of traffic congestion stands as a notable challenge in the advancement of our society. Various initiatives have been undertaken to tackle this dilemma. Our objective is to present an innovative solution aimed at mitigating road traffic congestion and enhancing its overall efficiency. This study has undertaken a credible examination of the feasibility of utilizing a Reinforcement Learning (RL) approach to address the challenge of adapting and managing traffic lights. The research has utilized a validated and realistic traffic simulator to establish a conducive environment for training and evaluating the RL agent. Our research has led us to propose a new solution for the dynamic management of intelligent signaling by using a hybridization of two learning algorithms, namely, reinforcement learning (Q learning) and Deep Learning. The proposed system has been tested for various traffic flow rates, and it was observed that our system is adaptive to high traffic flow, thus meeting the intended objective of alleviating congestion during peak hours. Future research efforts will concentrate on improving the achieved results. In the long term, investigations will be directed towards understanding the implications of integrating multiple Reinforcement Learning (RL) agents within a road network. This exploration will delve into the potential coordination of their efforts to achieve global enhancements over local ones. Additionally, the study will assess the impact on the vehicle population, considering how it may perceive changes in the infrastructure and adapt to exploit new opportunities. It is crucial to conduct thorough analyses in this research trajectory to comprehend the feasibility, potential benefits, and unintended negative consequences associated with the real-world implementation of such a self-adaptive system.

References

1. Behrisch, M., Bieker, L., Erdmann, J., & Krajzewicz, D., *SUMO–simulation of urban mobility: an overview*, In Proceedings of SIMUL 2011, the Third International Conference on Advances in System Simulation. ThinkMind, (2011).
2. Wu, J., Abbas-Turki, A., & El Moudni, A., *Traffic control in urban intersections using discrete methods*, In Proceedings of the 69th Vehicular Technology Conference (VTC Spring), 1–5. IEEE, (2009).
3. Glankwamdee, W., *Branch and Bound on computational grids*, Doctoral thesis at Lehigh University, (2008).
4. Aibeche, Y., *Development of an Intelligent Traffic Light Management System Using Genetic Algorithms*, (2020).
5. Khankhour, H., Abdoun, O., & Abouchabaka, J., *Parallel genetic approach for routing optimization in large ad hoc networks*, International Journal of Electrical and Computer Engineering, 12(1), 748–755, (2022).
6. Genders, W., *Reinforcement Learning for Traffic Signal Control*, Doctoral dissertation, McMaster University, Hamilton, Ontario, Canada. Department of Civil Engineering, (2018).
7. Naeem, M., Rizvi, S. T. H., & Coronato, A., *A gentle introduction to reinforcement learning and its application in different fields*, IEEE Access, vol. 8, 209320–209344, (2020).
8. Vinyals, O., Babuschkin, I., Chung, J., Mathieu, M., Jaderberg, M., Czarnecki, W., & Silver, D., *Alphastar: Mastering the real-time strategy game starcraft ii*, DeepMind blog, vol. 2, 20, (2019).
9. Kalashnikov, D., Irpan, A., Pastor, P., Ibarz, J., Herzog, A., Jang, E., & Levine, S., *Scalable deep reinforcement learning for vision-based robotic manipulation*, In Conference on robot learning, 651–673. PMLR, (2018, October).
10. Yau, K.-L. A., Qadir, J., Khoo, H. L., Ling, M. H., & Komisarczuk, P., *An overview of reinforcement learning models and algorithms applied to the control of traffic signals*, ACM Computing Surveys (CSUR), vol. 50, no. 3, 34, (2017).
11. Genders, W., & Razavi, S., *Assessment of state representations in reinforcement learning for adaptive traffic signal control*, Procedia Computer Science, vol. 130, 26–33, (2018).
12. Gao, J., Shen, Y., Liu, J., Ito, M., & Shiratori, N., *Adaptive traffic signal control: Deep reinforcement learning algorithm with experience replay and target network*, arXiv preprint arXiv:1705.02755, (2017).
13. Mousavi, S. S., Schukat, M., & Howley, E., *Traffic light management employing deep policy-gradient and value-function-based reinforcement learning*, IET Intelligent Transport Systems, vol. 11, no. 7, 417–423, (2017).
14. Li, L., Lv, Y., & Wang, F.-Y., *Traffic Signal Timing Using Deep Reinforcement Learning*, IEEE/CAA Journal of Automatica Sinica, vol. 3, no. 3, 247–254, (2016).
15. Wei, H., Zheng, G., Yao, H., & Li, Z., *Intellilight: A Reinforcement Learning Approach for Intelligent Traffic Light Control*, In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2496–2505, ACM, (2018).

16. Meng, Z., Du, X., Sottovia, P., Foroni, D., Axenie, C., Wieder, A., & Sommer, C., *Topology-preserving simplification of OpenStreetMap network data for large-scale simulation in SUMO*, In SUMO Conference Proceedings, vol. 3, 181–197, (2022).

17. Dudiak, D., Szabóová, M., & Magyar, J., *Exploring Q-Learning in Social Robots for English-Slovak Vocabulary Learning*, In 2023 IEEE 23rd International Symposium on Computational Intelligence and Informatics (CINTI), 000025–000030, IEEE, (2023, November).

18. Zhang, Z., *Introducing Reinforcement Learning to High School Students Through the Integration of Physical Robots and Virtual Interfaces*, Doctoral dissertation, Tufts University, (2023).

19. Chaudhuri, H., Masti, V., Veerendranath, V., & Natarajan, S., *A comparative study of algorithms for intelligent traffic signal control*, In Machine Learning and Autonomous Systems: Proceedings of ICMLAS 2021, 271–287, Singapore: Springer Nature Singapore, (2022).

Hala Khankhour: *Research laboratory in computer science,*
Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco.
E-mail: hala.khankhour@uit.ac.ma

Najat Rafalia: *Research laboratory in computer science,*
Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco.
E-mail: najat.rafalia@uit.ac.ma

Jaafar Abouchabaka: *Research laboratory in computer science,*
Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco.
E-mail: Jaafar.abouchabaka@uit.ac.ma