Deep Reinforcement Learning Approach for V2X Managed Intersections of Connected Vehicles

Alexandre Lombard, Ahmed Noubli, Abdeljalil Abbas-Turki, Nicolas Gaud, and Stéphane Galland

Abstract— **Intersections are major bottlenecks for road traffic, as well as the origin of many accidents. Efficient management of traffic at intersections is required to ensure both safety and efficiency. Yet, the traditional solutions (static signs, traffic lights) are limited in their efficiency as they consider the flow of vehicles and not the vehicles at the microscopic level. By using inter-vehicular communication of connected vehicles, recent works have shown the possibility to have a great increase in the number of evacuated vehicles thanks to the possibility to give an individual right-of-way directly to each vehicle. In this context of intersections of cooperative vehicles, the scheduling of this right-of-way in order to maximize the throughput of the intersection is still a challenging task, with regard to the hybrid and dynamic aspects of the problem. In this paper, we propose an approach based on Deep Reinforcement Learning (DRL) to efficiently distribute the right-of-way to each vehicle. A Markov Decision Process model of intersections of cooperative vehicles, enabling the application of DRL, is proposed. The performance of the DRL-based scheduling is then compared with classic traffic lights, and with two state-of-the-art cooperative scheduling policies, showing the benefits of the approach (increase of the flow, reduction of CO2 emissions).**

Index Terms— **Autonomous vehicle, deep reinforcement learning, traffic management, V2X.**

I. Introduction

THE emergence of automation and connectivity of commercial vehicles is quite recent, and there remains a lot of work before reaching the SAE (Society of Automotive Engineers) Levels 4 and 5 of autonomous vehicles [1]. In the same time, vehicular communication protocols are quickly evolving, notably supporting Wireless Access in Vehicular Environment (WAVE), itself supported by IEEE 802.11p and IEEE 802.11bd, and enabling Vehicle-to-Everything (V2X) communication (vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), vehicle-to-pedestrian (V2P) and vehicle-to-network (V2N)). Hence, the idea of using the control abilities of these vehicles along with their ability to communicate, allows thinking of a safer, more efficient traffic management, especially at intersections.

Intersection management (IM) is a challenging problem. Most of the fatal and injurious collisions happen to occur close to intersections [2]. Intersections also act as a bottleneck of traffic flow, inducing congestion and an increase of pollutant emissions. Preventing collisions requires a careful and efficient management system to distribute the right-of-way (ROW) to the vehicles, but the classical solutions (either static signs or traffic lights) are limited as they do neither consider traffic dynamics at the microscopic level, nor the routes of the different vehicles.

Cooperative Intersection Management (CIM) aims to replace the classic signalization at intersections (the road signs, as well as the traffic lights). It harnesses the connectivity vehicles to provide more accurate management of their conflicts. In a CIM system, V2V as well as V2I are used to distribute the right-of-way to the vehicles. While CIM is still at the experimental state for road vehicles, due to its requirements in terms of infrastructure deployments and vehicle equipments, it is actually used for industrial Automated Guided Vehicles (AGV) and is planned to be extended to autonomous shuttles system.

Several CIM approaches differ, yet they rely on a common core of concepts:

- The right-of-way is individualized (as opposed to traffic lights where the right-of-way is given to a lane).
- To determine this right-of-way, the vehicles transmit their position, speed, and all relevant data to an "intersection manager", which uses this data to transmit authorization to vehicles.

Most of the differences between the CIM approaches lie in the:

- Communication protocol: what information is transmitted, by whom, and when.
- Scheduling policy that is used to define which vehicle has the right of way: which vehicle should cross the intersection first.
- Vehicle's trajectory control: how the authorization impacts the displacement of the vehicle. If the velocity of the vehicle is automatically controlled according to the right-of-way, it is an Autonomous Intersection Management (AIM).

Usually, in a CIM system, the vehicles approaching the intersection will send a request to an intersection manager (roadside unit or RSU) to get the right-of-way. The RSU replies to each vehicle with adequate information to organize

Manuscript received 22 August 2022; revised 6 January 2023 and 27 February 2023; accepted 3 March 2023. Date of publication 16 March 2023; date of current version 7 July 2023. This work was supported by the Université de Technologie de Belfort-Montbéliard in the context of the Inter-Universités de Technologie (UT) Project SMART-E2AU. The Associate Editor for this article was L. Wang. *(Corresponding author: Alexandre Lombard.)*

The authors are with the Université de Technologie de Belfort-Montbéliard (UTBM), Laboratoire Connaissance et Intelligence Artificielle Distribuées (CIAD), F-90010 Belfort cedex, France (e-mail: alexandre.lombard@utbm.fr; ahmed.noubli@utbm.fr; abdeljalil.abbas-turki@utbm.fr; nicolas.gaud@utbm.fr; stephane.galland@utbm.fr).

Digital Object Identifier 10.1109/TITS.2023.3253867

the crossing of the intersection. Therefore, RSU is in charge of deciding the sequence of passage of the vehicles. The scheduling policy is the key point of the optimization of the cooperative intersection: defining which vehicle has the right-of-way, ensures the prevention of collisions, and enables the optimization of the intersection according to a chosen criterion.

Several scheduling algorithms, like dynamic programming and Mixed-Integer Linear Programming (MILP), have been proposed to compute the optimal schedule within a given objective function, such as the evacuation time (C_{max}), the sum of tardiness ($\sum T_j$), and so on [3], [4], [5], [6], [7]. Nevertheless, the computation time of these algorithms doesn't fit the (hard) real-time constraint of CIM. Heuristics have been proposed to overcome the issue [8], [9], [10], [11], but the computation time is not the only shortcoming of both optimal/near optimal approaches. Choosing a suitable scheduling approach must address the challenge of the traffic dynamic. Each vehicle arrival questions the already planned schedule if it still optimal. This holds for the dynamic of vehicle trajectory. More precisely, variations of the vehicle's speed and position modify the considered exit time. The two challenges, namely computation time and traffic dynamics, have motivated most of the CIM's works to use simple scheduling rules (see section II).

Moreover, global efficiency of a scheduling policy depends on several parameters, like the topology of the road, or more generally the local traffic conditions (traffic density, heterogeneous traffic flow, etc.) [12], or the latency of the wireless communication (as studied by [13]). Thus, defining an optimal scheduling policy able to dynamically adapt to local traffic conditions, while ensuring the safety of drivers, is a challenging task.

The present paper proposes and a new solution for the design of a suitable scheduling policy, and compares it with current state-of-the-art solutions. For this purpose, we propose an approach based on **Deep Reinforcement Learning (DRL)** algorithms. These algorithms rely on a deep neural network to learn a nearly-optimal policy to control a system through a trial-and-error approach. Unlike supervised learning, DRL does not depend on data labeled by an expert, but allows a system to automatically define a strategy according to a simple reward, making it theoretically easier to set up and to generalize to various domains. This explains the recent successes of DRL based approach in domains which were, until recently, out-of-the-range of classical AI and optimization approaches [14], [15].

In practice, designing a system to be controlled by DRL, and the hyperparameters of such system is challenging. Some specific aspects of complex systems are known to be difficult to manage for most of the DRL approaches [16], and to this day there is no possibility to presume to the applicability of DRL to a specific system. Moreover, there is neither a defined methodology to design a system in such a way that it could be easily controlled by a DRL approach, nor a methodology to define the hyperparameters of the DRL for a given system.

The purpose of this paper is to propose the definition of a DRL-based algorithm for the scheduling task of CIM and to evaluate its impact on traffic performance. Its highlights are:

- The design of a DRL environment for controlling vehicles in intersections with V2X based signalization
- The calibration of the related hyperparameters, and the evaluation of the training process
- The feasibility study in simulation and a comparison with other V2X based signalization approaches, showing the potential benefits of the approach

For designing a DRL-based solution, the following steps must be done: 1) identify a model of the environment (state representation, action space) 2) shape a reward 3) select an appropriate DRL policy 4) train and evaluate the policy

Thus, the paper is structured according to the following. Section II briefly reviews previous work on CIM. The definitions and assumptions related to the problem of the scheduling's optimization of CIM are then detailed in section III. In section IV, a Markov Decision Process (MDP) of the CIM is proposed to be able to apply DRL approach to the CIM scheduling task. In section V an experimental protocol through simulation is presented to evaluate the potential benefits of using DRL according to several criteria, then the results of simulations are detailed and analyzed. Finally, section VI sums up the present paper and concludes our study.

II. Related Work

The concept of CIM of connected and autonomous vehicles (CAV) is not new, as one of its first mention is in [17]. Thereafter, the original idea has been widely studied, and the design of the system and the inherent possibilities of traffic optimization (reduced collisions, increased throughput) have been successively refined by the studies of [13], [18], and [19], leading to the real-world implementation on three unmanned vehicles presented in [20]. The work on the topic is still ongoing, as the field opens a lot of computational possibilities to optimize the traffic, and new, updated, approaches are frequently presented [21], [22]. Even more recent works generalize the concept and consider different kinds of agents to include the pedestrians in the management solution [23], [24], [25].

The scheduling policy is the strategy used to order the right-of-way of the vehicles. In the case of traffic lights, the scheduling is based on fixed-phases, or adaptive phases. For the CIM, among the most commonly used policies, one can cite the *First-Come First-Serve* (FCFS) policy [18], where the vehicles cross the intersection according to their order of arrival; Or the *Distributed Clearing Policy* (DCP) [26], which tries to favor vehicles' platooning to limit the impact of the commutation delay when switching between lanes. However, even if the platoon is mathematically proved to be an effective approach [26], [27], it remains the issue of determining the rules to form an efficient group of vehicles that will cross the intersection together. The answer must take into account the present traffic state, and the probable next ones. Moreover, several parameters need to be considered to form platoons, such as the upper bound of the size of the platoon, the absolute and relative positions and speeds of the selected vehicles, the number of potential conflicts with others, to quote a few. Instead of defining static rules like the ones used by the vehicle-actuated signals [28], [29], this paper questions the ability of the DRL to form these groups as conveniently as possible.

In the domain of traffic management, several papers have studied the applicability of RL and DRL to the scheduling of adaptive traffic lights [30], [31], [32], even considering multiple connected intersections [33], and it is now known that DRL can be used to define the phases of traffic lights efficiently, but these papers do not use the potential offered by the use of V2X communication.

Recently, some interesting work started to study the application of the Reinforcement Learning methods to improve the performance of CIM and Autonomous Intersection Management (AIM) systems. Reference [34] proposed to use RL to control the behavior of an autonomous vehicle through the intersection, but does not consider the other vehicles or the whole traffic as the system to optimize.

An exciting approach is proposed by [35], based on tabular Q-Learning and Multiagent Reinforcement Learning (MARL) to let the vehicles coordinate themselves through the intersection. They use a multiagent formulation to face the challenging aspect (in the sense of computational tractability) of a centralized optimization based on an intersection manager [36]. But, the usage of a centralized system is known to offer an improved safety, especially in case of communication issues.

To our best knowledge, the usage of deep reinforcement learning for a centralized cooperative intersection management has not been studied. While the system is more complex due to a larger action space, the potential benefits should supposedly be higher than existing approaches as the CIM is known to provide better performance than adaptive traffic lights, and as centralized CIM offers a better resilience to issues in the network communication layer.

III. Definitions and Assumptions

This section presents the relevant definitions used throughout this paper, and the assumptions made to model the system to optimize.

An **intersection** I is defined as the junction of several roads, each road being composed of one or more lanes. A lane always has a direction: it can be directed toward the center of the intersection (in this case it is an incoming lane, also called entry zone), or the opposite (in this later case it is an outgoing lane, also called exit zone). In is the set of incoming lanes and Out is the set of outgoing lanes, thus $I = \{In, Out\}$.

The center of the intersection (from the end of an incoming lane, to the start of an exit lane) is presently called the **conflict zone** (CZ); when vehicles are in an incoming/outgoing lane, they have to follow the trajectory of the lane, yet in the conflict zone they follow a trajectory from the end of an incoming lane to the beginning of an outgoing lane. A couple $(entry, exit) \in In \times Out$ is called a **route**. If there is an intersection between two routes r_1 and r_2, i.e. if a vehicle trying to get to the exit of r_1 coming from the associated incoming lane may collide with a vehicle trying to follow r_2, then the two routes are said to be conflicting. We consider that a route is not conflicting with itself, as two vehicles following the same route can be reduced to a car-following problem.

The aim of the traffic management at intersections is to distribute the **right-of-way** (ROW) to the vehicles. When a vehicle has the ROW, it can follow its route without having the risk of colliding with other vehicles. The ROW can be defined as a reservation of a spatially limited part of the route for a limited amount of time. The modalities of this reservation are defined by the **protocol** of intersection management (like traffic lights, or a Cooperative Intersection Management system).

The strategy of the distribution of the ROW to the different vehicles in the incoming lanes is called the **scheduling policy**. An optimal scheduling policy, for a given protocol, is a policy which maximizes the number of vehicles able to leave the intersection for a given amount of time. For instance, FCFS [18] is a common policy where the right-of-way is given to the vehicles according to their order of arrival, but for dense traffic, it is not the most efficient one.

In CIM, the protocol can be either centralized or decentralized. In a centralized approach, there is an **intersection manager** which has the role to distribute the right-of-way to the vehicles, while in a decentralized approach, the vehicles negotiate directly the right-of-way. Also, for centralized CIM, as well as traffic lights (and unlike Autonomous Intersection Management (AIM)), the control on the vehicles is limited to the distribution of the right-of-way. The application of the implied command is up to the driver or to the control system of the vehicle.

To assess the performance of the couple protocol/scheduling policy, several metrics can be used. One can cite the **average waiting time** which is the average number of seconds a vehicle has been stopped before crossing the intersection, and the **throughput** of the intersection which is the total number of vehicles evacuated for a given amount of time.

The stated problem is to optimize the scheduling policy (with regard to the average waiting time and the throughput) of an intersection, using a centralized intersection manager.

Thus, in the present paper, the following assumptions are made:

- The intersection, whose throughput is being optimized, is equipped with an intersection manager distributing the ROW using V2X communication (i.e. a roadside unit)
- The vehicles are equipped with V2X onboard units, data related to the position, the speed and the path of the vehicle are communicated to roadside unit by V2X; this data can be imprecise due to the limited accuracy of the sensors, this is taken into account by the intersection manager
- The communication delay is assumed to be below $500ms$: considering the WAVE model and the performance of the underlying communication layers [37], it is reasonable to consider this assumption will be met. Moreover, the protocol serving as a basis for the optimization is specifically designed to be resilient to message losses and delays [13]; a delay above $500ms$ would eventually reduce the overall performance, but not impact the feasibility of the approach
- **The ROW is individualized**: the intersection manager distributes the ROW to individual vehicles; the ROW can

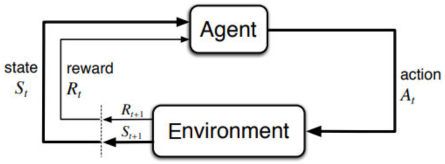


Fig. 1. Reinforcement Learning cycle [38].

be displayed with an onboard signalization system, or by acting on the actuators of a semi-autonomous vehicle. Traffic lights have the inherent limitation of giving the ROW to a whole lane, by individualizing the ROW per vehicle, the intersection manager gains more granularity in its control.

IV. DRL-CIM: Deep Reinforcement Learning for the Scheduling Policy of the Cooperative Intersection

A typical Reinforcement Learning (RL) system is a Markov Decision Process (MDP) $\{S, A, T, R\}$ (State, Action, Transition, Reward). To apply RL, a first step is to identify the properties of the system that can be used to design the corresponding MDP.

A. Principles of Value-Based Deep Reinforcement Learning

S is the state space, A the action space, T the transition function and R the reward function. The MDP can be divided into two parts (Figure 1):

- the *agent*, which performs an action $A_t \in A$ according to an observed state $S_t \in S$
- the *environment*, which updates itself according to the action of the agent, and returns a new state S_{t+1} and a corresponding reward $R_{t+1} \in \mathbb{R}$ to the agent

In the present situation, the decision process of the agent is driven by a deterministic policy $\pi : S \to A$ which tells the action to apply according to a given state. The purpose of the RL is to find the optimal policy π^* which maximizes the reward in the long time.

Thus, the first step to apply a RL algorithm to a given system is to define the state space S, the action space A, the transition function T, and the reward function R.

Thereafter, several approaches exist to find the optimal policy, mostly depending on the shape of the state space S and of the action space A. As detailed in Section IV-B, the state space can be seen as continuous while the action space is discrete, which allows the application of Deep Q-Learning (DQN) [39].

The DQN approach is a Deep RL method that was adapted from the Q-Learning algorithm [40]. The core idea of DQN is to use a neural network to approximate a Q-function giving the estimated value of actions for a given state. Using a neural network to approximate the Q-function allows working with a continuous state space. The approach gained a lot of interest thanks to its impressive versatility: the same algorithm could be applied to a wide range of Atari games without any adaptation, and had a performance superior to human players for some games. Yet, the proposed DQN was (at first) limited to Atari games, and eventually appeared to be inapplicable for several games (where it failed to outperform random policies).

The present study chooses to focus on this approach to demonstrate the feasibility of DRL for CIM, while keeping in mind that if DQN works, the improved DQN approaches should work too (notably Double DQN [41] or DQN with prioritized experience replay [42]).

The idea behind Q-Learning and therefore DQN is to determine a function $Q : S \times A \to \mathbb{R}$ which computes the "value" of an action in a given state. This function is determined through trial and error during an exploration phase, where random actions are taken for the encountered states. In DQN, this Q-function is approximated by a neural-network. The optimal parameters of the Q-function are determined according to Bellman equation (1) ($\gamma \in [0, 1]$ being the update factor and s' being the state following s after the action a). The learning of this function is made through offline learning, meaning that at a fixed frequency, the error between the estimated Q-function and the computed value is used for backpropagation, improving the estimation of the Q-function.

$$Q(s, a) \leftarrow r + \gamma \max_{a'} Q(s', a') \tag{1}$$

After enough experience, the policy $\pi(s) \to \arg\max_a Q(s, a)$ should take the actions maximizing the expected reward.

The definitions and necessary adaptations of DQN for the case of CIM are detailed in Section IV-B.

B. Markov Decision Process Design for Cooperative Intersection Management

The design of the MDP related to a given system directly influences the performance of the DRL algorithms which will exploit the proposed model.

For CIM, we consider that the agent (i.e. the controllable system) is the intersection manager, while the environment is composed of all the vehicles trying to cross the intersection. In the real CIM system, when a vehicle approaches the intersection, a request is sent to the intersection manager. The later one then decides to let the vehicle pass (or not). When the vehicle leaves the intersection, a notification is sent to the manager. Between the entrance and the exit, the vehicle is supposed to regularly send its position and velocity to the intersection manager. Depending on the chosen protocol, the ROW can be permanent or cancellable.

The CIM system is illustrated in Fig. 2 [43], with vehicles transmitting their relative position (d_e, d_{el}) to the intersection before approaching the local conflict zone to the intersection manager (following the communication protocol illustrated in Fig. 3). This one relies on an ordered passage sequence to distribute the ROW.

The definition of the **state** S must provide to the agent all the required information to efficiently compute the action A to apply. In the present paper, the state is represented as a picture-like discrete 2D grid where each cell can contain a vehicle, the tint of the cell being used to represent the route chosen by the vehicle, and its luminosity is used to represent

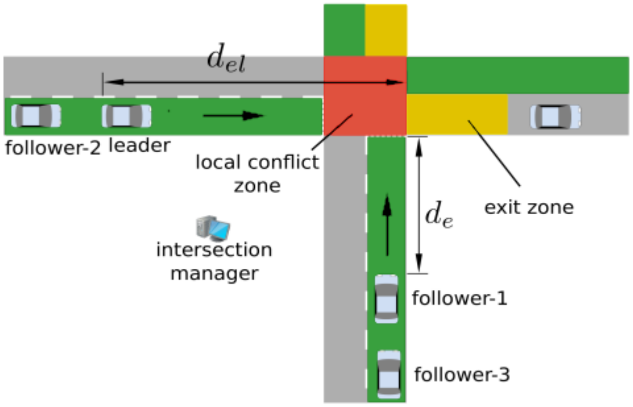


Fig. 2. Cooperative Intersection Management (CIM).

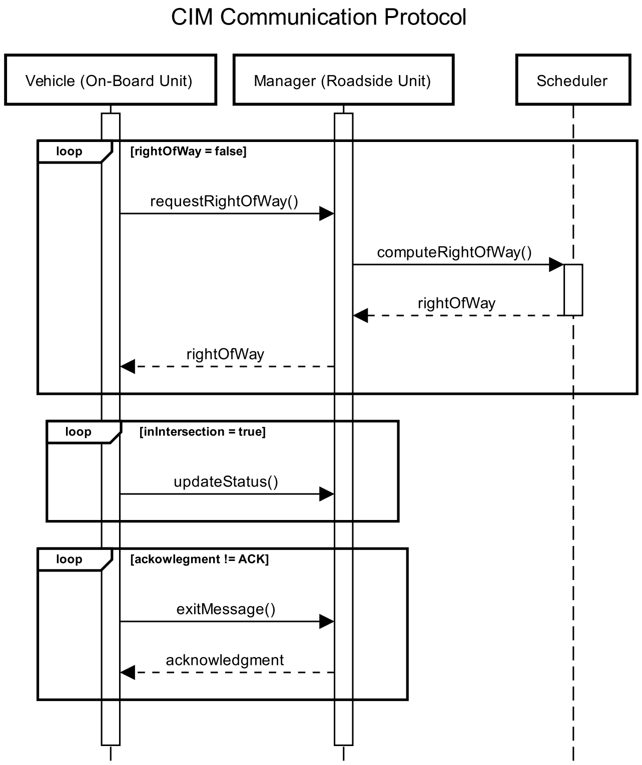


Fig. 3. UML sequence diagram of the communication protocol used for CIM: it describes the exchanged message of a vehicle approaching, crossing and leaving the intersection.

its speed (the higher the speed, the higher the luminosity). In a real-world scenario, all these data could easily be collected thanks to V2I communication. While it could look like that all the relevant features could be easily transmitted by the network, the choice is made to convert the data to a 2D picture representation:

- the state size is thus fixed, and not dependent on the number of vehicles in the intersection (this fixed shape is required by the neural network used by DQN), while a fixed-size vector padded with null values could have been used, it would have limited the number of vehicles considered;
- also, as the picture conveys data about the road geometry, it allows generalization to various road geometries and/or intersection topologies (without changing the state definition or the underlying neural network).

The result of the transformation from the continuous space state (from the simulation) to the discrete space is illustrated in Fig. 4. In the proposed implementation, the state represents an area of 100m per 100m reduced to 50 pixels per 50 pixels (1 pixel is equal to a square of 2m per 2m, and is lit on if the center of the vehicle is located within the corresponding area). The limited resolution of the grid allows considering the inaccuracy of the data provided by the vehicles' sensors.

The definition of possible **actions** A is used to define the possibilities of control available to the agent. The intersection manager should have the ability to control the ROW of all vehicles, i.e. tell if a vehicle can cross or not. The action space is inherently discrete, but the variable number of vehicles present in the network at a given instant is not compliant with the DRL approach, which requires a fixed size for the action. To overcome this limit, we decided to restrict the distribution of the ROW to a limited amount of vehicles: for each lane, the intersection manager only focuses on the two closest vehicles. Thus, for a given intersection whose incoming lane number l is known, we can easily deduce the size of the action space as any combination of 2 vehicles (2^2) for l lanes, i.e. 4^l. The choice to consider 2 vehicles instead of a single one offers the possibility for the intersection manager to build convoys of vehicles (limiting the phases of deceleration/acceleration). By default, for safety reasons, vehicles which are not in the first 2 vehicles won't have the ROW (and thus, will stop before the conflict area), they may gain it once one of their leader vehicles has exited the conflict area, if the intersection manager decides so.

To be noted, for safety reasons the ROW is definitive: when a vehicle has received the ROW it keeps it until it is able to leave the intersection. In a real case scenario, a cancellation of a ROW emitted by the intersection manager may not be received by the concerned vehicles, leading to a potentially hazardous situation.

For CIM, the **transition** function of the environment $T : (S, A) \rightarrow S$ is deterministic. Yet, the choice of the transition function is important as it can be observed that if the application of a transition has a small impact on the evolution of the state, it may create a large temporal distance (in terms of number of successive steps) between the application of an action and the consequent reward. This is why we have chosen to not use fixed time steps, which is usually the classical approach, and instead we have defined the transition function as follows: given a state $s_t \in S$ and an action $a_t \in A$, $T(s_t, a_t) = s_{t+1}$ where s_{t+1} is the state of environment following s_t after the occurrence of one of these events:

1) A new vehicle is approaching the intersection
2) All vehicles with the ROW crossed the intersection
3) No vehicle is allowed to cross the intersection and a fixed small amount of time Δt has passed (this avoids being stuck in a situation where no vehicle has the ROW)

To be noted that with this definition, the duration of a step in simulation time is variable (i.e. there is a variable number

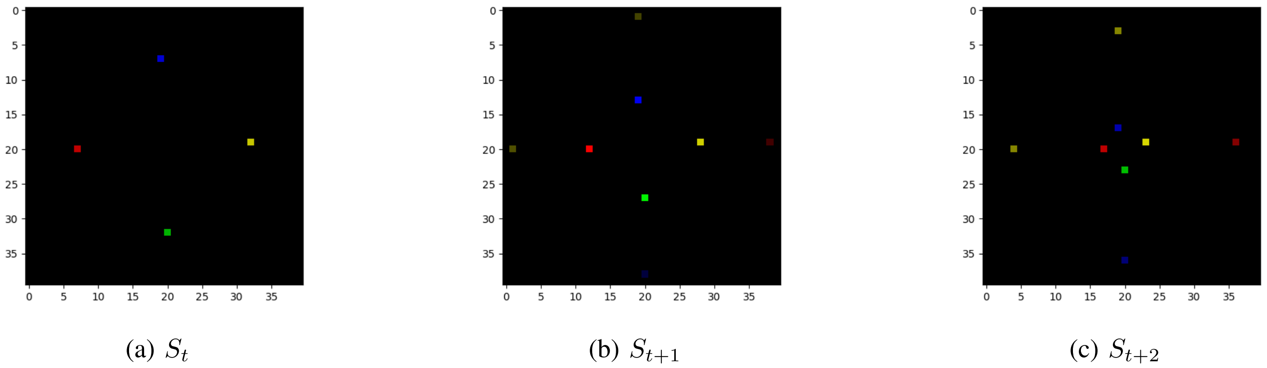


Fig. 4. Graphical visualization of three successive values for the state vector S. The colored pixels represent vehicles, the tint is the selected route and the luminosity matches the speed.

of *simulation steps* between two *MDP steps*). The last transition event is necessary to prevent the simulation from being blocked after an action that would prevent the occurrence of the first two transition events.

The definition of the **reward** function R is crucial, as it will implicitly define the criterion to optimize. To improve the overall traffic efficiency, the choice is made is to penalize the waiting time and to favor the evacuation of vehicles from the intersection. This leads to the function $r : (S, A) \rightarrow \mathbb{R}$ described by Equation (2) with $T_{avg\ waiting\ time}$ being the average waiting time of vehicles present at the intersection during the transition, n_{out} being the total number of vehicles having left the intersection during the transition. Given that several simulation steps can occur for a single MDP step (or *transition*), these values are averaged, i.e. they are summed for all simulation steps of the transition, then divided by the number of simulation steps that occurred during the transition.

$$r(s, a) = -100 \times T_{avg\ waiting\ time} + 10 \times n_{out} \quad (2)$$

The coefficients -100 and $+10$ of the two components of the reward are used to weight the influence of each component. The calibration of these values has been made empirically during the design of the system. Also, in our case, the prevention of collision is not taken into account in the definition of the reward, thus the DQN is not rewarded for preventing collisions. This part is discussed more in details in IV-C, where a safety management strategy is presented.

C. Safety Management

Some actions could potentially lead to collisions between vehicles, and the *randomness* of DRL based approach would let open the possibility of this kind of actions. This would clearly prevent the application of the system in a real-world scenario, as a system which could authorize simultaneously two vehicles with conflicting movements would be hazardous.

A so-called “safe mode” is then introduced. It acts as a filter for the application of actions: if an action would potentially lead to a collision, because it allows at least two vehicles with conflicting movements to cross simultaneously the intersection, then it is ignored. A $safe : (S, A) \rightarrow Boolean$ function is then introduced, whose value is $True$ if an action a_t is safe in state s_t, and $False$ otherwise. The vehicles do not have the ROW by default, and an action is required to provide it to a vehicle. Thus, if an action could lead to a collision, i.e. a vehicle is gaining the ROW while a conflicting vehicle already have it, ignoring the action will prevent the vehicle from getting the ROW and will make it to stop. Hence, ignoring the actions leads to an increase of the average waiting time and a decrease of the evacuated vehicles, causing a relative decrease of the reward which discourages the corresponding behavior.

The movements of two vehicles are considered as conflicting if both following conditions are met:

- the vehicles do not belong to the same lane
- there is an intersection between the trajectory of the two vehicles

The transition function T is modified to T_{safe} whose behavior is as follows:

$$T_{safe}(s_t, a_t) = \begin{cases} s_t \text{ if } safe(s_t, a_t) = False \\ T(s_t, a_t) \text{ otherwise} \end{cases} \quad (3)$$

D. Neural Network and Learning Hyperparameters

The model of the neural network and the hyperparameters of the learning strategy are critical for the success of the learning process (i.e. establishing an efficient policy over a limited amount of time).

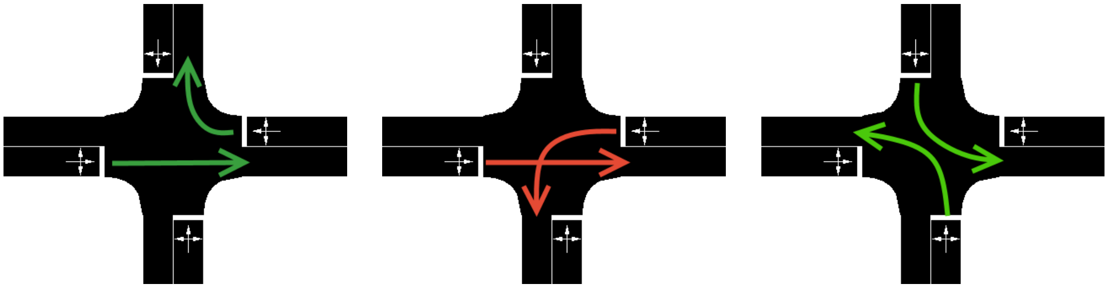
As the input (i.e. the state) is represented as a picture, a classical Convolutional Neural Network (CNN) is used to approximate the Q-function. The neural network used for the DRL is structured as detailed in Fig. 6.

The hyperparameters of the DQN algorithm are defined as described in Table I. The ϵ-random steps is the number of steps where random actions are taken, while ϵ-greedy steps is the number of steps where the actions are either chosen randomly or according to the learned policy (with a probability decreasing from 100% to 10% during the ϵ-greedy steps).

Due to the lack of an accurate deterministic methodology to define these parameters, the ones proposed here were obtained through trial-and-error.

V. Simulation and Results

To perform the learning and the evaluation, the SUMO simulation platform is used [45], as it can efficiently evaluate the



(a) Non-conflicting movements between two vehicles

(b) Conflicting movements: there is an intersection between the two trajectories

(c) Non-conflicting movements: the geometry of the road allow simultaneous opposite left-turns

Fig. 5. Illustration of conflicting and non-conflicting movements at intersection.

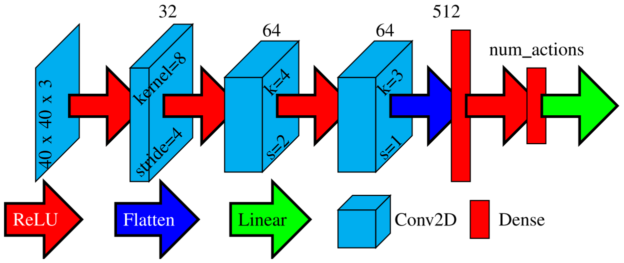


Fig. 6. Neural Network model used for approximating the Q-function of the DQN.

TABLE I

HYPERPARAMETERS FOR DQN

Hyperparameter	Value
ϵ-random steps	$50'000$
ϵ-greedy steps	$4'000'00$
Max memory length	$100'000$
Batch size	32
Frequency for Q network update	Every 4 steps
Frequency for target network update	Every 10000 steps
Gamma	0.99
Learning rate	$2.5e^{-4}$
Optimizer	Adam [44]
Loss function	Huber

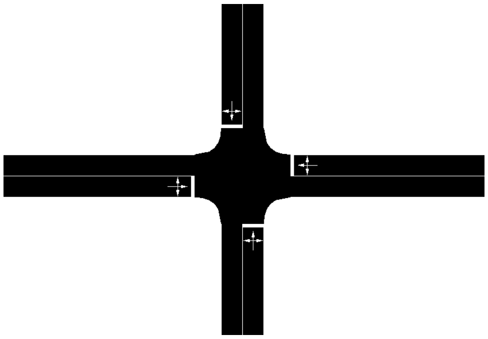


Fig. 7. The road network used for simulations, as displayed by SUMO-GUI: a 4-lane intersection with three possible movements for each lane.

TABLE II

VEHICLES CHARACTERISTICS

Attribute	Value
Maximum acceleration	$2m.s^{-2}$
Maximum deceleration	$-9m.s^{-2}$
Target velocity	$50km/h$
Length	$5m$
CO2 emission class	PC_G_EU4

performance of traffic management solutions and is commonly used for DRL related works on traffic management [46].

A. Simulation Environment

Fig. 7 presents the road network being used for evaluation. It is a classic symmetric 4-way intersection which will serve as a proof of concept.

The present road network is 200 meters wide (both along the north-south axis and the east-west axis). For each incoming lane, three movements are allowed (left-turn, right-turn or straight forward). Vehicles are spawned at the beginning of each lane, with a rate of d_{veh} between $100veh/hour$ and $600veh/hour$ (randomly chosen at the beginning of an episode). This value defines the probability of generation of a vehicle for a given amount of time, yet when a lane is full and cannot accept more vehicles, new vehicles will not be spawned. The simulation step length is fixed to $1s$ and a decision is taken by the intersection manager at every step.

The route of the vehicle is defined when it is spawned, and re-routing vehicles is not allowed. For a given vehicle, each one of the three available route has a fixed probability to be selected, these probabilities are constant for a given episode and are defined this way: between 10% and 33% of left-turn, and an equal distribution of forward and right-turn movements.

The attributes of the vehicles are given by Table II.

B. Experimentation and Measures

Four scenarios were designed in order to evaluate the potential benefits of DRL:

- In the first one, classic traffic lights are used.
- In the second one, CIM with a *manual* scheduling approach is used, as specified by [47]. The *First-Come First-Served* scheduling policy is used. This approach, not using DRL, is used as a reference to estimate the improvements offered by DRL.

- In the third one, CIM with another manual scheduling approach is used: Distributed Clearing Policy. This state-of-the-art approach is known to perform better than FCFS for dense traffic and is also used as a point of comparison. DCP is similar to FCFS except that if a vehicle is closely following another vehicle having the ROW (with a distance below 30 meters), it will also get the ROW even if it has reached the intersection after other vehicles from conflicting lanes, thus favoring convoy of vehicles.
- In the last one, a DQN approach is used. Following the model of the environment as an MDP described in section IV, a DQN is trained and used to control the intersection.

It can be noted that the classic traffic lights approach is used as a reference to illustrate the potential benefits of the proposed approach, but it is not at the core of the present contribution. Indeed, traffic lights are inherently limited by the fact they distribute the ROW to a whole lane instead of a single vehicle, unlike CIM with individualized ROW. Thus, for the sake of brevity, the choice is made to not compare different approaches of traffic light scheduling strategies (including DRL based ones), which would not be relevant in the present paper. Instead, we focus on comparisons between CIM based approaches with individualized ROW.

Regarding the measures, simulations are run for a period of $T_{ep} = 1000s$ (episode duration). During this period, the following measurements are performed using built-in SUMO functions:

- Average wait time (s) per vehicle
- Cumulative wait time (s)
- Number of vehicles evacuated (the total numbers of vehicles having left the intersection)
- Total CO_2 emission (g) (estimated according to [48], giving the emission in g/km as a function of speed considering a 0% road gradient and Euro IV gasoline powered passenger vehicles)

The measures are made regularly following a fixed time-step $\Delta_{stat} = 1s$ (independent of the transition function of MDP), then are averaged for the whole episode.

C. Learning Performance

In this section, we describe the evolution of the performance of the learning process.

The evolution of the reward with the episodes, as shown in Fig. 8, indicates that a policy is effectively learned to optimize the reward, with an initial value around $-8e-6$ and a final value around $-1e-4$. the average reward of the first 10% episodes is $-3.949e-5$ ($\sigma = 1.170e-5$) and the average reward of the last 10% is $-3.339e-4$ ($\sigma = 2.517e-4$), showing a relative convergence. Due to the stochastic nature of the environment (vehicles being spawned randomly with random routes), it cannot be expected to have a perfect convergence to a defined value.

In the same time, the retained traffic metrics are observed, and the results are presented in Fig. 9, 10, 11, and 12 (the moving average over 100 episodes is highlighted to ease the reading of the charts). The average waiting time and the number of evacuated vehicles are directly integrated in the reward used for the learning process, and the evolution of both measures show the effectiveness of the DQN to improve these metrics: $9.81s$ ($\sigma = 2.46$) → $1.15s$ ($\sigma = 0.7$) for the average waiting time, 273.8 ($\sigma = 37.9$) → 446.2 ($\sigma = 66.5$) for the number of evacuated vehicles.

Eventually, the observations show a reduction of the CO_2 emissions, even though this measure is not directly integrated in the reward. For the presented learning, this value goes from $8.93e4g$ ($\sigma = 1.68e4$) to $5.39e4g(\sigma = 2.19e4)$.

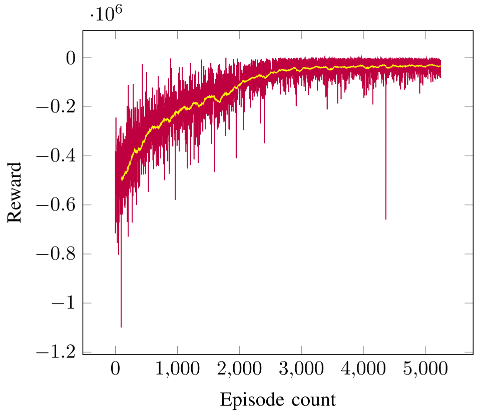


Fig. 8. Evolution of the reward as a function of episodes (the moving average over 100 episodes is highlighted).

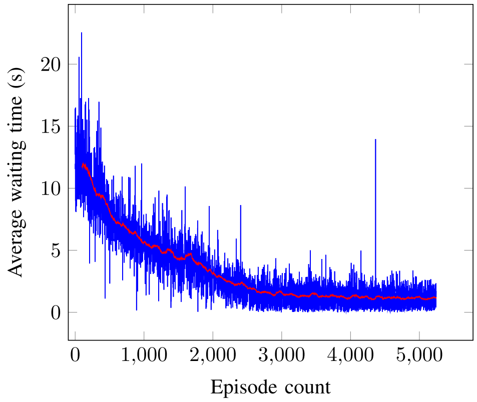


Fig. 9. Evolution of the average waiting time during the training process (moving average over 100 episodes).

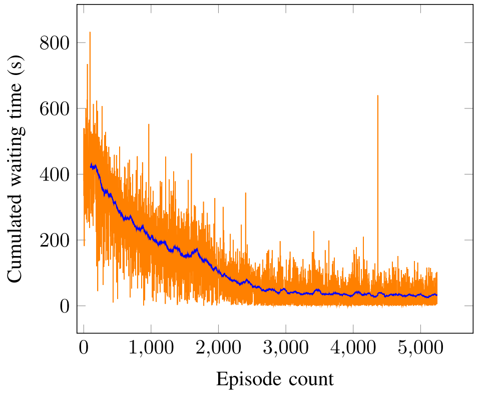


Fig. 10. Evolution of the cumulated waiting time during the training process (moving average over 100 episodes).

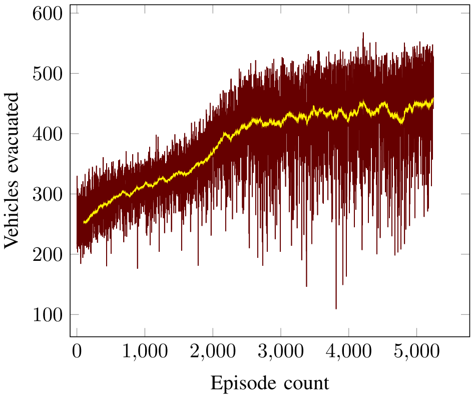


Fig. 11. Evolution of the number of evacuated vehicles during the training process (moving average over 100 episodes).

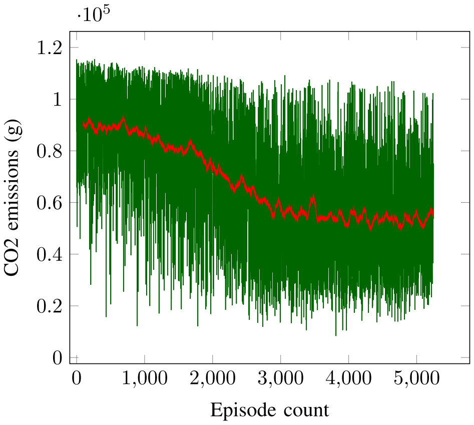


Fig. 12. Evolution of the CO2 emissions during the training process (moving average over 100 episodes).

D. Effectiveness Comparison

In this section, we evaluate the performance of the trained **DQN** model and compare it with a traffic light approach and two other CIM approaches. To sum up, the following strategies are compared:

- **Traffic lights** managed intersection
- Cooperative intersection based on the CVAS protocol with the **FCFS** scheduling policy
- Cooperative intersection based on the CVAS protocol with the **DCP** scheduling policy
- Cooperative intersection based on the CVAS protocol with a **DQN** scheduling policy

For traffic lights, the timing of the phases is adjusted using usual formulas according to the maximum expected flow. For FCFS, it can be noted that the vehicles cross the intersection according to their order of arrival, yet, if the vehicles have non-conflicting routes, they can cross the intersection simultaneously. DCP will try to build convoys of vehicles if they are following each other with a distance below 30 meters.

For each strategy, we compare the performance for different flows: from 100 vehicles per hour and per lane (vhl) to 600 vehicles per hour and per lane (as the performances can vary according to the flow). For each scenario, 20 simulations are performed, the results are then averaged.

TABLE III
BENCHMARK OF INTERSECTION SCHEDULING POLICIES

Strategy	Flow	Avg. waiting time (s)	Total waiting time (s)	Evacuated vehicles	CO2 emissions (g)
TL	100	18.28	102.82	108	12,580.7
FCFS	100	0	0	112	3,639.21
DCP	100	0	0	112	6,662.37
DQN	100	$6.79{\cdot}10^{-2}$	0.47	111.8	13,647.32
TL	200	17.11	213.36	214	26,623.78
FCFS	200	0	0	219.8	12,376.62
DCP	200	0	0	220	8,467.83
DQN	200	$2.59{\cdot}10^{-2}$	0.16	219.25	16,173.17
TL	300	16.47	338.6	315.35	42,934.14
FCFS	300	0.56	8.51	324.7	29,366.35
DCP	300	$5.43{\cdot}10^{-6}$	$4.35{\cdot}10^{-5}$	329.9	20,496.68
DQN	300	$8.61{\cdot}10^{-2}$	0.77	327.85	22,771.52
TL	400	20.27	732.1	386.8	69,521.16
FCFS	400	3.25	139.82	340.45	$1.03 \cdot 10^5$
DCP	400	$1.67{\cdot}10^{-2}$	0.2	438.6	26,188.07
DQN	400	0.2	2.35	436	31,332.84
TL	500	24.37	1,043.22	396.05	83,103.62
FCFS	500	3.52	159.86	338.9	$1.13 \cdot 10^5$
DCP	500	2.74	96.73	511.45	79,788.57
DQN	500	0.41	6.43	544.05	40,669.5
TL	600	25.45	1,123.25	405.1	86,260.2
FCFS	600	3.62	167.28	339.1	$1.15 \cdot 10^5$
DCP	600	4.92	221.85	514.9	$1.11 \cdot 10^5$
DQN	600	1.92	57.55	614.8	70,731.62

The different measures are summed up in Table III for the different flows and illustrated in Fig. 13, 14, 15, and 16. The different approaches (TL, FCFS, DQN) can be viewed in the following video: https://youtu.be/Z4EXQHwTMh4.

It can be observed that traffic lights are generally outperformed by FCFS, DCP and DQN, for reduced flows. Yet, above 400 vhl the FCFS approach is outperformed by traffic lights in terms of evacuated vehicles (Fig. 16). DCP, on the other hand, performs better in terms of evacuated vehicles than FCFS when the traffic increases, but is outperformed by DQN. The difference between FCFS, DCP and DQN is smaller for small flows (with a small - $<$ 2% - advantage for FCFS, in terms of evacuated vehicles, for a flow of 100 vhl), yet the advantage of DQN over the other strategies increases when the traffic flows are increasing, for all metrics.

Interestingly, the DQN approach shows a clear reduction of CO2 emissions (Fig. 15) compared to the two other approaches, even though this parameter was not considered during the training of the DQN. This can be explained by the reduced waiting times (Fig. 13).

To sum up the results in few words and numbers, FCFS, DCP and DQN are almost equivalent for small traffic flows ($\leq$ 300 vhl), but for a dense traffic (600veh), the average waiting time is reduced by 47.0% compared to FCFS and 61.0% compared to DCP; the number of evacuated vehicles is respectively increased by 81.3% and 19.4%; the CO2 emissions are respectively reduced by 38.7% and 36.9%.

VI. Discussion and Future Works

This paper presents a novel approach for managing intersections using inter-vehicular communications. We introduce the possibility to use Deep Reinforcement Learning to manage the process of scheduling the right-of-way of the vehicles. After a

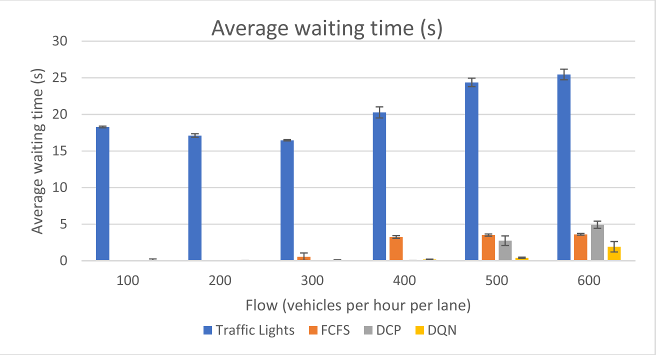


Fig. 13. Comparison of the average waiting time for several strategies and flows.

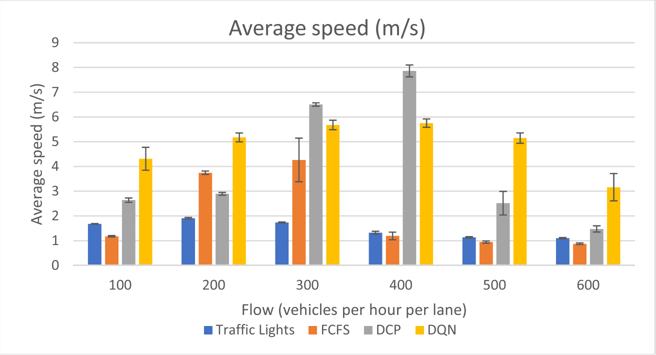


Fig. 14. Comparison of the average vehicles' speeds for several strategies and flows.

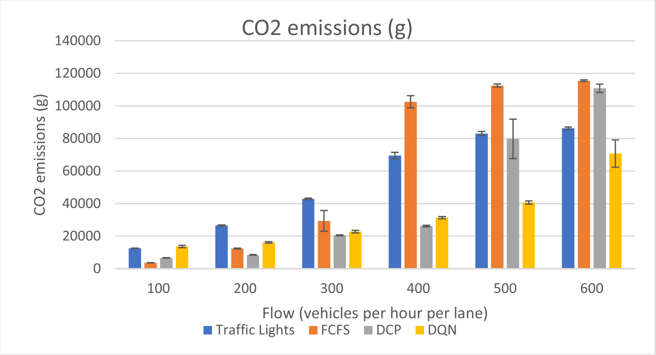


Fig. 15. Comparison of the average CO_2 emissions for several strategies and flows.

detailed presentation of the approach, defining the underlying Markov Decision Process and the specific adaptations required to be able to apply DRL approaches, the performances of the DQN approach are compared with the performances that could be expected from classic traffic lights, and with a first-come first-served cooperative intersection approach.

The present study shows that, for reduced traffic flows, the DQN approach does not offer a great advantage over the FCFS approach. Yet, as soon as the traffic flows are increasing, the FCFS approach is outperformed, in terms of waiting time, number of evacuated vehicles and CO2 emissions. The DQN approach also appears to be able to efficiently manage various traffic conditions.

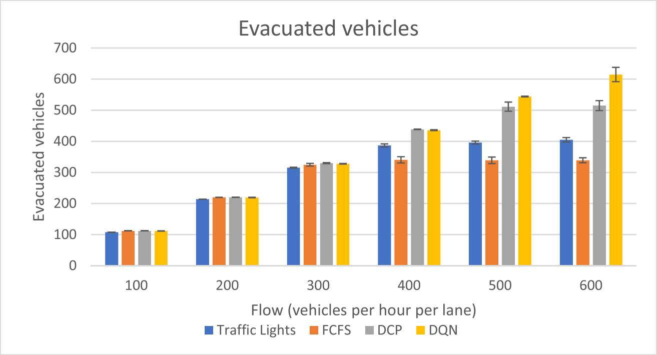


Fig. 16. Comparison of the average number of evacuated vehicles for several strategies and flows.

The ability of the proposed system to adapt to various traffic conditions is an improvement over traditional strategies: for instance, the calibration of traffic lights depends on the estimation of traffic flows, and a temporary change in these conditions (for example, in the case of road works), can greatly impact the performance of the intersection. With DQN, once trained in the simulation for a given intersection, the system does not need to be re-calibrated in case of a change in the traffic conditions.

Moreover, the proposed model for the cooperative intersection makes the DQN approach applicable to various types of intersections: even though the study is conducted on a 4-way intersection, a change in the geometry would just require a new training, while a change in the number of lanes would require an adjustment of the action space and a new training. Indeed, it can be noted that the geometry of the intersection does not have to be explicitly provided, all relevant data being encoded in the state vector. Yet, the increased size of the action space when the number of lanes is increased could affect the performance of the training process. More work is required to measure the impact of the number of lanes on the speed of convergence, a potential solution being the use of a DQN variant able to handle large action space [49].

To sum up, the DQN approach for CIM offers the following advantages, compared to traffic lights and FCFS AIM approaches:

- Improved performance in terms of throughput of the intersection
- Reduced CO2 emissions
- Ability to adapt to various traffic conditions (without any manual adjustments)
- Generalizable to various kinds of intersections (with a new training in simulation)

As DRL is showing interesting potential benefits for the scheduling of vehicles with CIM, an eventual perspective would be to apply other DRL models to the problem of scheduling's optimization of CIM. Moreover, as the vehicles are becoming more and more automated, it can be imagined to use the DRL to directly control the velocity of the vehicles approaching the intersection instead of the right-of-way. Further work is required to assess the feasibility of such an approach, and to evaluate the potential benefits.

Another interesting research direction would be feeding the DRL algorithm with predictions regarding the behavior of the vehicles [50] to ease the training process. Finally, the approach could eventually be applied to a network of intersections, but some work is required to limit the impact on the performance of the extended action space.

References

[1] SAE International, "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles," Soc. Automot. Eng., Warrendale, PA, USA, Tech. Rep. J3016_202104, 2021.

[2] Federal Highway Administration. (2019). *Intersection Safety*. [Online]. Available: https://cms7.fhwa.dot.gov/research/research-programs/safety/intersection-safety

[3] J. Wu, A. Abbas-Turki, A. Correia, and A. El Moudni, "Discrete intersection signal control," in *Proc. IEEE Int. Conf. Service Oper. Logistics, Informat.*, Aug. 2007, pp. 1–6.

[4] J. Wu, F. Perronnet, and A. Abbas-Turki, "Cooperative vehicle-actuator system: A sequence-based framework of cooperative intersections management," *IET Intell. Transp. Syst.*, vol. 8, no. 4, pp. 352–360, Jun. 2014.

[5] B. Chachuat, "Mixed-integer linear programming (MILP): Model formulation," McMaster Univ. Dept. Chem. Eng. Tech. Rep., Jul. 2019, vol. 17, pp. 1–26.

[6] S. Soleimaniamiri and X. Li, "Scheduling of heterogeneous connected automated vehicles at a general conflict area," in *Proc. Transp. Res. Board 98th Annu. Meeting Transp. Res. Board*, 2019.

[7] Z. Yao, H. Jiang, Y. Cheng, Y. Jiang, and B. Ran, "Integrated schedule and trajectory optimization for connected automated vehicles in a conflict zone," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 1841–1851, Mar. 2020.

[8] J. Wu, A. Abbas-Turki, and A. El Moudni, "Cooperative driving: An ant colony system for autonomous intersection management," *Appl. Intell.*, vol. 37, no. 2, pp. 207–222, 2012.

[9] F. Yan, M. Dridi, and A. E. Moudni, "Autonomous vehicle sequencing problem for a multi-intersection network: A genetic algorithm approach," in *Proc. Int. Conf. Adv. Logistics Transp.*, May 2013, pp. 215–220.

[10] T.-H. Nguyen and J. J. Jung, "Ant colony optimization-based traffic routing with intersection negotiation for connected vehicles," *Appl. Soft Comput.*, vol. 112, Nov. 2021, Art. no. 107828.

[11] L. Cruz-Piris, M. A. Lopez-Carmona, and I. Marsa-Maestre, "Automated optimization of intersections using a genetic algorithm," *IEEE Access*, vol. 7, pp. 15452–15468, 2019.

[12] J. Chodur and K. Ostrowski, "Assessment of traffic conditions at signalized intersections," *Arch. Transp.*, vol. 18, no. 2, pp. 5–24, 2006.

[13] F. Perronnet, A. Abbas-Turki, and A. El Moudni, "A sequenced-based protocol to manage autonomous vehicles at isolated intersections," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2013, pp. 1811–1816.

[14] V. Mnih et al., "Playing Atari with deep reinforcement learning," in *Proc. Conf. Neural Inf. Process. Syst.*, 2013, pp. 1–9.

[15] S. S. Mousavi, M. Schukat, and E. Howley, "Deep reinforcement learning: An overview," in *Proc. SAI Intell. Syst. Conf.* Cham, Switzerland: Springer, 2016, pp. 426–440.

[16] Z. Ding and H. Dong, "Challenges of reinforcement learning," in *Deep Reinforcement Learning*. Singapore: Springer, 2020, pp. 249–272.

[17] R. Naumann, R. Rasche, and J. Tacken, "Managing autonomous vehicles at intersections," *IEEE Intell. Syst. Appl.*, vol. 13, no. 3, pp. 82–86, May 1998.

[18] K. Dresner and P. Stone, "Multiagent traffic management: An improved intersection control mechanism," in *Proc. 4th Int. Joint Conf. Auto. Agents Multiagent Syst.*, Jul. 2005, pp. 530–537.

[19] J. Gregoire, S. Bonnabel, and A. De La Fortelle, "Optimal cooperative motion planning for vehicles at intersections," in *Proc. IEEE 4th Workshop Navigat., Accurate Positioning Mapping Intell. Vehicles*, 2013, pp. 1–6.

[20] A. Lombard, F. Perronnet, A. Abbas-Turki, A. El Moudni, and R. Bouyekhf, "V2X for vehicle speed synchronization at intersections," in *Proc. 22nd Intell. Transp. Syst. World Congr.*, 2015, pp. 255–262.

[21] E. Namazi, J. Li, and C. Lu, "Intelligent intersection management systems considering autonomous vehicles: A systematic literature review," *IEEE Access*, vol. 7, pp. 91946–91965, 2019.

[22] Y. Li and Q. Liu, "Intersection management for autonomous vehicles with vehicle-to-infrastructure communication," *PLoS ONE*, vol. 15, no. 7, Jul. 2020, Art. no. e0235644.

[23] M. Zhang, A. Abbas-Turki, A. Lombard, A. Koukam, and K.-H. Jo, "Autonomous vehicle with communicative driving for pedestrian crossing: Trajectory optimization," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–6.

[24] R. Chen, J. Hu, M. W. Levin, and D. Rey, "Stability-based analysis of autonomous intersection management with pedestrians," *Transp. Res. C, Emerg. Technol.*, vol. 114, pp. 463–483, May 2020.

[25] M. I.-C. Wang, C. H.-P. Wen, and H. J. Chao, "Roadrunner+: An autonomous intersection management cooperating with connected autonomous vehicles and pedestrians with spillback considered," *ACM Trans. Cyber-Phys. Syst.*, vol. 6, no. 1, pp. 1–29, Jan. 2022.

[26] M. Ahmane et al., "Modeling and controlling an isolated urban intersection based on cooperative vehicles," *Transp. Res. C, Emerg. Technol.*, vol. 28, pp. 44–62, Mar. 2013.

[27] J. Wu, F. Yan, and J. Liu, "Effectiveness proving and control of platoon-based vehicular cyber-physical systems," *IEEE Access*, vol. 6, pp. 21140–21151, 2018.

[28] G. F. Newell, "Properties of vehicle-actuated signals: I. One-way streets," *Transp. Sci.*, vol. 3, no. 1, pp. 30–52, Feb. 1969.

[29] X. B. Wang, K. Yin, and H. Liu, "Vehicle actuated signal performance under general traffic at an isolated intersection," *Transp. Res. C, Emerg. Technol.*, vol. 95, pp. 582–598, Oct. 2018.

[30] E. Van der Pol and F. A. Oliehoek, "Coordinated deep reinforcement learners for traffic light control," in *Proc. Learn., Inference Control Multi-Agent Syst. (NIPS)*, 2016, pp. 21–38.

[31] J. Zeng, J. Hu, and Y. Zhang, "Adaptive traffic signal control with deep recurrent Q-learning," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1215–1220.

[32] H. Wei, G. Zheng, H. Yao, and Z. Li, "IntelliLight: A reinforcement learning approach for intelligent traffic light control," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 2496–2505.

[33] H. Wei et al., "PressLight: Learning max pressure control to coordinate traffic signals in arterial network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 1290–1298.

[34] D. Isele, R. Rahimi, A. Cosgun, K. Subramanian, and K. Fujimura, "Navigating occluded intersections with autonomous vehicles using deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2034–2039.

[35] Y. Wu, H. Chen, and F. Zhu, "DCL-AIM: Decentralized coordination learning of autonomous intersection management for connected and automated vehicles," *Transp. Res. C, Emerg. Technol.*, vol. 103, pp. 246–260, Jun. 2019.

[36] M. W. Levin, H. Fritz, and S. D. Boyles, "On optimizing reservation-based intersection controls," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 3, pp. 505–515, Mar. 2017.

[37] W. Anwar, N. Franchi, and G. Fettweis, "Physical layer evaluation of V2X communications technologies: 5G NR-V2X, LTE-V2X, IEEE 802.11bd, and IEEE 802.11p," in *Proc. IEEE 90th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2019, pp. 1–7.

[38] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[39] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Feb. 2015.

[40] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, 1992.

[41] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 30, no. 1, pp. 1–7, 2016.

[42] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," in *Proc. 4th Int. Conf. Learn. Represent. (ICLR)*, San Juan, Puerto Rico, 2016, pp. 1–21.

[43] X. Hao, A. Abbas-Turki, F. Perronnet, and R. Bouyekhf, "V2I-based velocity synchronization at intersection," *Math. Methods Comput. Techn. Sci. Eng.*, vol. 37, no. 1, pp. 67–72, Nov. 2014.

[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, 2015.

[45] P. A. Lopez et al., "Microscopic traffic simulation using SUMO," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2575–2582. [Online]. Available: https://elib.dlr.de/124092/

[46] M. Khayatian et al., “A survey on intersection management of connected autonomous vehicles,” *ACM Trans. Cyber-Phys. Syst.*, vol. 4, no. 4, pp. 1–27, Oct. 2020.

[47] F. Perronnet, A. Abbas-Turki, J. Buisson, A. El Moudni, R. Zeo, and M. Ahmane, “Cooperative intersection management: Real implementation and feasibility study of a sequence based protocol for urban applications,” in *Proc. 15th Int. IEEE Conf. Intell. Transp. Syst.*, Sep. 2012, pp. 42–47.

[48] M. Keller et al., “Handbook emission factors for road transport 3.1 (HBEFA),” in *INFRAS*. Bern, Switzerland: INFRAS, 2010.

[49] Z. Zhao, Y. Liang, and X. Jin, “Handling large-scale action space in deep Q network,” in *Proc. Int. Conf. Artif. Intell. Big Data (ICAIBD)*, May 2018, pp. 93–96.

[50] H. Gao, Y. Qin, C. Hu, Y. Liu, and K. Li, “An interacting multiple model for trajectory prediction of intelligent vehicles in typical road traffic scenario,” *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Dec. 31, 2021, doi: 10.1109/TNNLS.2021.3136866.

Alexandre Lombard received the Ph.D. degree in computer science from the Université de Technologie de Belfort-Montbéliard (UTBM), France, in 2017. In 2019, he joined UTBM, as an Associate Professor. His work is mainly focused on the design of cooperation strategies of intelligent and connected vehicles in order to optimize the traffic efficiency at a microscopic and a macroscopic level. His research interests include intelligent transportation systems, cooperative traffic management and autonomous vehicles, as well as multiagent systems and reinforcement learning.

Ahmed Noubli is currently pursuing the degree in computer science with the Université de Technologie de Belfort-Montbéliard (UTBM), France. During his internship with the Laboratoire Connaissance et Intelligence Artificielle Distribuées (CIAD), he worked on collaborative intersections of vehicles, deep reinforcement learning, and multiagent-based simulations.

Abdeljalil Abbas-Turki received the Ph.D. degree in control and computer science from the Université de Technologie de Belfort-Montbéliard (UTBM), France, in 2003. He is currently an Associate Professor with CIAD-UBFC-UTBM. He is involved in many bus rapid transit projects and traffic flows design in industrial sites. He is also involved in developing cooperative intersection management of autonomous vehicles for encouraging new mobility systems. His research interests include discrete event dynamic systems, combinatorial optimization, control theory, and artificial intelligence applied to the traffic modeling and control. He was the Head of the X.icars demonstration at ITS WC 2015.

Nicolas Gaud received the Dr.Ing. and M.Sc. degrees and the Ph.D. degree in computer science from the Université de Technologie de Belfort-Montbéliard (UTBM), France, in 2007. He is currently a Senior Associate Professor with UTBM. He is involved in various industrial projects dealing with the simulation of virtual entities. His research interests include agent-oriented software engineering, holonic multiagent systems, and agent-based simulation.

Stéphane Galland received the Ph.D. degree in computer science from Mines Saint Étienne, France. He is currently a Full Professor with the Université de Technologie de Belfort-Montbéliard (UTBM), France. He is the Founder and the Deputy Director of the Distributed Artificial Intelligence and Knowledge Laboratory, Université Bourgogne Franche-Comté, France. He is the Head of the Smart Environment Research Group, UTBM. He has authored more than 80 research articles in international journals, conferences, and workshops. His research interests include agent-oriented software engineering and simulation, smart environments, and intelligent transport systems.