vehicles

MDPI

Article

Advantage Actor-Critic for Autonomous Intersection Management

John Ayeelyan [1], Guan-Hung Lee [1], Hsiu-Chun Hsu [2] and Pao-Ann Hsiung [1,*]

[1] Department of Computer Science and Information Engineering, National Chung Cheng University, Minxiong Township, Chiayi 621301, Taiwan
[2] Department of Information Management, National Chung Cheng University, Minxiong Township, Chiayi 621301, Taiwan
* Correspondence: pahsiung@cs.ccu.edu.tw

Abstract: With increasing urban population, there are more and more vehicles, causing traffic congestion. In order to solve this problem, the development of an efficient and fair intersection management system is an important issue. With the development of intelligent transportation systems, the computing efficiency of vehicles and vehicle-to-vehicle communications are becoming more advanced, which can be used to good advantage in developing smarter systems. As such, Autonomous Intersection Management (AIM) proposals have been widely discussed. This research proposes an intersection management system based on Advantage Actor-Critic (A2C) which is a type of reinforcement learning. This method can lead to a fair and efficient intersection resource allocation strategy being learned. In our proposed approach, we design a reward function and then use this reward function to encourage a fair allocation of intersection resources. The proposed approach uses a brake-safe control to ensure that autonomous moving vehicles travel safely. An experiment is performed using the SUMO simulator to simulate traffic at an isolated intersection, and the experimental performance is compared with Fast First Service (FFS) and GAMEOPT in terms of throughput, fairness, and maximum waiting time. The proposed approach increases fairness by 20% to 40%, and the maximum waiting time is reduced by 20% to 36% in high traffic flow. The inflow rates are increased, average waiting time is reduced, and throughput is increased.

Keywords: autonomous vehicles; intersection management system; reinforcement learning; fairness; traffic control

Citation: Ayeelyan, J.; Lee, G.-H.; Hsu, H.-C.; Hsiung, P.-A. Advantage Actor-Critic for Autonomous Intersection Management. *Vehicles* **2022**, *4*, 1391–1412. https://doi.org/10.3390/vehicles4040073

Academic Editor: Richard Romano

Received: 15 September 2022
Accepted: 6 December 2022
Published: 12 December 2022

1. Introduction

As the density of the urban population increases, the increasing number of vehicles causes traffic burdens, and how to cross road intersections more efficiently becomes an important problem that must be addressed. According to reports, traffic accidents are closely related to intersections [1]. Moreover, congestion is the most influential factor with respect to CO_2 emissions, which is the most important cause of greenhouse gas and air pollution from the transportation sector. According to recent reports [2], 2.5% and 10% of CO_2 emissions are due to traffic congestion and delays, respectively. Figure 1 shows the CO_2 emissions due to travel speed for different types of vehicle; it can be seen that lower speeds cause more emissions per kilometer. Advances in wireless networks are helpful in Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications. The IEEE has recently published standards on Wireless Access for Vehicle Environments (WAVE) [3,4] specifications for Dedicated Short-Range Communications (DSRC) technology. The biggest feature of DSRC is the low latency of its message transmission, which makes it suitable for vehicular environments that require real-time communication. The emergence of DSRC has created discussion and research in the area of autonomous intersection management (AIM).

With the help of wireless networks, better communication between V2V and V2I is contributing to the emergence of AIM, which is a new non-signalized approach to

intersection management that allows vehicles to cross intersections on the basis of design policy and without human intervention. Several studies have shown that AIM systems are superior to signal-based methods [3,5]; however, there are many improvements that remain to be made.

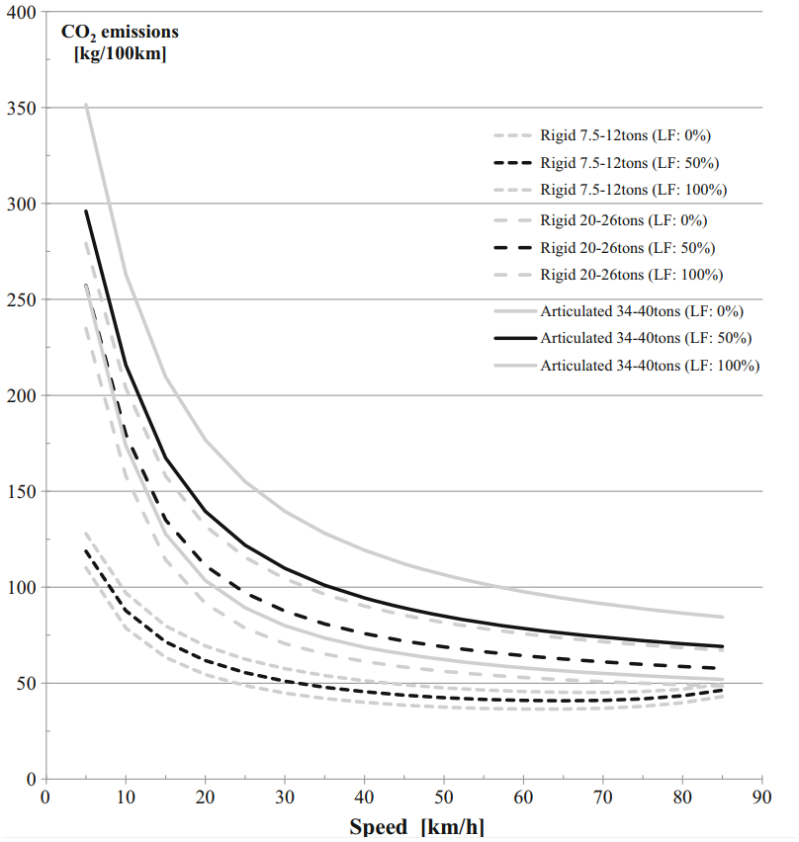

Figure 1. Vehicle speed and CO_2 emissions.

1.1. Background

Qian and Altché proposed a priority-based AIM system [6] to coordinate autonomous vehicles passing through ane intersection. An intersection controller assigns priority to incoming vehicles, and the vehicles send requests to controller to cross the intersection. When the controller receives a request, it decides whether to assign a priority according to the priority assignment policy. The vehicle must be assigned a priority for the intersection; prioritized vehicles passes through the intersection while maintaining a so-called brake safety status [6], by which vehicles respect the priority relations maximally through braking. Vehicles with lower priority brake to let higher priority vehicles pass if there exists any conflict in their respective paths.

In our research, we use reinforcement learning [7] to learn the priority policy. The aim is to let the system learn an effective priority assignment policy through interaction with the environment, allowing more vehicles to pass the intersection. Reinforcement learning is an effective machine learning technique that emphasizes how to perform an action effectively based on the environment by maximizing the needed benefits. Agents observe the environment, take actions, and are rewarded, and through this process develop a policy to select actions. The aim is to obtain the maximum cumulative reward. Reinforcement learning actively focuses on immediate planning, and seeks a balance between current knowledge (exploitation) and uncharted territory (exploration).

1.2. Motivation

Intersection management policy is an important issue in transportation. Various systems for autonomous intersection management have been proposed [3,6]. Although

these methods have good performance compared with signal-based approach, there remain many issues to discuss. Kamal et al. [3] proposed a predictive control model to coordinate automatic vehicles at non-signalized intersections in order to increase intersection capacity and avoid collisions. In this scheme, the state of a vehicle approaching the intersection is the considered framework, and the intersection area increases the vehicle's optimized trajectory. The vehicle may apply aggressive braking and acceleration in order to avoid a collision. However, such braking and aggressive acceleration increases fuel consumption, and may make the user uncomfortable. Qian et al. [6] proposed a priority and brake-safe control-based algorithm in which the intersection controller assigns priority to a vehicle, which then crosses the intersection while maintaining brake-safe distance from prioritized vehicles with respect to state. In the experimental results, their proposed method showed improved average time lost (ATL) and average queue length (AQL) compared to signal-based approaches with restricted traffic flow. However, with higher traffic volumes this performance was found to deteriorate. On the other hand, because the proposed policy attempts to maximize throughput, it leads to unfairness; in other words, vehicles may wait for a long time while others cross the intersection directly. Experimental results show that a vehicle may need to wait as long as 123 s before crossing an intersection, while other vehicles may pass directly through intersection without waiting. The average waiting time is around 30 s. Jain's fairness [8] index is used to evaluate fairness; the full range is from 0 to 1, and 1 means the system is 100% fair. This method, however, only has a fairness index value of 0.3.

In this paper, our object is to propose an AIM architecture based on reinforcement learning. This architecture has good control policy learning ability and can learn a good policy while the traffic flow changes from low to high. In addition, we aim to develop a good reward function that can enable the system to learn a fair and efficient intersection management policy, allowing vehicles can cross and move through the intersection with lower delay disparity.

The novelty of the proposed work lies in increasing the reward function used to encourage a more fair allocation of intersection resources, ensuring brake-safe control to ensure that autonomous moving vehicles are safe during travel, and reducing the wait time in high traffic flow. Our experimental results prove that our method increases fairness by 20% to 40% and reduces waiting time by 20% to 36% in high traffic flows compared to previous research methods.

The rest of this paper is organized as follows. In Section 2, we introduce works related to AIM and reinforcement learning. Section 3 provides the definitions and parameter settings used in our method along with assumptions made during our work. Section 4 presents the overall framework and details of our proposed intersection model. Our experiment results are explained and compared with other research works in Section 5. Finally, in Section 6, we provide conclusions and propose possible future works in this research area.

2. Related Work

In this section, we present previous approaches that address issues around the design of autonomous intersection management and reinforcement learning in traffic control, such as how to design the state space, action space, and reward function.

2.1. Autonomous Intersection Management

Autonomous Intersection Management (AIM) is a non-signal based intersection control system dedicated to improving traffic safety and efficiency. With the improvement of vehicle communication and autonomous driving technology, vehicles can share their current intention and state with the intersection facilities in order to further determine how to distribute intersection resources. In the field of AIM research, there are two primary methods, namely, the resource reservation method and the trajectory planning method. Cascetta et al. [9] described the functional design of a transportation supply system, includ-

ing such different theories and methods as transportation supply models, transportation demand models, traffic model assignments, demand travel flow, and dynamic models for transportation systems. Aaron Parks-Young and Guni Sharon [10] proposed a protocol for intersection management using mixed autonomous operated vehicles. The main objective of their work was increase safety in uncertain situations. In a related work on concentrated autonomous control policy for efficient autonomous intersection management, Bindzar et al. [11] proposed universal simulation tools for managing urban intersections. The main aim of their work was to reduce traffic congestion and adjust the signal plan for monitoring. Li et al. [12] proposed a model based on reinforcement learning for autonomous intersection management. Musolino et al. [13] proposed mobility as a service (MaaS) for an integrated transport system. MaaS restructures single or multiple transport models to integrate mobility operators' services into a single service. This work used dynamic strategies with various feedback rewards and time spans to avoid conjecture at multiple levels. Other major related works are described below.

2.2. Resource Reservation Methods

As shown in Figure 2, AIM was first proposed by Dresner et al. [5] as a centralized method using an intersection modeled as tiles. Tiles on a vehicle path are reserved for a period of time in order to ensure that the vehicle can safely pass the intersection without colliding. Two agents are introduced in this approach, namely, a vehicle agent (VA) to control the vehicle and an intersection reservation agent (RA) that resides at the intersection. When a vehicle approaches an intersection, the VA sends a reservation request to the RA, which includes information about the vehicle such as its time of arrival, velocity, position, vehicle dynamics, etc., in order to reserve tiles on its planned trajectory. The principle policy proposed in [5] is based on FCFS, which serves the earliest-arriving vehicle first. Schepperle et al. [14] introduced an auction-based initial time slot auction (ITSA) instead of FCFS; in this approach, the vehicle with the highest bid passes the intersection first. The main idea of this method is that there are different waiting times for each driver.

In addition to centralized reservation management, decentralized reservation management without intersection controllers has been considered. In [15,16], Naumann et al. proposed a collision region-based distributed reservation scheme. They introduced the token reservation and occupation concept, in which each token is associated with collision regions; one vehicle holds a token and occupies that region at a certain time. When the vehicle acquires the token, it broadcasts the occupancy information while crossing the related regions. Simultaneously, other vehicles continue to listen and detect token availability in order to avoid conflicts. VanMiddlesworth et al. [17] discussed similar concepts and introduced two types of messages, CLAIM and CANCEL, which are sent by vehicles. A vehicle constantly listens for CLAIM messages from other cars as it approaches the intersection and compares them with their planned trajectories. If there is no conflict, it broadcast the CLAIM message of the corresponding tile on its planned trajectory to other cars. This method allows CLAIM messages to dominate each other; for example, if two CLAIMs conflict with each other, the lower priority vehicle must relinquish its reservation to the higher priority one. Unless there is a higher priority CLAIM message, the vehicle is granted the right to reserve the tiles and can begin to cross the intersection. When the vehicle passes the intersection, it issues a CANCEL message to release the reserved tiles. Experimental results show that delay was significantly reduced compared to traffic light and four-way stop management. The authors of [16] presented a brief survey of intersection management techniques for connected vehicle and discussed different communication infrastructures for autonomous intersections. Their work summarizes signals, controls, and safety management for autonomous vehicles.

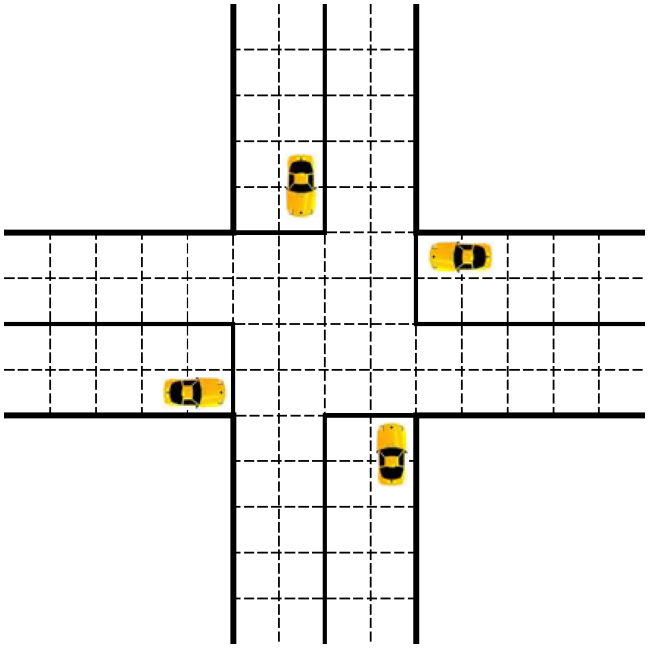

Figure 2. A four-way intersection discretized into a grid of tiles.

2.3. Trajectory Planning Methods

Trajectory planning is another important requirement for AIM. While resource reservation methods manage intersections by considering them through the scheduling of space tiles and time slots, the trajectory planning method is used for various moving objects in a conflict-free way. In [18], Grégoire et al. proposed a mathematical framework based on path–velocity decomposition and adopted priority to relative intersection passing vehicles between for path identification. A vehicle that wants to cross an intersection needs to send a request to the intersection controller in order to gain priority, and only vehicles with priority can pass. Then, a priority flow graph is constructed, including the relationships between the involved vehicles, and an algorithm is used to construct the optimal trajectory for the given priorities. While simulations confirm the safety effect, this approach may cause deadlocks at high flow densities. Moreover, it assumes that vehicles follow the control plan, and does not consider control uncertainty. In [19], the same authors revisited the priority-based robot motion planning method, presenting a control policy called brake-safe control. By introducing a feedback control law to ensure that all vehicles follow the priority relationship while crossing the intersection, collisions caused by mechanical failure, accidental control, and other uncertainties can be avoided. In [6], Qian et al. adapted [19] in presenting their policy called Fast First Service. In [20], the authors proposed a contradictory criteria and five pairs of criteria considered for designing trajectory planning systems. The same authors [21] proposed a manipulator trajectory method for work spaces using a kinematic model, and provided the solution for the independent area and overlapping areas, thereby avoiding the trajectory misalignment problem.

2.4. Reinforcement Learning for Traffic Control

Intersection signals using reinforcement learning control have been widely discussed in the literature [1,22–24]. The approaches in these studies differ in terms of the traffic network model and state definitions required for reinforcement learning. Here, we focus on discussing three important definitions which are important for reinforcement learning, namely state, action, and reward. Although the reinforcement learning studies presented here are used for traffic signal control, there are many other ideas that can be considered. The state, defined as $s_t^i \in S^i = \{s_1^i, s_2^i, s_3^i, \ldots, s_{|S^i|}^i\}$, is used to describe the state of intersection i at time t, which are an agent's decision-making factors. A state s_t^i can be represented by two main approach, the queue size and the position of the waiting vehicle. The action

definition can be stated as $a_t^i \in A^i = \left\{a_1^i, a_2^i, a_3^i, \ldots, a_{|A|}^i\right\}$, and is the action set that can be selected by an agent to change the traffic environment. An action a_t^i can be represented by two main approach, the traffic phase and traffic phase split. The traffic phase represents the selection of combinations of the traffic path which do not have conflicts and allow the lanes to receive a green light. As per Figure 3, the traffic phase ensures traffic safety and avoids traffic congestion. The authors of [25] designed different signal phases, as shown in Table 1. Zhang et al. [26] proposed a reinforcement learning approach to weakly control traffic systems, and used this to achieve optimal cooperation learning strategies.

Table 1. Phase Scheme [25].

Phases -I EWG	**Phases -II EWY**	**Phases -III EWLG**	**Phases -IV EWLY**
turn right from west or east	Wait	Turn left west or east	Wait
Phases -V SNG	Phases -VI SNY	Phases -VII SNLG	Phases -VIII SNLY
turn right from north or south	Wait	Turn left from north or south	Wait

The traffic phase can be used to split the representation into time intervals, which can then be assigned to a traffic phase at an intersection. The two methods are described below. First, an action consists of combinations of green timing of each phase, where the green timing is the time interval that allows vehicles to cross an intersection. The green timing consists of the sequence of a traffic phase, with the cycle time based on the traffic demand. Second, actions can be either (1) switching to a different traffic phase or (2) remaining in the current traffic phase [27]. This definition leads to the agent always choosing to remain in the current traffic phase unless executing this action does not receive the best reward, then changing to another traffic phase [28].

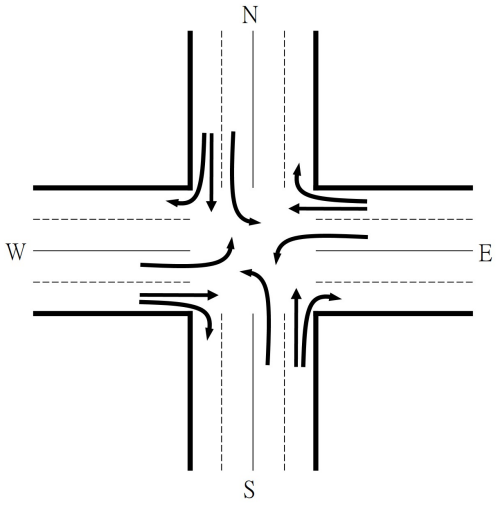

Figure 3. Single intersection model.

The reward definition, represented as reward r, can be a variable or a constant (e.g., $r = 1$ expresses a prize and $r = -1$ expresses a punishment [29]). As a variable, it can be a combination of different elements. There are three main methods for calculating variable reward values: first, a reward $r = r_1 + r_2$ which has two elements r_1 and r_2 with equal weights [30]; second, a reward with weight $r = [\epsilon_1 \times (r_1^i + r_2^i)] + [(1 - \epsilon_1) \times (r_1^j + r_2^j)]$, where $0 \leq \epsilon_1 \leq 1$ represents different priority levels between two elements (in this case, if $\epsilon_1 \leq 0.5$, then lane i has higher priority than lane j); and third, a reward with multiple

elements and weights, where the weights represent different priority levels for different elements [30]. There are four main definitions of rewards, such as variation of vehicular delay, waiting time, appropriateness of green time [22,23,29,31], and variation of queue size [24,29,32]. The authors of [33] provide a summary and brief details about reinforcement learning dynamic control, its environment, and its goals.

2.5. Intersection Throughput and Fairness

Although increasing intersection throughput is an important issue, its fairness needs to be addressed as well. In [34], Pasin et al. discussed the tradeoff between throughput and fairness in intersection control. They proposed two algorithms for intersection control issues, with a focus on the tradeoffs between fairness and throughput. They considered various traffic conditions in a single-intersection scenario and evaluated two platoon-based algorithms, namely, Longest Queue First (LQF) and Efficient Intersection Control (EIC). In their experiments, the platoon-based algorithms performed significantly better in throughput, while only EIC achieved good fairness. In [35], Wu et al. introduced a delay-based mechanism for traffic light management in transportation networks; they dealt with excessive delay and achieving better fairness using a queue control mechanism. The delay-mechanism traffic system isolated intersections and improved the throughput effectively. The advantage of a delay control mechanism is in dealing with excessive delay under different traffic patterns. Simulation results show that delay-based systems produce better fairness in terms of delay results, and that this improvement is more obvious under heterogeneous traffic conditions. In [8], Jain et al. proposed a formula called the fairness index for quantifying equality. The proposed formula applies to any system related to resource allocation. The authors provide a number of examples from different areas to illustrate applications of their fairness index in different situations. The fairness index ranges between 1 and 0, meaning it can be expressed as a percentage to continuously allocate changes in fairness. In this paper, the Jain fairness index is used to compare the proposed method with existing AIM methods. Victor Manuel Madrigal Arteaga proposed [36] an efficient adaptive intersection management approach using fuzzy logic. In this work, real time data were used for implementation and the effective flow rate was calculated.

The authors of [37] proposed a scheduling-based method for autonomous intersection control (AIC) for autonomous vehicles. In their approach, the intersection controller computes the inflow based on FCFS order and an optimization heuristic. Their work was simulated using MATLAB, and performance was evaluated using traffic flow, utilization (throughput) of the intersection, and delay in terms of seconds. The authors of [38] proposed an AIM system called Roadrunner+ that cooperates with continuous vehicles. The proposed work integrates dynamic lanes into AIM with roadside units. The proposed work was simulated using SUMO with different parameters, such as throughput and delay. Their proposed approach was able to attain 15.16% better throughput than other traditional methods. The authors of [39] proposed an AIC with global scheduling to protect autonomous vehicles from collisions. The particle swarm algorithm was used for optimization, and performance was compared using fairness, efficiency, delay, and throughput. The PSO-based approach achieved better throughput, fairness, and robustness. The authors of [40] proposed a hybrid approach for controlling dynamic intersections, which they called GAMEOPT. This method uses game theory and optimization methods to control cooperative intersections dynamically. This work included 10,000 vehicles, and used the SUMO simulator for simulations. Throughput, fairness, efficiency, and safety were the parameters used for performance evaluation.

3. Autonomous Intersection Management

In this section, our method, which we call Advantage Actor-Critic Autonomous Intersection Management (A2CAIM), is introduced in detail. Figure 4 shows the basic framework of our proposed mA2CAIM method in two parts: the vehicle intersection

controller and the brake-safe model. The intersection controller is responsible for managing the priority of vehicles, ensuring that vehicles with higher priorities can pass the intersection preferentially.

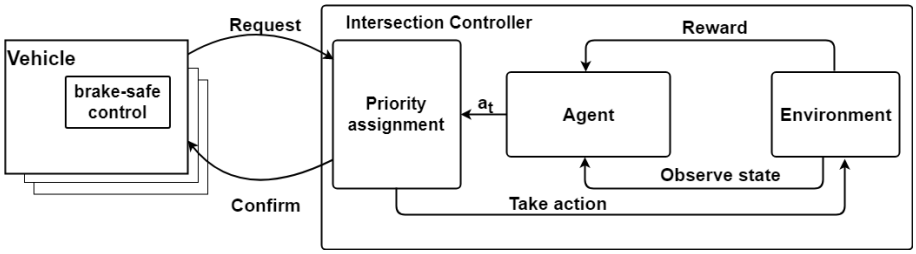

Figure 4. Deep Reinforcement Learning-based AIM architecture.

The intersection controller consists of the priority assignment model, agent, and environment, which are described below.

- Priority Assignment Model
  The model maintains a priority list and waiting list for assigned and unassigned vehicle. It receives requests from vehicles and stores then in the waiting list. At each time step, the priority assignment model assigns vehicles a priority in certain lanes, as determined by the agent, then adds the vehicles to the priority list. Generally, the different types of applicable intersection models are crossroad, roundabout, misaligned intersection, ramp merge, deformed intersection, X-intersection, T-intersection, and Y-intersection.
- Agent
  The agent is responsible for carrying out the priority assignment policy by selecting vehicles to cross the intersection. It continually revises the policy based on previous experience. At every timestamp, the agent collects the state information s_t from the environment and chooses vehicles to pass. The agent sends actions to the priority assignment model. After priority assignment, the action is executed and the vehicles which are allowed to pass receive the priority needed to cross the intersection. The state then changes to s_{t+1} and the agent receives a reward r_{t+1}. The main object of the agent used in this study is to develop an effective and optimal policy in order to increase its cumulative reward.
- Environment
  The environment refers to the information obtained from monitoring the traffic environment; it consists of the size of the queue, vehicle waiting times, and number of vehicles inside the intersection.

The break-safe model for vehicles [6] is described in Section 4.5. This model guarantees that vehicles can pass the intersection without any collisions occurring. When a vehicle enters the cooperative zone, it sends a request to cross the intersection, and only when it receives the confirmation message and priority list from the intersection controller is it able to pass the intersection. When the vehicle crosses the intersection, it must follow the priority in the list, that is, when the trajectories of two vehicles collide, the lower-priority vehicle must brake to let the higher-priority one pass.

4. Advantage Actor-Critic for Autonomous Intersection Management

In this section, we introduce the A2C model for AIM, which contains the state space, action space, reward, and learning algorithm.

4.1. State Space

In the design of the state space, it is necessary to consider the kind of design that can lead to a full description of the current state of the traffic environment. First, it is necessary to define the condition of each lane; here, $s_t^Q = \{s_t^{q,1}, \ldots, s_t^{q,n}\}$ is the queue length of

all the lanes 1 to n at time t. We then define the average wait time for the vehicles in all lanes $s_t^W = \{s_t^{w,1}, \ldots, s_t^{w,n}\}$. To obtain a comprehensive understanding of the traffic, we then use the number of vehicles inside the intersection to represent the situation at time t, that is, s_t^I. The state is now defined as the combination of the above-mentioned elements: $s_t = \{s_t^Q, s_t^W, s_t^I\}$.

4.2. Action Space

At each time step, the agent observes the state and chooses an action to interact with the environment. Compared to the different types of lane approaches, this lane approach action design allows vehicles that have sent requests to cross the intersection to be chosen. In this work, we consider five types of actions: (1) allow vehicles from the west or east to move straight or turn right, (2) allow vehicles from the west or east to turn left, (3) allow vehicles from the north or south to move straight or turn right, (4) allow vehicles from the north or south to turn left, and (5) not allow any vehicles to move. This design principle is intended to let the vehicles in different lanes cross the intersection without conflict. The intuition behind the design is to increase throughput by preventing vehicles from braking or waiting too frequently for other cars due to constant switching of lane permissions.

4.3. Reward

The reward function is described as follows:

$$R_{t+1} = \begin{cases} \alpha n_{t+1} - max_i w_{t+1}^i + \beta, & if\ max_i w_{t+1}^i < max_i w_t^i \\ \alpha n_{t+1} - max_i w_{t+1}^i, & otherwise \end{cases} \tag{1}$$

where n_{t+1} is the change in throughput between time t and $t+1$, $max_i w_{t+1}^i$ is the maximum average time at i and increase at $t+1$, and α, β are constants. The term n_{t+1} is used to enhance the efficiency of intersection crossing, and the term $max_i w_{t+1}^i$ is used to avoid starvation and unfairness. Note that the units of n_{t+1} and $max_i w_{t+1}^i$ are different; thus, the reward is a linear combination of n_{t+1} and $max_i w_{t+1}^i$. Moreover, in order to encourage agents to choose the vehicle that has been waiting for the longest time, we provide an additional reward of β if $max_i w_{t+1}^i < max_i w_t^i$, meaning that the vehicle with the longest wait time receives higher priority to cross the intersection. This approach can improve fairness by considering $max_i w_{t+1}^i$ through the additional reward β.

4.4. Learning

Next, we describe how Advantage Actor-Critic(A2C) learns. Actor-Critic contains two neural networks. The Actor is a strategy function $\pi(s)$ that outputs a set of action probabilities according to the state. The Critic function $V(s)$ is used to estimate the state. The Actor updates the strategy according to the value from the Critic; on the other hand, the Critic learns how to evaluate it more accurately based on the reward. The value function $V(s)$ represents the policy and π is the expected discounted, which is defined in [41] as follows:

$$V(s) = E_{\pi(s)}[r + \gamma V(s')] \tag{2}$$

Essentially, we weight-average the term $r + \gamma V(s')$ as possible action state s, where s' is the next state of s, r denotes the reward, and γ denotes the discount factor. The action value $Q(a, s)$ is defined as

$$Q(a, s) = \gamma V(s') + r \tag{3}$$

the action value means that after taking action a at state s, the reward $r + \gamma V(s')$ is obtained. Now, we define a new function, namely, the advantage function $A(s, a)$, as follows:

$$A(a, s) = Q(a, s) - V(s) = \gamma V(s') - V(s) + r \tag{4}$$

the policy gradient used to optimize the effective policy is π. The object function is defined as

$$J(\pi) = E_{\rho^{s_0}}[V(s_0)] \tag{5}$$

the gradient function tells us how good policy π is, where ρ is the distribution of the environment. We can then obtain the gradient of the $J(\pi)$ function using the method proposed in [42]:

$$\nabla_\theta J(\pi) = E_{s\sim\rho^\pi,\, a\sim\pi(s)}[A(s,a) \cdot \nabla_\theta \, log \, \pi(a|s)] \tag{6}$$

this formula is intuitively explained. It tells us that function $J(\pi)$ changes the direction of the weight of the neural network. On the right side of the formula, if the advantage function $A(s,a)$ is positive at state s, meaning that the value obtained by selecting a at s is higher than the average, then adjusting θ means that the probability of the agent selecting action a at state s is increased. As we are trying to maximize the function $J(\pi)$, we can say that the loss function is

$$L_\pi = -J(\pi) \tag{7}$$

then, we can rewrite the function $J(\pi)$ by treating $A(s,a)$ as a constant:

$$J(\pi) = E[A(s,a) \cdot log \, \pi(a|s)] \tag{8}$$

finally, we swap the expectation for average over all samples in a batch, meaning that the final loss function is then

$$L_\pi = -\frac{1}{n}\sum_{i=1}^{n} A(s_i,a_i) \cdot log \, \pi(a_i|s_i) \tag{9}$$

next, we define the value loss. In the n-step return scenario, the true function $V(s)$ should be as follows:

$$V(s_0) = r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots + \gamma^{n-1} r_{n-1} + \gamma^n V(s_n) \tag{10}$$

the value function $V(s)$, which we want to approximate, should converge to Equation (10), and we can write the error as

$$e = r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots + \gamma^{n-1} r_{n-1} + \gamma^n V(s_n) - V(s_0) \tag{11}$$

then, we can define the loss of the value function L_V as the mean squared error of all given samples, as follows:

$$L_V = \frac{1}{n}\sum_{i=1}^{n} e_i^2 \tag{12}$$

The module structure is shown in Figure 5. Actor and Critic are two neural networks, the input is the state s, and the output is the probability of each action $\pi(s)$ and a scalar $V(s)$, respectively. The neural used here consists of three hidden layers, and each layer includes 64 neurons. RELU was selected for use as the activation function in the model. The working process is presented in Algorithm 1 and Figure 6. In Figure 6, a_t is an action at time t, s_t is the state at time t, t is the time step, and T is the length of the simulation.

First, the agent copies the global parameters and observes the state. Then, it initiates training, interacts with the environment, and computes the gradient according to the data. Finally, it updates the gradient to the global model. Due to countinous updates to the global model, our proposed system can connect autonomous interconnection management and mobility as a service (maaS). The main components of MaaS are environmental learning, decision-making, and system planning. In our proposed approach, autonomous environment agent learning and rewards help to make for more effective decision-making. Using environmental learning and decision-making can aid in better and more autonomous management.

Algorithm 1: The A2C-based Intersection Management Algorithm.

Input:
N: Number of simulations;
sim_len: The length of simulation time step;
ϵ: A user-defined value that $0 < \epsilon < 1$;
Variable:
γ: discount factor;
a_t: An action at time t;
s_t: A state at time t;
r_{t+1}: a reward received after executing a_t;
episode: One of the simulations;
t: The time step;
e: a variable contains (s_t, a_t, r_{t+1});
t_{update}: the frequency of update neuronal net work;
θ: the parameter vector of π;
θ_v: the parameter vector of $V(s)$;
for $episode = 1$ **to** N **do**
  Observe state s_t;
  for $t = 1$ **to** t_{update} **do**
    Choose a_t randomly with probability ϵ, else perform a_t to policy $\pi(a_t|s_t;\theta)$;
    Take action a_t, observe r_{t+1}, s_{t+1};
    $s_t \leftarrow s_{t+1}$;
  end
  $R = V(s_t, \theta_v)$;
  for $i \in \{t-1, \ldots, t-t_{update}\}$ **do**
    $R \leftarrow r_i + \gamma R$;
    gradient:$\theta \leftarrow \theta + \nabla_\theta log\pi(a_i \| s_i;\theta)(R - V(s_i;\theta_v))$
    gradient:$\theta_v \leftarrow \theta_v + \delta(R - V(s_i;\theta_v))^2/\delta\theta_v$
  end
end

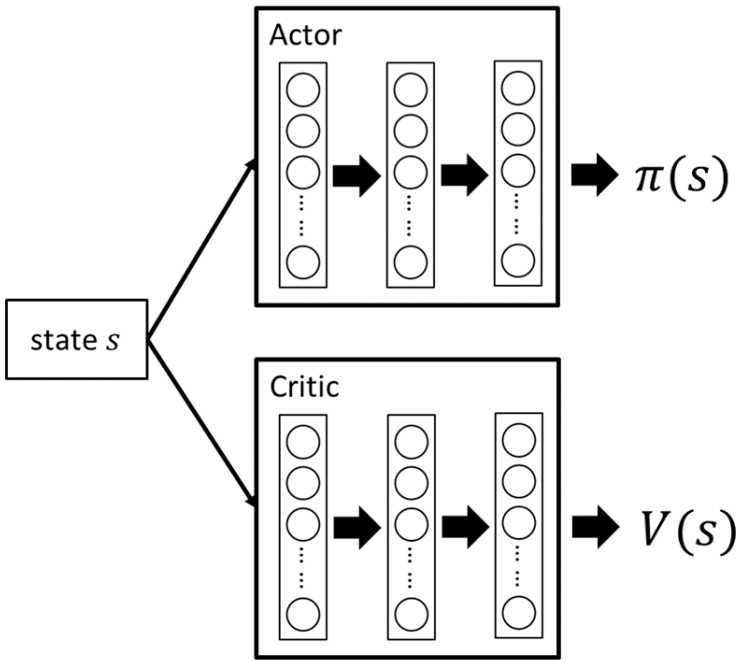

Figure 5. Module Structure of A2C.

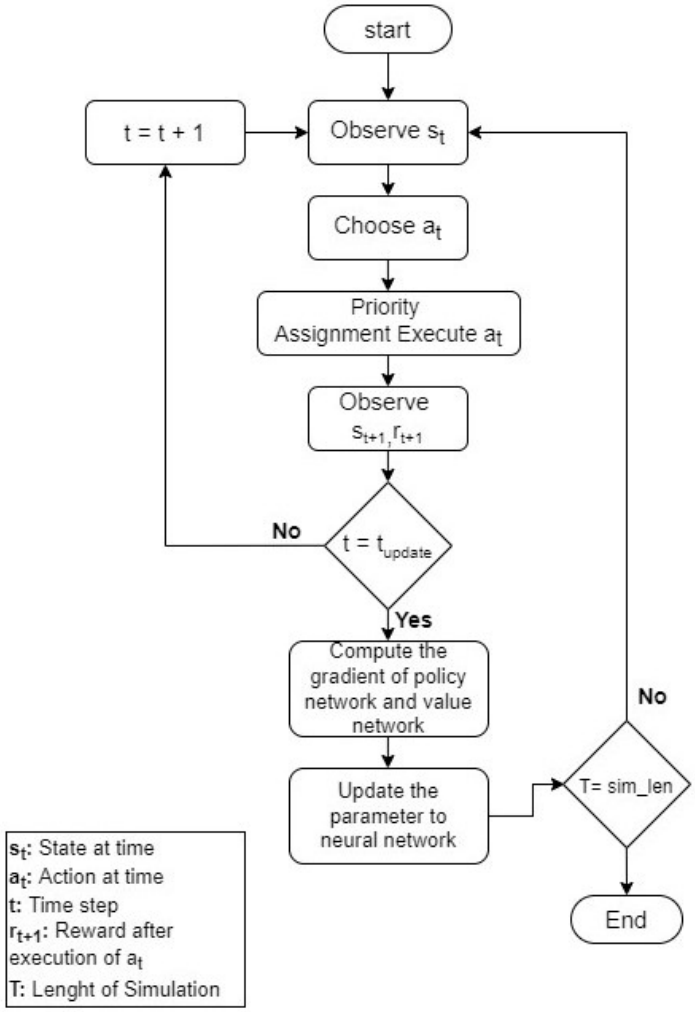

Figure 6. Flow of Intersection Control Model.

4.5. Brake-Safe Control

The brake-safe control used to cross intersections without collisions [6] is presented as follows. The model and control described here use differential equations and a set of assumptions. The differential equation of state is described as follows:

$$\dot{y}_i(t) = v_i(t) \tag{13}$$

$$\dot{v}_i(t) = u_i(t)\delta\Big(u_i(t), v_i(t)\Big) \tag{14}$$

where δ denotes the binary function and $v_i \in [0, \bar{v}_i]$ denotes the set of times. The velocity $v_i(t)$ is zero, and the acceleration $u_i(t)$ is negative; alternatively, the velocity $v_i(t)$ reaches the maximum value and the acceleration $u_i(t)$ is positive, in which case the output δ function is zero. Given two vehicles i, j, the completed obstacle region is defined as $R_{\sigma(j)<\sigma(i)}$, with positions (y_i, y_j) and leading priority $\sigma(j) < \sigma(i)$. For the control section, assuming vehicle j is in state $x_j(t_0)$ at time t_0, the authors note

$$B_{\sigma(j)<\sigma(i)}(x_j(t_0)) := \{x_i \in X_i | \forall t \geq t_0, \Big(y_i\big(t, \underline{u}_i, x_i(t_0)\big), y_j\big(t, \underline{u}_j, x_j(t_0)\big)\Big) \notin R_{\sigma(j)<\sigma(i)}\} \tag{15}$$

where $\underline{u}$ denotes the braking control. A vehicle i brakes at t_0 if

$$\forall j \in N, \sigma(j) < \sigma(i) \Rightarrow x_i(t_0) \in B_{\sigma(j)<\sigma(i)}(x_j(t_0)) \tag{16}$$

where N is the number of vehicles in the system. This means that vehicle i can cross safely without colliding with vehicle j, even if vehicle j uses the maximum brake control value. To ensure that all vehicles are always in brake-safe state, a control law g is introduced. Given the initial configuration $\sigma(j) < \sigma(i)$ of two vehicles i, j, the worst-case scenario is that vehicle j brakes and vehicle i accelerates during the next or future time slot. With this in mind, the control law can be described as follows.

Let $u_i^{impulse} \in U_i$ describe the impulse control function for vehicle i, defined by

$$u_i^{impulse}(t) = \begin{cases} \bar{u}_i & if\ t = 0 \\ \underline{u}_i & if\ t \geq 1 \end{cases} \tag{17}$$

The flow control law is described synthetically as follows:

$$g_i(x) = \begin{cases} \underline{u}_i & if\ \exists\, \sigma(j) < \sigma(i), \exists t \geq 0\ s.t. \left(y_j(t, \underline{u}_j, x_j), y_i(t, u_i^{impulse}, x_i) \in R_{\sigma(j)<\sigma(i)} \right) \\ \bar{u}_i & else \end{cases} \tag{18}$$

In Equation (18), vehicle i exerts maximum brake control if there exists a collision at time $t \geq 0$ in the worst-case scenario. Otherwise, it may accelerate in any case. Figure 7 shows the flow for how the brake-safe control is used in this work. Each vehicle starts with an initial speed of zero and an initial brake-safe state, according to Equation (15). Furthermore, each vehicle i computes the control law g_i using Equation (18) and uses g_i as the control from time step t to $t + 1$ until its position y_i reaches the destination $\bar{y}_i$. As far as matters of safety are concerned, the proposed work takes into consideration cooperative, connected, and automated mobility (CCAM) and a brake-safe control policy [18], with a focus on priorities and recommendations while crossing the intersection. This approach circumvents collisions caused by mechanical failure and takes into account accident control measures for ensuring road safety. With the proposed safety policy and the ongoing developments around the idea of "Sustainable Mobility as a Service" (S-MaaS) [43], users can share speed, public transportation mobility services [44], as well as other mobility services on a single digital platform. The proposed plan can help to reduce the travel time of mobility services and make automated intersection crossings safer.

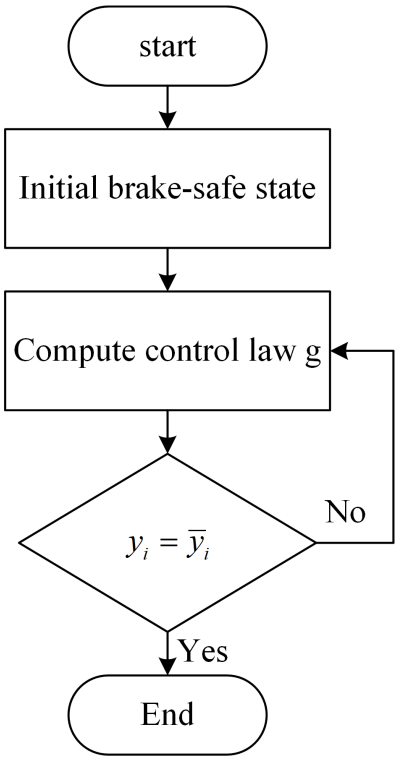

Figure 7. Flow of brake-safe control.

5. Experiments and Results Discussion

In this section, we present the simulation environment, simulation setup, simulation results, and evaluation of A2CAIM, then describe the traffic simulator information.

5.1. Simulation Environment

As shown in Table 2, our method was implemented using the Python programming language on a computer with an Intel(R) Core(TM) i5 CPU with 2.8 GHz and 16 GB RAM running the Windows 10 (64-bit) operating system. The Python programming language was used to realize the proposed Advantage Actor-Critic Agents Autonomous Intersection Management method.

Table 2. Experimental environment.

CPU	Intel(R) Core(TM) i5-8400 CPU @ 2.80GHz
Memory	16 GB - DDR4
Operating System	Windows 10 (64-bit)
Programming Language	Python 2.7

5.2. Simulation Setup

We simulated the proposed method using SUMO [45], which is a multi-modal and microscopic open-source traffic simulator. SUMO allowed us to simulate each vehicle with its own routes moving individually through the networks. We generated traffic demands and values of simulated vehicles, and generated the manipulated behaviors individually via the Traffic Control Interface (TCI), which is the interface between the outer codes and SUMO. Figure 8 shows the intersection model used in our experiments, which consists of twelve incoming lanes. The vehicles in the left-most lane turn left, while those in the middle lanes turn right or move straight. We used an Origin–Destination (OD) matrix, as shown in Table 3, to describe the traffic. An entry value of 0.2 for West and East that means that 20% of vehicles move West to East. The number of generated vehicles per minute is used as the parameter for the Poisson distribution to generation vehicles in the simulation; the basic parameters are summarized in Table 4.

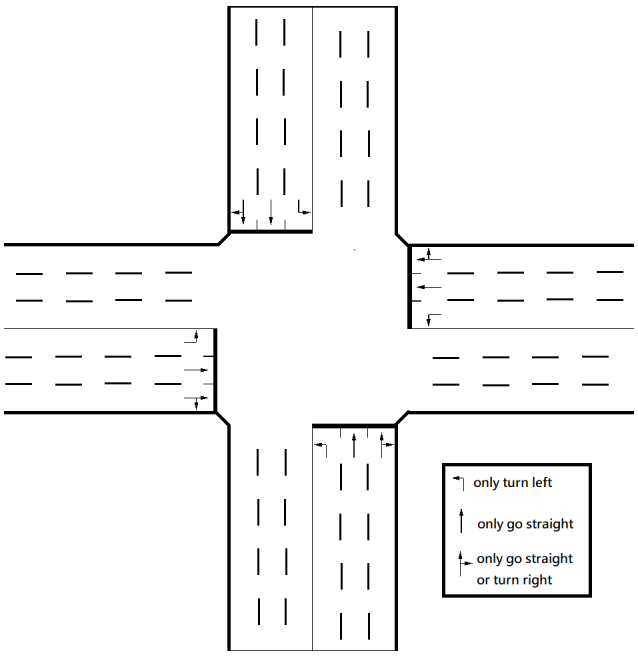

Figure 8. Intersection model used in our simulations.

Table 3. Origin–Destination matrix.

Origin \ Destination	West	East	South	North
West	-	0.3	0.035	0.085
East	0.3	-	0.085	0.035
South	0.085	0.035	-	0.2
North	0.035	0.085	0.2	-

Table 4. System simulation parameters.

Simulation Parameter	Value
Car moving length	5 m
Cooperative zone car length	50 m
Maximum velocity	10 m/s
Maximum accelerate	4 m/s^2
Minimum accelerate	-5 m/s^2
Brake-safe control time step	0.1 s

5.3. Simulation Results

This section evaluates our training results and compares them with other methods, namely, Fast First Service (FFS) (2017) [6] and GAMEOPT (2022) [3]. FFC is considered as the base comparison method in terms of fairness, average wait time, maximum wait time, and throughput. Below, we introduce the performance indicators used to evaluate the performance of these models.

- Average Wait Time (AWT): If the velocity of a vehicle in an incoming lane is smaller than 0.1 m/s, we regard the vehicle as in the wait queue. The Average Wait Time (AWT) is represented by

$$AWT = \frac{\sum_i w_i}{N} \tag{19}$$

  where w_i denotes the wait time of vehicle i and N denotes the number of vehicles waiting.
- Maximum Wait Time (MWT): The maximum wait time for vehicles during the simulation is called the Maximum Wait Time. It expresses how long a vehicle needs to wait in the worst case.
- Throughput: The vehicles exiting an intersection during the simulation is called the throughput.
- Fairness: The fairness metric used in our simulations is Jain's fairness index [8], represented by

$$Fairness = \frac{\left(\sum_{i=1}^{N} w_i\right)^2}{N \sum_{i=1}^{N} (w_i{}^2)} \tag{20}$$

  where w_i is a waiting vehicle i and N is the number of vehicles. If all vehicles have the same wait time, the fairness index is 1. The fairness decreases as the disparity in the wait time of vehicles increases.

5.4. Evaluation of A2CAIM

The experimental results show the performance of our proposed model using different parameter combinations, including the reward function and timestep. Table 5 specifies the

parameters used in this experiment. In each training epoch, we ran the simulation for 200 s using a high inflow rate (100 vehicle arrivals per minute).

Table 5. Parameters used in simulations.

Parameter	Value
Simulation length	200 s
Inflow rate	100 vel/min
Discount factor	0.99

The reward function used here is

$$R_{t+1} = \begin{cases} \alpha n_{t+1} - max_i w^i_{t+1} + \beta, & if\ max_i w^i_{t+1} < max_i w^i_t \\ \alpha n_{t+1} - max_i w^i_{t+1}, & else \end{cases} \quad (21)$$

where n_{t+1} is the change in throughput between time t and $t+1$, $max_i w^i_{t+1}$ is the average wait for lane i with increasing $t+1$, and α, β are constant values. The use of the element n_{t+1} is to maximize throughput in order to enhance the efficiency of the system, while $max_i w^i_{t+1}$ is used to avoid starvation in case a is not prioritized for long time (in which case the reward is very low). The use of α is to balance n_{t+1} and $max_i w^i_{t+1}$, as the value of n_{t+1} is around 0 to 5 and $max_i w^i_{t+1}$ is around 10 to 80. Finally, β is used to encourage the agent to select the lane that has been waiting for the longest time, and should be large enough to have an encouraging effect.

Figures 9 and 10 show the MWT and Fairness with different combinations of α and β. We used different combinations of α and β as the parameters of the reward function to train our model for 30 epochs, and used ten simulations to average the performance. It can be seen that our method outperforms the others when $\alpha = 15$. Then, we used $\alpha = 15$ and different β to train the model. As shown in Figure 11, our method outperformed the others in terms of the Maximum Wait Time when $\alpha = 15, \beta = 459$.

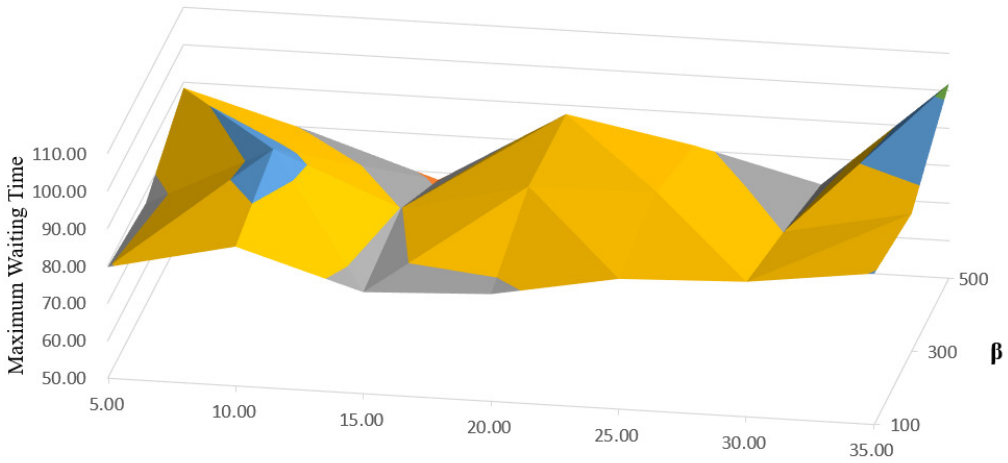

Figure 9. Maximum Wait Time difference with different α, β.

Based on these results, we chose $\alpha = 15, \beta = 459$ as the parameter for the reward function and used different action time steps to evaluate model performance. Table 6 shows the average performance with different action time steps. The experimental results show that a time step of 4.0 s results in the best MWT and throughput.

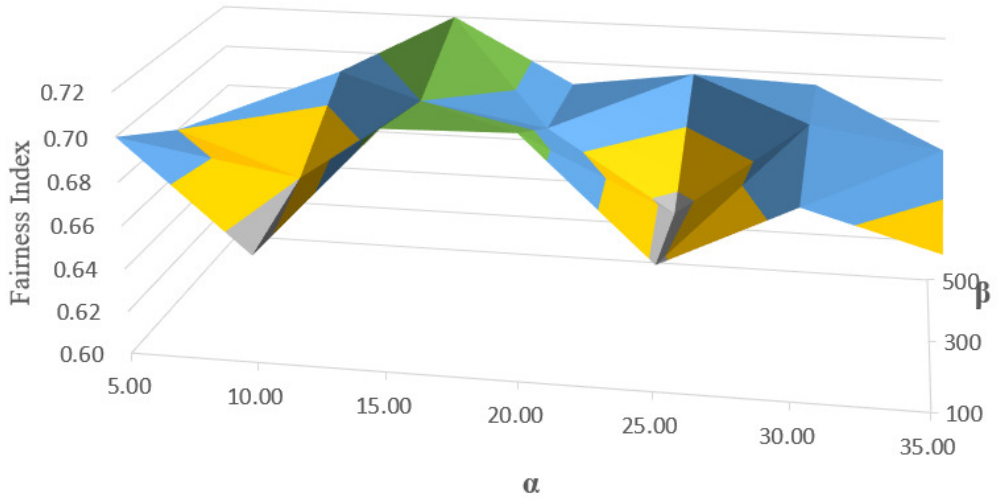

Figure 10. Fairness with different α, β.

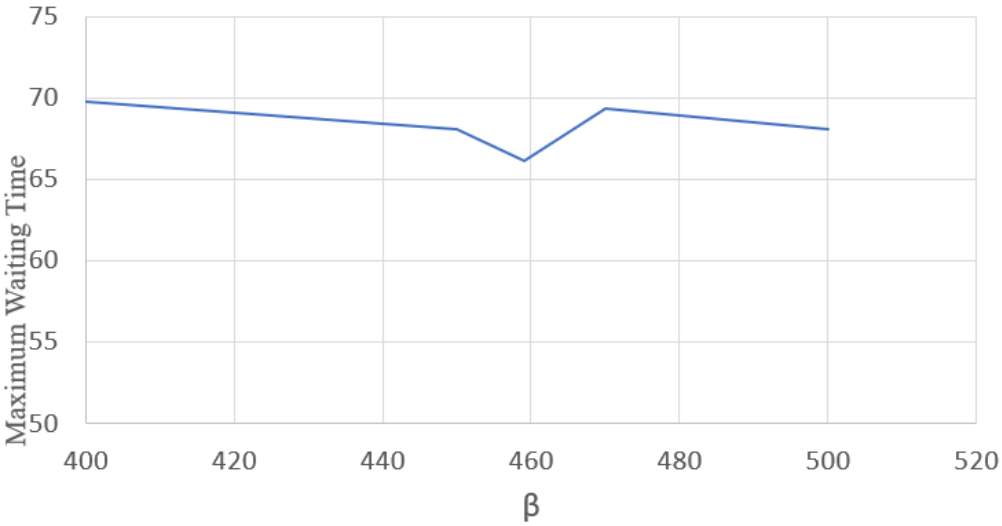

Figure 11. Maximum Wait Time with $\alpha = 15$ and different β.

Table 6. Average performance comparison with different action time steps.

Timestep (s)	AWT (s)	MWT (s)	Throughput (Vehicles)	Fairness
2.5	36.76	79.05	162.0	0.67
3.0	37.59	66.09	193.1	0.73
3.5	39.83	79.64	204.9	0.67
4.0	32.93	65.63	213.3	0.70

5.5. Comparison with FFS and GAMEOPT Methods

We compared our model with the FFS policy [6] and GAMEOPT [40] traffic flow without lights. In GAMEOPT, 50+ vehicles per minute were considered in the implementation. In our implementation, we considered 100+ vehicles per minute. In total, around 6000 vehicles per hour were considered in the simulations. The parameters used for our method were $\alpha = 15, \beta = 459$ and an action time step of 4.0 s. We simulated both methods using four different inflow rates ten times. Table 7 shows the average performance over ten simulations.

Table 7. Average performance results over ten simulations using four different inflow rates.

Inflow Rate (Vehicles/min)	125			100			75			50		
Method	**A2CAIM**	**GAMEOPT**	**FFS**	**A2CAIM**	**GAMEOPT**	**FFS**	**A2CAIM**	**GAMEOPT**	**FFS**	**A2CAIM**	**GAMEOPT**	**FFS**
AWT (s)	33.6	37.1	38	32.93	36	31.73	27.34	24	24.25	15.51	17.2	7.74
MWT (s)	65	110	120	65.63	108	108.51	63.32	82	89.56	59.14	70	43.01
Throughput (vehicles)	230.2	210.4	220	213.3	180	227.1	193.2	150	204	157.5	120	138.7
Fairness	0.78	0.6	0.4	0.7	0.5	0.32	0.63	0.42	0.24	0.43	0.4	0.09

Although the throughput and AWT are worse than FFS, our method has better MWT and Fairness at high inflow rates (125, 100, and 75 vehicles per minute). Figure 12c shows that the fairness of our method is 20% to 40% better than FFS and 15% better than GAMEOPT. Figure 12d shows the throughput results; while our method is initially somewhat worse than FFS and GAMEOPT, as the number of vehicles increases the throughput increases as well. For example, when vehicle flow is increased to 100 or 125 per minute, the throughput gradually increases. Figure 12a,b shows the AWT and MWT at different inflow rates. Our method has a higher AWT than FFS and GAMEOPT, although it has a lower MWT at a high inflow rate of up to 100 vehicles (a reduction of 25% to 36% up to inflow of 100). The inflow is increased automatically, and the average waiting time is reduced. This means that even though the average wait time of our method supports a higher number of vehicles, our proposed method produces better results compared to FFC and GAMEOPT. However, the difference in waiting time per vehicle is low, and considering the worst case from among the three methods, the worst-case vehicle waiting in our method crosses the intersection sooner than with FSS and GAMEOPT at high inflow rates. On the other hand, the AWT and MWT are worse than FFS and GAMEOPT when the inflow rate is low (25 or 50 vehicles per minute). With 100 or 125 vehicles, the AWT and MWT with our method are reduced.

In terms of efficiency, Table 7 shows the overall performance for our proposed approach with different overflow conditions. As the overflow rate increases, all of the performance metrics gradually increase as well. For example, compared to the FFS and GAMEOPT methods (Figure 12b,c), the fairness and maximum wait time and overall performance are always good with different inflow rates (50, 75, 100, and 125). The average wait time (Figure 12a of vehicles less good when the inflow rate is lower. As the inflow rate increases (100, 125), the average wait time decreases. The throughput is similar to that of FFC, and as the inflow rate increases (125), the throughput increases as well.

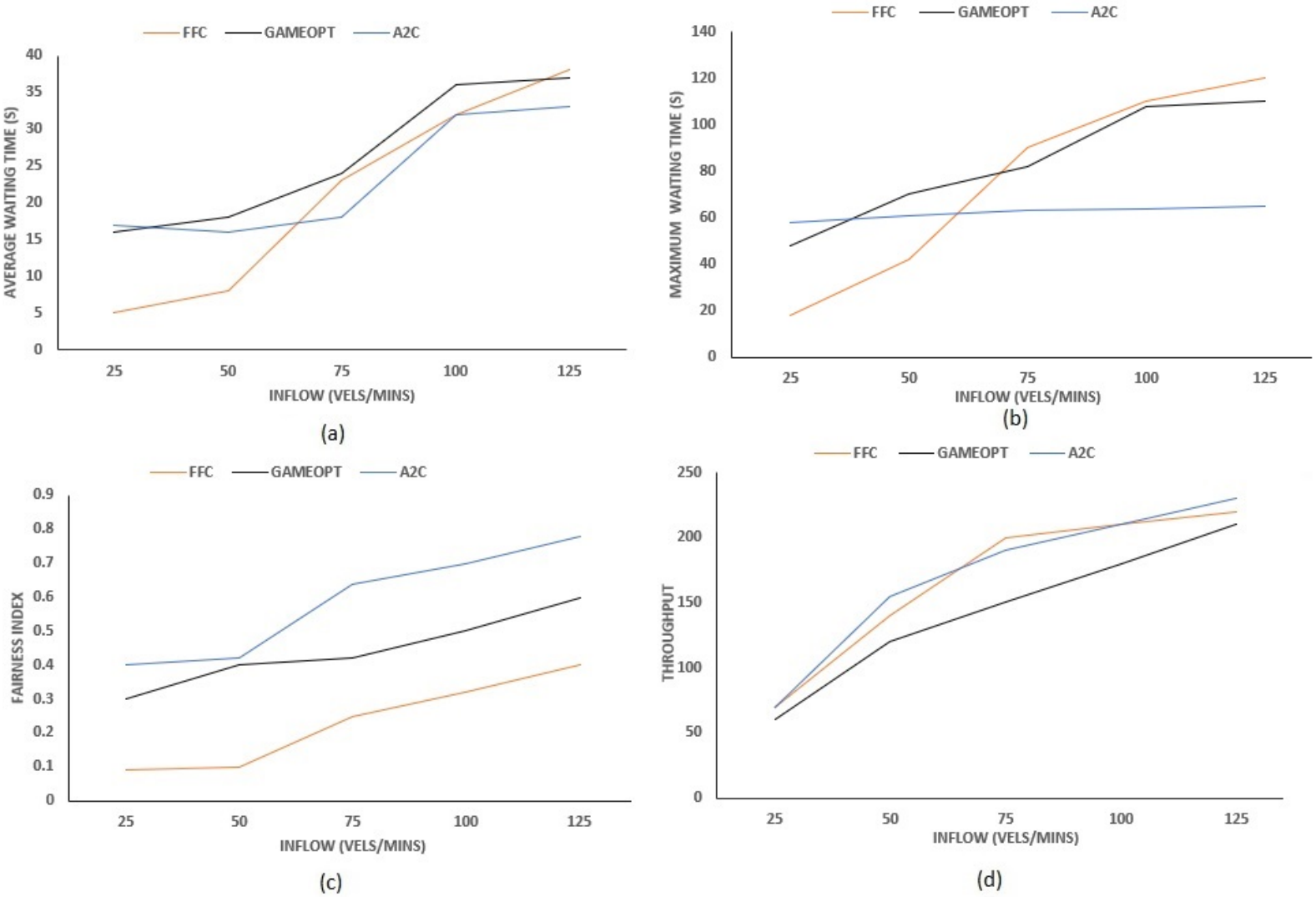

Figure 12. Performance results for different inflow rates: (**a**) AWT, (**b**) MWT, (**c**) Fairness, (**d**) Throughput.

6. Conclusions and Future Work

In this work, we propose an Autonomous Intersection Management system based on Advantage Actor-Critic (A2C). In this system, we use the brake-safe control model to avoid collisions and ensure the safety of vehicles as they travel through the intersection. We use A2C to help the Intersection Controller to learn a fair and efficient intersection management policy. In addition, we design a reward function to encourage the system to distribute intersection resources fairly while maintaining high throughput. We experimented with several different parameter configurations to train our model, ultimately using Simulation of Urban Mobility (SUMO) to simulate the proposed method at an isolated intersection. Our experimental results show that our method is able to improves the fairness index by 20% to 40% compared with FFS and GAMEOPT. The maximum wait time is reduced by 20% to 36% in high traffic flow, and at high inflow rates the average waiting time is reduced and throughput is increased.

In the future, it would be possible to consider simulating different types of intersections, such as roundabouts or other more complex intersections. In addition, emergency vehicles that must rush through intersections, such as police cars, ambulances, etc., could be considered.

Author Contributions: Conceptualization, methodology, G.-H.L.; data curation, writing—original draft preparation, J.A.; writing—review and editing, H.-C.H.; supervision,Hsiung, P.-A.H.; funding acquisition, P.-A.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research is partially supported by research projects from the Ministry of Science and Technology, Taiwan under research project grants MOST 110-2643-F-194-006, MOST 110-2420-H-194-002-TH, MOST 110-2927-I-194-001, and MOST 108-2221-E-194-033-MY3

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chang, G.L.; Xiang, H. *The Relationship between Congestion Levels and Accidents*; Technical Report, MD-03-SP 208B46; State Highway Administration: Baltimore, MD, USA, 2003.
2. Kellner, F. Exploring the impact of traffic congestion on CO_2 emissions in freight distribution networks. *Logist. Res.* **2016**, *9*, 21. [CrossRef]
3. Kamal, M.A.S.; Imura, J.; Hayakawa, T.; Ohata, A.; Aihara, K. Intersection Coordination Scheme for Smooth Flows of Traffic Without Using Traffic Lights. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 1136–1147. [CrossRef]
4. *IEEE Std 1609.0-2013*; IEEE Guide for Wireless Access in Vehicular Environments (WAVE)—Architecture. IEEE: Piscataway, NJ, USA, 2014; pp. 1–78.
5. Dresner, K.; Stone, P. A Multiagent Approach to Autonomous Intersection Management. *J. Artif. Intell. Res.* **2008**, *31*, 591–656. [CrossRef]
6. Qian, X.; Altché, F.; Grégoire, J.; de La Fortelle, A. Autonomous Intersection Management systems: Criteria, implementation and evaluation. *IET Intell. Transp. Syst.* **2017**, *11*, 182–189. [CrossRef]
7. Mnih, V.; Badia, A.P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In Proceedings of the 33rd International Conference on Machine Learning (PMLR), New York, NY, USA, 19–24 June 2016.
8. Jain, R.K.; Chiu, D.M.W.; Hawe, W.R. *A Quantitative Measure of Fairness and Discrimination*; Eastern Research Laboratory, Digital Equipment Corporation: Hudson, MA, USA, 1984.
9. Cascetta, E. *Transportation Systems Engineering: Theory and Methods*; Springer Science & Business Media: Berlin, Germany, 2013; Volume 49.
10. Parks-Young, A.; Sharon, G. Intersection Management Protocol for Mixed Autonomous and Human-Operated Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2022**, 1–11. [CrossRef]
11. Bindzar, P.; Macuga, D.; Brodny, J.; Tutak, M.; Malindzakova, M. Use of Universal Simulation Software Tools for Optimization of Signal Plans at Urban Intersections. *Sustainability* **2022**, *14*, 2079. [CrossRef]
12. Li, G.; Wu, J.; He, Y. ActorRL: A Novel Distributed Reinforcement Learning for Autonomous Intersection Management. *arXiv* **2022**, arXiv:2205.02428.
13. Musolino, G.; Rindone, C.; Vitetta, A. Models for Supporting Mobility as a Service (MaaS) Design. *Smart Cities* **2022**, *5*, 206–222. [CrossRef]
14. Schepperle, H.; Böhm, K. Agent-based traffic control using auctions. In *International Workshop on Cooperative Information Agents*; Springer: Berlin, Germany, 2007; pp. 119–133.

15. Naumann, R.; Rasche, R.; Tacken, J.; Tahedi, C. Validation and simulation of a decentralized intersection collision avoidance algorithm. In Proceedings of the IEEE International Conference on Intelligent Transportation Systems, Boston, MA, USA, 12 November 1997; pp. 818–823.
16. Naumann, R.; Rasche, R.; Tacken, J. Managing autonomous vehicles at intersections. *IEEE Intell. Syst. Their Appl.* **1998**, *13*, 82–86. [CrossRef]
17. VanMiddlesworth, M.; Dresner, K.; Stone, P. Replacing the stop sign: Unmanaged intersection control for autonomous vehicles. In Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems, Estoril, Portugal, 12–16 May 2008; Volume 3, pp. 1413–1416.
18. Gregoire, J.; Bonnabel, S.; de La Fortelle, A. Optimal cooperative motion planning for vehicles at intersections. *arXiv* **2013**, arXiv:1310.7729.
19. Gregoire, J.; Bonnabel, S.; De La Fortelle, A. Priority-Based Coordination of Robots. CoRR. abs/1602.01783. 2014. Available online: https://hal.archives-ouvertes.fr/hal-00828976/file/priority-based-coordination-of-robots.pdf (accessed on 5 December 2022)
20. Chen, L.; Englund, C. Cooperative intersection management: A survey *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 570–586. [CrossRef]
21. Chen, L.; Englund, C. Manipulator trajectory planning based on work subspace division *Concurr. Comput. Pract. Exp.* **2022**, *34*, 570–586.
22. Li, C.-G.; Wang, M.; Sun, Z.-G.; Lin, F.-Y.; Zhang, Z.-F. Urban Traffic Signal Learning Control Using Fuzzy Actor-Critic Methods. In Proceedings of the Fifth International Conference on Natural Computation, Tianjin, China, 14–16 August 2009; Volume 1, pp. 368–372.
23. Jin, J.; Ma, X. Adaptive Group-Based Signal Control Using Reinforcement Learning with Eligibility Traces. In Proceedings of the IEEE 18th International Conference on Intelligent Transportation Systems, Gran Canaria, Spain, 15–18 September 2015; pp. 2412–2417.
24. Mikami, S.; Kakazu, Y. Genetic Reinforcement Learning for Cooperative Traffic Signal Control. In Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence, Orlando, FL, USA, 27–29 June 1994; Volume 1; pp. 223–228.
25. Ha-li, P.; Ke, D. An intersection signal control method based on deep reinforcement learning. In Proceedings of the 10th International Conference on Intelligent Computation Technology and Automation (ICICTA), Changsha, China, 9–10 October 2017; pp. 344–348.
26. Zhang, C.; Jin, S.; Xue, W.; Xie, X.; Chen, S.; Chen, R. Independent Reinforcement Learning for Weakly Cooperative Multiagent Traffic Control Problem *IEEE Trans. Veh. Technol.* **2021**, *7*, 7426–7436. [CrossRef]
27. Chanloha, P.; Usaha, W.; Chinrungrueng, J.; Aswakul, C. Performance Comparison between Queueing Theoretical Optimality and Q-Learning Approach for Intersection Traffic Signal Control. In Proceedings of the Fourth International Conference on Computational Intelligence, Modelling and Simulation, Kuantan, Malaysia, 25–27 September 2012; pp. 172–177.
28. Liu, W.; Liu, J.; Peng, J.; Zhu, Z. Cooperative Multi-agent Traffic Signal Control system using Fast Gradient-descent Function Approximation for V2I Networks. In Proceedings of the IEEE International Conference on Communications (ICC), Sydney, Australia, 10–14 June 2014; pp. 2562–2567.
29. Teo, K.T.K.; Yeo, K.B.; Chin, Y.K.; Chuo, H.S.E.; Tan, M.K. Agent-Based Traffic Flow Optimization at Multiple Signalized Intersections. In Proceedings of the 8th Asia Modelling Symposium, Taipei, Taiwan, 23–25 September 2014; pp. 21–26.
30. Prashanth, L.A.; Bhatnagar, S. Threshold Tuning Using Stochastic Optimization for Graded Signal Control. *IEEE Trans. Veh. Technol.* **2012**, *61*, 3865– 3880. [CrossRef]
31. El-Tantawy, S.; Abdulhai, B. An Agent-based Learning Towards Decentralized and Coordinated Traffic Signal Control. In Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems, Funchal, Portugal, 19–22 September 2010; pp. 665–670.
32. Araghi, S.; Khosravi, A.; Johnstone, M.; Creighton, D. Q-learning Method for Controlling Traffic Signal Phase Time in a Single Intersection. In Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), The Hague, The Netherlands, 6–9 October 2013; pp. 1261–1265.
33. Yau, K.A.; Qadir, J.; Khoo, H.L.; Ling, M.H.; Komisarczuk, P. A Survey on Reinforcement Learning Models and Algorithms for Traffic Signal Control. *ACM Comput. Surv.* **2017**, *50*, 1–34. [CrossRef]
34. Pasin, M.; Scheuermann, B.; Moura, R.F.d. Vanet-based Intersection Control with a Throughput/Fairness Tradeoff. In Proceedings of the 8th IFIP Wireless and Mobile Networking Conference (WMNC), Munich, Germany, 5–7 October 2015; pp. 208–215.
35. Wu, J.; Ghosal, D.; Zhang, M.; Chuah, C. Delay-Based Traffic Signal Control for Throughput Optimality and Fairness at an Isolated Intersection. *IEEE Trans. Veh. Technol.* **2018**, *67*, 896–909. [CrossRef]
36. Madrigal Arteaga, V.M.; Pérez Cruz, J.R.; Hurtado-Beltrán, A.; Trumpold, J. Efficient Intersection Management Based on an Adaptive Fuzzy-Logic Traffic Signal. *Appl. Sci.* **2022**, *12*, 6024. [CrossRef]
37. Guney, M.A.; Raptis, I.A. Scheduling-based optimization for motion coordination of autonomous vehicles at multilane intersections. *J. Robot.* **2020**, *2020*, 6217409. [CrossRef]
38. Wang, M.I.C.; Wen, C.H.P.; Chao, H.J. Roadrunner+: An Autonomous Intersection Management Cooperating with Connected Autonomous Vehicles and Pedestrians with Spillback Considered. *ACM Trans. Cyber-Phys. Syst. (TCPS)* **2021**, *6*, 1–29. [CrossRef]

39. Zhang, Y.; Liu, L.; Lu, Z.; Wang, L.; Wen, X. Robust autonomous intersection control approach for connected autonomous vehicles. *IEEE Access* **2020**, *8*, 124486–124502. [CrossRef]
40. Suriyarachchi, N.; Chandra, R.; Baras, J.S.; Manocha, D. GAMEOPT: Optimal Real-time Multi-Agent Planning and Control at Dynamic Intersections. *arXiv* **2022**, arXiv:2202.11572.
41. Janisch, J. Let's Make an A3C: Theory. Available online: https://jaromiru.com/2017/02/16/lets-make-an-a3c-theory/ (accessed on 16 Febuary 2017).
42. Thomas, P.S.; Brunskill, E. Policy Gradient Methods for Reinforcement Learning with Function Approximation and Action-Dependent Baselines. *arXiv* **2017**, arXiv:abs/1706.06643.
43. Vitetta, A. Sustainable Mobility as a Service: Framework and Transport System Models. *Information* 2022, *13*, 346. [CrossRef]
44. Abdoos, M.; Mozayani, N.; Bazzan, A.L.C. Traffic Light Control in Nonstationary Environments Based on Multi Agent Q-learning. In Proceedings of the 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), Washington, DC, USA, 5–7 October 2011; pp. 1580–1585.
45. Krajzewicz, D.; Erdmann, J.; Behrisch, M.; Bieker, L. Recent Development and Applications of SUMO—Simulation of Urban MObility. *Int. J. Adv. Syst. Meas.* 2012, *5*, 128–138.