Optimal Control of Urban Intersection Scheduling for Connected Automated Vehicles

Shenghao Jiang[1]

***Abstract*— We propose a novel urban congestion-aware intersection scheduling model based on vehicle to infrastructure communication (V2I) for automated and connected vehicles. In this model, a combinational optimized model which combines passing order and vehicular motion control together is proposed. In order to resolve the intersection conflict issue and improve traffic capacity, driving tube and potential conflict matrix is applied in the schedule optimization model. Take the global average waiting time as optimized object, we propose state encoding approach to collect all the vehicle's information in the intersection. Then Deep Q Network (DQN) method is applied to resolve the scheduling problem, which outputs the driving tubes enable vector and subsequently $7th$ polynomial based motion planning trajectory planning is exploited to generate the most comfortable and most efficient trajectory for active vehicles. The optimal time cost profile will be feed back to intersection manager via V2I channel for next time scheduling decision. The performance of this framework is evaluated based on a typical Chinese complicated urban scenario with extensive simulation, our framework achieves encouraging results in terms of average waiting time and peak traffic throughput.**

I. INTRODUCTION

Intersection management is one of the most representative applications of intelligent vehicles with connected and autonomous functions. The connectivity provides environmental information that a single vehicle cannot sense, and the autonomy supports precise vehicular control that a human driver cannot achieve. Intersection management is crucial for safety and traffic efficiency. To prevent collisions and improve intersection passing efficiency, a combinational optimization model which consists of global congestion-aware vehicles passing order scheduler and vehicular dynamic controller is quite desirable.

With the development of autonomous driving and intelligent city, intelligent intersection scheduling, where physical traffic signals turn into cyber ones, has become a popular research topic in recent years. Traditionally, the intersection management focuses on adapting the lengths of traffic signals to real-time traffic conditions such as queue lengths, flow speeds, and information from neighboring intersections [1], [2], [3], [4]. Based on bilateral vehicle-to-infrastructure communication (V2I), the dynamics of vehicles in the intersection control zone can be easily obtained by intersection manager and a centralized intersection manager can talk to individual vehicles and allocate specific time slots for

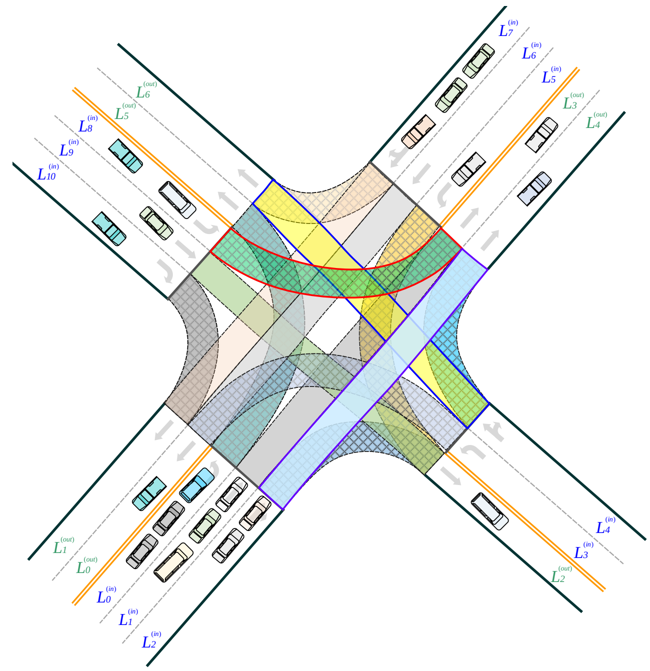


Fig. 1. A typical intersection consists of 11 incoming lanes and 7 outgoing lanes, ($M = 11, N = 7$).

them to pass the intersection after processing the information from all vehicles [5], [6], [7], [8], [9]. The manager functions as a virtual traffic light that can change at infinite frequency. The centralized intersection scheduling problem is usually modeled as an optimization model under different constraints. In [7], the scheduling problem is formulated as a mixed-integer linear program (MILP), and is solved by IBM CPLEX optimization package. In [10], the meta-heuristic method of Particle Swarm Optimization (PSO) model is exploited to solve the constrained optimization problem. The advantage of these centralized scheduling methods is that individual vehicles no longer need to worry about spatial confliction. Besides these centralized approaches, it is also possible for vehicles to communicate with each other through vehicle-to-vehicle(V2V) communication channel and decide the passing order by themselves distributively [11], [12], [13], [14], [15]. In general, a centralized approach has better controllability, reliability, and overall system awareness, but a distributed approach is more adaptive, especially when the infrastructure cost is a concern. There are lots of mainstream rules used in above works when scheduling conflicts exist, such as first-in-first-out, low priority vehicle yields to higher priority vehicles, etc. These rules are not suitable for dif-

[1]Shenghao Jiang is with School of Engineering and Applied Sciences, Harvard University, 29 Oxford Street Cambridge, MA 02318, USA. shjiangbupt@gmail.com

ferent intersection scenarios, and the traffic efficiency can be further improved if the passing order and passing time are adapted to real-time traffic. Meanwhile, existing methods remain in the strategic level and neglects vehicles motion dynamics, whose scalability and applicability in complex real traffic scenarios are limited. To alleviate these problems, a combinational optimization problem consists of intersection scheduling optimization and vehicular motion control optimization is formulated in this paper. Markov Decision Process (MDP) is then exploited to model the relationship between global reward and every schedule action. A multiple layers regression based Q value estimator is trained to map the intersection state and most appropriate scheduling action.

The remainder of this paper is organized as follows. Section II formulates the congestion-aware scheduling problem and vehicle motion control model, in addition, the optimal time cost profile is introduced. Section III introduces the Deep Q-Learning approach to solve the scheduling optimization model. Section IV presents numerical simulation to proof the advantage of our proposed congestion-aware intersection scheduling approach, and subsequently the performance is reported. Section VII gives a brief conclusion and lists our future works.

II. Congestion-Aware Scheduling & Vehicle Motion Control Problem Formulation

We consider a typical intersection scenario which contains M incoming lanes and N outgoing lanes, as is illustrated in Fig. 1. In this paper, driving tube is defined as the linked virtual lane in junction area. Linking matrix L is defined below:

$$L = \begin{bmatrix} f_{0,0} & f_{0,1} & \cdots & f_{0,N-1} \\ f_{1,0} & f_{1,1} & \cdots & f_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ f_{M-1,0} & f_{M-1,1} & \cdots & f_{M-1,N-1} \end{bmatrix} \tag{1}$$

where $f_{i,j} \in \{0,1\}, \forall i \in Z, 0 \leq i < M, j \in Z, 0 \leq j < N$. $f_{i,j} = 1$ indicates that there exists a driving tube between incoming lane $L_i^{(in)}$ and outgoing lane $L_j^{(out)}$. To model the potential confliction during vehicles flow in this intersection, a conflict matrix C is used below:

$$C = \begin{bmatrix} c_{0,0} & c_{0,1} & \cdots & c_{0,Q-1} \\ c_{1,0} & c_{1,1} & \cdots & c_{1,Q-1} \\ \vdots & \vdots & \ddots & \vdots \\ c_{Q-1,0} & c_{Q-1,1} & \cdots & c_{Q-1,Q-1} \end{bmatrix} \tag{2}$$

Q is the amount of linked driving tubes. $c_{i,j} \in \{0,1\}, \forall i \in Z, 0 \leq i < Q, j \in Z, 0 \leq j < Q$. $c_{i,j} = 1$ implies the driving tube i intersects with driving tube j.

A. Congestion-Aware Intersection Scheduling Problem

Increasing the throughput to its maximum capacity at an intersection is one desired goal to reduce traffic congestion. It can be achieved through scheduling optimization for all the vehicles located inside the intersection control zone. Due to the conflict of driving tubes, only vehicles flow from one direction could be allowed to pass the intersection with safety distance constraint at a time. Suppose that there are totally Q ($Q \leq MN$) linked driving tubes, $\boldsymbol{a}_t = [a_0^{(t)}, a_1^{(t)}, \cdots, a_{Q-1}^{(t)}]^T, a_i \in \{0,1\}$ denotes the active status of driving tubes at time t, $a_k = 0$ indicates the k^{th} driving tube Γ_k is disable at time t, while $a_k = 1$ implies that driving tube Γ_k is active and the most front vehicle is allowed to pass the intersection along Γ_k. $\pi = [\boldsymbol{a}_0, \boldsymbol{a}_1, \cdots, \boldsymbol{a}_T]^T$ is the vehicles scheduling strategy along the global time horizon T. The intersection scheduling optimization model can be formulated as follows:

$$min_\pi \quad J(t_{\forall i}^{(arr)}, t_{\forall i}^{(in)}, t_{\forall i}^{(out)}) \tag{3}$$

where $t_{\forall i}^{(arr)}, t_{\forall i}^{(in)}, t_{\forall i}^{(out)}$ imply vehicle i's arrive time of intersection zone (as the visible area show in Fig. 1), its driving tube enter time and driving tube leave time, respectively. The intersection scheduling strategy should be subjected to following constraints for each arbitrary t:

$$c_{i,j} = 0, \ \forall (i,j) \in \psi(a_i^{(t)} = 1, a_j^{(t)} = 1, i \neq j) \tag{4}$$

$$c_{i,j} = 0, \ \forall (i,j) \in \psi(a_i^{(t)} = 1, o_j^{(t)} = 1, i \neq j) \tag{5}$$

$$|s_{l(ref)}^{(t)} - s_l^{(t)}| > \delta, \ \forall l \in \Gamma_i, \forall i \in \{0, 1, \cdots, Q-1\} \tag{6}$$

Optimization target addressed in Eq. 3 aims to minimize the cost function from the global time horizon perspective. Eq. 4 and Eq. 5 are the spatial conflict constraints which indicate that the driving tubes intersecting with each other could not be enable at the same time, $o_j^{(t)}, \forall j \in \{0, 1, \cdots, Q-1\}$ is the occupation status (1 for occupation and 0 for free) of driving tube Γ_j. ψ is the collection of driving tubes' index pair which satisfy its conditions. Eq. 6 formulates the constraint that two adjacent vehicles (vehicle l and its leading reference vehicle $l^{(ref)}$) in the same driving tube should always keep a safety distance δ. If vehicle i locates at the most front of driving tube Γ_i's entrance and $a_i^{(t)}$ is enable at time t, then the driving tube Γ_i's enter time $t_i^{(in)}$ and leave time $t_i^{(out)}$ of vehicle i can be figured out:

$$t_i^{(in)} = t, \ \forall i \in Z, 0 \leq i < K \tag{7}$$

$$t_i^{(out)} = t + \mathbb{P}_i, \ \forall i \in Z, 0 \leq i < K \tag{8}$$

where K is the maximum vehicles amount in the scenario, $\mathbb{P}_i$ is the time cost profile of vehicle i passing driving tube. The cost function is defined below, it can be easily adopted by specific requirement.

$$\begin{aligned} J(t_{\forall i}^{(arr)}, t_{\forall i}^{(in)}, t_{\forall i}^{(out)}) = & w_1 \cdot \frac{1}{K} \sum_{i=0}^{K-1} P_i (t_i^{(in)} - t_i^{(arr)}) \ + \\ & w_2 \cdot max\{t_i^{(out)}\}, \forall i \in Z, 0 \leq i < K \end{aligned} \tag{9}$$

$t_i^{(arr)}$ is the time when vehicle i arrives in the intersection control zone. P_i is the priority of vehicle i. Higher priority vehicle are desired to be scheduled preferentially such as ambulance, fire truck, etc. w_1 and w_2 are weight coefficients.

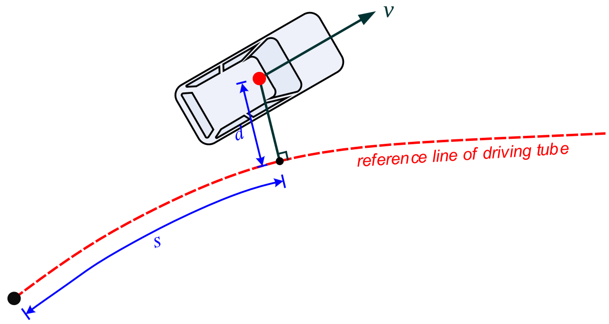


Fig. 2. Frenet coordinate system: Vehicle's dynamics are described with lateral $d(t)$ and longitudinal $s(t)$, respectively.

B. Vehicular Optimal Motion Planning Problem

In order to solve the intersection scheduling problem in Section II-A, we need to figure out the optimal time cost profile of each vehicle when it passes the driving tube. The optimal time cost profile should consider not only passing efficiency but also comfortability and safety issue. In this paper, polynomial based motion planning algorithm, which has the advantage of being realizable on automotive ECU and convergence is guaranteed [16], is exploited to generate the optimal time cost profile and guarantee safety. Usually one is not interested in the vehicle pose in Cartesian coordinate system but rather in the position of vehicle relative to the driving tube. As is shown in Fig. 2, we formulate the vehicle pose in both lateral and longitudinal direction. Thus the movement of the vehicle can be described as an optimal control problem with the output $d(t)$ and $s(t)$, respectively.

Both lateral and longitudinal position is modeled with $7th$ order polynomials, $7th$ polynomial based trajectory planning is used here for fitting complex trajectories. Its alternative models could be lower order polynomials model, i.e.,

$$p(t) = C \cdot T \tag{10}$$

$p(t)$ indicates $d(t)$ or $s(t)$. $C = [c_0, c_1, c_2, \cdots, c_7]$ is $7th$ polynomials coefficients vector and $T = [1.0, t, t^2, t^3, t^4, t^5, t^6, t^7]^T$. According to the initial state $x_i(t_i) = [p(t_i), \dot{p}(t_i), \ddot{p}(t_i), \dddot{p}(t_i)]^T$ and final state $x_e(t_e) = [p(t_e), \dot{p}(t_e), \ddot{p}(t_e), \dddot{p}(t_e)]^T$, the coefficients vector can be determined,

$$C = W^{-1} \begin{bmatrix} x_i(t_i)^T & x_e(t_e)^T \end{bmatrix}^T \tag{11}$$

The optimal control problem is now to determine the final time t_e and corresponding state $x_e(t_e)$, where

$$W = \begin{bmatrix} 1 & t_i & t_i^2 & t_i^3 & t_i^4 & t_i^5 & t_i^6 & t_i^7 \\ 0 & 1 & 2t_i & 3t_i^2 & 4t_i^3 & 5t_i^4 & 6t_i^5 & 7t_i^6 \\ 0 & 0 & 2 & 6t_i & 12t_i^2 & 20t_i^3 & 30t_i^4 & 42t_i^5 \\ 0 & 0 & 0 & 6 & 24t_i & 60t_i^2 & 120t_i^3 & 210t_i^4 \\ 1 & t_e & t_e^2 & t_e^3 & t_e^4 & t_e^5 & t_e^6 & t_e^7 \\ 0 & 1 & 2t_e & 3t_e^2 & 4t_e^3 & 5t_e^4 & 6t_e^5 & 7t_e^6 \\ 0 & 0 & 2 & 6t_e & 12t_e^2 & 20t_e^3 & 30t_e^4 & 42t_e^5 \\ 0 & 0 & 0 & 6 & 24t_e & 60t_e^2 & 120t_e^3 & 210t_e^4 \end{bmatrix} \tag{12}$$

We assume that all vehicles are driving along the center line of driving tubes and no lane change is activated when passing intersection. The optimal time cost profile for each vehicle passing the driving tube can be divided into 2 cases:

1) Driving tube is idle: For a driving tube Γ_i, if there is no vehicles inside it and a vehicle is enable to pass it at time t, the final state is set to $(s(t_e), v^{(desire}, 0.0, 0.0)$, $s(t_e)$ is the longitudinal length of Γ_i, $v^{(desire}$ is the desire velocity of vehicle when it leaves Γ_i. By sampling time t_e we can obtain a bundle of trajectories, and cost function J_s is defined to select the best motion trajectory, and its corresponding t_e^* is the optimal time cost profile:

$$J_s = \frac{1}{2} \int_t^{t_e} ((s^{(4)}(t))^2) dt + w_{t_e} t_e \tag{13}$$

where $\int_t^{t_e} ((s^{(4)}(t))^2) dt$ is the integral over 4^{th} derivative, it indicates the comfort level and $w_{t_e} t_e$ measures the driving tube passing efficiency with w_{t_e} as weight coefficient.

2) Driving tube is occupied by other vehicles: When a vehicle is enable to enter the driving tube with other vehicles inside, the car following model with safety distance $[D_0 + \tau \dot{s}_{lv}(t)]$ is applied for every vehicle (δ in Eq. 6 is adapted to be dynamic according to leading vehicle's velocity $s_{lv}(t)$), i.e., for each vehicle passing the driving tube, a reference longitudinal position $s^{(ref)}$ is defined to ensure safety.

$$s^{(ref)} = s_{lv}(t) - [D_0 + \tau \dot{s}_{lv}(t)] \tag{14}$$

D_0 is a given standstill distance and τ is the configured time headway. The cost function J_s then is formulated as:

$$J_s = \frac{1}{2} \int_t^{t_e} ((s^{(4)}(t))^2) dt + w_{ref} (s^{(ref)} - s(t_e))^2 + w_{t_e} t_e \tag{15}$$

where $w_{ref}(s^{(ref)} - s(t_e))^2$ is the loss measuring the L^2 Euclidean distance between planned position and target position at time t_e. We vary the end constraints with different $s^{(ref)} \pm \Delta s^{(i)}$ and $t_e^{(j)}$ and search the optimal trajectory according to Eq. 15. Based on Eq. 13 and Eq. 15, the optimal time cost profile for each vehicle in Eq. 8 can be figured out.

III. Optimization of Intersection Congestion-Aware Scheduling

A. V2I Communication Based Architecture

Fig. 3 illustrates the architecture for solving intersection scheduling problem based on V2I communication. For vehicle i in intersection control zone, it sends its planned longitudinal trajectory $s_i(t)$, its arrive time $t_i^{(arr)}$, current velocity v_i, physical priority P_i, location L_i, destination lane id D_i to intersection manager. Once receive all the vehicles' message, intersection manager will extract features to encode the scene as a state $\boldsymbol{S}_t$ and subsequently a solver is invoked to figure out the schedule decision A_i^{in} (1 indicates scheduling enable and 0 implies waiting) with desire velocity $v_i^{(desire)}$ or reference longitudinal position $s_i^{(ref)}$.

B. Markov Decision Process

To solve the congestion-aware scheduling problem with optimal motion trajectory planning, Markov Decision Process (MDP) is applied to reformulate the joint problem, in which the constraints are encoded into state transition, and

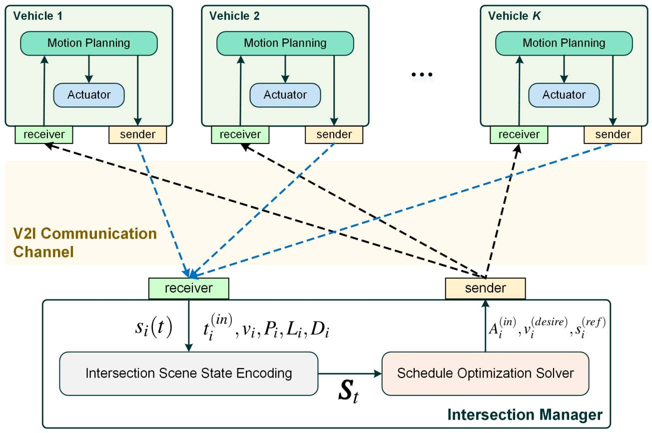


Fig. 3. V2I Architecture for Solving Intersection Scheduling Problem.

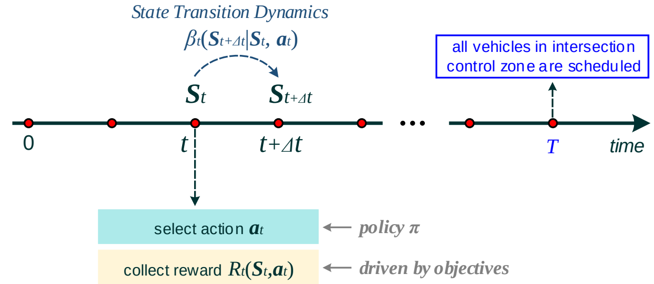


Fig. 4. Markov Decision Process: Based on different policy π, time horizon T and the vehicles' average waiting time in intersection control zone will diverse vastly.

objective function is subject to action reward. A typical MDP is defined through $\{\mathbb{A}, \mathbb{S}, \beta_t(\boldsymbol{S}_{t+\Delta t}|\boldsymbol{S}_t, \boldsymbol{a}_t), R_t(\boldsymbol{S}_t, \boldsymbol{a}_t), 0 \le t \le T, \boldsymbol{S}_t \in \mathbb{S}, \boldsymbol{a}_t \in \mathbb{A}\}$. $\mathbb{A}$, $\mathbb{S}$ are all possible actions and states collection, respectively. T indicates the finite time horizon of MDP, $\beta_t(\boldsymbol{S}_{t+\Delta t}|\boldsymbol{S}_t, \boldsymbol{a}_t)$ and $R_t(\boldsymbol{S}_t, \boldsymbol{a}_t)$ are transition probability from state $\boldsymbol{S}_t$ to state $\boldsymbol{S}_{t+\Delta t}$ with action $\boldsymbol{a}_t$ at time t and its corresponding immediate reward, respectively. As illustrated in Fig. 4, different decision policy will diverse MDP time horizon T and influence vehicles's average waiting time significantly (i.e., Eq. 9). With the assumption that the future rewards are discounted by a factor of $\lambda \in (0,1]$ per time iteration, and define the future cumulative discounted payoff G_t at time t as:

$$G_t = \sum_{u=t}^{T_\pi} \lambda^{u-t} R_u \tag{16}$$

where T_π is the time step at which the intersection scheduling task terminates based on policy π ($\pi : \boldsymbol{S}_t \to \boldsymbol{a}_t$ maps state into proper action). R_u is the immediate reward under policy π at time u. The goal of MDP solver is to find an optimal policy π^* over entire markov deterministic policies space $\mathscr{P}^\infty$ in a way that maximises G_t [17]:

$$\pi^* = \arg\max_{\pi \in \mathscr{P}^\infty} \mathbb{E}[G_t|\pi] \tag{17}$$

$\mathbb{E}[G_t|\pi]$ implies the expected cumulative G_t under policy π.

C. State Vector

The MDP state $\boldsymbol{S}_t$ of intersection scheduling scene at given time t contains fully observable context collected by V2I channel, which is defined as follows:

$$\begin{aligned} \boldsymbol{S}_t = (&t_0^{(wait)}, P_0, B_0, D_0, v_0 \\ &t_1^{(wait)}, P_1, B_1, D_1, v_1 \\ &\cdots, \\ &t_{M-1}^{(wait)}, P_{M-1}, B_{M-1}, D_{M-1}, v_{M-1} \\ &s_0^{(ref)}, s_1^{(ref)}, \cdots, s_{Q-1}^{(ref)})_t^T \end{aligned} \tag{18}$$

where $\boldsymbol{S}_t \in \mathbb{R}^{5M+Q}$, $t_i^{(wait)}, P_i, B_i, D_i, v_i, \forall 0 \le i < M, i \in Z$ indicate the waiting time of most front vehicle at incoming lane L_i^{in}, its physical priority property, vehicles amount in the waiting queue at incoming lane L_i^{in}, its destination outgoing lane id, its current velocity. $s_i^{(ref)}, \forall 0 \le i < Q, i \in Z$ represents the incoming vehicles' reference longitudinal position of driving tube Γ_i.

D. Actions

Actions in the intersection scheduling process are defined as $\boldsymbol{a}_t = [a_0, a_1, \cdots, a_{Q-1}]_t^T, a_i \in \{0,1\}$, $\boldsymbol{a}_t \in \mathbb{A}$. The driving tube Γ_i at time t will be enabled if i^{th} element of $\boldsymbol{a}_t$ equals to 1. If one driving tube is active, then the 7^{th} polynomial based motion planner will be invoked to generate the optimal control reference trajectory for vehicle to pursuit. At the same time, the vehicles' planned $s(t)$ and its velocity v_i, location L_i. will be feed back to the intersection manager then. It is worth to mentioning that the action space $\mathbb{A}$ should exclude action candidates where driving tube conflicts exist.

E. Reward Design

In the MDP model, the action reward rules should express the objective function of Eq. 9 appropriately. If and only if the intersection scheduling process terminates (i.e., all vehicles in intersection control zone are scheduled with safety longitudinal distance and passed the driving tubes safely), the MDP decision agent receives a positive reward R:

$$R = \begin{cases} w_1 \frac{1}{K} \sum_{i=1}^{K} \frac{1}{P_i \cdot t_i^{(wait)}} + w_2 \frac{1}{T_d}, & if\ schedule\ done \\ -1.0, if\ |s_i - s_j| < \xi, & \forall (i,j), i \ne j \\ 0, & otherwise \end{cases} \tag{19}$$

where K is the total vehicles amount in the scheduling phase, ξ is the minimum acceptable safety distance, if two vehicles in the same driving tube are with smaller longitudinal distance less than ξ, the reward will be negative and schedule terminates.

F. Deep Q Network Solver

By observing that the MDP state $\boldsymbol{S}_t \in \mathbb{R}^{5M+Q}$ and $\boldsymbol{a}_t \in \mathbb{R}^{2^Q}$, it is impossible to store all action-value pairs under massive states with tabular format. Deep Q network (DQN) is a more efficient approach to build the action-value mapping with

Algorithm 1: Deep Q Learning for Congestion-Aware Intersection Scheduling

Initialize replay memory $\mathbb{D}$ to capacity N
Initialize action-value estimation network Q with random weights ϑ
Initialize target action-value estimation network $\hat{Q}$ with weights $\vartheta^{-} = \vartheta$
for *episode* $= 1, \phi$ **do**
    Initialize state $\boldsymbol{S}_1$, initialize $t = t_0$
    while *True* **do**
        With probability ε select random action $\boldsymbol{a}_t \in \mathbb{A}$
        Otherwise select action $\boldsymbol{a}_t = \arg\max_{\boldsymbol{a} \in \mathbb{A}} Q(\boldsymbol{a}|\boldsymbol{S}_t; \vartheta)$
        Execute action $\boldsymbol{a}_t$ in emulator
        Observe immediate reward $R_t(\boldsymbol{S}_t, \boldsymbol{a}_t)$ and next state $\boldsymbol{S}_{t+\Delta t}$
        Store transition $(\boldsymbol{S}_t, \boldsymbol{a}_t, R_t(\boldsymbol{S}_t, \boldsymbol{a}_t), \boldsymbol{S}_{t+\Delta t})$ in $\mathbb{D}$
        Sample random mini-batch of transitions $(\boldsymbol{S}_k, \boldsymbol{a}_k, R_k(\boldsymbol{S}_k, \boldsymbol{a}_k), \boldsymbol{S}_{k+\Delta t})$ from $\mathbb{D}$
        Set $y_k = R_k(\boldsymbol{S}_k, \boldsymbol{a}_k)$ if intersection scheduling terminates at time step $k + \Delta t$
        Otherwise set $y_k = R_k(\boldsymbol{S}_k, \boldsymbol{a}_k) + \lambda max_{\boldsymbol{a}'} \hat{Q}(\boldsymbol{a}'|\boldsymbol{S}_{k+\Delta t}; \vartheta^{-})$
        Perform a gradient descent step on $\frac{1}{2}(y_k - Q(\boldsymbol{a}_k|\boldsymbol{S}_k; \vartheta))^2$ with respect to ϑ
        Set $\vartheta^{-} = \vartheta$ every C steps
        $t \leftarrow t + \Delta t$
        if *intersection scheduling terminates at* $\boldsymbol{S}_t$ **then**
            break

multiple layers perception (MLP) network [17]. The Q value $Q^{\pi}(\boldsymbol{S}_t, \boldsymbol{a}_t)$ under policy π is modelled by:

$$Q^{\pi}(\boldsymbol{S}_t, \boldsymbol{a}_t) = \mathbb{E}[G_t|\pi] \tag{20}$$

Algorithm 1 illustrates the procedure of MLP training to fit the optimal action-value mapping function based on stochastic gradient descent. ε-greedy policy is applied to handle the trade off between exploration and exploitation. The loss function is defined below:

$$\begin{aligned} L_k(\vartheta_i) = \frac{1}{2}[R_k(\boldsymbol{S}_k, \boldsymbol{a}_k) + \lambda max_{\boldsymbol{a}'} \hat{Q}(\boldsymbol{a}'|\boldsymbol{S}_{k+\Delta t}; \vartheta_i^{-}) \\ - Q(\boldsymbol{a}_k|\boldsymbol{S}_k; \vartheta_i)]^2 \end{aligned} \tag{21}$$

where L_k, ϑ_k^{-} represents the loss value and network weights at iteration k. *max* indicates the maximum reward over all possible actions a' in target network. After training the DQN, the action-value estimation network will always select the optimal action under current state according to the historical experience learning.

IV. Numerical Experiments

A. Simulation Setup

In order to evaluate the effectiveness of our proposed intersection scheduling strategy for connected vehicles, a typical scenario of Fig. 1 with 11 incoming lanes and 7 outgoing lanes is simulated with extensive experiments. The intersection scene could be changed with time because of two aspects: 1). vehicles from unknown outside environment driving into intersection zone will increase the queue size of each incoming lanes, also possible leads to congestion, 2). intersection scheduling will reduce the queue size and facilitate the traffic clear. Therefore, an outside environment updating model is built for simulating the dynamics of vehicles driving into the intersection control zone, given a time horizon T with granularity Δt and the total vehicles amount V, the traffic time sequence follows the gaussian distribution and the ingoing lanes follows the uniform distribution , i.e.,

$$T_i^{(in)} \sim N(\mu, \sigma), \quad L_i^{(in)} \sim U(0, 11) \tag{22}$$

where $\mu = T/2$ implies the peak value of intersection traffic occurs at time $T/2$. $U(0, 11)$ means that each vehicle could be placed in the 11 incoming lanes with same probabilities. Fig. 5 illustrates 3 typical traffic flow distribution of the intersection. Table I and Table II list the driving tubes length and the parameters configuration in simulation. The DQN network is a 3-layer MLP with neurons $64, 48, 48$, respectively. The proposed DQN based scheduling strategy is compared against other mainstream approaches as list below:

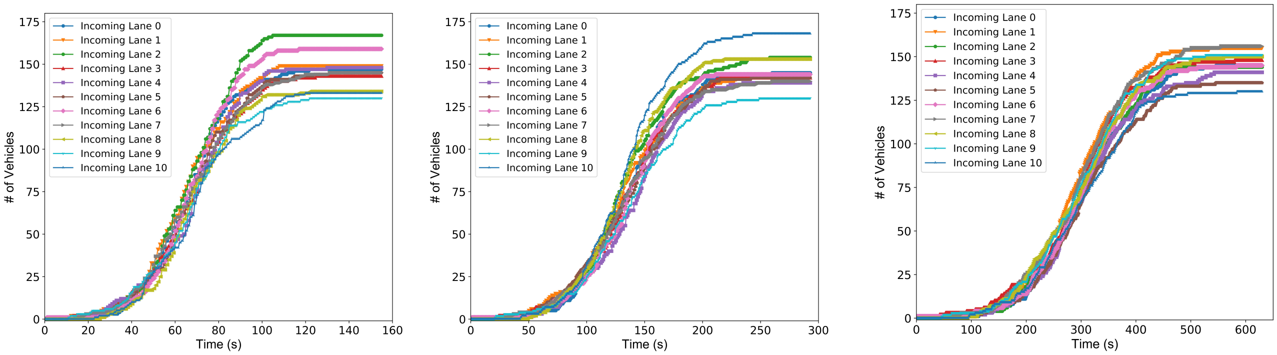
TABLE I
Driving Tube Length in Simulator

Γ_i	$L_i^{(in)}$	$L_i^{(out)}$	Len. (m)	Γ_i	$L_i^{(in)}$	$L_i^{(out)}$	Len. (m)
0	0	5	69.82	7	5	2	55.74
1	1	3	73.98	8	6	0	68.91
2	2	2	38.87	9	7	1	68.24
3	2	4	74.63	10	7	6	34.54
4	3	0	58.53	11	8	3	61.75
5	4	4	36.76	12	9	2	61.75
6	4	6	62.66	13	10	1	32.76

TABLE II
Parameters Configuration in Simulator

Parameter	Value	Parameter	Value
w_{t_e}	1.0	D_0	4.0
τ	0.8	w_{ref}	2.0
learning rate	0.01	λ	0.9
ε	0.9	batch size	512
w_1	10.0	w_2	1.0

- TL-5 [15]: Intersection scheduling with traffic light controlling mechanism in which traffic lights colors change in 5 seconds period.
- TL-10 [15]: Intersection scheduling with traffic light controlling and traffic lights colors change in 10 seconds period.
- AMP-IP [18]: Intersection scheduling with the advanced maximum progression intersection protocol (AMP-IP). A low priority vehicle is allowed to go first if it is anticipated to leave all conflict zones earlier than the arrival time of all higher priority vehicles.
- RWA [9]: A schedule strategy based on right-of-way assignment in which two vehicles are allowed to pass

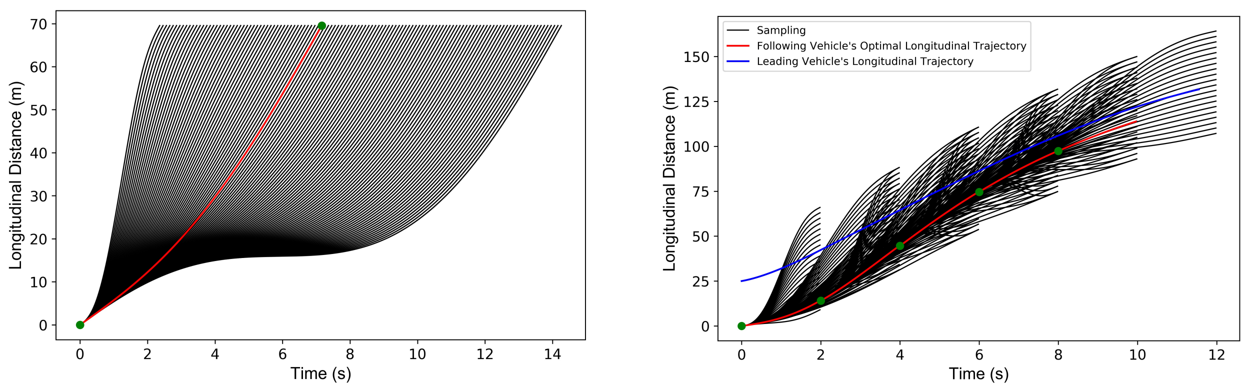


(a) $V = 1600, \sigma = 20$.

(b) $V = 1600, \sigma = 40$.

(c) $V = 1600, \sigma = 90$.

Fig. 5. Three typical intersection traffic flow distribution in simulator.



(a) Optimal longitudinal trajectory planning of driving tube 0: Incoming velocity $= 20km/h$, outgoing velocity $= 50km/h$, optimal time cost profile $= 7.178$s.

(b) Leading vehicle velocity converges to $50km/h$ in driving tube 3, the following vehicle performs sampling to find optimal trajectory and optimal time cost profile based on Eq. 15, the longitudinal gap will converges to $12.4m$.

Fig. 6. Two examples of optimal trajectory planning.

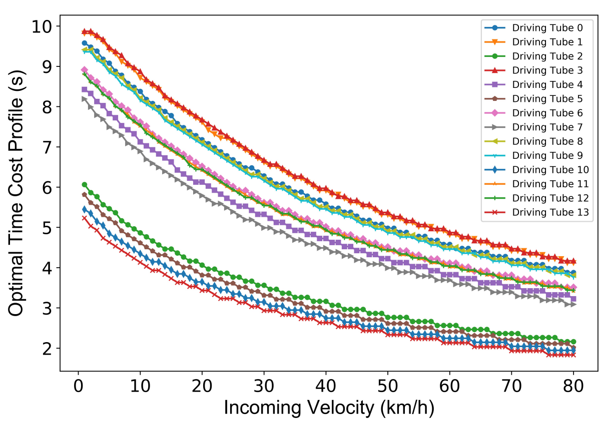


Fig. 8. Optimal time consumption profile for 14 driving tubes with various incoming velocity when driving tubes are idle (desired velocity is $50km/h$).

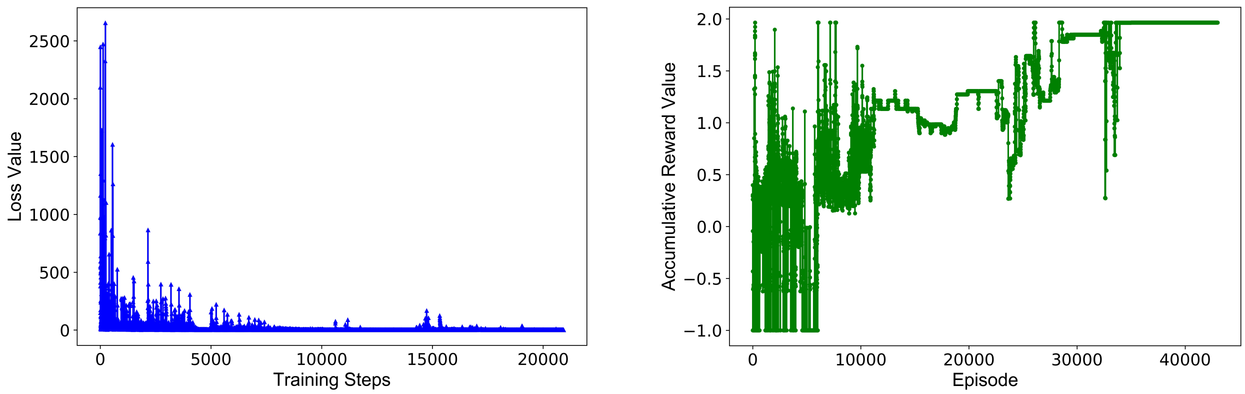
the junction at the same time. It roughly follows FIFO rule, but shorten the safety gap via appropriate right-of-way assignment (low priority vehicle yields to higher priority ones).

B. Optimal Time Consumption Profile for Driving Tubes

When the driving tube is idle (not occupied by other vehicles), the polynomial based motion planning algorithm only samples the time of end state. As is shown in Fig. 6(a), the optimal time cost for a vehicle with incoming velocity $20km/h$ and outgoing desired velocity $50km/h$ passing driving tube 0 is 7.178s. Fig. 6(b) depicts a vehicle passing driving tube 3 where exist a leading vehicle in front of it. The longitudinal gap between two vehicles converges to $12.4m$ to ensure safety. Fig. 8 plots the optimal time cost profile for 14 driving tubes when the driving tube is idle. If a driving tube is occupied by other vehicles before, then vehicles scheduled to enter this driving tube will perform car following based on polynomial based motion planning algorithm, we sample the end state with different $s^{(ref)} \pm \Delta s^{(i)}$ and $t_e^{(j)}$ and find the optimal trajectory and time cost according to Eq. 15.

C. DQN Training & Scheduling Results

Based on parameters configuration of Table I and Table II, we train the DQN with $V = 1600, \sigma = 10$, smaller σ implies heavier vehicles congestion and the intersection state $\boldsymbol{S}_t$ will be covered more comprehensively. Fig. 7 plots the training loss and reward distribution, from which we can observe



(a) DQN Training Loss vs. Steps

(b) Accumulative Reward vs. Episodes

Fig. 7. DQN train phase with $V = 1600, \sigma = 10$.

TABLE III
AVERAGE WAITING TIME & PEAK TRAFFIC THROUGHPUT (PER SECOND) WITH DIFFERENT INTERSECTION SCHEDULING STRATEGIES (VEHICLES AMOUNT IN SIMULATOR = 1600)

σ	TL-5 [15]	TL-10 [15]	AMP-IP [18]	RWA [9]	Proposed
90	$0.42s \pm 1.47s/2.69$	$0.86s \pm 2.70s/2.37$	$0.33s \pm 0.91s/2.16$	$0.31s \pm 0.90s/2.99$	$0.32s \pm 0.59s/2.31$
80	$1.28s \pm 2.15s/3.07$	$1.25s \pm 3.22s/3.30$	$0.45s \pm 1.04s/3.19$	$0.42s \pm 1.12s/3.65$	$0.39s \pm 1.18s/3.07$
70	$3.53s \pm 2.24s/3.26$	$6.94s \pm 4.61s/3.67$	$1.96s \pm 1.42s/3.62$	$1.77s \pm 1.51s/3.70$	$1.86s \pm 1.44s/4.28$
60	$4.44s \pm 1.57s/4.83$	$8.89s \pm 3.10s/3.19$	$2.72s \pm 0.91s/3.04$	$2.60s \pm 0.95s/3.11$	$2.57s \pm 0.97s/6.14$
50	$4.77s \pm 1.15s/4.44$	$9.42s \pm 2.27s/5.09$	$2.81s \pm 0.78s/4.74$	$2.81s \pm 0.66s/4.32$	$2.73s \pm 0.69s/6.80$
40	$4.85s \pm 0.83s/5.70$	$9.73s \pm 1.75s/5.85$	$2.91s \pm 0.54s/7.97$	$2.93s \pm 0.51s/8.70$	$2.88s \pm 0.52s/9.78$
30	$5.71s \pm 12.41s/6.74$	$11.01s \pm 20.12s/6.21$	$3.30s \pm 2.77s/9.20$	$3.39s \pm 2.93s/9.26$	$3.20s \pm 2.61s/9.92$
25	$6.59s \pm 19.50s/6.69$	$12.42s \pm 34.32s/6.63$	$3.66s \pm 4.39s/9.23$	$4.01s \pm 6.42s/9.10$	$3.32s \pm 4.77s/10.27$
20	$7.65s \pm 29.84s/5.97$	$14.76s \pm 55.13s/6.46$	$4.71s \pm 9.41s/9.69$	$5.26s \pm 11.12s/9.53$	$4.29s \pm 9.37s/10.37$
15	$10.74s \pm 49.91s/6.76$	$20.40s \pm 94.71s/6.60$	$5.43s \pm 12.42s/9.83$	$6.95s \pm 16.53s/9.81$	$4.93s \pm 12.09s/10.61$
10	$15.13s \pm 73.14s/7.14$	$34.71s \pm 163.73s/6.95$	$8.52s \pm 20.27s/9.58$	$9.72s \pm 23.75s/9.62$	$8.26s \pm 20.07s/10.55$

that at the beginning of training, the accumulative reward can occasionally achieve high value because of exploration mode and finally it converges to an optimal reward due to the action-value mapping Q network's weights is already stable. Table III reports the performance comparison of average waiting time and peak traffic throughput per second based on different intersection scheduling strategies. The peak traffic throughput is calculated as the number of vehicles that passing the intersection area at the peak traffic arrival time in simulator (i.e., at time μ of Eq. 22). The average waiting time in Table III is reported with mean value and standard deviation format. Our proposed DQN based scheduling strategy outperforms TL-5 and TL-10 mechanisms significantly because of that when the traffic density is low, the traffic light based scheduling lacks of flexibility and increase vehicles' waiting time. For example, in our proposed approach, driving tubes Γ_4 and Γ_9 are allowed to be enable simultaneously because these two driving tubes do not conflict spatially, but only one direction is legal in traffic light controlling scenario. TL-10's performance is worse than TL-5 because when traffic density goes up, 10 seconds period can cause lots of vehicles in congestion and more waiting time will be cost. Compared with AMP-IP and RWA strategies, our method achieves similar performance when traffic density is low, but as the σ becomes smaller, it outperforms AMP-IP and RWA. RWA follows roughly FIFO rule and lower priority vehicle yields to higher one rule, it lacks of real-time traffic adaptation ability, for example, considering a vehicle with higher velocity conflicts with a vehicle with smaller velocity spatially, they arrive the intersection at the same time but the slow vehicle have higher priority, so RWA will schedule it firstly, and it causes the fast vehicle has to slow down and wait for next time scheduling. However, the fast vehicle will be scheduled firstly in our proposed method because its velocity is encoded in intersection state $\boldsymbol{S}_t$ and it can pass the driving tube quickly (with desired velocity $50km/h$). As for AMP-IP, it is based on pre-defined rules and vehicles in congestion lane can not be guaranteed to be scheduled in time when traffic flow is heavy. The peak traffic throughput in our proposed method can achieve ~ 10.5 per second. It is worth to mentioning that decrease D_0 and τ can increase traffic throughput and reduce average waiting time, but it sacrifice safety to some extent because smaller D_0 and τ imply closer following gap when vehicles passing the driving tubes.

V. CONCLUSIONS AND FUTURE WORKS

In this paper, a novel urban congestion-aware intersection scheduling algorithm is proposed. It solves two problems: 1). efficient scheduling for complex urban intersections and 2). vehicles dynamics optimal problem. By exploiting the polynomial based motion planning algorithm to generate the optimal time cost profile, the intersection scheduling can ensure driving comfortability and safety. The experiments are evaluated on a large intersection with 11 incoming lanes and 7 outgoing lanes with extensive experiments. The numerical simulation results demonstrate that it achieves high peak traffic throughput and low waiting time. The methodology (optimal time cost profile generation, DQN based scheduling solver) can be easily adopted to different urban intersections.

REFERENCES

[1] R. Wunderlich, C. Liu, I. Elhanany, and T. UrbanikII. A novel signal-scheduling algorithm with quality-of-service provisioning for an isolated intersection. *IEEE Transactions on Intelligent Transportation Systems*, 9(3):536–547, Sep. 2008.

[2] M. Firdous, F. U. Din Iqbal, N. Ghafoor, N. K. Qureshi, and N. Naseer. Traffic light control system for four-way intersection and t-crossing using fuzzy logic. In *2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, pages 178–182, March 2019.

[3] S. C. Chang, M. W. Tsai, and G. W. Huang. A ga based intelligent traffic signal scheduling model. In *2007 IEEE Symposium on Computational Intelligence in Scheduling*, pages 93–97, April 2007.

[4] A. Aljaafreh and N. A. Oudat. Optimized timing parameters for real-time adaptive traffic signal controller. In *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, pages 244–247, March 2014.

[5] Y. Xu, H. Zhou, B. Qian, H. Wu, Z. Zhang, and X. S. Shen. When automated vehicles meet non-signalized intersections: A collision-free scheduling solution. In *2018 IEEE/CIC International Conference on Communications in China (ICCC)*, pages 709–713, Aug 2018.

[6] D. Miculescu and S. Karaman. Polling-systems-based autonomous vehicle coordination in traffic intersections with no traffic signals. *IEEE Transactions on Automatic Control*, pages 1–1, 2019.

[7] S. A. Fayazi and A. Vahidi. Mixed-integer linear programming for optimal scheduling of autonomous vehicle intersection crossing. *IEEE Transactions on Intelligent Vehicles*, 3(3):287–299, Sep. 2018.

[8] F. Altch and A. de La Fortelle. Analysis of optimal solutions to robot coordination problems to improve autonomous intersection management policies. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 86–91, June 2016.

[9] Yadong Xing, Can Zhao, ZhiHeng Li, Yi Zhang, Li Li, Fei-Yue Wang, Xiao Wang, Yujing Wang, Yuelong Su, and Dongpu Cao. A right-of-way based strategy to implement safe and efficient driving at non-signalized intersections for automated vehicles, 2019.

[10] M. A. Guney and I. A. Raptis. Scheduling-driven motion coordination of autonomous vehicles at a multi-lane traffic intersection. In *2018 Annual American Control Conference (ACC)*, pages 4038–4043, June 2018.

[11] M. D. Vaio, P. Falcone, R. Hult, A. Petrillo, A. Salvi, and S. Santini. Design and experimental validation of a distributed interaction protocol for connected autonomous vehicles at a road intersection. *IEEE Transactions on Vehicular Technology*, 68(10):9451–9465, Oct 2019.

[12] J. Shi, Y. Zheng, Y. Jiang, M. Zanon, R. Hult, and B. Houskal. Distributed control algorithm for vehicle coordination at traffic intersections. In *2018 European Control Conference (ECC)*, pages 1166–1171, June 2018.

[13] V. Savic, E. M. Schiller, and M. Papatriantafilou. Distributed algorithm for collision avoidance at road intersections in the presence of communication failures. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1005–1012, June 2017.

[14] B. Yang and C. Monterola. A simple distributed algorithm for lightless intersection control based on non-linear interactions between vehicles. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6, Oct 2017.

[15] C. Liu, C. Lin, S. Shiraishi, and M. Tomizuka. Distributed conflict resolution for connected autonomous vehicles. *IEEE Transactions on Intelligent Vehicles*, 3(1):18–29, March 2018.

[16] M. Werling, J. Ziegler, S. Kammel, and S. Thrun. Optimal trajectory generation for dynamic street scenarios in a frent frame. In *2010 IEEE International Conference on Robotics and Automation*, pages 987–993, May 2010.

[17] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. 2013. cite arxiv:1312.5602Comment: NIPS Deep Learning Workshop 2013.

[18] S. Azimi, G. Bhatia, R. Rajkumar, and P. Mudalige. Reliable intersection protocols using vehicular networks. In *2013 ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, pages 1–10, April 2013.