FREIGHT MOBILITY RESEARCH INSTITUTE
College of Engineering & Computer Science
Florida Atlantic University

Project ID: Y5R5-21

COORDINATED INTERSECTION CONTROL THROUGH REINFORCEMENT LEARNING WITH SPECIAL CONSIDERATION OF FREIGHT TRAFFIC

Final Report

by

Chaolun Ma
Cma16@tamu.edu
Zachry Department of Civil and Environmental Engineering,
Texas A&M University

Bruce Wang, Ph.D.
bwang@tamu.edu
979-845-9901
Zachry Department of Civil and Environmental Engineering,
Texas A&M University

Yunlong Zhang, Ph.D.
bwang@tamu.edu
979-845-9902
Zachry Department of Civil and Environmental Engineering,
Texas A&M University

for

Freight Mobility Research Institute (FMRI)
Florida Atlantic University
777 Glades Rd.
Boca Raton, FL 33431

December 14, 2021

ACKNOWLEDGEMENTS

This project was funded by the Freight Mobility Research Institute (FMRI), one of the twenty TIER University Transportation Centers that were selected in this nationwide competition, by the Office of the Assistant Secretary for Research and Technology (OST-R), U.S. Department of Transportation (US DOT).

DISCLAIMER

TABLE OF CONTENTS

EXCUTIVE SUMMARY **1**
1.0 INTRODUCTION **3**
2.0 LITERATURE REVIEW **5**
2.1 FIXED-TIME CONTROL WITH COORDINATION 5
2.2 ADAPTIVE CONTROL 5
2.3 CONTROL WITH REINFORCEMENT LEARNING 6
2.4 SUMMARY OF OPTIMIZATION OBJECTIVES 7
3.0 METHODOLOGY **9**
3.1 LYAPUNOV OPTIMIZATION 9
3.2 BACKPRESSURE 10
3.3 Q-LEARNING 13
3.4 REINFORCEMENT LEARNING AND DOUBLE DQN 14
3.5 BACK PRESSURE WITH REINFORCEMENT LEARNING 15
4.0 SIMULATION **17**
4.1 SIMULATION SETTINGS 17
4.2 TESTED ALGORITHMS IN SIMULATION 20
4.3 AGENT PERFORMANCE ON SCENARIOS WITH UNIFORM PASSENGER VEHICLE FLOW 20
4.4 AGENT PERFORMANCE ON SCENARIOS WITH TRUCK FLOW 23
4.4.1 10% truck volume 23
4.4.2 25% truck flow 24
4.4.3 40% truck flow 25
4.5 DISCUSSION 26
5.0 CONCLUSION **29**
6.0 REFERENCES **31**

LIST OF TABLES

Table 1 Traffic volume in the simulation 19
Table 2 Vehicle Type parameter defaults in the simulation 19
Table 3 Average vehicle delay in arterial and grid network case with uniform passenger vehicle flow (in seconds) 22
Table 4 Average vehicle delay in arterial and grid network case with 10% truck volume (in seconds) 23
Table 5 Average vehicle delay in arterial and grid network case with 25% truck volume (in seconds) 24
Table 6 Average vehicle delay in arterial and grid network case with 40% truck volume (in seconds) 25

LIST OF FIGURES

Figure 1: Permitted flow movement for left-turn phase on major arterial 10

Figure 2: Intersection simulation environment 17
Figure 3: Available phases in traffic signal control of the simulation 18
Figure 4: Arterial and grid network environment 19
Figure 5 Average backpressure and waiting time during training process over 300 episodes in arterial case 21
Figure 6: Average backpressure and waiting time during training process over 300 episodes in the grid network case 22
Figure 7: Average vehicle delay in arterial case (in seconds) 23
Figure 8: Average vehicle delay in grid network case (in seconds) 23
Figure 9 Average vehicle delay through the arterial with 10 % truck volume (in seconds)..... 24
Figure 10 Average vehicle delay through the grid network with 10 % truck volume (in seconds) 24
Figure 11 Average vehicle delay through the arterial with 25 % truck volume (in seconds)..... 25
Figure 12 Average vehicle delay through the grid network with 25 % truck volume (in seconds) 25
Figure 13 Average vehicle delay through the arterial with 40 % truck volume (in seconds)..... 26
Figure 14 Average vehicle delay through the grid network with 40 % truck volume (in seconds) 26

EXCUTIVE SUMMARY

This project specially studies signal timing with special consideration of freight traffic in urban areas. The rationale is that freight logistics are critical to quality of life and economies. However, freight mobility, especially along major freight corridors in urban areas, rarely gets special consideration in signal timing. The advent of the Internet of Things (IoT) makes vast information collection a reality. The rich data environment, combined with the boost in computational power, has brought unprecedented opportunities closer to reality than ever before for real-time, information-driven intersection traffic control under variants of traffic scenarios.

The research advances the conventional traffic signal control through introduction of control theories and reinforcement learning methods to design highly efficient network control algorithms. This research focuses on developing a new traffic responsive network signal control in general, and specially with freight traffic considered. When dealing with network signal control, unlike the traditional formulations that either face challenges to quantify promptly such as total delay, or using simple linear combinations of observations as reinforcement learning's reward that lack a theoretical basis (e.g., sum of weighted waiting time and queue length). Hence, this study first utilizes Lyapunov optimization to minimize the long-term average queue across the network and proposes backpressure as the network performance measure. Then the study builds a network signal control algorithm with reinforcement learning (RL) that utilizes backpressure as reward and uses double Deep Q-Network (Double-DQN) in the training process. The proposed algorithm is compared with traditional transportation methods and other RL-based methods.

The numerical tests are conducted on two types of networks, a single corridor, and a local grid network under three traffic demand scenarios from low, medium, to heavy. Numerical test via simulation shows the benefits of the developed model and algorithms under different scales of truck traffic. The effect of different truck ratios (0%, 10%, 25%, and 40%) on each control algorithm was tested simultaneously for the same major and minor traffic volume scenarios. The tests consistently show that the proposed algorithm outperforms the others in terms of the average vehicle waiting time on the network when traffic volume is relatively high.

1.0 INTRODUCTION

Freight logistics are critical to the quality of life and economies. However, freight mobility, especially along major freight corridors in urban areas, rarely gets special consideration in signal timing. Signals are intended to better facilitate traffic flow at level crossings from different approaches, thereby reducing traffic delay and vehicle emissions. The advent of the Internet of Things (IoT) makes vast information collection a reality. The rich information collected through sensors and inter-vehicle communication has enabled the large-scale application of reinforcement learning, a proven powerful tool for efficient and responsive decision-making. This research will build a traffic control algorithm based on the Lyapunov optimization theory and reinforcement learning in the context of big data with a specific objective of improving freight mobility along corridors and grid networks.

The signal control is realized by implementing a control policy. A control policy determines the durations and sequences of phases at each intersection to facilitate traffic movement. Here, a signal phase has three time intervals: green, yellow, and red. A phase refers to a time interval in which the traffic right of way under green and yellow does not change for traffic from all approaches. The signal control policy can take the form of a fixed-time control and vehicle actuated control or adaptive control driven by real-time traffic. Isolated intersection control only considers traffic local to the intersection, while network traffic control considers the coordinated effect of signals. Control implementation is through the control box at each intersection. When the intersections communicate with each other through a traffic control center that they are connected to, a network controller may be developed. Many conventional control methods in literature deal with idealistic traffic, such as uniform and constant traffic. Traffic variations are usually not considered enough. Although the isolated intersection has been intensively studied, unfortunately, little research on isolated intersections has revealed its network implications by now.

The research models the vehicles in the signalized road network as a network queueing process and find the near-optimal control policy by the Lyapunov optimization theory, which create a metric called “backpressure” to measure the performance of the control policy in the network. The Lyapunov optimization theory provides sound theoretical basis for the control method and reward design for the reinforcement learning. Another major element in the proposed method is introduction of reinforcement learning to the control method. Recently, there is an increasing interest in academia to apply reinforcement learning techniques to improve control problems, especially in network control problems (Mannion et al., 2016). The network control problem is suitable for the reinforcement learning’s (RL) trial-and-error framework. By exporting and interacting with the environment after tremendous times, the RL agent can learn for the best policy with large probability. RL can automatically bridge the gap after substantial iterations as long as this relationship between the states and reward is expected to be relatively reliable and stable. Meanwhile, the RL agent can automatically learn control policies from the observed data. The policy can be manually evaluated before implementation and easily to be transferred.

However, it may face a time-consuming parameter tuning process and may be hard to interpret the policy trained by the RL model. The fundamental problem is the state and reward design.

This research will focus on developing a new traffic responsive network signal control in general, but with freight traffic considered in particular, and provide a new metric for network signal control by directly translating general delay minimization into the maximization of intersection throughput, and thus provide a solid theoretical basis for the subsequent reward design of the reinforcement learning. Finally, combine transport theory with reinforcement learning methods to design highly efficient network control algorithms. Numerical tests via simulation will be conducted to show the benefits of the developed model and algorithms under different scales of truck traffic. The new control proposes new measures for optimal switching points for network signal control by directly translating general delay minimization into intersection throughput maximization, and thus to provide a basis for the subsequent reward design in the reinforcement learning. Also, the method can deal with traffic with uncertainty or randomness and be distributed implemented.

The report is organized as the following:

Section 2 summarizes the literature review. Some investigated research topics include fixed-time control, actuated control, adaptive control, and reinforcement learning-based control.

Section 3 is about algorithm development, consisting of three parts. The first part focus on the origin of reinforcement learning, elaborating on how and why Double-DQN is chosen. The second part focus on the stochastic optimization and reward design for reinforcement learning. We utilize the Lyapunov optimization theory to get the time-invariant measure - backpressure for network delay minimization, which provides a solid theoretical basis for the following reward design of reinforcement learning. Finally, the work combines traffic theory and control theory with reinforcement learning methods to design highly efficient network control algorithms. We will show that the proposed measure meets states' definition and reward in reinforcement learning and is relatively reliable and stable.

Section 4 tests the algorithm on the arterial and grid network case via simulation under various traffic volume and truck percentage cases, respectively, and compared with traditional algorithms (i.e., PASSER V) and reinforcement learning-based algorithms (i.e., RL-Queue).

Section 5 summarizes the results and conclusions, briefly discusses the limitations of the study and presents directions and potential improvement for future work.

2.0 LITERATURE REVIEW

Efficient trucking contributes to American economic viability. Trucking freight is ranked first among all the freight modes by both tonnage and value. Freight vehicles have significantly different characteristics in kinetic movement, economic values, and environmental effects. However, freight traffic's impact is rarely well-addressed in developing control strategies in the literature for either individual intersections or continuous intersections on arterials. On the other hand, today's technological developments offer unprecedented opportunities for new theories and models for traffic control. Equipment on the intersection or vehicles (e.g., cameras, detectors, GPS, etc.) can identify vehicle types and allow to take into account an increasing amount of real-time traffic data when adjusting signal timing to real-time traffic.

2.1 FIXED-TIME CONTROL WITH COORDINATION

Under fixed-time control, each controller has a predetermined timing plan. Fixed-time control is based on past traffic surveys and does not timely respond to real-time traffic conditions. Two strategies are generally employed to develop timing plans for an arterial street: progression-based methods (bandwidth maximization) and flow profile methods (delay and stops minimization). Green bandwidth maximization is essentially a geometry problem, which manipulates cycles in time-space diagrams to enable network intersecting coordination (Ficklin, 1969; Petterman, 1947). Morgan and Little et al. first formulate the bandwidth maximization optimization as a mixed-integer linear programming problem, develop MAXBAND to an arterial and network by adding cycle constraints (Little, 1966; Little et al., 1981; Morgan and Little, 1964). Gartner considers the specific features of each link and develops MULTIBAND, which optimizes all the signal control variables and bandwidth progressions on each roadway segment ("A multi-band approach to arterial traffic signal optimization," 1991; Gartner et al., 1990). PASSER V explicitly optimizes over the set of possible phase sequences to maximize progression or minimize total delay and works smoothly under both undersaturated and oversaturated traffic conditions (N.A. Chaudhary et al., 2002). P.D. Whiting first uses the delay-offset relationship and applies network topology theory to derive the network offsets (Hillier and Holroyd, 1965). The method is improved by incorporating disaggregate and dynamic programming techniques (Allsop, 1968; N.H.Gartner, 1972). Example of flow profile method includes TRANSYT-7F (Robertson, 1969) and SYNCHRO.

2.2 ADAPTIVE CONTROL

Traffic patterns depend on various external factors such as time, weather, and unpredictable situations such as accidents. These factors used to be indirectly considered in the adaptive traffic control system. Numerous adaptive systems have arisen over the past decades. SCOOT is a centralized system based on data collected from far upstream detectors. It uses the TRANSYT optimization method and prediction algorithm to produce cycles, offsets. It splits to

maintain the saturation rate of the intersection around the "ideal" value, but it suffers extensive calibration work (Hunt et al., 1981; Stevanovic et al., 2009). OPAC eliminates loops, offsets, and split constraints. Instead, OPAC maximizes the number of vehicles passing through an intersection by solving dynamic programming of the link state. OPAC maximizes performance by continually optimizing the system rather than periodically updating local controller settings (Gartner, 1982; Gartner et al., 2002). RHODES uses data collected from upstream and stop-line detectors for each approach to calculate loads on links and predict future platoon sizes, route choice, and network geometry(Head et al., 1992; Mirchandani and Head, 2001). Varaiya introduced the max pressure (MP) algorithm to reduce the risk of over-saturation and maximize the network's throughput by minimizing the pressure for a signalized network with multiple intersections. The 'pressure' of a phase is defined as the difference between the total queue length on incoming and outgoing approaches. Green time is given to phases with the most pressure to release (Varaiya, 2013). Although the algorithm requires only queue information at the intersection and has been tested in simulation under various cases, it still relies on assumptions to simplify the traffic condition and does not guarantee optimal results in the real world ("Adaptive Max Pressure Control of Network of Signalized Intersections," 2016; Wei et al., 2019a). DORAS-Q is a real-time, traffic responsive control applied to isolated intersections. When making a switch decision, the controller chooses the phase with the highest efficiency, calculated based on the existing queues, short-term predictions for the current approach arrivals rates, and average historical arrival rates for other phases. DORAS-Q is much less data demanding but does require knowledge of the existing queues (Wang et al., 2017).

2.3 CONTROL WITH REINFORCEMENT LEARNING

Recently, there is an increasing interest in academia to apply reinforcement learning technique to improve control problem, especially in network control problem (Mannion et al., 2016). The application of reinforcement learning (RL) methods opens a new window for solving the traffic signal control problem. There are extensive attempts to improve traffic signal control performance by reinforcement learning to outperform the traditional transportation methods (Abdoos et al., 2014) (Abdulhai et al., 2003) (Brys et al., 2014) (El-Tantawy et al., 2013) (Chen et al., 2020).

The reinforcement learning is capable of automatically learning high-quality control policies by interacting with the environment without an explicit performance model and with little system-specific knowledge. Meanwhile, the RL agent can automatically learn control policies from the observed data with little system-specific knowledge or unrealistic assumptions. The policy can be manually evaluated before implementation and be easily transferred. A key question is how to formulate a problem under RL's framework, i.e., the state, action and reward definition. This is a scientifically difficult but practically worthy effort.

Various kinds of elements have been proposed to describe the environment state in the traffic signal control problem, such as queue length, waiting time, volume, speed, position of vehicles, delay, phase, and duration. The criterion for a good design of states and rewards is to enable the agent to extract useful information and direction of optimization from the environment (Wei et al., 2019c). Also, long short-term memory (LSTM) layer is added to the training network to represent the spatial-temporal characteristics of the traffic state (Li et al., 2020). The easiest way

is to define reward as a weighted sum of state components such as queue length, waiting time, and delay(Wei et al., 2018). There is a rich literature trying to find the optimal reward function (Yau et al., 2017)(Teo et al., 2014) (Araghi et al., 2013)(Jin and Ma, 2015). However, there are additional challenges to resolve about how to take freight vehicles into consideration.

The problem is two-fold. First, one common issue of the current RL-based traffic signal control approaches is that the setting is often heuristic and lacks proper theoretical justification from transportation literature. Thus, there is a gap or need to connect reward with measurement endorsed by transportation theory that can be effectively observed after an action. Potential ideas include link congestion (Xu et al., 2020) with difference between upstream and downstream flows (Wei et al., 2019b).

Second, although the ultimate goal is to minimize the delay or travel time of all vehicles trespassing the intersection or intersections, the delay or travel time is hard to use as an effective reward function in RL. The delay or travel time of a vehicle may hardly be directly observed. Although the link performance function (BPR function) may provide an estimate of the delay, the delay can be influenced by several other factors such as free-flow speed, platoon dispersion, travel patterns, and vehicle component. Model's parameter adjustment may result in inaccurate estimation and thus lead to unsatisfactory performance of real-time signal control. All of the mentioned factors may bring additional randomness and map the same states to dramatically varying total rewards, thus fail the model from convergence.

2.4 SUMMARY OF OPTIMIZATION OBJECTIVES

Optimization of traffic signal control, as in other mathematical applications, uses an objective function to determine the optimal solution from a set of feasible groups, thereby maximizing (or minimizing) some metric. Traffic signal control's objective is to facilitate vehicles' efficient movement through the intersection or a roadway network. Various measures have been proposed to quantify the intersection or network's efficiency from different perspectives, such as maximizing the green bandwidth on major arteries. However, most of them fall into the following categories: travel time, delay, queue length, number of stops, and throughput (Wei et al., 2019c).

Common goals are either to minimize the average travel time of vehicles or to maximize the total number of vehicles through the network. These two correspond to travel time and throughput, respectively. Another similar measurement is the total delay, which is the time a vehicle has traveled within the environment minus the expected travel time. Besides, the number of stops and total queue length is the description of the intersection states.

The typical approach that transportation researchers take is to cast traffic signal control as an optimization problem under certain assumptions about the traffic model, e.g., vehicles come in a uniform and constant rate. In transportation studies, the inaccuracy arises from many assumptions in the models and approximations used to measure critical parameters. While many of the methods and models discussed above may provide the best solution within the defined space, the optimal values are subject to constant change due to the strong assumptions and the traffic flow's non-stationary nature. For simplicity without loss of generality, the research will convert truck to passenger vehicle equivalent by conversion factor in the simulation

(Transportation Research Board, 2016). The report will focus on addressing the near-optimal control mechanism and evaluating the effectiveness and robustness of the designed algorithm.

3.0 METHODOLOGY

Traffic is a random process in terms of timing, volume, route choice and vehicle following, etc. therefore, traffic control is a complex process. The early literature often approaches the control problem by assuming a constant and deterministic traffic in order to develop approximation models. Models gradually progress by explicitly dealing with random traffic. We first briefly introduce Lyapunov optimization, a technique widely used in stochastic optimization problems for optimal control before we present our control methods.

3.1 LYAPUNOV OPTIMIZATION

Lyapunov optimization refers to the use of a Lyapunov function to optimally control a dynamical system. The Lyapunov function is a non-negative scalar measure of multidimensional state vectors describing the system. Usually, the function is defined as whose value becoming larger as the system moves to an undesired state. System stability is achieved by taking control actions that make the Lyapunov function drift in the negative direction towards zero. The core idea of Lyapunov optimization is to decompose a long-term optimization index with long-term constraints into sub-optimization problems according to time slices. Tassiulas and Ephremides first bring the Lyapunov techniques into queuing network control problems and propose stable routing and scheduling policies, backpressure routing, and max-weight scheduling algorithms, which stabilize the network whenever possible. It is worth noticing that the policy only requires knowledge of the current network states, and they do not require knowledge of the probabilities associated with future random events (Neely et al., 2005; Tassiulas and Ephremides, 1993, 1992). The following is a synopsis of it.

Consider a network of N queues and let $\boldsymbol{\Theta(t)} = \big(\Theta_1(t), \dots, \Theta_N(t)\big)$ be the queue backlog vector at any time $t > 0$, where $\Theta(t)$ is a vector of actual queues in the network. Assume the $\Theta(t)$ vector evolves over time slot $t \in T$. As a scalar measure of the "size" of the vector $\Theta(t)$, a quadratic Lyapunov function $L[\Theta(t)]$ is defined as $\sum_{n=1}^{N}[\Theta_n(t)]^2$, and the expected change in the Lyapunov function over one slot, Lyapunov drift is defined as

$$\Delta\big(\boldsymbol{\Theta(t)}\big) = E\big[L\big(\boldsymbol{\Theta(t+1)}\big) - L\big(\boldsymbol{\Theta(t)}\big)\big] \tag{1}$$

Consider the quadratic Lyapunov function and assume $E[L(0)] < \infty$. Suppose there are constants $B > 0$, $\epsilon \geq 0$, such that the following drift condition holds for all slots $\tau \in T$ and all possible $\Theta(\tau)$:

$$\Delta\big(\Theta(\tau)\big) \leq B - \epsilon \sum_{n=1}^{N} \Theta_n(\tau) \tag{2}$$

Then all queues $\theta_n(t)$ are mean rate stable, more strictly, if $\epsilon > 0$, all queues are strongly stable and

$$sup \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{n=1}^{N} E[\Theta_n(\tau)] \leq \frac{B}{\epsilon} \tag{3}$$

3.2 BACKPRESSURE

The algorithm is motivated by backpressure routing introduced in the network control (communication and grid) of electrical engineering (Neely et al., 2005; Tassiulas and Ephremides, 1992; Wongpiromsarn et al., 2012). One of the compelling features of backpressure routing is that it leads to maximum network throughput based on directly observable variables and requiring little knowledge about vehicle-specific information, which guarantees ease of and reliability of the implementation. The algorithm has been proven to inherit a key property of backpressure routing, maximizing network throughput, even if the signal at each node is determined in a distributed and independent manner. Such a distributed system reduces the computational difficulties and avoids the curse of dimensionality. Furthermore, it does not require any knowledge about vehicle-specific information (e.g., OD info, real-time arrival prediction).

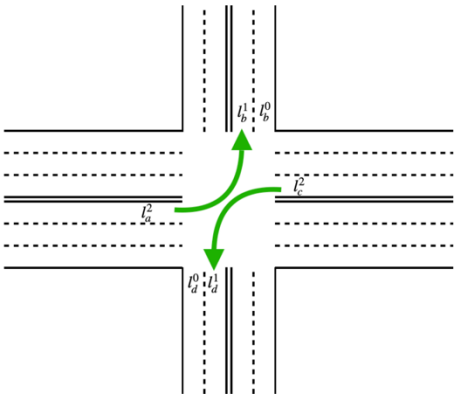


Figure 1: Permitted flow movement for left-turn phase on major arterial

Consider a network with N intersections and time-varying flow, each intersection $i \in 1,2, \dots, m$ consist of $\mathcal{L}$ links, and each link corresponds to a phase, where $\mathcal{L} = \{l_1, l_2, \dots, l_m \dots, l_M\}$. Each link l_m consist of k lanes, denoted as l_m^k. There are M links in the network in total. For simplicity, we omit the lane marks in the following deduction. Flow from link l_a to link l_b denotes the corresponding lane flow through the intersection plotted in **Figure 1**.

Each intersection i can be described by a tuple (M_i, P_i, Z_i), where $M_i \subseteq L^2$is a set of all possible movements through junction i; P_i is a set of all the possible signal phases of intersection i; $\phi_{ij} \in P_i$ is the phases controlling the approach k at intersection i. Each flow movement through the intersection i is defined by a pair (l_a, l_b), where $l_a, l_b \in \mathcal{L}^2$. S_i is the set of the traffic state. For the simplicity of implementation, the traffic state here refers to a state that can be easily and quickly observed by the detectors at the intersection. For example, queue length, vehicle location. Each flow movement through the intersection i is defined by a pair (l_a, l_b), where $(l_a, l_b) \in \mathcal{M}_i$. The pair (l_a, l_b) denotes the approach from link l_a thorough intersection i to link l_b, which is plotted in **Figure 1**.

The signal controller works in a discretized time horizon $[0, T]$. At each time step t, vehicles may enter and exit the link. For each link $l_i \in L$, $f_i(\phi_{ij}, l_a, l_b, z)$ gives the flow that to go from l_a to l_b through intersection i under state z when phase ϕ_{ij} is activated. With link length and density known, the link flow can be estimated by the Greensheild model:

$$f(t) = v_f d(t) - \left[\frac{v_f}{k_{jam}}\right] d(t)^2 \tag{4}$$

Where,

v_f is the free-flow speed,

$d(t)$ is the density on the link at time step *t*, d_{jam} is the jam density.

Define $W_{ab} = \Theta_a(t) - \Theta_b(t)$ as the differential queue backlog between link l_a and link l_b. Then, for each phase ϕ_{ij}, the backpressure at time step t is:

$$D_\phi(t) = \sum_{\phi_{ij} \in P_i} W_{ab} f_{ab}(\phi_{ij}, l_a, l_b, s_t) \tag{5}$$

We now leave the backpressure here and conduct the Lyapunov drift analysis of the signal-controlled network. Define $L(\Theta) = \sum_i \Theta_i^2$ as the Lyapunov function, representing a scalar measure of the network congestion. For a given control policy and network at time step t, the Lyapunov drift is:

$$\Delta(\Theta(t)) = E[L(\Theta(t+1)) - L(\Theta(t))] \tag{6}$$

This scheme does not require knowledge of the individual vehicle's real-time arrival rates or OD-specific information. Previous literature has shown that backpressure routing leads to maximum network throughput (Neely, 2010).

For any control policy, the Lyapunov drift at any time step t satisfies:

$$\Delta\big(\Theta(\tau)\big) \le B - 2\big[\Phi\big(\Theta(t)\big) - \Gamma\big(\Theta(t)\big)\big] \tag{7}$$

where,

$$B = (z_{max}^{out})^2 + \left(z_{max}^{in}\right)^2 \tag{8}$$

$$\Phi\big(\Theta(t)\big) = \sum_{i}^{\square} Q_i(t)\, E[\sum_{b}^{\square} z_{ib}^{out}(t) - \sum_{a}^{\square} z_{ai}^{in}(t)\,|\boldsymbol{\Theta}(t)]\} \tag{9}$$

$$\Gamma\big(\Theta(t)\big) = \sum_{i}^{\square} \Theta_i(t)\, E[\Theta(t)] \tag{10}$$

z_{max}^{out} and z_{max}^{in} are the max input and output flow, respectively. Notice that $\sum_i^{\square} \Theta_i\,(t)\big[\sum_b^{\square} z_{ib}^{out}(t) - \sum_a^{\square} z_{ai}^{in}(t)\big] = \sum_{ab}^{\square} f_{ab}\,(t)[\Theta_a(t) - \Theta_b(t)]$, therefore, for every t, we find the connection between the Lyapunov drift and the backpressure. The difference between queue is the pressure. Backpressure can be interpreted as the flow weighted pressure. **Equation 9** becomes the backpressure defined in **Equation 5**. Therefore, when the network is stable, minimizing the Lyapunov drift is equivalent to maximize the backpressure, thus maximizing the network throughput. The network capacity region Λ is the closure of the set of all matrices λ_i that can be stably supported over the network, considering all possible algorithms, it has proven that no control algorithm can achieve stability beyond the set Λ, even if the entire set of future events is known in advance (Tassiulas and Ephremides, 1992). The signal control problem assumes that the road network is not beyond its capacity, which is a reasonable assumption. If it exceeds its capacity, an overpass should be considered at the intersection. The capacity region of the network is given by the set Γ, such that there exists a policy that makes the network stable and has

$$\lambda_i + \epsilon \le \sum_{b}^{\square} z_{ib}^{out}(t) - \sum_{a}^{\square} z_{ai}^{in}(t) \tag{11}$$

Also notice in **Equation 9**, we have

$$\Phi\big(\Theta(\tau)\big) = \sum_{i}^{\square} \Theta_i(t) \left[\sum_{b}^{\square} z_{ib}^{out}(t) - \sum_{a}^{\square} z_{ai}^{in}(t)\right] \ge \epsilon \sum_{i}^{\square} \Theta_i(\tau) \tag{12}$$

Therefore, to minimize the $\Phi\big(\Theta(t)\big)$ is equivalent to maximize the backpressure, and thus limiting the size of weighted queue backlog at time step t, $\sum_i^{\square} \Theta\,(t)$. We assume there exists a state-only policy $\alpha^*(t)$, which satisfies:

$$E[Y_i(t)] = E[Y_i(\alpha^*(t), s_t)] \le \epsilon \quad \forall i \in \{1,2,\dots,K\} \tag{13}$$

Where,

$$y_i(\alpha^*(t), s_t) = \sum_{b}^{\square} z_{ib}^{out}(t) - \sum_{a}^{\square} z_{ai}^{in}(t) \tag{14}$$

$$f_{ij}(t) = f_{ij}(\alpha^*(t), s_t) \tag{15}$$

The state-only policy refers to the optimal action $\alpha^*(t)$ is chosen by relying only on the observed state s_t in each time slice *t*. We can show that:

$$\Delta\big(\Theta(\tau)\big) \leq B - 2\sum_{i}^{\square} \Theta_i(t)\, E[Y_i(t)] \tag{16}$$

$$E\big[\Delta\big(\Theta(\tau)\big)\big] \leq B - 2\epsilon \sum_{i}^{\square} E\,[\Theta_i(t)] \tag{17}$$

Summing the **Equation 17** from $\tau = 0$ to $\tau = T$ and notice the definition of Lyapunov drift $\Delta\big(\Theta(t)\big) = \Theta(t+1) - \Theta(t)$ and assume $\Theta(0) = 0$, we have

$$0 \leq E[\Theta(T)] \leq tB - 2t\epsilon \sum_{\tau=0}^{t-1} \sum_{i}^{\square} E[\Theta_i(t)] \tag{18}$$

$$\frac{1}{t}\sum_{\tau=0}^{t-1} \sum_{i}^{\square} E\,[\Theta_i(t)] \leq \frac{B}{\epsilon} \tag{19}$$

Notice that B is defined in **Equation 8**. Little's theorem provides a base for analyzing the queueing delay. The theorem states that when a network reaches a steady state, the average number of jobs in a queue is equal to the product of the average arrival rate of the jobs and the average time a job is kept in the queue. By Little's theorem, minimizing delay achieving maximizing the throughput.

3.3 Q-LEARNING

The Q-learning approach to solving the Bellman equation:

$$V^{\pi}(s) = \sum_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} p(s'|s,a)[r(s,a,s') + \gamma V^{\pi}(s')] \tag{20}$$

$$Q^{\pi}(s,a) = \sum_{s' \in \mathcal{S}} p(s'|s,a)\left[r(s,a,s') + \gamma \sum_{a' \in \mathcal{A}} \pi(a'|s') Q^{\pi}(s',a')\right] \tag{21}$$

Q-Learning poses an idea of assessing the quality of an action that is taken to move to a state rather than determining the possible value of the state being moved to. The Q-learning algorithm makes the following update:

$$Q(s,a) = Q(s,a) + \alpha[r(s,a,s') + \gamma max_{a'} Q(s',a') - Q(s,a)] \quad (22)$$

The quantity in square brackets in **Equation 22** is exactly zero when a' is the optimal action to take under states s'. In other word, $Q(s',a')$ is the optimal action-state value pair. The quantity in the square brackets can be interpreted as the "Bellman error", the error term describes how far off the target quantity $r(s,a,s') + \gamma max_{a'} Q(s',a')$ is from the estimates $Q(s,a)$ in the current step. Q-learning algorithm iteratively updates $Q(s,a)$ by **Equation 22** to reduce the Bellman error until reach a converged solution.

However, to store all the action-state pair value s, Q-learning requires a finite state and action space where it is possible to maintain a table lookup the estimated Q-value. However, it is not always the case where we have finite states and/or actions. When we have an infinite state space and/or action space, specifically in developing the traffic signal control policy, then it becomes impossible to store all the value pairs. An elegant way is to use function approximation to generalize across states and store the approximation function, which is typically done using a deep neural network due to their expressive power. And thus, we will introduce Deep Q-Network (DQN) in the next subsection.

3.4 REINFORCEMENT LEARNING AND DOUBLE DQN

The basic idea behind many reinforcement learning algorithms is to estimate the action-value function, by using the Bellman equation as an iterative update, e.g. $Q_{t+1}(s,a) = E\left[r(s,a,s') + \gamma \max_{a \in A_s} Q_t(s',a') \middle| s,a\right]$. Such value iteration algorithms converge to the optimal action-value function, $Q_t(s,a) \rightarrow Q^*(s,a)$ as $t \rightarrow \infty$ (Mnih et al., 2015; Richard S. Sutton and Andrew G. Barto, 2015).

In DQN, the experience replay memory and the target network were decisive in allowing the neural network to learn the tasks through RL. Their drawback is that they drastically increase the sample complexity and overestimate the target Q value. This over-estimation is inevitable in regular Q-learning, and therefore the double DQN is proposed (Hasselt, 2010). Applying double learning to DQN is straightforward: there are already two value networks: the trained and target networks. Instead of using the target network to both select the greedy action in the next state and estimate its Q-value, here the trained network weights θ is used to select the greedy action $a^* = argmax_{a'} Q_\theta(s',a')$ while the target network only estimates its Q-value. This induces only a minor modification of the DQN algorithm and significantly improves its performance and stability. The main idea of double DQN is to train independently two value networks: one will be used to find the action with the max Q-value and estimate the Q-value itself. Even if the first network chooses an over-estimated action as the greedy action, the other might provide a less over-estimated value for it, resulting in a better solution.

3.5 BACK PRESSURE WITH REINFORCEMENT LEARNING

One major issue of current RL-based traffic signal control approaches is that the setting is often heuristic and lacks proper theoretical justification from transportation literature. Common goals are either to minimize the average travel time of vehicles or delay. However, these goals are either "delayed" or heavily rely on estimation, resulting in a mismatch between state and reward, which leads to poor performance. The algorithm put forward in this section is based on RL but theoretically grounded by the backpressure method mentioned above. Backpressure has the property of being based on real-time observable quantities, a perfect fit for the state-reward pair design in reinforcement learning. The only information required is the queue backlog on each lane of the intersection in the roadway network, the lane density, and lane length. Detectors can acquire the first two. Greenshields model can calculate the flow with a known density. The RL formulation is demonstrated as the following:

State: current phase ϕ, the total number of vehicles and stopped vehicles (speed < 0.1 m/s, or 0.22 mph) on each incoming lane (l_a) and outgoing lanes (l_b).

Action: at each time t, each agent chooses a phase as its action a_t from action set A, indicating the traffic signal should be set as current phase ϕ. Each action candidate a_i is represented as a one-hot vector.

Reward: the reward for an intersection is the backpressure. The backpressure of an intersection i is defined as the absolute sum of the backpressure over all phases, denoted as:

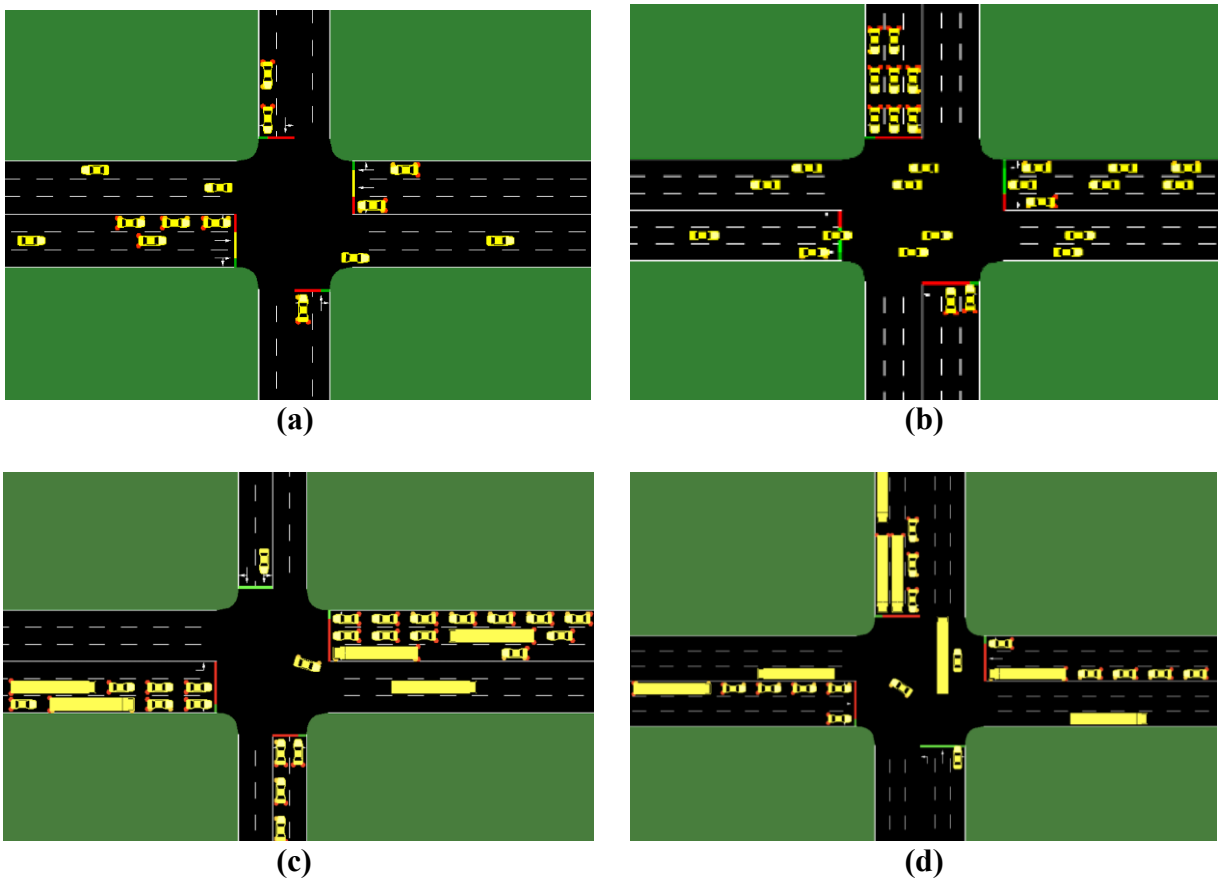
$$R_i(t) = -\sum_{\phi \in P_i} \left| D_\phi(t) \right| \tag{23}$$

In RL, the long-term reward is the objective for optimization, and the solution is derived from the trial-and-error search. We adopt Double-DQN as a function approximator. To stabilize the training process, we maintain an experience replay memory by adding the new data samples and removing the old samples occasionally. Periodically, the agent will take samples from memory and use them to update the network.

4.0 SIMULATION

4.1 SIMULATION SETTINGS

Numerical tests are conducted on two types of networks: a single corridor and a grid network. The reason for separately testing on a single corridor is that single corridors are often the major means of dealing with urban traffic. We have utilized the SUMO 1.8 micro-simulator in conjunction with TraCI (Traffic Control Interface) 1.8 for modeling the case. SUMO (Simulation of Urban MObility) is an open-source, microscopic and continuous traffic simulation package designed to handle large traffic networks simulation with a large set of tools for scenario creation (Lopez et al., 2018). SUMO allows us to create a traffic simulation environment and track every vehicle. TraCI implements RL-based real-time signal control possible. The RL agent is built and trained in Pytorch 1.7.1 and Python 3.8.



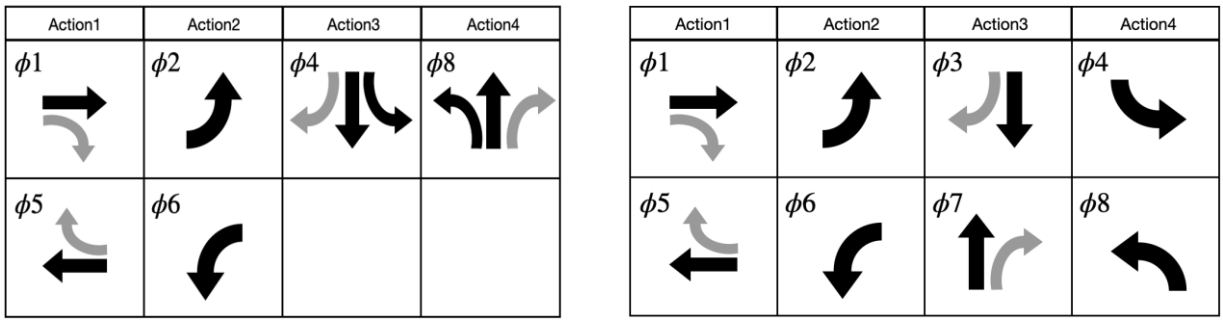
(a) **(b)** **(c)** **(d)**

Figure 2: Intersection simulation environment

Figure 2 (a) and **(c)** presents the layout of an individual intersection of major and minor arterial without and with truck volume, respectively, in the simulation network. **Figure 2 (b)** and

(d) present the intersection of two major arterials, without and with truck volume, respectively. The link on minor arterial at each intersection consists of two through lanes. Each of the through lanes also servers the turning traffic. The link on major arterial at each intersection consists of one left-turn lane and two through lanes. One of the through lanes also servers the right-turning traffic. In reality, there are singular dominating corridors that can be easily identified.

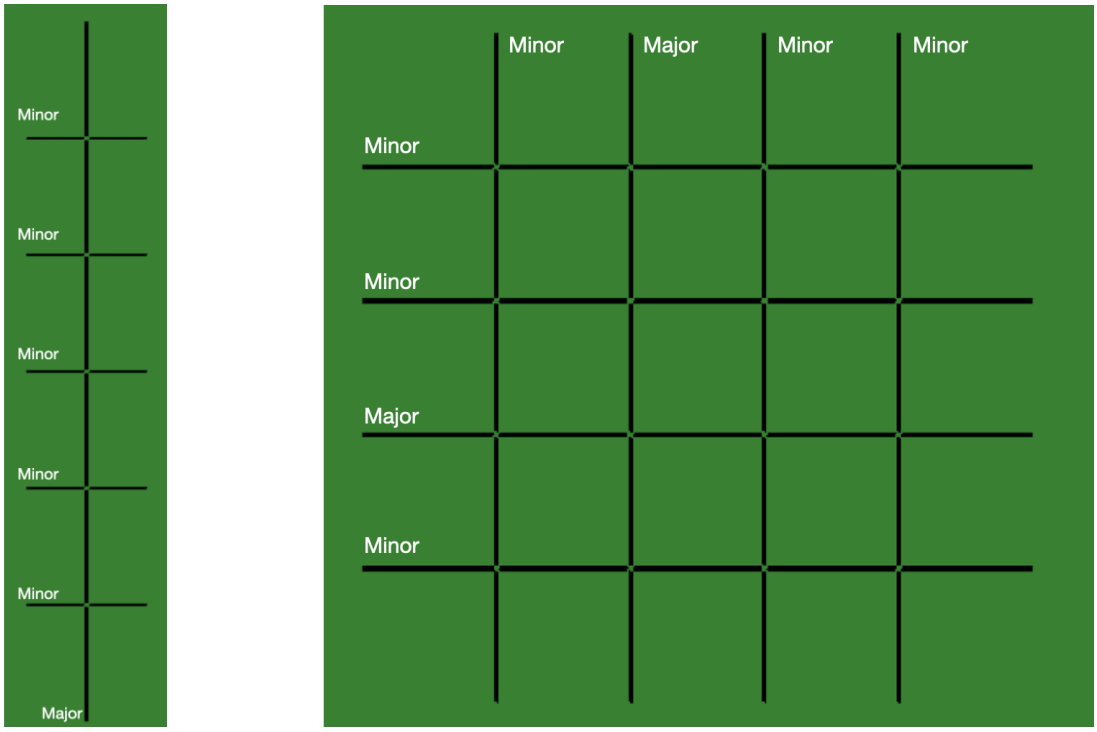
Figure 3 (a) presents the available phase timing plan for the intersection of major and minor arterial, and **(b)** shows the one for the intersection of two major arterials. The phase plan also contains the available action for selecting the RL agent. The amber interval is set as 5 seconds and represents the time between two consecutive phases to clear the intersection, consisting of 3 seconds yellow and 2 seconds all-red interval. The min green time is 5 seconds, and the max green time is 30 seconds. The ring-and-barrier diagram is for illustrative purposes and presents the phase plan for the simulation. In reality, the RL algorithm neither requires four phases nor a fixed sequence.



(a) **(b)**

Figure 3: Available phases in traffic signal control of the simulation

Figure 4 illustrates the arterial **(a)** and grid network **(b)** used in the simulation. The test arterial consists of five intersections and the grid network contains $4 \times 4 = 16$ intersections. The arterial in the numerical test consists of one major arterial road with higher traffic volumes and five minor roads with lower volumes. The minor street crossings spaced 1640 ft (500 m) along the major arterial with free-flow speed v_f 50 mph. The grid network in the simulation makes of two major arterial roads with higher volumes and three minor roads with lower volumes in each direction. The distance between roads, free-flow speed, and the normal travel times are the same with the arterial. Three traffic scenarios, high, medium, and low, are used for the test, as indicated in **Table 1**.



(a) **(b)**

Figure 4: Arterial and grid network environment

Table 1 Traffic volume in the simulation

Traffic Scenario	Major Roads(veh/h)	Minor Roads(veh/h)
Low	500	200
Medium	900	300
High	1300	400

Also, in each scenario, the effect of different truck ratios (0%, 10%, 25%, and 40%) on each control algorithm was tested simultaneously for the same major and minor traffic volume scenarios. The research will convert truck to two passenger vehicle in the simulation (Federal Highway Administration, 2017). The vehicle type defaults are shown in **Table 2**.

Table 2 Vehicle Type parameter defaults in the simulation

Vehicle Type	Length Width Height	MinGap	Acceleration	Deceleration	Emergency Deceleration
Passenger	5 m 1.8 m 1.5 m	2.5 m	2.6 m/s^2	4.5 m/s^2	9 m/s^2
Truck	16.5 m 2.55 m 4 m	2.5 m	1.1 m/s^2	4 m/s^2	7 m/s^2

4.2 TESTED ALGORITHMS IN SIMULATION

We compare our model with the following two categories of methods: non-machine learning transportation methods and RL-based methods. Non-machine learning methods include Fixed timing plan with coordination, DORAS-Q, MaxPressure and Backpressure. We directly optimize the waiting time and average queue length by double DQN and use them as baselines of the reinforcement learning based algorithms. All algorithms are fine-tuned. And all RL-based algorithms are trained by Double DQN.

Fixed-time: Fixed-timing plan and offsets optimized with PASSER V. Fixed timing plan with green wave progression is the most classical approach achieving coordination on arterial in practice.

DORAS-Q: DORAS-Q is designed for isolated intersection control and may be applied to the network as a distributed control system in which each intersection only optimizes its control and the entire system adapts gradually, it requires the existing queue length, short-term (usually 5 seconds) and the average historical arrival rates for each phase to estimate the switch-to efficiency and phase efficiency. Then decide on changing or keeping the current phase based on the discharge efficiency.

Max Pressure (Varaiya, 2013): Max pressure defines the differences of queue between the current and the downstream intersection as pressure of the phase, and greedily chooses the phase with the maximum pressure.

Back pressure: Consider the flow on the link, greedily chooses the phase with the maximum backpressure.

RL-WaitingTime: Directly define the waiting time as the reward function. Train RL agent to minimize the waiting time of all vehicles in the network.

RL-Queue: Directly define the average queue length as the reward function. Train RL agent to minimize the average queue length of each link in the network.

RL-MaxPressure: Directly define the total network pressure as the reward function. Train RL agent to maximize the negative of absolute maxpressure.

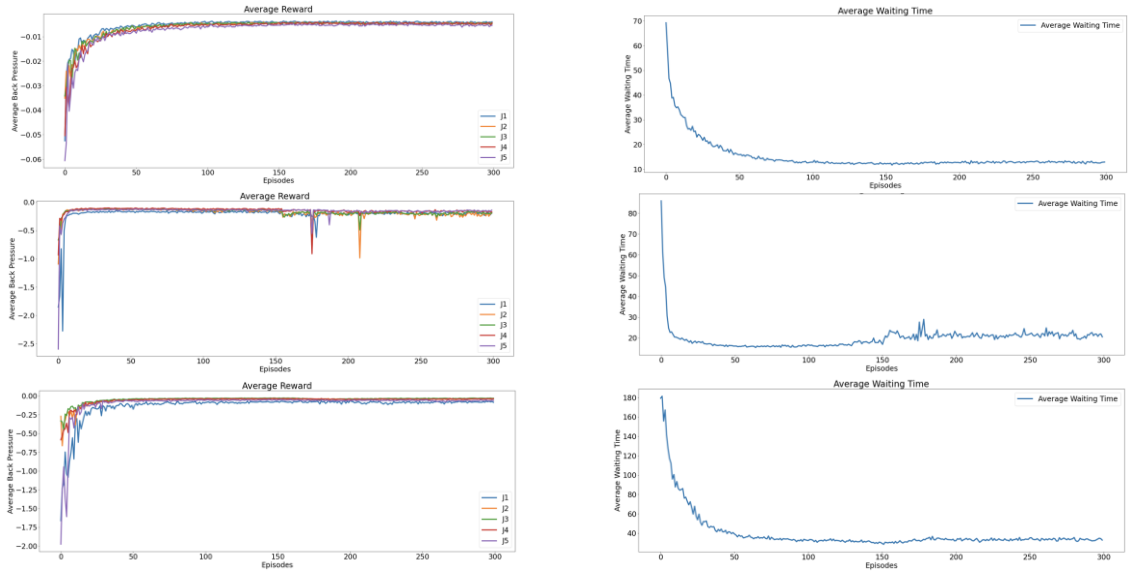
RL-BackPressure: Directly define the total network backpressure as the reward function. Train RL agent to maximize the negative of absolute backpressure defined in **Equation 23**.

All algorithms are fine-tuned. And all RL-based algorithms are trained by Double DQN. Section 4.3 illustrates the results of arterial case and section 4.4 presents the results of grid network case.

4.3 AGENT PERFORMANCE ON SCENARIOS WITH UNIFORM PASSENGER VEHICLE FLOW

Figure 5 shows the agent's performance and the fast convergence in the arterial environment during the training process. The horizontal axis of the figures reflects the episodes.

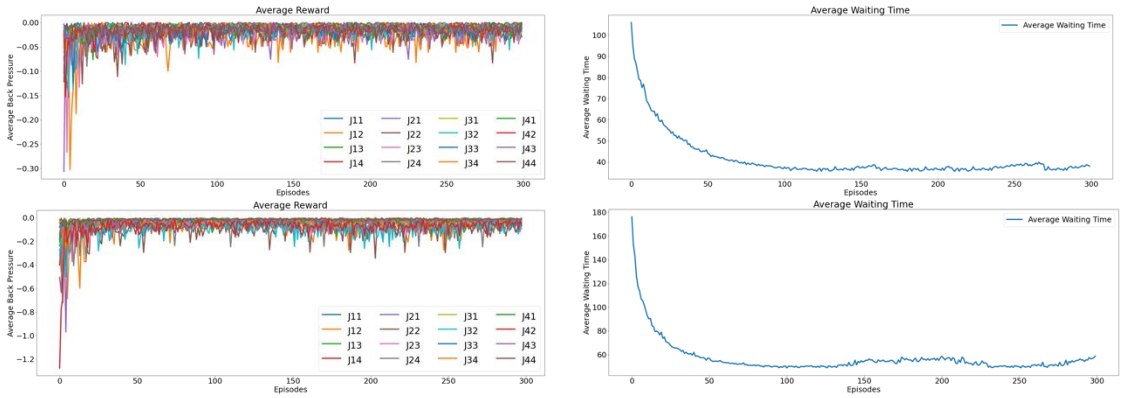
The top to the bottom of the figure, corresponds to low, medium, and high scenarios, respectively. The vertical axis of **Figure 5 (a)** reflects the average back pressure of an intersection, while the vertical axis of **Figure 5 (b)** reflects the average waiting time of each vehicle. The figures show that the training agent in the arterial environment converges after 100 episodes. **Figure 5** illustrates the convergence curve of our agents' learning process with respect to the average waiting time of each episode. Compared with the backpressure curves, we can see that the travel time is closely correlated with pressure. Convergence curve of average duration and our reward design (back pressure).

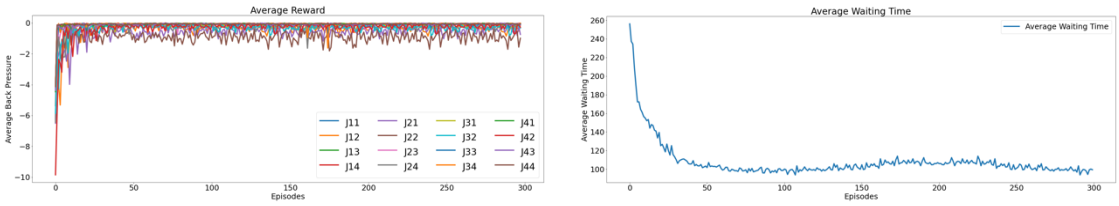


(a) **(b)**

Figure 5 Average backpressure and waiting time during training process over 300 episodes in arterial case

Figure 6 is the coverage curve of average backpressure and waiting time in the gird network case. The convergence trend is similar to that of the arterial case. The training processing converges around 100 episodes. Because the structure of the network is more complex than arterial so that the fluctuation after convergence is more unstable and violent. The correlation between backpressure and travel time can also be found in the grid network case.





(a) **(b)**

Figure 6: Average backpressure and waiting time during training process over 300 episodes in the grid network case

We compare our model with the following two categories of methods: non-machine learning methods and RL-based methods. Non-machine learning method includes Fixed timing plan with coordination, DORAS-Q and MaxPressure. Fixed timing plan with green wave progression is the most classical approach to achieving coordination on the arterial in practice. Fixed-timing plans and offsets are optimized with PASSER V. DORAS-Q (Wang et al., 2017) is designed for isolated intersection control, and may be applied to the network as a distributed control system in which each intersection only optimizes its control and the entire system adapts gradually, it requires the existing queue length, short-term (usually 5 seconds) and the average historical arrival rates for each phase to estimate the switch-to efficiency and phase efficiency. Then decide on changing or keeping the current phase based on the discharge efficiency. MaxPressure (Varaiya, 2013) defines the differences of the queue between the current and the downstream intersection as the pressure of the phase and greedily chooses the phase with the maximum pressure. All algorithms are fine-tuned. **Table 3** illustrates the results of the simulation.

Table 3 Average vehicle delay in arterial and grid network case with uniform passenger vehicle flow (in seconds)

	Low Volume		Medium Volume		High Volume	
	Arterial	Network	Arterial	Network	Arterial	Network
Fixed-time	30.93	93.73	38.06	139.87	89.82	191.16
DORAS-Q	23.25	72.77	36.64	84.82	76.64	148.92
Max Pressure	19.88	48.22	31.59	56.71	71.62	167.61
Back Pressure	18.99	46.94	26.92	56.49	56.86	145.04
RL-WaitingTime	15.61	39.73	34.16	69.27	80.13	140.84
RL-Queue	15.38	40.67	26.47	60.88	66.47	140.87
RL-MaxPressure	16.36	44.56	25.57	51.43	51.56	107.43
RL-Backpressure	**13.46**	**37.34**	**22.64**	**49.72**	**46.07**	**99.37**

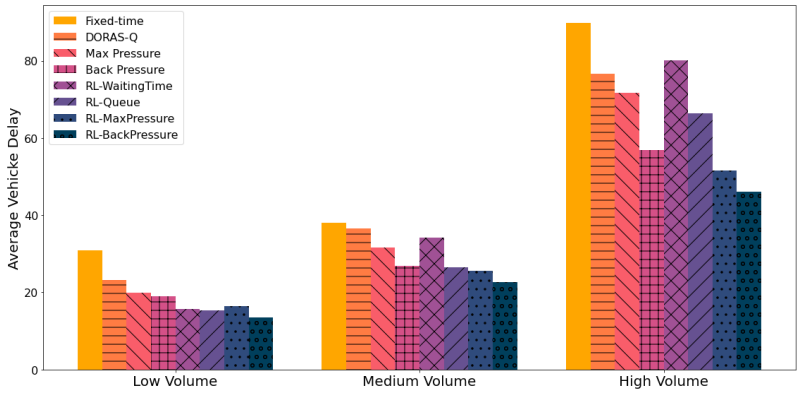


Figure 7: Average vehicle delay in arterial case (in seconds)

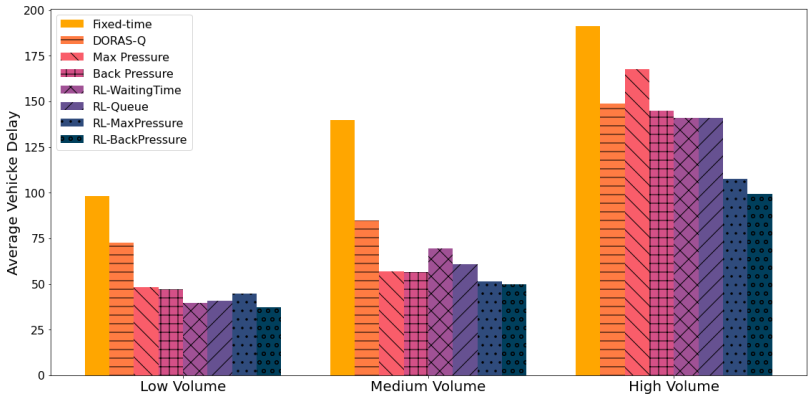


Figure 8: Average vehicle delay in grid network case (in seconds)

Figure 7 and 8 presents RL-Backpressure outperforms the other four signal control algorithms in both arterial and grid network cases. Not surprisingly, fixed-time control performs at the bottom, but it does not stop using it as a benchmark. Under all scenarios, DORAS-Q and MaxPressure outperform the fixed time control with coordination. RL baseline have satisfied performance in low volume scenarios, are exceeded by the RL-MaxPressure and RL-BackPressure under medium and high-volume cases.

4.4 AGENT PERFORMANCE ON SCENARIOS WITH TRUCK FLOW

We also investigate the effect of different truck ratios (0%, 10%, 25%, and 40%) on each control algorithm. All settings are the same as the uniform passenger vehicle flow case except for the various truck ratio. We also conduct the simulation in the low, medium, and high traffic flow scenarios. Specifically, the high traffic volume of 25% means that the traffic volume on the major/minor arterials remains at 1300/400 vehicles/hour, with 975/300 trucks/hour on the major/minor arterials and 325/100 vehicles/hour passenger vehicles on major/minor arterials. When calculating the queue, we assume a truck equals two passenger vehicles.

4.4.1 10% truck volume

Table 4 illustrates the results of the simulation under all scenarios with 10% of truck volume.

Table 4 Average vehicle delay in arterial and grid network case with 10% truck volume (in seconds)

	Low Volume		Medium Volume		High Volume	
	Arterial	Network	Arterial	Network	Arterial	Network
Fixed-time	36.96	99.26	52.72	153.79	132.38	226.39
DORAS-Q	28.87	74.82	49.76	96.09	130.47	184.25
Max Pressure	20.96	52.56	41.50	75.43	125.14	185.82
Back Pressure	19.87	54.61	37.14	77.58	113.65	165.09
RL-WaitingTime	**15.35**	45.99	40.76	89.58	108.69	143.35
RL-Queue	15.45	**43.56**	32.21	88.17	106.03	147.22
RL-MaxPressure	18.47	51.12	34.87	78.77	91.72	125.19
RL-BackPressure	17.82	46.73	**30.73**	**75.24**	**90.55**	**120.86**

Figure 9 and Figure 10 presents RL-BackPressure outperforms the other four signal control algorithms in both arterial and grid network cases. Not surprisingly, fixed-time control performs at the bottom, but it does not stop using it as a benchmark. Under all scenarios, DORAS-Q and MaxPressure outperform the fixed time control with coordination. RL baseline have satisfied performance in low volume scenarios, are exceeded by the RL-MaxPressure and RL-BackPressure under medium and high-volume cases.

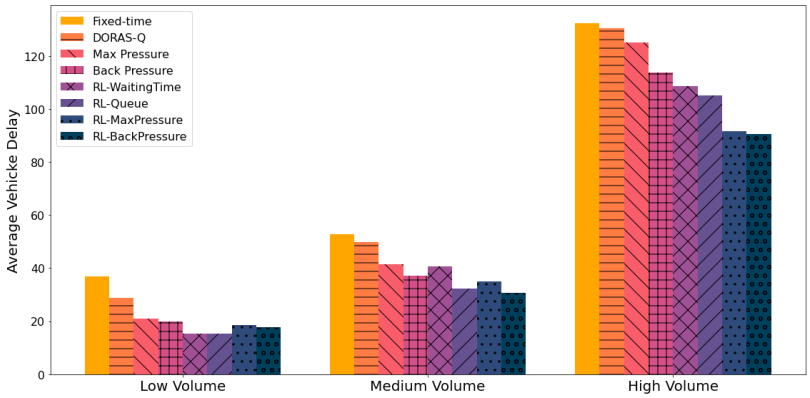


Figure 9 Average vehicle delay through the arterial with 10 % truck volume (in seconds)

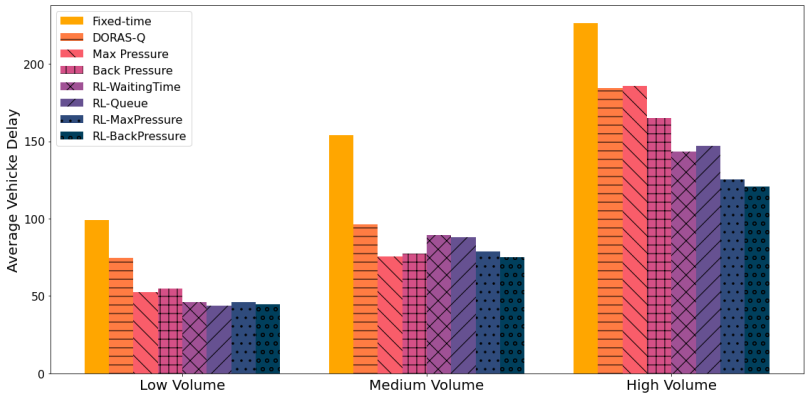


Figure 10 Average vehicle delay through the grid network with 10 % truck volume (in seconds)

4.4.2 25% truck flow

Table 5 illustrates the results of the simulation under all scenarios with 25% of truck volume.

Table 5 Average vehicle delay in arterial and grid network case with 25% truck volume (in seconds)

	Low Volume		Medium Volume		High Volume	
	Arterial	Network	Arterial	Network	Arterial	Network
Fixed-time	42.13	100.62	84.94	181.88	148.03	258.91
DORAS-Q	31.14	75.38	57.52	128.95	118.36	227.39
Max Pressure	23.18	59.65	41.92	109.28	122.27	207.33

Back Pressure	21.95	61.12	36.58	96.58	119.61	200.69
RL-WaitingTime	19.29	**35.61**	47.26	89.97	120.31	178.33
RL-Queue	**17.78**	36.23	35.39	87.11	119.78	176.39
RL-MaxPressure	19.39	51.47	39.01	88.06	115.39	172.83
RL-BackPressure	18.50	50.55	**34.58**	**85.84**	**113.48**	**165.39**

Figure 11 and Figure 12 presents RL-BackPressure outperforms the other four signal control algorithms in both arterial and grid network cases. Similar to the 10% truck volume case, DORAS-Q and MaxPressure outperform the fixed time control with coordination. RL baseline have outperformance other algorithms under low volume scenarios. RL-MaxPressure still have the best performance under medium and high-volume cases.

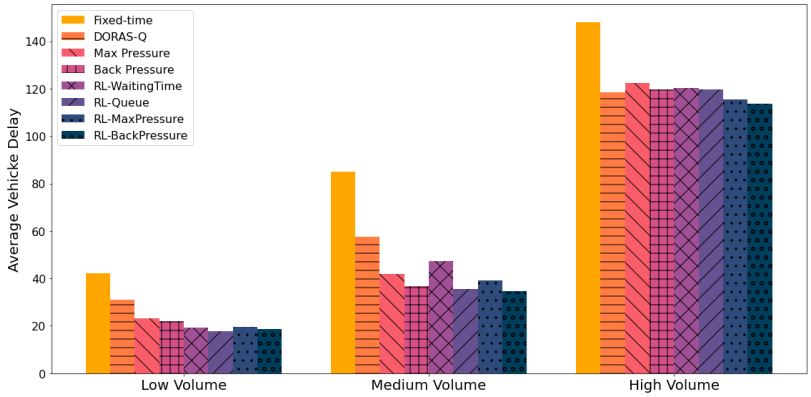


Figure 11 Average vehicle delay through the arterial with 25 % truck volume (in seconds)

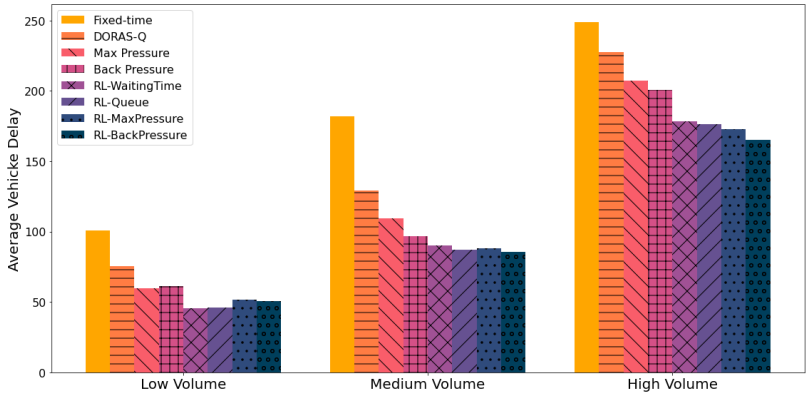


Figure 12 Average vehicle delay through the grid network with 25 % truck volume (in seconds)

4.4.3 40% truck flow

Table 6 illustrates the results of the simulation under all scenarios with 40% of truck volume.

Table 6 Average vehicle delay in arterial and grid network case with 40% truck volume (in seconds)

	Low Volume		Medium Volume		High Volume	
	Arterial	Network	Arterial	Network	Arterial	Network
Fixed-time	69.04	106.25	129.64	192.63	267.68	274.39

DORAS-Q	34.02	76.68	120.33	146.14	225.39	259.32
Max Pressure	27.93	59.80	127.96	128.93	239.37	254.30
Back Pressure	23.25	62.95	117.83	126.27	216.21	252.03
RL-WaitingTime	25.11	52.94	119.72	162.89	232.39	263.18
RL-Queue	23.74	51.14	114.71	165.79	232.79	247.28
RL-MaxPressure	22.92	50.46	120.05	125.82	216.39	235.32
RL-BackPressure	**22.18**	**48.06**	**111.05**	**119.37**	**208.72**	**228.07**

Figure 13 and **Figure 14** presents RL-BackPressure outperforms the other four signal control algorithms in both arterial and grid network cases. With higher percentage of truck volume, RL-MaxPressure have the best performance across all the scenarios.

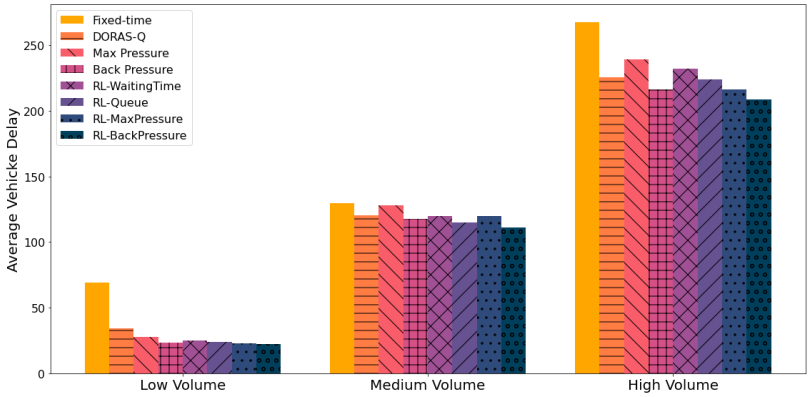


Figure 13 Average vehicle delay through the arterial with 40 % truck volume (in seconds)

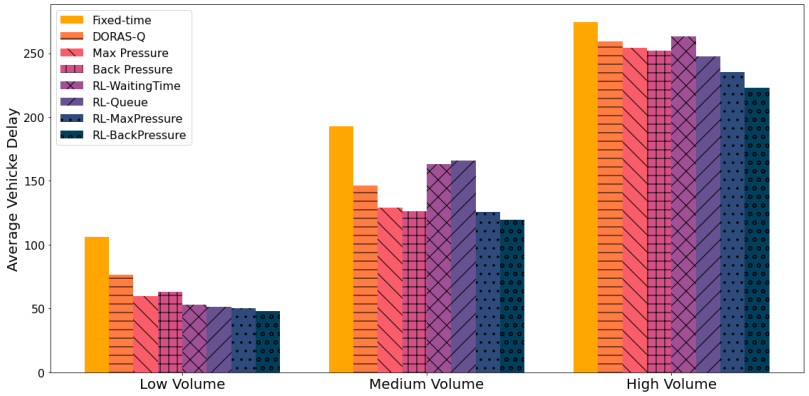


Figure 14 Average vehicle delay through the grid network with 40 % truck volume (in seconds)

4.5 DISCUSSION

We summary the simulation results in section 4.3 and 4.4. In general, considering the truck flow may increase the simulation may increase the total delay at the intersection. The higher the truck volume percentage, the more delay driver may experience at the intersection with the area. RL based algorithm have overall good performance in terms of total delay in the medium and high-volume scenarios. Also, the performance of the algorithms with the addition of reinforcement learning algorithms improves for the corresponding non-reinforcement learning algorithms. In

most scenarios, the RL-BackPressure algorithm outperforms all other comparable algorithms. The RL-BackPressure algorithm has excellent performance in the med and high-volume scenarios. RL baselines have the best performance in the low-volume scenario. We will analyze each algorithm's performance in detail.

Without a doubt, the green wave facilitates the vehicle's movement on the arterial and gird network with signalized intersections. However, even though the green wave is added to it, then the fixed timing design does not inherently consider the dynamic changes of traffic flow or responsiveness to the current situation at the intersection. Not surprisingly, fixed-time control performs at the bottom, but it does not stop using it as a benchmark.

DORAS-Q estimates the intersections' efficiency within a certain period and predicts future arrivals. The inaccurate estimation or prediction inevitably leads to flawed decisions. Even if the controller can make accurate predictions (e.g., during a fully connected vehicle), DORAS-Q optimizes based only on the predicted arrival stream information. On the one hand, the predictions cannot be 100 percent accurate, and in the absence of other information, there is no correction mechanism for the decision. Minor errors will gradually accumulate so that the decision does not match the future situation. Further, the vehicle will still interact with the controller, and this interaction will change the arrival stream, thus rendering the long-ago beyond-intersection predictions useless and degrading the decision. In addition, vehicles usually arrive in the form of a platoon. The current phase drops rapidly to zero after clearing the queue, allowing the signal to switch to the next phase based on DORAS-Q. Myopic switching frequently occurs during the signal control cycle, thus defeats the original design intent. However, it does not consider the coordination between intersections, and thus the performance has only slightly improved. In the grid network case, the DORAS-Q performs much better than fixed-time control because each intersection could utilize the arrival steam information from nearby intersections. Nonetheless, the performance is still not good enough due to the inaccurate prediction and errors accumulation. On the other hand, the truck volume may decrease the algorithm's performance compared with the uniform passenger vehicle flow case. Using conversion factor to convert truck to passenger vehicle in the queue length estimation may not work well in the algorithm. Although the algorithm is based on the queue length estimation, more truck characteristics may need to take into consideration and improve the mechanism of the algorithm.

MaxPressure utilizes the current information at the intersection, which overcomes the shortcoming of prediction. However, the algorithm greedily chooses the phase with the maximum pressure without utilizing historical knowledge so that the solution may be the local optimum rather than the global optimum. The more complex the road network, the more likely it is to converge to the local optimum. That is why the MaxPressure surprisingly under-performs DORAS-Q in the gird network case. In the simulation, either pure MaxPressure or pure back Pressure algorithm is susceptible to the flow pattern. With the same initial conditions, the difference in realizations is significant, especially under high-volume traffic. This inspires us to use the backpressure, whose discharge flow rate changes with the link's current flow. With the reinforcement added to it, the RL-MaxPressure, the deficiency is compensated by many learning iterations, resulting in an approximate optimal solution in the same state. Backpressure considers the potential flow through the node and refining the control theory in queuing networks. Neither of the pressure-like control requires input flow prediction or forecast. The only info required is the current state. A slight delay in the information collected will not affect the performance.

With truck volume in consideration, the truck volume may bring more impact or turbulence on the downstream traffic volume, which may be why the pressure series algorithm relatively underperforms in the low volume and low truck percentage scenarios. The addition of truck volume may increase the overall delay compared with the pure passenger flow case. There are two ways to explain the situation. First, a truck is usually longer than a passenger vehicle. A conversion factor or truck coefficient is typically used to convert the truck to passenger vehicle equivalent in transportation engineering. With the same traffic volume, the traffic flow with trucks will actually be higher than the equivalent pure passenger vehicle flow. Second, trucks are slower to accelerate and decelerate than passenger vehicles. When trucks approach or leave the intersection, the inhomogeneous may heterogeneity the traffic flow. The conversion factor may not be the same when the controller decides the phase. Also, the truck volume brings more disturbance to the coverage of the RL baseline algorithms. The results variation is more considerable than that of pure passenger vehicle flow case. And it takes more time to converge in each RL enhanced algorithm.

Considering that the traffic movement process at each intersection is stable, the system is also stable accordingly. In the main road environment without turnarounds, the actions taken by the RL agents do not create loops or block the network, making the imbalance between the intersections decrease and thus allowing efficient use of green light time. A common pseudo-refutation scenario is that if each intersection has the same long queue in each direction, then the pressure at each intersection is 0. Wouldn't the agent exacerbate the congestion? It is worth noting that this so-called refutation scenario can only exist in a single moment and is very unstable. Even in such a case, once a phase turns green, the backpressure in that direction will gradually decrease. After a certain point, the RL agent will choose to end the green light in that direction and give it to the next phase to earn a higher "reward", and the whole network system will evolve again in the direction of the lowest overall flow and pressure. Therefore, for a given period T, our RL agent can provide the maximum throughput, thus minimizing the travel time of all vehicles in the system.

5.0 CONCLUSION

Traffic signal control is installed to manage traffic flow, alleviate traffic congestion and increase road network efficiency. Deep reinforcement learning algorithms have recently become increasingly popular and have been widely recognized as an effective tool to solve the traffic control problem.

Overall, several measurements may be used for the network performance evaluation, including but not limited to the total travel time of vehicles in the network, average vehicle delay and the number of stops, and average travel speed. Several metrics are usually needed to validate each other for the network control performance evaluation. Thus, signal control's core problem is how to translate these well-known measurements (i.e., minimum delay, minimum stops) confidently into a timely observable, explicit controllable variable. This study proposes a new measurement to evaluate the network performance by directly translating general delay minimization into the maximization of intersection throughput based on Lyapunov optimization and then designing efficient traffic control algorithms, combining traffic theory and control theory with the reinforcement learning method.

The Lyapunov optimization is introduced in queuing control problems to reduce the impact of inaccurate prediction and increase the robustness to queuing. The state of the system is the total number of vehicles and stopped vehicles on each incoming and outgoing lane. The action is designed timing plan including four actions. The reward for the intersection is the backpressure to reduce the imbalance across the network queues. The Double DQN was adopted as a function approximator for a higher accuracy. The result shows that the RL-Backpressure performs better than fixed-time, DORAS-Q, MaxPressure, Backpressure, and RL-MaxPressure under varying volumes, especially under high volumes. RL based algorithm have overall good performance in terms of total delay in the medium and high-volume scenarios. Also, the performance of the algorithms with the addition of reinforcement learning algorithms improves for the corresponding non-reinforcement learning algorithms. Considering the truck flow may increase the simulation may increase the total delay at the intersection. The higher the truck volume percentage, the more delay driver may experience at the intersection with the area. The average travel time in the signal intersection of the grid network is higher than that of the arterial case due to the coordination of surrounding signals. Besides, the travel time and backpressure are closely correlated. As the pressure gradually approach to zero, the travel time also gradually decreases.

The addition of truck volume may increase the overall delay compared with the pure passenger flow case. Truck volume may bring more impact or turbulence on the downstream traffic volume, which may decrease the pressure series algorithm’s performance in the low volume and low truck percentage scenarios. However, RL-BackPressure is still robust in the high volume and high truck volume percentage case.

There are a few limitations to this study. First, the RL-BackPressure's performance is not outstanding for light traffic loads in the case of both arterial and gird networks. It worth

investigating how to improve the algorithm when little network effect exists. Second, a conversion factor may not fully represent trucks' characteristics. Considering detailed vehicles specific characteristic and performance may facilitate a better comprehensive study and robust traffic signal control algorithm in future studies. It may be necessary to incorporate Monte Carlo tree search into the algorithm to increase its robustness under various scenarios.

6.0 REFERENCES

Abdoos, M., Mozayani, N., Bazzan, A.L.C., 2014. Hierarchical control of traffic signals using Q-learning with tile coding. Appl Intell 40, 201–213. https://doi.org/10.1007/s10489-013-0455-3

Abdulhai, B., Pringle, R., Karakoulas, G.J., 2003. Reinforcement Learning for *True* Adaptive Traffic Signal Control. Journal of Transportation Engineering 129, 278–285. https://doi.org/10.1061/(ASCE)0733-947X(2003)129:3(278)

Adaptive Max Pressure Control of Network of Signalized Intersections, 2016. . IFAC-PapersOnLine 49, 19–24. https://doi.org/10.1016/j.ifacol.2016.10.366

Allsop, R.E., 1968. Selection of Offsets to Minimize Delay to Traffic in a Network Controlled by Fixed-Time Signals. Transportation Science 2, 1–13. https://doi.org/10.1287/trsc.2.1.1

Araghi, S., Khosravi, A., Johnstone, M., Creighton, D., 2013. Q-learning method for controlling traffic signal phase time in a single intersection, in: 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013). Presented at the 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), pp. 1261–1265. https://doi.org/10.1109/ITSC.2013.6728404

Brys, T., Pham, T.T., Taylor, M.E., 2014. Distributed learning and multi-objectivity in traffic light control. Connection Science 26, 65–83. https://doi.org/10.1080/09540091.2014.885282

Chen, C., Wei, H., Xu, N., Zheng, G., Yang, M., Xiong, Y., Xu, K., Li, Z., 2020. Toward A Thousand Lights: Decentralized Deep Reinforcement Learning for Large-Scale Traffic Signal Control. Proceedings of the AAAI Conference on Artificial Intelligence 34, 3414–3421. https://doi.org/10.1609/aaai.v34i04.5744

El-Tantawy, S., Abdulhai, B., Abdelgawad, H., 2013. Multiagent Reinforcement Learning for Integrated Network of Adaptive Traffic Signal Controllers (MARLIN-ATSC): Methodology and Large-Scale Application on Downtown Toronto. IEEE Transactions on Intelligent Transportation Systems 14, 1140–1150. https://doi.org/10.1109/TITS.2013.2255286

Federal Highway Administration, 2017. Comprehensive Truck Size and Weight Study.

Ficklin, W.E.P., N.C., 1969. The Analog Traffic Signal Model. Traffic Engineering 54–58.

Gartner, N., Assman, S., Lasaga, F., Hou, D., 1991. A multi-band approach to arterial traffic signal optimization. Transportation Research Part B: Methodological 25, 55–74. https://doi.org/10.1016/0191-2615(91)90013-9

Gartner, N.H., 1982. Demand-Responsive Decentralized Urban Traffic Control. Part I: Single-Intersection Policies.

Gartner, N.H., Assmann, S.F., Lasaga, F., Hous, D.L., 1990. MULTIBAND–A VARIABLE-BANDWIDTH ARTERIAL PROGRESSION SCHEME. Transportation Research Record.

Gartner, N.H., Pooran, F.J., Andrews, C.M., 2002. Optimized Policies for Adaptive Control Strategy in Real-Time Traffic Adaptive Control Systems: Implementation and Field Testing. Transportation Research Record 1811, 148–156. https://doi.org/10.3141/1811-18

Hasselt, H. van, 2010. Double Q-learning, in: Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 2, NIPS'10. Curran Associates Inc., Red Hook, NY, USA, pp. 2613–2621.

Head, K.L., Mirchandani, P.B., Sheppard, D., 1992. Hierarchical framework for real-time traffic control. Transportation Research Record.

Hillier, J.A., Holroyd, J., 1965. THE GLASGOW EXPERIMENT IN AREA TRAFFIC CONTROL. Traffic Engineering and Control 7, 569–571.

Hunt, P.B., Robertson, D.I., Bretherton, R.D., Winton, R.I., 1981. SCOOT - A TRAFFIC RESPONSIVE METHOD OF COORDINATING SIGNALS. Publication of: Transport and Road Research Laboratory.

Jin, J., Ma, X., 2015. Adaptive Group-Based Signal Control Using Reinforcement Learning with Eligibility Traces, in: 2015 IEEE 18th International Conference on Intelligent Transportation Systems. Presented at the 2015 IEEE 18th International Conference on Intelligent Transportation Systems, pp. 2412–2417. https://doi.org/10.1109/ITSC.2015.389

Li, S., Wei, C., Yan, X., Ma, L., Chen, D., Wang, Y., 2020. A Deep Adaptive Traffic Signal Controller With Long-Term Planning Horizon and Spatial-Temporal State Definition Under Dynamic Traffic Fluctuations. IEEE Access 8, 37087–37104. https://doi.org/10.1109/ACCESS.2020.2974885

Little, J.D.C., 1966. The Synchronization of Traffic Signals by Mixed-Integer Linear Programming. Operations Research 14, 568–594. https://doi.org/10.1287/opre.14.4.568

Little, J.D.C., Kelson, M.D., Gartner, N.M., 1981. Maxband: a program for setting signals on arteries and triangular networks, in: Transportation Research Record.

Lopez, P.A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., Wiessner, E., 2018. Microscopic Traffic Simulation using SUMO, in: 2018 21st International Conference on Intelligent Transportation Systems (ITSC). pp. 2575–2582. https://doi.org/10.1109/ITSC.2018.8569938

Mannion, P., Duggan, J., Howley, E., 2016. An Experimental Review of Reinforcement Learning Algorithms for Adaptive Traffic Signal Control, in: McCluskey, T.L., Kotsialos, A., Müller, J.P., Klügl, F., Rana, O., Schumann, R. (Eds.), Autonomic Road Transport Support Systems, Autonomic Systems. Springer International Publishing, Cham, pp. 47–66. https://doi.org/10.1007/978-3-319-25808-9_4

Mirchandani, P., Head, L., 2001. A real-time traffic signal control system: architecture, algorithms, and analysis. Transportation Research Part C: Emerging Technologies 9, 415–432. https://doi.org/10.1016/S0968-090X(00)00047-4

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D., 2015. Human-level control through deep reinforcement learning. Nature 518, 529–533. https://doi.org/10.1038/nature14236

Morgan, J.T., Little, J.D.C., 1964. Synchronizing Traffic Signals for Maximal Bandwidth. Operations Research 12, 896–912. https://doi.org/10.1287/opre.12.6.896

N.A. Chaudhary, V.G. Kovvali, C. Chu, S.M. Alam, 2002. Software for Timing Signalized Arterials. Texas A&M Transportation Institute.

Neely, M.J., 2010. Stochastic Network Optimization with Application to Communication and Queueing Systems. Synthesis Lectures on Communication Networks 3, 1–211. https://doi.org/10.2200/S00271ED1V01Y201006CNT007

Neely, M.J., Modiano, E., Rohrs, C.E., 2005. Dynamic power allocation and routing for time-varying wireless networks. IEEE Journal on Selected Areas in Communications 23, 89–103. https://doi.org/10.1109/JSAC.2004.837349

N.H.Gartner, 1972. Optimal synchronization of traffic signal networks by dynamic programming. Traffic Flow and Transportation.

Petterman, J.L., 1947. Timing Progressive Signal Systems. Traffic Engineering 29, 194–199.

Richard S. Sutton and Andrew G. Barto, 2015. Reinforcement Learning: An Introduction, Second. ed. The MIT Press, Cambridge, Massachusetts.

Robertson, D.I., 1969. “TRANSYT” method for area traffic control. Traffic Engineering & Control 11.

Stevanovic, A., Kergaye, C., Martin, P.T., 2009. SCOOT and SCATS: Closer Look into Their Operations.

Tassiulas, L., Ephremides, A., 1993. Dynamic server allocation to parallel queues with randomly varying connectivity. IEEE Transactions on Information Theory 39, 466–478. https://doi.org/10.1109/18.212277

Tassiulas, L., Ephremides, A., 1992. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. IEEE Transactions on Automatic Control 37, 1936–1948. https://doi.org/10.1109/9.182479

Teo, K.T.K., Yeo, K.B., Chin, Y.K., Chuo, H.S.E., Tan, M.K., 2014. Agent-Based Traffic Flow Optimization at Multiple Signalized Intersections, in: 2014 8th Asia Modelling Symposium.

Presented at the 2014 8th Asia Modelling Symposium, pp. 21–26. https://doi.org/10.1109/AMS.2014.16

Transportation Research Board, 2016. Highway Capacity Manual 6th Edition: A Guide for Multimodal Mobility Analysis. https://doi.org/10.17226/24798

Varaiya, P., 2013. The Max-Pressure Controller for Arbitrary Networks of Signalized Intersections, in: Ukkusuri, S.V., Ozbay, K. (Eds.), Advances in Dynamic Network Modeling in Complex Transportation Systems, Complex Networks and Dynamic Systems. Springer, New York, NY, pp. 27–66. https://doi.org/10.1007/978-1-4614-6243-9_2

Wang, X.B., Cao, X., Wang, C., 2017. Dynamic optimal real-time algorithm for signals (DORAS): Case of isolated roadway intersections. Transportation Research Part B: Methodological 106, 433–446. https://doi.org/10.1016/j.trb.2017.06.005

Wei, H., Chen, C., Zheng, G., Wu, K., Gayah, V., Xu, K., Li, Z., 2019a. PressLight: Learning Max Pressure Control to Coordinate Traffic Signals in Arterial Network, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '19. Association for Computing Machinery, New York, NY, USA, pp. 1290–1298. https://doi.org/10.1145/3292500.3330949

Wei, H., Chen, C., Zheng, G., Wu, K., Gayah, V., Xu, K., Li, Z., 2019b. PressLight: Learning Max Pressure Control to Coordinate Traffic Signals in Arterial Network, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '19. Association for Computing Machinery, New York, NY, USA, pp. 1290–1298. https://doi.org/10.1145/3292500.3330949

Wei, H., Zheng, G., Gayah, V., Li, Z., 2019c. A Survey on Traffic Signal Control Methods. arXiv e-prints 1904, arXiv:1904.08117.

Wei, H., Zheng, G., Yao, H., Li, Z., 2018. IntelliLight: A Reinforcement Learning Approach for Intelligent Traffic Light Control, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '18. Association for Computing Machinery, New York, NY, USA, pp. 2496–2505. https://doi.org/10.1145/3219819.3220096

Wongpiromsarn, T., Uthaicharoenpong, T., Wang, Y., Frazzoli, E., Wang, D., 2012. Distributed traffic signal control for maximum network throughput, in: 2012 15th International IEEE Conference on Intelligent Transportation Systems. pp. 588–595. https://doi.org/10.1109/ITSC.2012.6338817

Xu, M., Wu, J., Huang, L., Zhou, R., Wang, T., Hu, D., 2020. Network-wide traffic signal control based on the discovery of critical nodes and deep reinforcement learning. Journal of Intelligent Transportation Systems 24, 1–10. https://doi.org/10.1080/15472450.2018.1527694

Yau, K.-L.A., Qadir, J., Khoo, H.L., Ling, M.H., Komisarczuk, P., 2017. A Survey on Reinforcement Learning Models and Algorithms for Traffic Signal Control. ACM Comput. Surv. 50, 34:1-34:38. https://doi.org/10.1145/3068287