Master's Thesis

Adaptive Traffic Control Algorithm Based on Back-Pressure and Q-Learning

Arnan Maipradit

September 13, 2019

Graduate School of Information Science
Nara Institute of Science and Technology

A Master's Thesis
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
MASTER of ENGINEERING

Arnan Maipradit

Thesis Committee:

Professor Minoru Ito	(Supervisor)
Professor Keiichi Yatsumoto	(Co-supervisor)
Associate Professor Naoki Shibata	(Co-supervisor)
Assistant Professor Tomoya Kawakami	(Co-supervisor)

Adaptive Traffic Control Algorithm Based on Back-Pressure and Q-Learning*

Arnan Maipradit

Abstract

Nowadays traffic congestion has increasingly been a significant problem, which results in longer travel time and aggravates air pollution. Available work showed that back-pressure based traffic control algorithms can effectively reduce traffic congestion. However, those work control traffic based on either inaccurate traffic information or local traffic information, which causes inefficient traffic scheduling. In this paper, we propose an adaptive traffic control algorithm based on back-pressure and Q-learning, which can efficiently reduce congestion. Our algorithm controls traffic based on accurate real-time traffic information and global traffic information learned by Q-learning. As verified by simulation, our algorithm significantly decreases average vehicle traveling time from 17% to 37% when compared with state-of-the-art algorithm under tested scenarios.

Keywords:

Back-pressure, Q-Learning, Vehicle routing

*Master's Thesis, Graduate School of Information Science,
Nara Institute of Science and Technology, September 13, 2019.

Contents

List of Figures

1 Introduction

1.1 Background

Currently most traffic light control systems use fixed time cycle scheduling that lead to congestion and thus increasing vehicle travel time [1]. Because this kind of algorithm does not consider real-time or real-situation traffic information. Congestion can be reduced by smartly controlling traffic signals [2]. With the development of technology of intelligent transportation system (ITS) and Internet of Things (IoT), researchers have adopted such technology to improve the efficiency of transportation. Intelligent transportation system (ITS) is a traffic management system that uses intelligent algorithm to reduce vehicle travel time, improve traffic safety.

The intelligent traffic control systems currently implemented in urban road network are SCOOT [3], [4] and SCATS [5]. Theses systems use adaptive traffic signals that consider real-time traffic information [6], more effective than fixed cycle signal control. But these adaptive traffic signals still cannot provide performance guarantee [11]. Genetic Algorithm [7], Fuzzy Logic Control [8,9] are also considered as the solution to smartly controlling traffic signals. However, these algorithms [7-9] are centralized that do not suit with a large urban road network which requires decentralized algorithms.

Recently, decentralized traffic control algorithms based on back-pressure have been proposed [10-14]. Back-pressure based traffic signal control algorithm was showed to be superior to signal control of fixed time cycles in [11,19]. These back-pressure based traffic control algorithms do not consider adaptive control of vehicle routes, e.g., shortest path algorithm easily results in traffic congestion especially during rush hours. Some research considered jointly controlling traffic signals and vehicle routing [16,18]. But these work focused on giving individual vehicles adaptive route guidance only. Coordination between different vehicles will further reduce traffic congestion.

Some work also proposed back-pressure based algorithms to coordinate different vehicles [15,21]. In road network vehicles need time to travel from one road to another road which depends on vehicle speed and road length. Directly applying back-pressure algorithm to control traffic as in [15] is not appropriate. Based on this observation, [21] proposed an adaptive traffic control algorithm which adapts back-pressure algorithm by considering vehicle traveling time on a road. Specifically, they control traffic signal and vehicle routes based on real-time traffic information, like vehicle speed and vehicle position. As a result, their algorithm significantly reduces traffic congestion.

However, their work controls traffic lights and vehicle routes based on local traffic information only, i.e., every control agent considers information of vehicles only around its own junction. Therefore, their algorithm is short-sighted, since they do not use global traffic information. For more efficient traffic control, global traffic information and coordination between different junction agents are needed.

1.2 Back-Pressure and Q-Learning

Back-Pressure routing is an algorithm originally for routing packets based on queue length differentials (also called pressure gradients) in wireless communication networks. Back-Pressure routing usually refers to a data network, but in this work, we apply Back-Pressure routing to a road network which the pressure of the road network is a number of vehicles on the road network.

There are multiple algorithms available for Reinforcement Learning. Q-learning algorithm is the most widely used in Reinforcement Learning methods. Q-Learning is a basic form of Reinforcement Learning which uses Q-values to improve the behavior of the learning agent. First, an agent chooses an action at a given state based on a Q-value, which is a weighted reward based on the expected highest reward after getting Q-value it will store in Q-table The values in the Q-table are updated each time an agent selects an action. This Q-value will be iterratively computed by TD-Update or Temporal Difference rule as follows:

$$Q(S, A) \longleftarrow Q(S, A) + \alpha(R + \gamma Q(\bar{S}, \bar{A}) - Q(S, A)) \quad (1.1)$$

where current state of the agent is S and A is the current action picked according to the policy, $\bar{S}$ and $\bar{A}$ are next stage of the agent and next action picked using current Q-value estimation, R is the current reward that related to current action.

1.3 Contributions

In this work, we further improve [21] and propose an adaptive traffic control algorithm that controls traffic based on accurate real-time traffic information and global traffic information, where neighboring junction agents exchange traffic information to learn global traffic information.

1.4 Problem Formulation

1.4.1 Main Question

What is the performance of the algorithm controls traffic based on accurate real-time traffic information (achieved by using shadow network) and global traffic information (achieved by using Q-learning)?

we propose an adaptive traffic control algorithm based on back-pressure and Q-learning, which can efficiently reduce congestion. Our algorithm controls traffic based on accurate real-time traffic information and global traffic information learned by Q-learning. As verified by simulation, our algorithm significantly decreases average vehicle traveling time from 17% to 37% when com-pared with state-of-the-art algorithm under tested scenarios.

1.4.2 Sub-Question 1

What happens if the road network consists of self-driving vehicles that follow our algorithm and human-driving vehicles that not follow our algorithm?

We concern about this issue because in the urban road network, it impossible that all vehicles in the road network will be a self-driving vehicle. In this work, we simulate to show the performance of our algorithm for scenarios with both self-driving vehicles and human-driving vehicles.

1.4.3 Sub-Question 2

How does each agent know the destination of each vehicle and collect vehicle speed or position?

In this work we assume that all vehicles are self-driving vehicles, that mean driver has to send a destination to server. To support this idea wireless communication is important to connect each agent to the server and also image processing[24] to detect and collect a number of the vehicle in each road.

2 Road Network System

A road system consists of Roads (R) and Junctions (J), where $\mathbb{R} = \{R_1, R_2, R_3, ...R_{max}\}$ denotes roads, $\mathbb{J} = \{J_1, J_2, J_3, ...J_{max}\}$ denotes junctions. It is assumed that each R_i consists of 3 lanes L_{ij}, an example is given in Fig. 2.1. Vehicles of a traffic flow (f) have the same starting road (o) and destination road (d). We define $\mathbb{F}$ as the set of all flows, $\mathbb{O} = \{o(f), f \in \mathbb{F}\}$ as the set of all starting roads, $\mathbb{D} = \{d(f), f \in \mathbb{F}\}$ as the set of all destinations and $\lambda_f(t)$ as the number of vehicles of flow f that enter road network at time slot t.

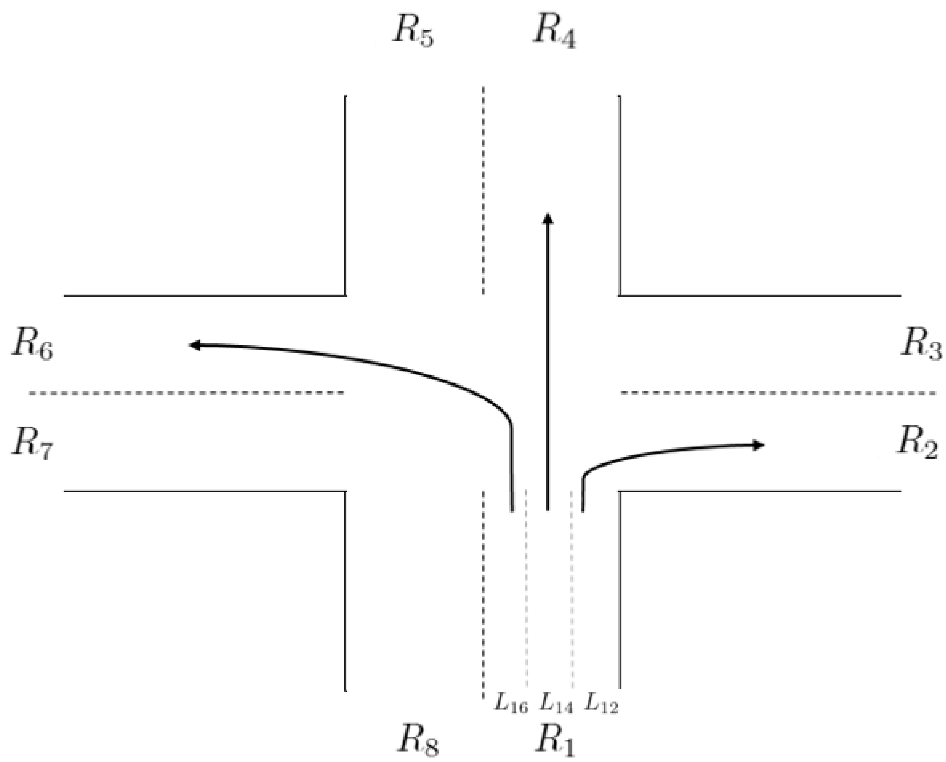

Figure 2.1: Fig1 An example of a junction with roads of three lanes.

We define a traffic movement (R_i, R_j) at a junction to be the process of a vehicle moving from R_i to R_j. We define a traffic phase to include all traffic movements that can happen simultaneously. Fig. 2.2. shows all possible phases at a junction. For a junction J_a, we define $\mathbb{M}_a$ as the set of all possible movements and $\mathbb{P}_a$ as the set of all possible traffic phases. Traffic signals at junction J_a are controlled by activating a traffic phase p_i^a from $\mathbb{P}_a$.

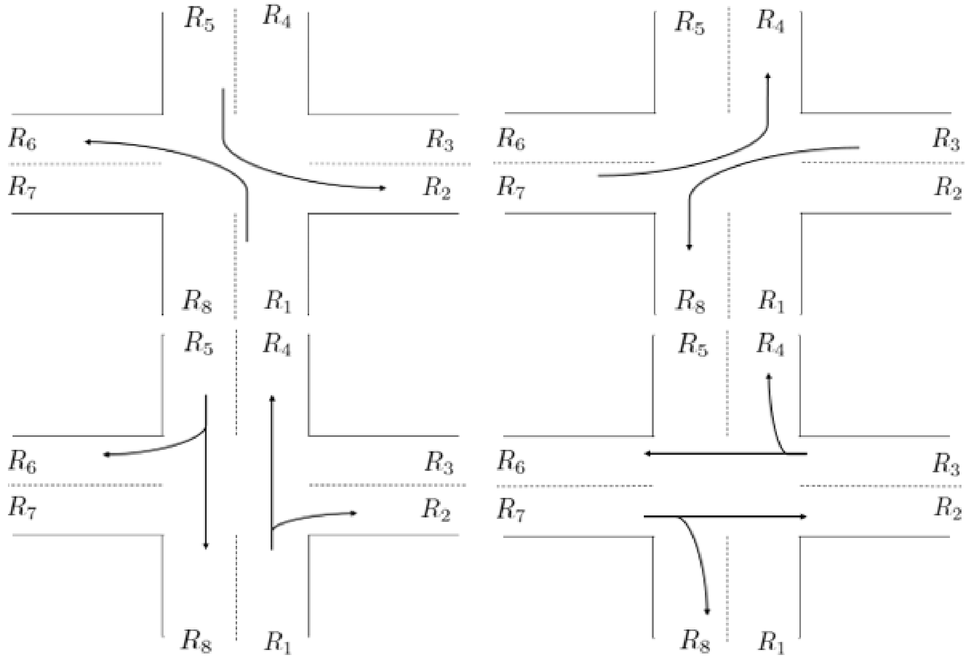

Figure 2.2: All possible phases at a junction.

3 Adaptive Traffic Control Based On Back-Pressure And Q-Learning

Our algorithm uses real-time traffic information and global traffic information in road network.

Each junction has a control agent that collects information of vehicle speed and vehicle position every time slot for traffic control. At each time slot, every control agent performs the following three tasks sequentially. Task 1 (Learning Global Congestion Information): It exchanges congestion level information with neighboring agents. Based on exchanged congestion information, the agent updates its own congestion estimate based on Q-learning. Through this kind of congestion information exchange and update, all agents will finally obtain global congestion information which can be used in the following two tasks. Task 2 (Traffic Phase Selection): The agent selects a traffic phase based on back-pressure algorithm. Task 3 (Vehicle Routing): After a vehicle passes through the junction and enters next road under the traffic phase selected in task 2, the agent determines which lane of that road the vehicle should join. Since each lane determines vehicle turning direction, i.e., going straight, turning left or turning right, the process of determining lanes for a vehicle to join forms the routing process of that vehicle. The following shadow network is constructed to perform three tasks.

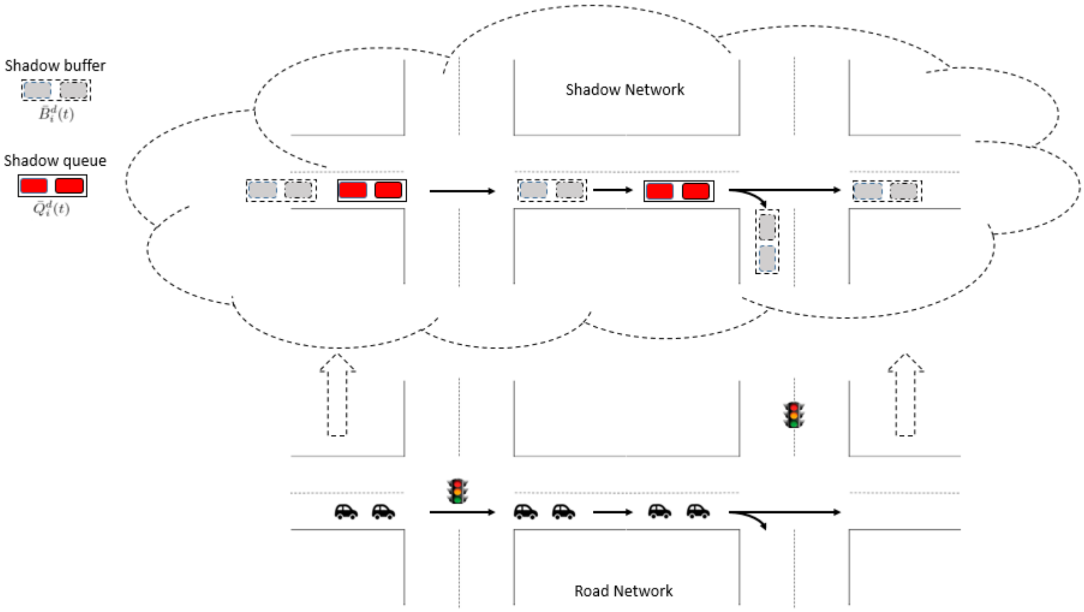

Figure 3.1: An example of a shadow network.

3.1 Shadow Network

An example of shadow network is given in Fig. 3.1, where a virtual shadow vehicle in shadow network corresponds to an actual vehicle in road network, a shadow buffer corresponds to the beginning part of one real road (a vehicle just passing through a junction will enter this part of road) and a shadow queue corresponds to the end part of one real road (a vehicle running close to next junction will enter this part of the road).

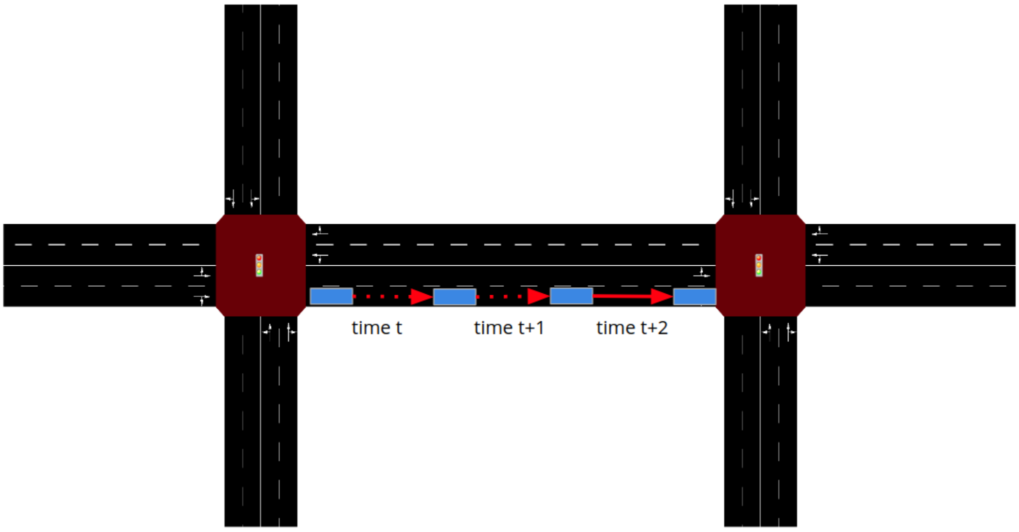

Figure 3.2: A vehicle needs time to travel across a road.

In the shadow network, whenever a real vehicle enters the road network, a shadow vehicle is generated and enters the shadow network. Furthermore, one more shadow vehicle is generated with probability $\epsilon, 0 < \epsilon < 1$ and also enters the shadow network. This operation makes sure that algorithm is stable, i.e., queue size will not go to infinite [15,17].

When an actual vehicle goes into road network from starting road R_i at t and wants to go to destination $d \in \mathbb{D}$, a shadow vehicle will also go into the shadow buffer $\bar{B}_i^d(t)$ for destination d and road R_i. When that vehicle approaches the end part of road R_i, the shadow vehicle first leaves shadow buffer $\bar{B}_i^d(t)$ and then enters shadow queue $\bar{Q}_i^d(t)$ associated with destination d and road R_i. We say a vehicle approaches end part of one road if its speed is less than 5 Km/h or it is within the range of 100 meters to next junction.

Similarly, after an actual vehicle destined for destination $d \in \mathbb{D}$ leaves road R_i and goes into adjacent road R_j at t, a shadow vehicle will leave shadow queue $\bar{Q}_i^d(t)$ of road R_i and goes into shadow buffer $\bar{B}_j^d(t)$ of R_j. Movement of virtual shadow vehicles in the shadow network can be seen as control information exchange, based on which a agent performs its three tasks (details are given in the following section).

3.2 Adaptive Traffic Control Algorithm Based on Back-Pressure and Q-Learning

Our adaptive traffic control algorithm based on back-pressure and Q-Learning (ARD-BP-Q) is decentralized and agent at each junction runs the following algorithm independently.

3.2.1 Task 1 Learning Global Congestion Information

At each time slot t, an agent performs the following three tasks sequentially. The agent at a junction is responsible for estimating route congestion level $R_{ij}^d(t)$ for all route to destination d from road i and by the way of the neighbor road j. Each agent maintains a table $\boldsymbol{R}$ to store the value of $R_{ij}^d(t)$. At the beginning of each time slot, the agent exchanges information of the number of vehicles $\bar{Q}_j^d(t)$ at upstream roads around that junction and the table $\boldsymbol{R}$ with neighboring agents. After exchanging those information, the agent updates its route congestion estimate $R_{ij}^d(t)$ as follows:

$$R_{ij}^d(t) \rightarrow (1-\alpha)R_{ij}^d(t-1) + \alpha[\bar{Q}_j^d(t) + \gamma \min_k R_{jk}^d(t)] \tag{3.1}$$

where α and γ are Q-learning parameters, $0 < \alpha, \gamma <= 1$. If $R_{ij}^d > C_{max}$, set $R_{ij}^d = C_{max}$, C_{max} is a positive constant. Each agent then calculates a bias quantity $C_i^d(t)$ as follows:

$$C_i^d(t) = \min_j R_{ij}^d(t) \tag{3.2}$$

Finally, the bias quantity $C_i^d(t)$ will be used in Traffic Phase Selection.

3.2.2 Task 2 Traffic Phase Selection

The agents at each junction compute traffic pressure $w_{ij}^d(t)$ for all destinations and traffic movement. Traffic pressure in our algorithm ARD-BP-Q (Algorithm 1) is defined as follows:

$$w_{ij}^d(t) = max\{(\bar{Q}_i^d(t) + C_i^d(t)) - (\bar{Q}_j^d(t) + C_j^d(t)), 0\} \tag{3.3}$$

Then the agent select the destination d^*_{ij} that return maximizes traffic pressure $w^d_{ij}(t)$ defined as follows:

$$d^*_{ij}(t) = \arg\max_{d} w^d_{ij}(t) \tag{3.4}$$

From above equation agents define $w_{ij}^{d^*_{ij}(t)}(t)$ as the weight of traffic movement which corresponds to one $d^*_{ij}(t)$ at time slot t.

Finally, the agent selects and activates the phase $p^{a*}(t) \in \mathbb{P}_a$ that releases the most traffic pressure defined as follows:

$$p^{a*}(t) = \arg\max_{p^a_l \in \mathbb{P}_a} \Sigma_{(R_i,R_j)\in p^a_l} w_{ij}^{d^*_{ij}(t)}(t) s_{ij}(t) \tag{3.5}$$

where s_{ij} is the number of vehicles that can move from road R_i to road R_j at time slot t

3.2.3 Task 3 Vehicle Routing

Vehicle will follow the routing probabilities $P^d_{ij}(t)$ based on $\hat{\sigma}^d_{ij}(t)$ defined as follows:

$$P^d_{ij}(t) = \frac{\hat{\sigma}^d_{ij}(t)}{\Sigma_{k:\ (R_j,R_k)\in\mathbb{M}_a} \hat{\sigma}^d_{ik}(t)} \tag{3.6}$$

where $\hat{\sigma}^d_{ij}(t)$ is the estimated value of expected number of shadow vehicles of destination d that moves from shadow queue $\bar{Q}^d_i(t)$ to shadow buffer $\bar{B}^d_j(t)$ which corresponds to road R_i and R_j. $\hat{\sigma}^d_{ij}(t)$ is updated by the agent of junction J_a for all destination $d \in \mathbb{D}$ and traffic movement $(R_i, R_j) \in \mathbb{M}_a$ as follows :

$$\hat{\sigma}^d_{ij}(t) = (1-\beta)\hat{\sigma}^d_{ij}(t-1) + \beta\sigma^d_{ij}(t) \tag{3.7}$$

where $0 < \beta < 1$. After vehicle enters road R_i at time slot t it will join lane L_{ij} with routing probability $P^d_{ij}(t)$.

Since our goal is to reduce vehicle traveling time, a heuristic is that we should let vehicles with longer traveling time pass through a junction first. Thus, we also propose the following Adaptive Traffic Control Algorithm Based on Back-Pressure and Q-Learning with Vehicle traveling time (ARD-BP-QV Algorithm 2), which is the same with Algorithm 1 except that traffic pressure is defined as follows:

$$w_{ij}^{d}(t) = max\{(\bar{V}_i^d(t) + C_i^d(t)) - (\bar{V}_j^d(t) + C_j^d(t)), 0\} \tag{3.8}$$

where $\bar{V}_i^d(t)$ is the normalized value of the sum of traveling time of vehicles in shadow queue $\bar{Q}_i^d(t)$, the normalized value is within range from 50-100. We need to normalize vehicle traveling time to make it comparable to the quantity of bias $C_i^d(t)$ and $C_j^d(t)$.

4 Simulation Setup and Results

In this section, we compare the performance of our algorithm with other algorithms in an open-source simulator SUMO (Simulation of Urban MObility) [20].

- ☐ Traffic signal control with fixed-cycles (FC)
- ☐ Back-pressure and shortest path based traffic control algorithm (SP-BP) [11]
- ☐ Back-pressure based adaptive traffic signal control and vehicle routing without real-time control information update (AR-BP) [15].
- ☐ Back-pressure based adaptive traffic signal control and vehicle routing with real-time control information update (ARD-BP)[21].
- ☐ Adaptive Traffic Control Algorithm Based on Back-Pressure and Q-Learning (ARD-BP-Q).
- ☐ Adaptive Traffic Control Algorithm Based on Back-Pressure and Q-Learning with Vehicle traveling time (ARD-BP-QV).

4.1 Configuration

We implement road network that mimic from a real Stockholm road network which, given by OpenStreetMap that can export topology of road network [22,23].The road network consists of three and four way junctions as shown in Fig. 4.1. All roads have different lengths (400-1600 meters) and speed limits (60-140 km/h). Roads in this network are bi-directional. There are 6 origin and destination pairs $\{(o_1, d_1), (o_2, d_2), (o_3, d_3), ..., (o_6, d_6)\}$. All vehicles arrive at starting roads with the same rates (360-2520 vehicles/hour). Duration of a slot is configured to be 15 seconds. Shadow vehicle generating probability ϵ is configured to be 0.02 and vehicle routing parameter β is configured to be 0.02 (routing parameter β is explained in [21]).

We define vehicle traveling time to be the time it takes a vehicle to travel from its starting road to its destination. For algorithms AR-BP, ARD-BP, ARD-BP-Q and ARD-BP-QV, parameter $\alpha = 2.5$.

During simulations we collect the following data: vehicle speed, number of vehicles in road network, number of arriving vehicles at destinations and vehicle traveling time. Vehicle traveling time is the time it takes a vehicle to travel from its origin to its destination.

For algorithms FC and SP-BP, we run simulation for 12200 seconds. We collect simulation data of vehicles that enter road network before 7200 seconds only, because vehicles entering road network after 7200 seconds may not arrive at destinations.

For algorithms AR-BP , ARD-BP , ARD-BP-Q and ARD-BP-QV, we run simulation for 18200 seconds. We collect simulation data of vehicles that enter road network from 6000-13200 seconds only, because these algorithms need time to learn vehicle routing probabilities and reach a stable routing policy.

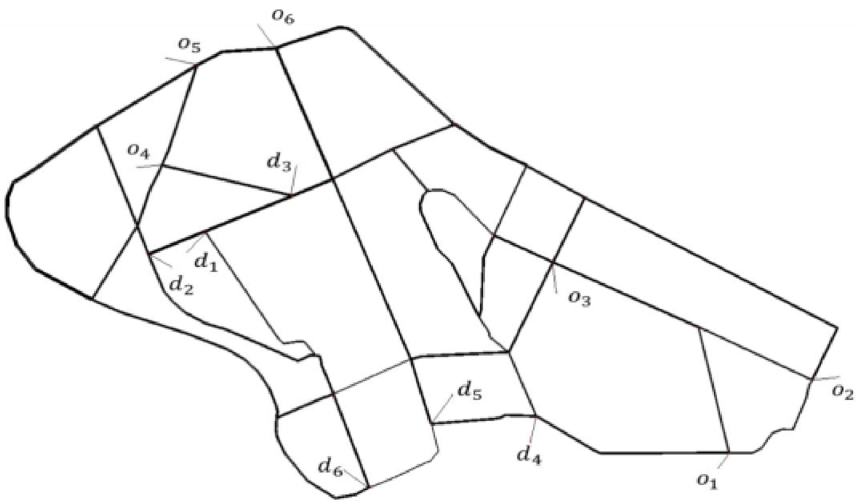

Figure 4.1: Road network structure of Stockholm city that use in SUMO with 6 pairs of origin and destination.

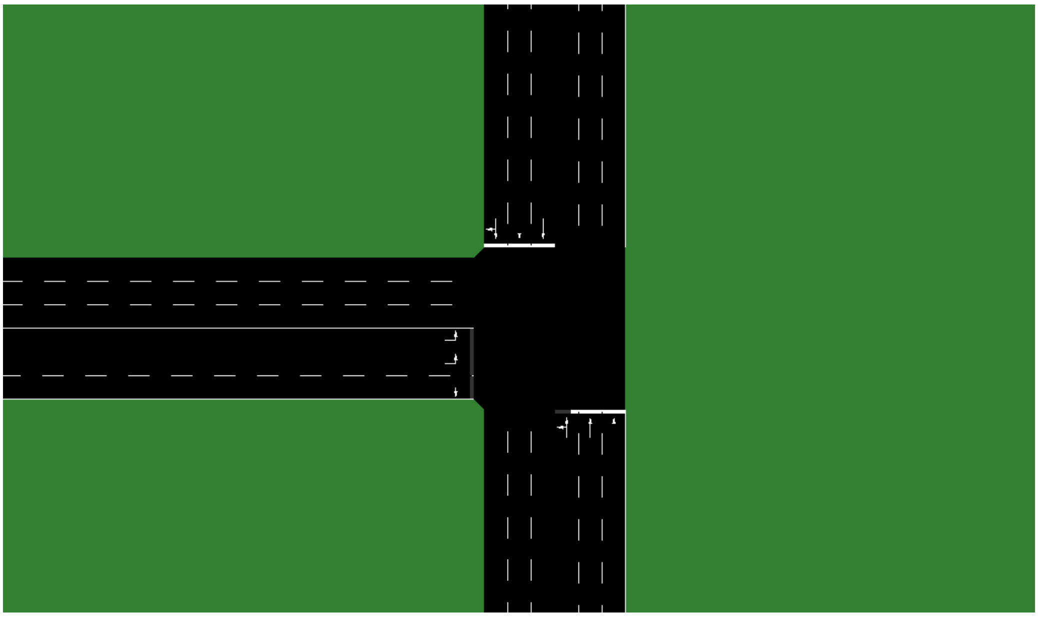

Figure 4.2: Three-way intersection that implemented in SUMO simulation.

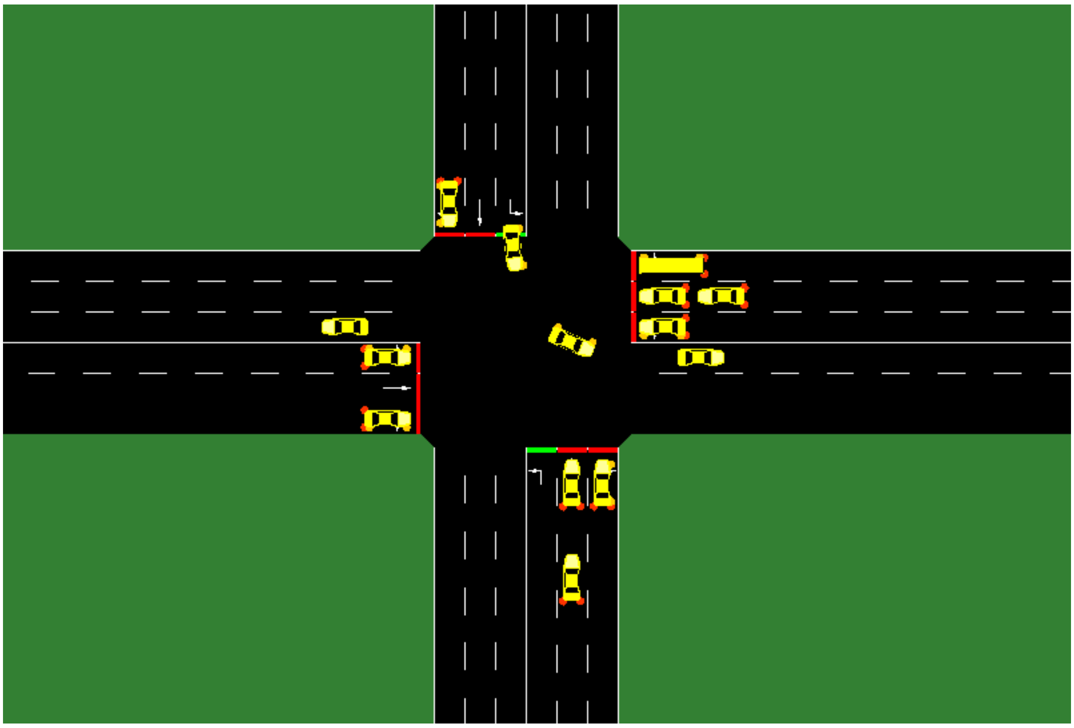

Figure 4.3: Four-way intersection that implemented in SUMO simulation.

4.2 Simulation Result and Analysis

In Fig. 4.4, our algorithm ARD-BP-QV achieves almost the lowest average traveling time under different vehicle arrival rates. Compared to ARD-BP, our algorithm ARD-BP-QV decreases average vehicle traveling time by 17% to 37%. Compared to ARD-BP-Q, algorithm ARD-BP-QV decreases average vehicle traveling time by 7% to 18%. This indicates that the heuristic of letting vehicles with longer traveling time pass through junction first is indeed an effective way to reduce vehicle traveling time.

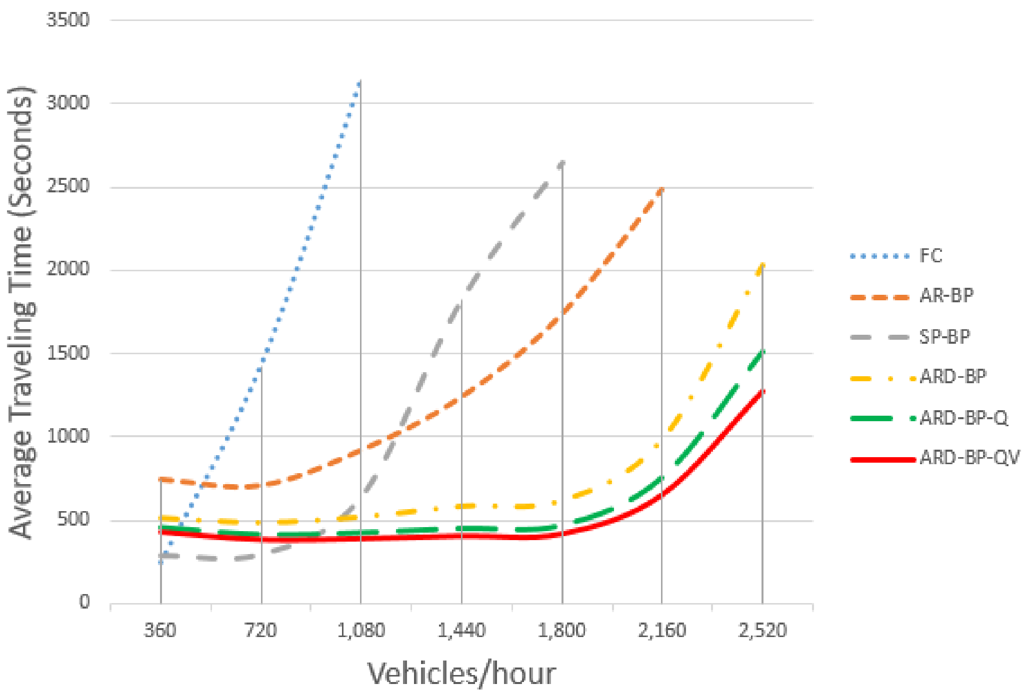

Figure 4.4: Average vehicle traveling time under different vehicle arrival rates.

Fig. 4.5 shows simulation results of average number of vehicles in road network. This figure shows that the number of vehicles in road network under ARD-BP-QV algorithm is smaller than other algorithms, meaning less traffic congestion.

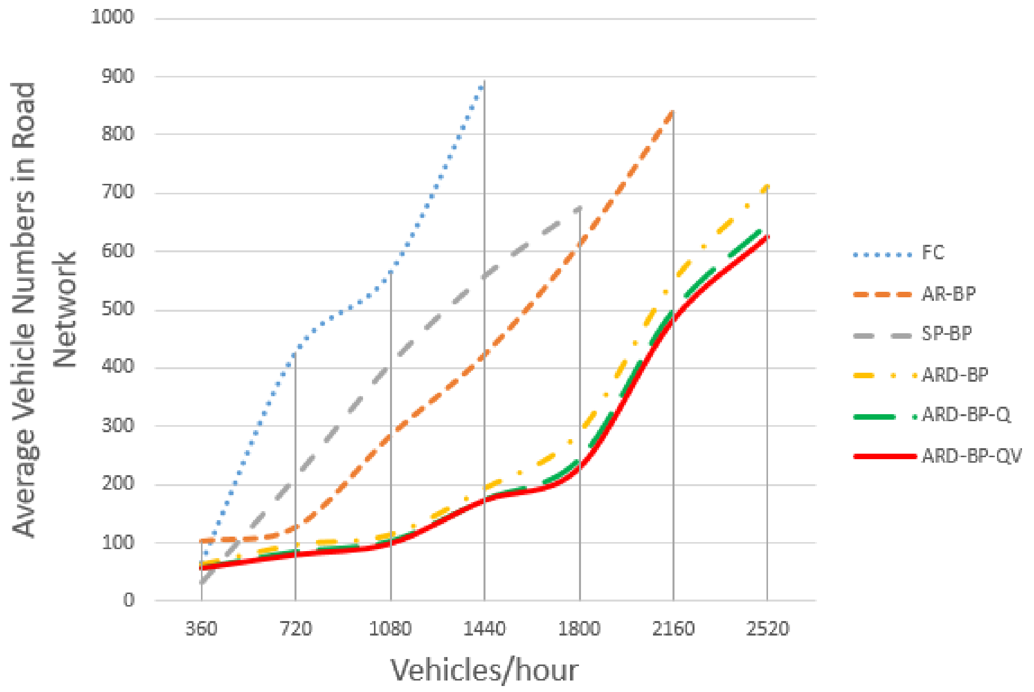

Figure 4.5: Average number of vehicles in road network under different vehicle arrival rates.

Fig. 4.6 shows that more vehicles can arrive at destinations under our algorithm ARD-BP-QV, meaning that more vehicles under other algorithms get stuck in road network.

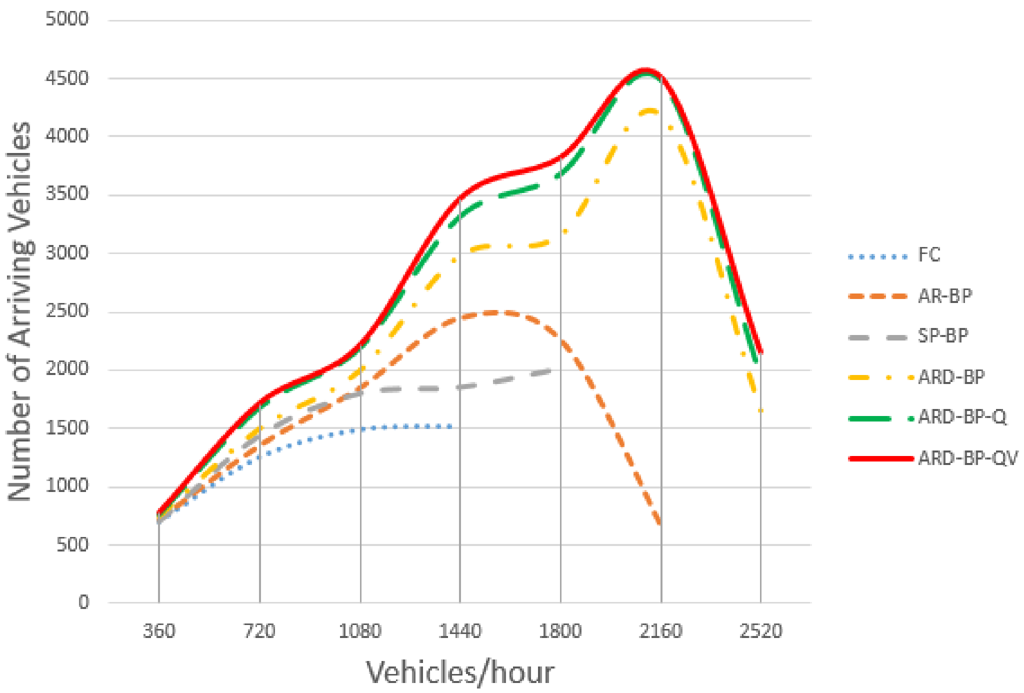

Figure 4.6: Number of vehicles arriving at destinations.

We also evaluate the fairness of our algorithm. From Fig. 4.7, we see that most of vehicles arrive at their destinations within 700 seconds, which is less than twice the average traveling time (385 seconds). So, our algorithm is fair for most vehicles.

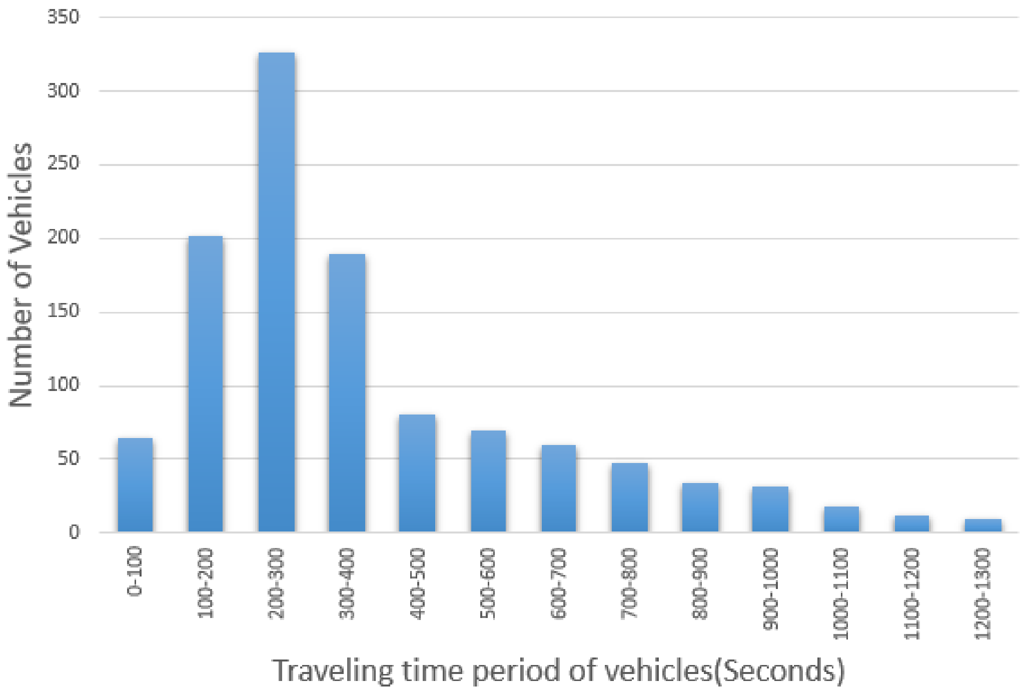

Figure 4.7: Histogram of number of vehicles of different travelling time. Vehicle arrival rate is set to be 1080 vehicles/hour and the average traveling time is 385 seconds.

We also run simulations to check the impact of parameter α to ARD-BP-QV performance. As shown in Fig. 4.8. we need to properly set α in our algorithm to achieve the optimal performance.

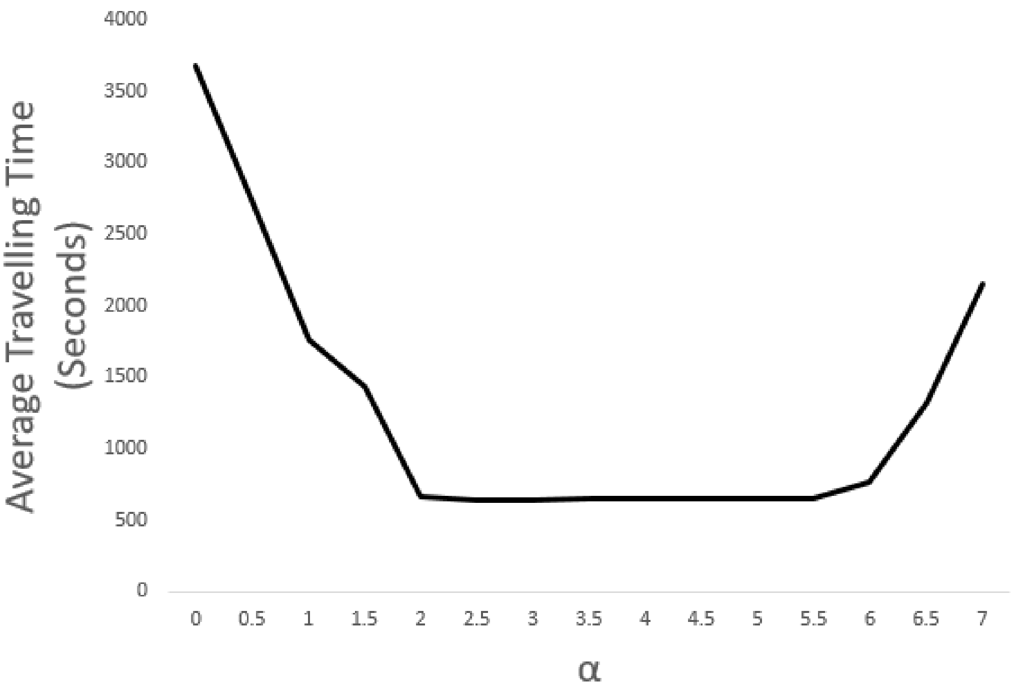

Figure 4.8: Performance under parameter α with rate of 450 vehicles/hour.

Finally, we run simulation to check our algorithm under scenarios with both self-driving or human driving vehicles, where all human-driving vehicles follow shortest path route and the percentage of human-driving vehicles ranges from 10% to 60%. The simulation results are summarized in Fig. 4.9.

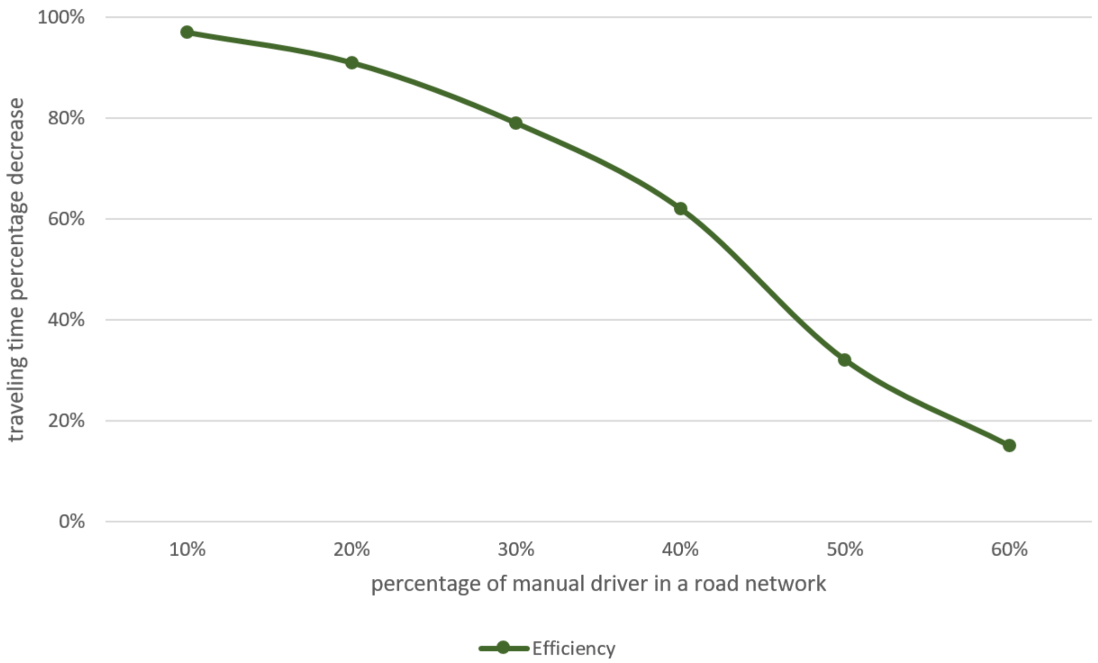

Figure 4.9: Vehicle traveling time of ARD-BP-QV under scenarios with both self-driving or human driving vehicles.

5 Conclusions

In this paper, we proposed an adaptive traffic control algorithm based on backpressure and Q-learning. Our algorithm controls traffic based on accurate real-time traffic information (achieved by using shadow network) and global traffic information (achieved by using Q-learning). Our algorithm can greatly decrease traffic congestion and is superior to other state-of-the-art algorithms.

Our algorithm is suitable for self-driving vehicles because all vehicles need to completely follow our algorithm. For scenarios with both self-driving vehicles and human-driving vehicles, simulation results show that vehicle traveling time increases as percentage of human-driving vehicles increase. How to improve algorithm efficiency under these scenarios will be our future work.

Acknowledgements

Upon the completion of this thesis, I would like to express my gratitude to all of those people who have made this dissertation exist.

First, I would like to thanks Professor Minoru Ito, my supervisor, for guidance and advice during 2 years in Mobile Computing Laboratory. He also finds scholarship and writes recommend letter to the scholarship. Moreover, I thank him for all of his support in the laboratory such as thesis review, daily life, financial support. It is my pleasure to continue my doctoral course in his laboratory.I would like to thank Professor Keiichi Yasumoto for comment and advice on my presentation which makes my research improve. To Associate Professor Naoki Shibata, I am truly gain a lot of knowledge from his class and he also helps to give a lot of advice and suggestion.

I would like also like to thanks Assistant Professor Tomoya Kawakami for all support for my research and suggestion for all-conference. And also help me to translate Japanese language documents which take a lot of time in spite of his tight schedule.

I also thanks Mr. Juntao Gao. During my master course, he use a lot of time to discuss and suggest about my research, After he resign from Nara Institue of Science and Technology he still help me to complete my thesis.

I would like to thanks all of NAIST staff and teacher that teach and support me during master course and in the future also. I am thankful to all of the members in Mobile Computing Laboratory for everything. I would like to express my gratitude to my family who support me.

References

[1] L. Li, D. Wen and D. Yao, "A survey of traffic control with vehicular communications," IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 1, pp. 425–432, Feb. 2014.

[2] A. Ejaz and H. Gharavi, "Cooperative Vehicular Networking: A Survey," IEEE Transactions on Intelligent Transportation Systems, vol.19, no.3, pp: 996–1014, 2018.

[3] D.I. Robertson, I. Dennis and R.D. Bretherton, "Optimizing networks of traffic signals in real time-the SCOOT method," IEEE Transactions on vehicular technology, vol.40, no.1, pp. 11–15, 1991.

[4] I. Day, "Scoot-split, cycle and offset optimization technique," Proceedings of TRB committee AHB25 adaptive traffic control, 1998.

[5] P.R. Lowrie, Scats, sydney co-ordinated adaptive traffic system: A traffic responsive method of controlling urban traffic, Darlinghurst, NSW, Australia : Roads and Traffic Authority NSW, Traffic Control Section, 1990.

[6] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos and Y. Wang, "Review of road traffic control strategies," Proceedings of the IEEE, vol. 91, no.12, pp. 2043–2067, 2003.

[7] X. CHEN and Z. SHI, "Real-coded genetic algorithm for signal timings optimization of a signal intersection," in Proceedings of the First International Conference on Machine Learning and Cybernetics, 2002.

[8] J. Niittymaki, "Installation and experiences of field testing a fuzzy signal controller," European Journal of Operation Research, vol. 131, 2001.

[9] E. Pappis, C.P. Mamdani, “A fuzzy logic controller for a traffic junction,” IEEE Transactions on Systems, Man and Cybernetics, vol. 7, no. 10, pp. 707–717, 1977.

[10] T. Le, P. Kovács, et al., “Decentralized signal control for urban road networks,” Transp. Res. C, Emerging Technol, vol. 58, pp. 431–450, 2015.

[11] T. Wongpiromsarn, T. Uthaicharoenpong, Y. Wang, E. Frazzoli and D. Wang, “Distributed traffic signal control for maximum network throughput,” in Proc. IEEE ITSC, pp. 588–595, 2012.

[12] J. Gregoire, E. Frazzoli, A. de La Fortelle and T. Wongpiromsarn, “Back-pressure traffic signal control with unknown routing rates,” IFAC Proceedings Volumes, vol. 47, no. 3, pp. 11332–11337, 2014.

[13] J. Gregoire, X. Qian, E. Frazzoli, A. deLaFortelle and T. Wongpiromsarn, “Capacity-aware back-pressure traffic signal control,” IEEE Trans. Control Netw. Syst, vol. 2, pp. 164–173, 2015.

[14] Kulcsár, K. Ampountolas and A. Dabiri, “Single region robust perimeter traffic control,” in Proc. IEEE Eur. Control Conf, pp. 2628–2633, 2015.

[15] A. Zaidi, B. Kulcsár and H. Wymeersch, “Back-pressure traffic signal control with fixed and adaptive routing for urban vehicular networks,” IEEE Transactions on Intelligent Transportation Systems, vol.17, no.8, pp. 2134–2143, 2016.

[16] H. Chai, H. M. Zhang, D. Ghosal and C. N. Chuah, “Dynamic traffic routing in a network with adaptive signal control,” Transportation Research Part C: Emerging Technologies, vol. 85, pp. 64–85, 2017.

[17] E. Athanasopoulou and L. Bui, et. al., "Back-pressure-based packet-by-packet adaptive routing in communication networks," IEEE/ACM transactions on networking, vol. 2, no. 1, pp. 244–257, 2013.

[18] S. Kim, M.E. Lewis and C. White, “Optimal vehicle routing with real- time traffic information,” IEEE Trans. Intell. Transp. Syst, vol. 6, no.2, pp. 178–188, 2005.

[19] L. Tassiulas and A. Ephremides, “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks,” IEEE Transactions on Automatic Control, vol.37, no.12, pp. 1936–1948, 1992.

[20] D. Krajzewicz, G. Hertkorn, C. Rössel and C, P. Wagner, “SUMO (Simulation of Urban MObility)-an open-source traffic simulation,” in Proceedings of the 4th middle East Symposium on Simulation and Modelling, 2002, pp. 183–187.

[21] Y. Liu, J. Gao and M. Ito, "Back-Pressure Based Adaptive Traffic Signal Control and Vehicle Routing with Real-Time Control Information Update," 2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES), Madrid, 2018, pp. 1–6.

[22] M. Haklay, "How Good is Volunteered Geographical Information? A Comparative Study of OpenStreetMap and Ordnance Survey Datasets," Environment and Planning B: Planning and Design, vol. 37 no. 4, pp. 682–703, 2010.

[23] M. Haklay, P. Weber, "OpenStreetMap: User-Generated Street Maps," IEEE Pervasive Computing vol. 7, no. 4, pp. 12–18, 2008.

[24] S. Messelodi, C. Modena, M. Zanin, "A computer vision system for the detection and classification of vehicles at urban road intersections," Pattern Analysis and Applications, 8(1-2), 17–31. 2005.

Publication List

[1] Arnan Maipradit, Juntao Gao, Tomoya Kawakami, Minoru Ito, "Back-PressureおよびQ-Learningに基づく適応的交通制御アルゴリズム," 研究報告マルチメディア通信と分散処理（DPS）no. 21 pp. 1–6, 2019.
[2] A. Maipradit, J. Gao, T. Kawakami, and M. Ito, Adaptive Traffic Control Algorithm Based on Back-Pressure and Q-Learning, Proc. of the 22nd IEEE Intelligent Transportation Systems Conference (ITSC 2019), 6 pages, 2019 (to appear).