SELF-LEARNING ACYCLIC ADAPTIVE TRAFFIC SIGNAL CONTROL

Samah El-Tantawy, Baher Abdulhai, University of Toronto

Introduction

Population is steadily increasing worldwide. Consequently the demand for mobility is increasing, traffic congestion is deteriorating, and undesirable changes in the environment are becoming major concerns. Infrastructure improvement have been has been the primarily method to cope with congestion throughout the recent decades. However, tight constraints on financial resources and physical space, as well as environmental considerations, have accentuated the consideration of a wider range of options. Therefore, the emphasis has shifted to improving the existing infrastructure by optimizing the utilization of the available capacity. Advancements in Intelligent Transportation Systems (ITS) have the potential to significantly alleviate traffic congestion and long queues at the intersections through innovative traffic signal control strategies.

Pre-timed and actuated traffic signal control systems are the most common control systems for isolated intersections. Pre-timed signal control does not adapt to fluctuations in traffic flows. Actuated signal control partially reacts to changes in the demand patterns by green extension to the direction being served. In congested grid-like networks, actuated control might result in very long queues on other movements (Zhang *et al.*, 2005).

Adaptive traffic signal control on the other hand adjusts signal timing parameters in response to real-time traffic flow fluctuations. therefore, has a great potential to outperform both pre-timed and actuated control (McShane *et al.*, 1998). Several methods of adaptive signal control have been reported in the literature. Reinforcement Learning (RL) has shown great potential for self-learning traffic signal control method, the main advantage of which is the ability to perpetually learn and improve service over time (Abdulhai and Kattan, 2003).

The paper starts with a brief background on methods used to solve the adaptive traffic signal control problems. Then a generic RL-based platform is proposed and tested on a real-world multi-phase intersection in downtown Toronto. The results of the testbed intersection are then presented and compared to the common pre-timed control strategy as a bench mark. The paper ends with conclusion and directions for future research.

Background

Due to the stochastic nature of the traffic system, a closed-loop control strategy that is adaptive to the most recent traffic conditions is paramount. Dynamic Programming (DP) is viewed as plausible approach to tackle the stochastic control problem (Gosavi, 2003). A significant portion of the adaptive signal control systems that have been proposed are based on dynamic programming, for instance, PRODYN (Farges *et al.*, 1983), OPAC (Gartner, 1983), RHODES (Head *et al.*, 1992). However, DP-based traffic signal control systems suffer from two major limitations; first, DP methods require a state transition probability model for the traffic environment which is difficult to obtain because of the stochastic nature of the traffic arrivals at the intersections; second, the number of states that represent various traffic conditions is typically massive. Therefore, DP algorithms are computationally intractable (Sutton and Barto, 1998; Gosavi, 2003).

Reinforcement Learning (RL), an Artificial Intelligence (AI) technique, overcomes the DP limitations; since RL is capable of solving/modelling the stochastic control problem without assuming a perfect model of the environment and with less computational effort (Sutton and Barto, 1998). In RL, a control agent interacts with the environment to learn and achieve the optimal mapping between the environment’s *state* and the corresponding optimal control *action*, offering a closed loop optimal control law This mapping from states to actions is also referred to as *policy*. The agent iteratively receives a feedback *reward* for the actions taken and adjusts the control policy until it converges to the optimal control policy. Since the RL control agent would learn from its own experience and adapt itself to the environment, it appears to offer promising results in traffic environment for adaptive signal control where optimal real-time adaptive control is a key element in improving the effectiveness and efficiency (Abdulhai and Kattan, 2003).

Abdulhai *et al.* (Abdulhai *et al.*, 2003) and Thrope (Thorpe, 1997) introduced the Q-Learning and SARSA, respectively, for isolated adaptive traffic signal control. In Bingham, (2001), a neuro-fuzzy traffic signal controller is used in which RL is used for learning the neural network. Oliveira *et al*. (De Oliveira *et al.*, 2006) proposed an RL-based method that learns in non-stationary scenarios using an approach that can detect context changes in the traffic network. In most of these studies, the algorithms are implemented on hypothetical simplified two-phase intersections. Also, to authors' best knowledge, all the previous studies that used RL for isolated traffic control (Thorpe, 1997; Bingham, 2001; Abdulhai *et al.*, 2003; De Oliveira *et al.*, 2006; Lu *et al.*, 2008) are designed to solve fixed phasing sequence intersections. Considering fixed phasing sequence signals can significantly reduce the dimension of action space and consequently shorten the computation time of the RL algorithm. However, these systems lack the flexibility to fully adapt to traffic flow fluctuations due to the phase sequence constraint.

To address these limitations, the proposed RL controller is designed to account for variable phasing sequence in which the control action is no longer an extension or a termination of the current phase as in the fixed phasing sequence approach. Instead, the algorithm extends the current phase or switches to any other phase according to the fluctuations in traffic, possibly skipping unnecessary phases. Therefore, this algorithm is envisioned as an acyclic timing scheme with variable phasing sequence in which not only the cycle length is variable but also the phasing sequence is not predetermined. Also, the proposed algorithm is tested on a simulation of real-world multi-phase intersection in downtown Toronto.

Q-Learning for Acyclic Adaptive Traffic Signal Control with Variable Phasing Sequence

Q-Learning is one of the most commonly used RL algorithms in the traffic control problem (Wiering, 2000; Abdulhai *et al.*, 2003; Zhang and Xu, 2005; Jacob and Abdulhai, 2006; Lu *et al.*, 2008; Wen *et al.*, 2009). The Q-Learning agent learns the optimal mapping between the environment's (e.g., transportation network) state s and the corresponding optimal control action a based on accumulating rewards *r(s,a)*. Each state-action pair (s, a) has a value called *Q-Factor* that represents the cumulative reward for the state-action pair (s, a). In each iteration, *k*, the agent observes the current state *s*, chooses and executes an action a that belongs to the available set of actions *A*, and then the *Q-Factors* are updated according to the reward *r(s,a)* and the state transition to state s' as follows (Sutton and Barto, 1998);

$$Q^k(s,a) = (1-\alpha)Q^{k-1}(s,a) + \alpha\left[r(s,a) + \gamma \max_{a' \in A} Q^{k-1}(s',a')\right]$$

where $\alpha, \gamma \in (0,1]$ referred to as the learning rate and discount rate, respectively.

The agent can simply choose the *greedy* action at each iteration based on the stored Q-Factors, as follows;

$$a \in \arg\max_{b \in A}[Q(x,b)]$$

However, the sequence Q^k is proven to converge to the optimal value under certain condition that is the agent has to visit the state–action pair an infinite number of iterations (Gosavi, 2003). This means that the agent must sometimes *explore*, that is, try other actions, rather than *exploit* the best actions. To balance the exploration and exploitation in Q-Learning, algorithms such as ϵ-greedy and softmax are typically used (Sutton and Barto, 1998).

The design elements of the proposed algorithm in terms of the typical RL structure (i.e., state, action, reward,...etc) are discussed next;

- **State:**

Three Q-Learning models are developed; each considering different possible state representations as follows;

State Definition1: Arrival of vehicles to the current green direction and Queue Length at red directions

This state is represented by a vector of N components, where N is the number of phases. One of the state vector components is the maximum arrivals in the green phase and the other components are the maximum queue lengths for the red phases. This definition can be represented as follows:

$$s_i = \begin{cases} \underset{l \in L(i)}{max}\, Q_l \quad if\ i \neq green\ phase \\ \underset{l \in L(i)}{max}\, Ar_l \ \ if\ i = green\ phase \end{cases} \quad \forall\ i \in \{1,2,\dots,N\}$$

where Ar_l and q_l are the number of arriving and queued vehicles in lane l , respectively. The maximum is taken over all lanes l that belong to the set of lanes corresponding to phase i, $L(i)$. The vehicle is considered at a queue if its speed is below 5 kph. Similar state definition is used in (Bingham, 2001).

State Definition 2: Queue length

State definition 1 may have a drawback. In case the arrivals in the green directions outweigh the queued vehicles in the red directions, the algorithm may favour extending the green for the current phase. However, in some cases, the best action could be switch to another phase with less number of queued vehicles but larger value of the cumulative delay for these those vehicles (a few vehicles that have been waiting for a while). This is due to the fact that arrivals are not proportionally related to the delay experienced by the vehicles in the intersection. Therefore, it is plausible to consider the queue lengths as a better representation for the delay than state definition 1. Hence, state definition 2 is represented by a vector of N components that are the maximum queue length associated with each phase.

$$s_i = \max_{l \in L(i)} q_l \quad \forall\ i \in \{1,2,\dots,N\}$$

In the RL-based signal control literature, this state definition is the most common (Abdulhai *et al.*, 2003)

State Definition 3: Cumulative Delay

The vehicle cumulative delay CD^v is the total time spent by this vehicle *(v)* in a queue. The cumulative delay for phase *i* is the summation of the cumulative delay of all the vehicles that are travelling on the *L(i).* This state is also represented by a vector of N components where each component is the cumulative delay of the corresponding phase.

$$s_i = \sum_{v\ travelling\ \ on\ l\ \in L(i)} CD^v \qquad \forall\ i \in \{1,2,\dots,N\}$$

This state definition is motivated by the observation that there are two cases in which the delay of some vehicles is not properly captured by state definition 2:

> ***Case 1:*** The maximum queue lengths in two (or more) different approaches are equal while their cumulative delay is significantly different.
>
> ***Case 2:*** If there is no queue at the current phase but there are still some vehicles with high cumulative delay that did not pass the stop line, while the queue lengths of other approaches are high but the cumulative delays are low.

In state definition 2, the queue length might be a myopic representation of the cumulative delay encountered by vehicles at the intersection; a concern that we attempt to address by considering the cumulative delay as a state representation as in state definition 3.

- **Action:**

As discussed previously, a variable phasing sequence is used and the action is the phase that should be in effect next.

$$a = i\ , \qquad i \in \{1,2,\dots,N\}$$

It is worth noting that if the action is the same as the current green phase, this means that the green time for that phase will be extended by 1 sec (time interval). Otherwise, the green light will be switched to phase *a* after accounting for the yellow, all red, and the minimum green times. Therefore, the decision point varies according to the sequence of actions taken.

- **Reward:**

The immediate reward is defined as the change (saving) in the total cumulative delay, i.e., the difference between the total cumulative delays of two successive decision points. The total cumulative delay at time t is the summation of the cumulative delay, up to time t, of all the vehicles that are currently in the system. Vehicles leave the system once they clear the stop line. If the reward has a positive value, this means that the delay is reduced by this value after executing the action. However, a negative reward value indicates that the action results in an increase in the total cumulative delay.

A typical reward function considers the delay experienced by the vehicles between two successive decision points (e.g. Abdulhai *et al.*, 2003; Lu *et al.*, 2008). This typical definition however does not consider how long the vehicles were delayed before the last decision point. Table 1 demonstrates the difference between the reward definition of Abdulhai *et al.*, (2003) and Lu *et al.*, (2008) referred to as Reward Definition 1 and the our proposed definition, referred to as Reward Definition 2.

Table 1: Illustrative Example for Two-Phase Intersection

<table>
<tr><td>Iteration k-1</td><td colspan="2">Intersection Approach 1</td><td colspan="2">Intersection Approach 2</td></tr>
<tr><td>Queue Length</td><td colspan="2">2</td><td colspan="2">10</td></tr>
<tr><td>Cumulative delay</td><td colspan="2">60</td><td colspan="2">0</td></tr>
<tr><td>Action</td><td colspan="2">Switch to Phase 2 (after minimum green of 10 sec)</td><td colspan="2">Extend Phase 1 (by 1 sec time interval)</td></tr>
<tr><td>Iteration k</td><td>Approach 1</td><td>Approach 2</td><td>Approach 1</td><td>Approach 2</td></tr>
<tr><td>Queue Length</td><td>2</td><td>0</td><td>0</td><td>20</td></tr>
<tr><td>Delay experienced between the two iterations</td><td>2*10=20</td><td>0</td><td>0</td><td>20*1=20</td></tr>
<tr><td>Cumulative delay</td><td>60+20=80</td><td>0</td><td>0</td><td>0+20=20</td></tr>
<tr><td>Reward Definition 1</td><td colspan="2">-20</td><td colspan="2">-20</td></tr>
<tr><td>Reward Definition 2</td><td colspan="2">(60+0)-(80+0)=**-20**</td><td colspan="2">(60+0)-(0+20)=**+40**</td></tr>
</table>

It is clearly shown from the above example that the first reward definition does not differentiate between the two actions taken by the agent (switch to phase 2, extend phase 1). However, the second reward definition does not only differentiate clearly between the two actions, but also have two opposite signs for the corresponding rewards. This is primarily because the proposed reward definition reflects the change in the total cumulative delay while the fist definition considers only the absolute value of the delay for the time interval between the successive decision points.

It is also expected that the proposed definition would speed up the Q-Factors convergence since all the Q-Factors are initiated by zero values, and hence the actions that result in negative reward values will have the lowest probability to be chosen by the Q-Learning agent compared to those of highly positive values.

- **Action selection method:**

In each iteration, the ϵ-greedy method (Sutton and Barto, 1998) is used for action selection in which an ϵ-greedy learner selects the greedy action most of the time except for ϵ amount of the time, it selects a random action uniformly. The value of ϵ is chosen to decrease gradually with iterations (from 0.9 to 0.1). This will result in more exploration at the beginning of the learning process which enables the Q-Learning agent to search the overall state-action space and gradually emphasizes exploitation as the agent converges to the optimal policy.

Testbed Intersection

The agent is tested on a major intersection (4-approachs 3-lanes including an exclusive left turn lane) in Downtown Toronto in the heart of the financial district (Front and Bay Street, see Figure 1). This intersection is chosen as an example of an important mutli-phase intersection. The morning rush hour observed traffic demand data for year 2006 is attached to the figure in a form of an Origin-Destination (OD) matrix. Each of EB/WB and NB/SB has separate through and left-turn operations, resulting in four phase (movement) combinations as shown in Figure 4. The performance of the widely used fixed time control is used as a bench mark and is compared to the Acyclic Q-Learning control agent. The fixed time signal plan is optimized using Webster method (Webster, 1958). Paramics, a microscopic traffic simulator, is used to build the testbed intersection. The RL platform is developed as a standalone program. The interaction between the Q-Learning agent and the Paramics Environment is implemented through the Application Programming Interface (API) functions in Paramics.   While it is practically infeasible to continue the learning process indefinitely, a stopping criterion is specified to bring the Q-

Learning to an end. In this implementation, the learning process is terminated after 2000 *one-hour* simulation runs.

Four discrete intervals are used in state definitions 1 and 2 ([0-1), [1-3), [3-6), and [6-10]) which results in 256 states. Because of the high variability in state 3 (cumulative delay) six discrete state intervals are defined to cover wide range of states ([0-5), [5-10), [10-50), [50-100), [100-300), and [300-500]) which results in 1296 states.

Test Scenario Design

Two demand levels are modelled in this experiment; one represents the actual observed demand from field data and the other represents a 50% increase in the demand level. The latter mimics a future forecasting scenario or a severely congested intersection. For each demand level, two demand profiles (i.e. the temporal arrival pattern) are considered; uniform profile in which the demand is spread uniformly across simulation time, and variable profile in which each movement has differently randomized arrival rates around its mean arrival rate. This results in total of 12 test scenarios (3 state representations x 2 demand levels x 2 demand profiles).

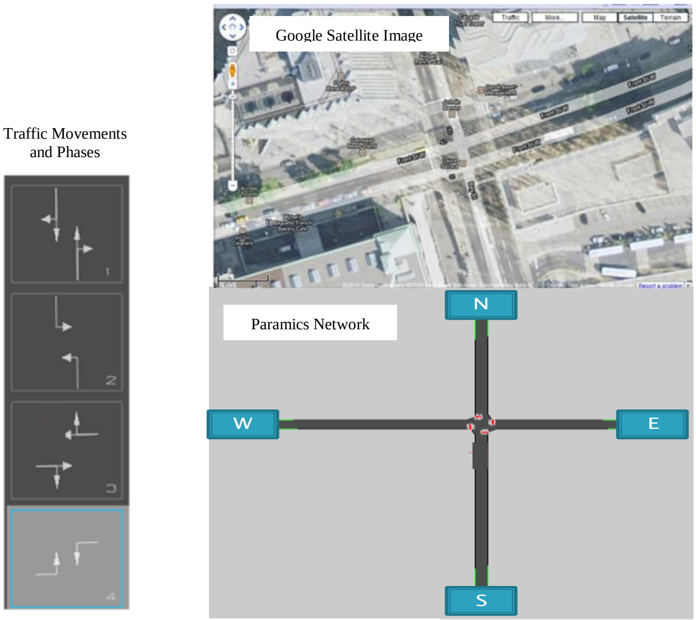
The models (RL and the Paramics APIs) are designed to output both aggregate and disaggregate level results as needed. At the aggregate level, the Q-Factors (Q-Tables) and the total cumulative delay are reported for the whole simulation runs and for each run, respectively. At the disaggregate level, the queue length for each lane, the average delay per vehicle for each lane, and the signal status (the current green phase and its current length) are reported.

Results and Analysis

Figure 2 demonstrates the convergence of the Q-Learning values. It can be seen from Figure 2 Total Vehicles Delay with Simulation Runs for (a) actual demand level, and (b) high demand level that the proposed acyclic Q-Learning approach consistently and considerably outperforms the pre-timed signal plan. For the actual demand case, compared to the fixed signal plan, the acyclic Q-learning approach reduces the total delay by 36% and 43% for the uniform profiles and variable profiles, respectively. The effectiveness of the acyclic Q-

Learning algorithm is more vivid in the variable profile case compared to the uniform profile which is intuitive.

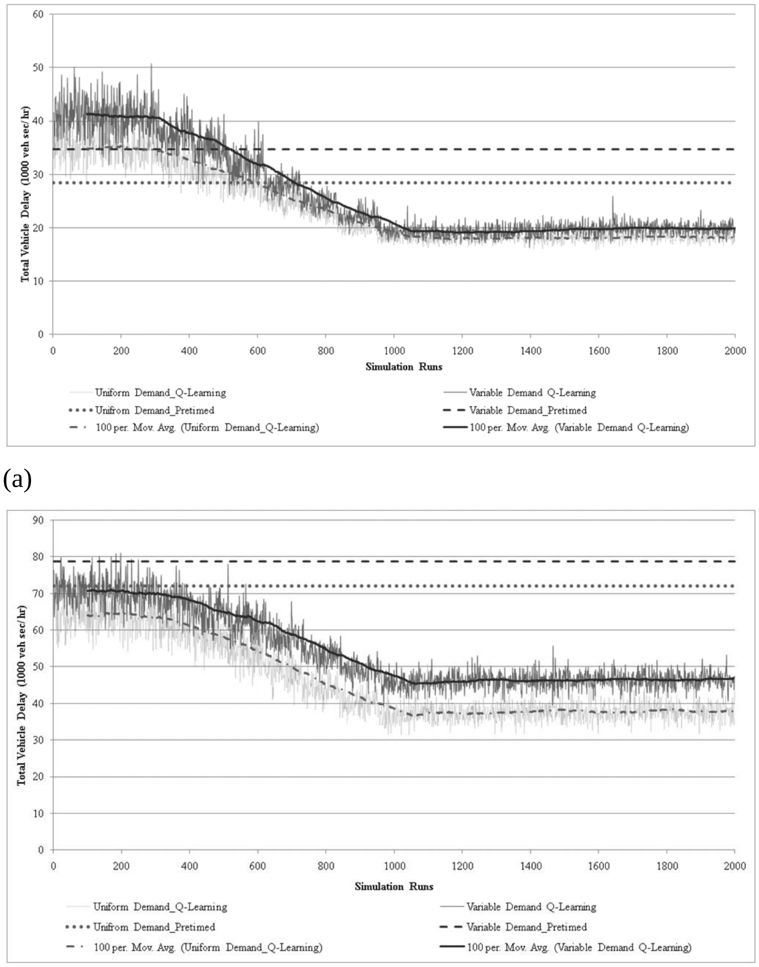
For the high demand level, similar trends are observed with proportional increase in the total delay values due to the increase in the demand level.



Traffic Volumes (Origin Destination Demand Matrix)

	E	S	N	W	Total
E	0	43	48	267	358
S	98	0	757	88	943
N	125	376	0	91	592
W	433	100	97	0	630
Total	656	519	902	446	2523

Figure 1 Testbed Intersection



(a)

(b)

Figure 2 Total Vehicles Delay with Simulation Runs for (a) actual demand level, and (b) high demand level

Figure 3 represents an example for the green time allocated for phases 1 using the acyclic Q-Learning approach compared to the fixed signal plan. It is clearly shown from Figure 3that the acyclic approach green splits are adapted to the demand profile. On the other hand, the fixed plan assigns a constant green time for each phase based on the flow per hour regardless of the demand variability within that hour.

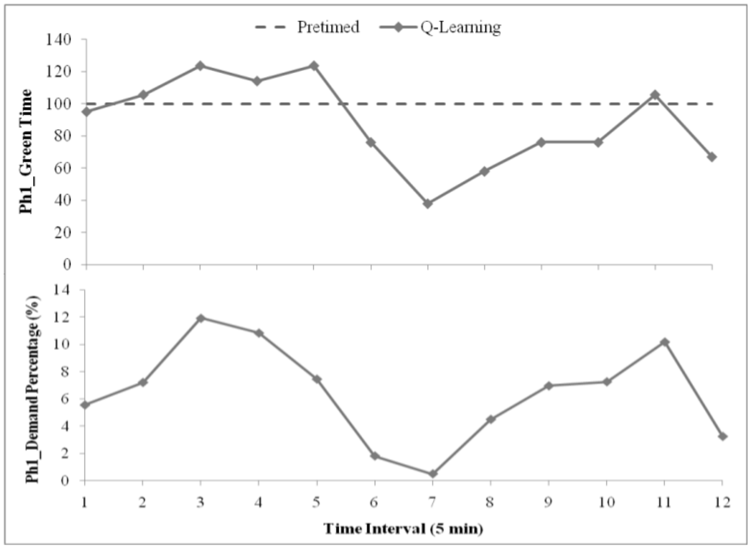


Figure 3 Allocated Green Time and Demand Arrival Percentages

Figure 4 illustrates the sum of the average approach delays (average intersection delay) for the 12 scenarios. It is shown that all state representations outperform the pretimed plan. No significant difference is observed between state definitions 2 and 3 in the actual demand case. However, in the high demand case, state definition 3 outperforms state 1 and 2. This is primarily due to the higher probability of occurrence of cases 1 and 2 (stated above) in higher demands level. State definition 1 on the other hand has the highest average delay compared to the other state definitions. This might be attributed to the low correlation between the cumulative delay and the number of vehicles arriving to the intersection.

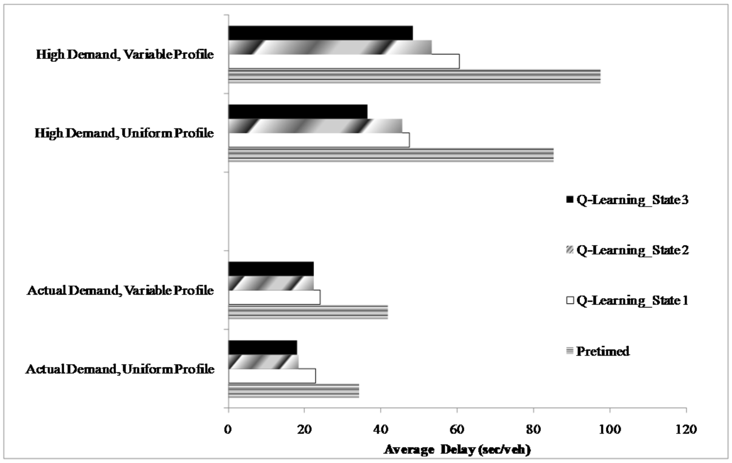


Figure 4 Average Delay Per Vehicle For Different Demand Levels and Profile

Conclusions

In this paper, Q-Learning-based acyclic signal control system is proposed that uses variable phasing sequence. Three models are developed; each with different state representation. A reward function that represents the cumulative delay reduction is proposed. The proposed models are tested on a simulation of a real-world multi-phase intersection in downtown Toronto and compared to the Webster-based pretimed signal control as a bench mark. The results showed that Q-learning approach consistently outperforms the pretimed signal plan with a wide margin regardless of state representations and the demand level. The effectiveness of the acyclic Q-learning approach is more vivid in the case of variable demand profile compared to uniform profile case; which reflects its adaptability to fluctuation in traffic conditions. For observed demand levels, there is no significant performance difference between state definitions 1 and 2. However, for higher demand levels state 3 (cumulative delay) is found to be more representative to the traffic conditions and produces better results when compared to states 1 and 2. From a practical perspective though, queue lengths are easier to measure than cumulative delay. The latter requires advanced sensing technology such as GPS or video tracking.

In this investigation, only the acyclic variable phasing approach is developed using the Q-learning algorithm; however, further research is needed to explore the performance of fixed phasing sequence cyclic Q-learning compared to the proposed approach. Various action selection algorithms (e.g. softmax) as well as different RL methods (e.g., SARSA, TD(λ)) can be investigated.

Acknowledgements

The authors gratefully acknowledge the financial support of the Connaught Scholarship of the University of Toronto and the OGSST scholarship. This research was enabled by the Toronto Intelligent Transportation Systems Centre.

References

Abdulhai, B. and L. Kattan (2003). "Reinforcement Learning: Introduction to Theory and Potential for Transport Applications." *Canadian Journal of Civil Engineering* 30(6): 981-991.

Abdulhai, B., R. Pringle and G. J. Karakoulas (2003). "Reinforcement Learning for True Adaptive Traffic Signal Control." *Journal of Transportation Engineering* 129(3): 278-285.

Bingham, E. (2001). "Reinforcement Learning in Neurofuzzy Traffic Signal Control." *European Journal of Operational Research* 131(2): 232-241.

De Oliveira, D., A. L. C. Bazzan, B. C. da Silva, E. W. Basso, L. Nunes, R. Rossetti, E. de Oliveira, R. da Silva and L. Lamb (2006). Reinforcement Learning-Based Control of Traffic Lights in Non-Stationary Environments: A Case Study in a Microscopic Simulator. *Proc. of EUMAS06, pp.31-42, 2006, Citeseer.*

Farges, J. L., J. J. Henry and J. Tufal (1983). The Prodyn Real-Time Traffic Algorithm. *Proc. of the IFAC Symposium, Baden-Baden.*

Gartner, N. H. (1983). "Opac: A Demand-Responsive Strategy for Traffic Signal Control." *Transportation Research Record: Journal of the Transportation Research Board* 906: 75-81.

Gosavi, A. (2003). *Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning*, Kluwer Academic Pub.

Head, K. L., P. B. Mirchandani and D. Sheppard (1992). "Hierarchical Framework for Real-Time Traffic Control." *Transportation Research Record* 1360: 82-88.

Jacob, C. and B. Abdulhai (2006). "Automated Adaptive Traffic Corridor Control Using Reinforcement Learning: Approach and Case Studies." *Transportation Research Record: Journal of the Transportation Research Board* 1959: 1-8.

Lu, S., X. Liu and S. Dai (2008). Incremental Multistep Q-Learning for Adaptive Traffic Signal Control Based on Delay Minimization Strategy. *Proceedings of the 7th World Congress on Intelligent Control and Automation June 25 - 27, 2008, Chongqing, China.*

McShane, W. R., R. P. Roess and E. S. Prassas (1998). *Traffic Engineering*, Prentice Hall.

Sutton, R. S. and A. G. Barto (1998). "Introduction to Reinforcement Learning." *MIT Press, Cambridge Mass.*

Thorpe, T. (1997). "Vehicle Traffic Light Control Using Sarsa." *Master's Project Rep., Computer Science Department, Colorado State University, Fort Collins, Colorado*.

Webster, F. V. (1958). *Traffic Signal Settings*, HMSO.

Wen, K., S. Qu and Y. Zhang (2009). A Machine Learning Method for Dynamic Traffic Control and Guidance on Freeway Networks. *2009 International Asia Conference on Informatics in Control, Automation and Robotics.*

Wiering, M. (2000). Multi-Agent Reinforcement Learning for Traffic Light Control. *Proceedings of the Seventeenth International Conference on Machine Learning Morgan Kaufmann Publishers Inc. San Francisco, CA, USA*

Zhang, L., H. Li and P. D. Prevedouros (2005). Signal Control for Oversaturated Intersections Using Fuzzy Logic. *Proceedings, the 84th Annual Meeting of The Transportation Research Board, Washington D.C.*

Zhang, Z. and J. M. Xu (2005). A Dynamic Route Guidance Arithmetic Based on Reinforcement Learning. *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 18-21 August.*