INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 10(5):627-642, October, 2015.

Traffic Signal Control with Cell Transmission Model Using Reinforcement Learning for Total Delay Minimisation

P. Chanloha, J. Chinrungrueng, W. Usaha, C. Aswakul

Pitipong Chanloha and **Chaodit Aswakul***
Wireless Network and Future Internet Research Group,
Department of Electrical Engineering, Chulalongkorn University,
Thailand, 10330.
*Corresponding author: chaodit.a@chula.ac.th

Wipawee Usaha
School of Telecommunication Engineering, Institute of Engineering,
Suranaree University of Technology,
Muang District, Nakhon Ratchasima,Thailand, 30000.
wipawee@sut.ac.th

Jatuporn Chinrungrueng
National Electronics and Computer Technology Center,
National Science and Technology Development Agency,
Klong Luang, Pathumthani, Thailand, 12120.
jatuporn.chinrungrueng@nectec.or.th

Abstract: This paper proposes a new framework to control the traffic signal lights by applying the automated goal-directed learning and decision making scheme, namely the reinforcement learning (RL) method, to seek the best possible traffic signal actions upon changes of network state modelled by the signalised cell transmission model (CTM). This paper employs the Q-learning which is one of the RL tools in order to find the traffic signal solution because of its adaptability in finding the real time solution upon the change of states. The goal is for RL to minimise the total network delay. Surprisingly, by using the total network delay as a reward function, the results were not necessarily as good as initially expected. Rather, both simulation and mathematical derivation results confirm that using the newly proposed red light delay as the RL reward function gives better performance than using the total network delay as the reward function. The investigated scenarios include the situations where the summation of overall traffic demands exceeds the maximum flow capacity. Reported results show that our proposed framework using RL and CTM in the macroscopic level can computationally efficiently find the proper control solution close to the brute-forcely searched best periodic signal solution (BPSS). For the practical case study conducted by AIMSUN microscopic traffic simulator, the proposed CTM-based RL reveals that the reduction of the average delay can be significantly decreased by 40% with bus lane and 38% without bus lane in comparison with the case of currently used traffic signal strategy. Therefore, the CTM-based RL algorithm could be a useful tool to adjust the proper traffic signal light in practice.
Keywords: Traffic Signal Control (TSC), Cell Transmission Model (CTM), Reinforcement Learning (RL).

1 Introduction

The opportunities in expanding physical transportation capacity within a well-established city are becoming practically limited. With continuing social and economic growth in metropolitan areas, many transportation facilities are being used to their full capabilities. The effects of demand management by using innovative local policies to match the unique nature of local

demand still remain to be explored. Without adding new facilities, attempts to operate and control the traffic by exploring existing capacity are challenging. Fortunately, the traffic problems can be handled by using advanced traffic information and control systems, which are among the most classical problems in traffic engineering. And computer technologies have been applied to find the optimal traffic signal timing for facilitating the traffic movements. More importantly, the main persistent challenge of the traffic problems is the ability to adapt traffic signals according to unexpectedly temporal traffic demand or road condition changes.

In this regard, intelligent *learning* methods to control the traffic signal and deal with the unexpected dynamics of road congestion status have been proposed in the literature. Of particular interest in this paper is an unsupervised-learning approach to find good traffic signal controls from experiences gained gradually by interacting directly with the road congestion environment. The herein adopted approach, reinforcement learning (RL) [1] and its potential to deal with the traffic engineering problems has been first proposed by [2]. More recently, in [3], Q-learning has been addressed as an RL technique to improve the control of integrated traffic corridor. For an isolated-intersection control [4], Q-learning has also been applied to control the traffic signal lights. All these existing literature have formulated RL with the road congestion environment that has been modelled in the detailed dynamics of individual mobility. Such behavior of microscopic traffic flow models can be too limited in their practical usages when the computational complexity becomes a major concern.

Alternatively, regarding the choices of environment models for RL, during the actual implementation phase of algorithm in the road network, RL can also be designed to measure directly its reward value from the observable abstraction of the considered road segments. However, a direct exposure of any learning algorithms to the actual system during a trial-and-error phase can lead to severe risks on road traffic problems and unsatisfactory public acceptance of the new control system. In this paper, we have proposed to integrate the RL concept with a computationally convenient macroscopic traffic flow model. In particular, the well-established *cell transmission model* or CTM [5] has been employed in this paper. The original version of CTM has been first proposed to model vehicle movements in an unsignalised network. Following developments of CTM to support a signalised road network have been proposed by [6], and further investigated by [7] for a TRANSYT system, and the signal optimisation with genetic algorithm has been proposed by [8] and by [9] based on a mixed-integer linear programming for two intersections. CTM has been refined recently to model the behaviors of multiple traffic classes [10] and with a more flexible topology mapping [11]. Recently, Q-learning has also been proposed with CTM to model the traffic flow dynamics [12] but the considered cases therein are successfully applicable to only a traffic route guidance problem, not to the signal optimisation problem as emphasised in this paper.

To study the traffic condition when the summation of overall traffic demand from all directions exceeds the maximum flow capacity, in the past we must rely on the whole network model. However, in this paper, by merely using a single intersection network scenario and with herein introduced boundary conditions to capture necessary vehicles backlog dynamics around the vicinity of the considered intersection, our proposed model can help reduce computational burdens a great deal. In addition, the developed CTM model can still maintain interesting insightful interpretations of control sequences. Particularly, with newly formulated RL environment using CTM parameters, three reward functions of RL to minimise the total network delay have been evaluated by considering the accumulative vehicle delays in only the directions facing the red light signal, receiving the green light signal, or in both directions. Our numerical experiments have shown that the choice of the red light delay has been found to outperform all the other choices in various traffic conditions. Moreover, the solution obtained from the macroscopic-viewpoint of RL has been compared with that from the best periodic signal solution (BPSS). Our comparison

using the microscopic simulator AIMSUN has helped confirm the applicability of the proposed CTM-based RL signal optimisation in this paper.

2 Problem Formulation

2.1 State Space

Suppose the vehicles in the systems belong to a single class e.g. personal cars. As shown

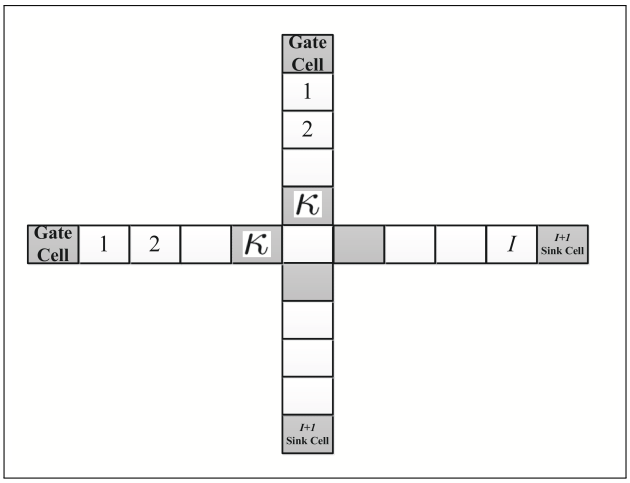

Figure 1: CTM boundaries and signalised cells

in Fig. 1, each road is partitioned into small cells $i = 1, ..., I$. The incoming demand patterns to an intersection is classified into P directions. Let $\mathcal{S}$ be the state space of the system. For each vehicle cell i in direction p at time slot t, define $s_i^p(t)$ as the number of vehicles. Let $\mathbf{s}(t) = [s_i^p(t),\ \forall(i,p)] \in \mathcal{S}$ be the state vector which represents the total number of vehicles in the system at time slot t. Note that in a real traffic scenario, the number of vehicles can be estimated from sensors on the road. To avoid the computational burden caused by the state space explosion, the quantisation technique is employed. The level of quantisations here can be represented by the number of deployed sensors in the road network. For simplification, let us define the quantised level of the total number of vehicles approaching the intersection from direction p at time slot t as $\tilde{s}^p(t) = \left\lceil \frac{\sum_{i=1}^{\kappa} s_i^p(t)}{\mathcal{C}} f \right\rceil + I(\sum_{i=1}^{\kappa} s_i^p(t) = 0)$, where $I(.)$ is the indicator function; $\mathcal{C}$ is the maximum number of vehicles totally allowable in each cell $i = 1, ..., \kappa$; and f is the total number of quantisation levels. The RL state can then be redefined as $\tilde{\mathbf{s}}(t) = [\tilde{s}^p(t),\ \forall p]$.

2.2 Cell Transmission Model

To incorporate the evolution of traffic dynamics in the system, a basic macroscopic model CTM is employed. The CTM parameters can be defined as follows [5].

Sending capability

Sending capability represents the ability to send the vehicles from cells to other cells, i.e., moving vehicles from beginning to ending cells. The sending capability can be defined as

$$\Lambda_i^p(t) = \min\left\{s_i^p(t), q_i^p(t)\right\}. \tag{1}$$

For cell i in direction p at time slot t, $\Lambda_i^p(t)$ is the sending capability; $s_i^p(t)$ is the number of vehicles; and $q_i^p(t)$ is the maximum number of vehicles that can flow through cell i.

Receiving capability

Receiving capability can be calculated by considering the remaining spaces in each cell and the maximum number of vehicles that can be present in the cell. Thus, for cell i in direction p at time slot t, its receiving capability can be defined as

$$\Psi_i^p(t) = \min\{q_i^p(t), \delta_i^p\,[c_i^p(t) - s_i^p(t)]\}, \tag{2}$$

where δ_i^p is the wave speed coefficient and $c_i^p(t)$ is the maximum number of vehicles that can be present. Note that the parameter $q_i^p(t)$ is influenced by the signal phase being chosen in cell i, direction p and time slot t in the action selection.

Cell cascading

This is the representation of the connection between two adjacent cells from the beginning cell $i-1$ and the ending cell i. The number of vehicles that flow in this cascading scenario can be calculated from the sending and receiving capability by

$$y_i^p(t) = \min\{\Lambda_{i-1}^p(t), \Psi_i^p(t)\}, \tag{3}$$

where $y_i^p(t)$ is the number of vehicles that flow into cell i in direction p at time slot t.

Flow conservation

Flow conservation is used to update the number of vehicles for the next time slot:

$$s_i^p(t+1) = s_i^p(t) + y_i^p(t) - y_{i+1}^p(t). \tag{4}$$

2.3 Action Space

To influence the system dynamics, for each time slot, the control agent (traffic controller) must select whether it would keep the current signal indication or change it. Such decision is called action. At state vector $\tilde{\mathbf{s}}$, an action must be selected from a state dependent set $\mathcal{A}(\tilde{\mathbf{s}})$. Specifically, $\mathcal{A}(\tilde{\mathbf{s}})$ is the set of all possible actions which a traffic controller can take at state $\tilde{\mathbf{s}}$. Define action a_t as the phase of signal light to be chosen (e.g, phase 1 for the green light from West to East and phase 2 for that from North to South) at time slot t. The indicator function $G^p(t)$ becomes one (zero) when vehicles in direction p get green (red) light in the chosen action at time slot t. Note that the action space $\mathcal{A}(\tilde{\mathbf{s}})$ must be defined such that all conflicting flows are not allowed to have green light at the same time.

The system dynamics are changed according to the traffic signal lights corresponding to the action taken $a_t \in \mathcal{A}(\tilde{\mathbf{s}})$. Assume that in one time slot, vehicles can move on average to the adjacent cells only. Let q_{max} be the maximum number of vehicles that can flow through each cell per time slot. For non-signalised cell i, the maximum number of vehicles that can flow through cell i in direction p at time slot t is given by $q_i^p(t) = q_{max}, \forall(p,t)$. For signalised cell i, the maximum number of vehicles that can flow through cell i is $q_i^p(t) = q_{max}$ when $G^p(t) = 1$ and $t - \tau_i(t) > L$. Otherwise, $q_i^p(t) = 0$. Note that L is the total starting/stopping loss time upon each signal change and $\tau_i(t)$ is the latest time instant where the traffic signal indication of cell i at time slot t has been changed.

Gate cell "0"

The boundary condition is here formulated by following [5]. At the boundary, input vehicle flows can be modelled by a cell pair ("00" and cell "0"). A source cell "00" with an infinite number of vehicles $s_{00}^p(t) = \infty$ ready to enter an initially empty gate cell "0" of infinite size, $c_0^p(t) = \infty$. The flow capacity $q_0^p(t)$ of the gate cell "0" is set to the desired link input flow. Thus, the boundary conditions can be can be expressed and written by $\Lambda_0^p(t) = \min s_0^p(t), q_0^p(t)$, $y_0^p(t) = q_0^p(t)$, $y_1^p(t) = \min\{\Lambda_0^p(t), \Psi_1^p(t)\}$ and $s_0^p(t+1) = s_0^p(t) + y_0^p(t) - y_1^p(t)$. Suppose that the output cell referred as the sink cell, for all exiting traffic has infinite size $c_{I+1}^p(t) = \infty$ and $q_{I+1}^p(t) = \infty$, The sink cell $I+1$ thus has the receiving capability of $\Psi_{I+1}^p(t) = \infty$.

2.4 Vehicle Delay

In RL, to quantify the consequence of the action taken at time slot t, an immediate reward in terms of vehicle delay is returned to the agent (traffic controller). Vehicle delay is defined as the number of vehicles that cannot move away from the present cell within each time slot. In this research, two types of vehicle delay are proposed, i.e., internal delay and external delay. At time slot t for each direction p, let $d_0^p(t)$ be the external vehicle delay and $d_i^p(t)$ be the internal vehicle delay in cell i. These delays can be expressed as

$$d_0^p(t) = s_0^p(t) - y_1^p(t), \tag{5}$$
$$d_i^p(t) = s_i^p(t) - y_{i+1}^p(t), \quad i = 1, 2, \ldots, I. \tag{6}$$

The external delay can be interpreted as the delay experienced by the vehicles that wait to enter the considered road network from its upstream neighbourhoods. The external delay value forms the boundary condition to capture necessary vehicle backlog dynamics around the vicinity of the considered intersection. The internal vehicle delay is the delay incurred within each cell along the considered road network. Combining both types of delay therefore reflects how well the action just taken by the agent (traffic controller) at state vector $\tilde{\mathbf{s}}$ is, by merely taking into account a single intersection. The next section provides the long term performance criteria in terms of these delay functions which will be optimised for the best possible traffic signal control by means of RL.

2.5 Performance Criteria

To evaluate the optimal policy (set of actions) that minimises the total network delay, the performance criteria $\Upsilon(t)$ at time slot t is defined as

$$\Upsilon(t) = \Upsilon_{red}(t) + \Upsilon_{green}(t), \tag{7}$$
$$\Upsilon_{red}(t) = \sum_{p=1}^{P}\sum_{i=0}^{I}(1 - G^p(t))d_i^p(t), \tag{8}$$
$$\Upsilon_{green}(t) = \sum_{p=1}^{P}\sum_{i=0}^{I} G^p(t)d_i^p(t), \tag{9}$$

where $\Upsilon_{red}(t)$ is the "red light delay" and $\Upsilon_{green}(t)$ is the "green light delay". The red (green) light delay is the total vehicle delay from all the cells in the directions that see the red (green) light.

3 Signal Optimisation By Q-learning Algorithm

Without loss of generality, let us index the signalised cells by κ as an example of CTM-based intersection model shown in Figure 1. Assume no turning movement is allowed at this intersection. The signalised cells κ are used to control the traffic flows from West to East and North to South. To tackle the road traffic problem where the system always changes, a well-known method that can learn directly from experiences is employed, namely, the Q-learning method [1]. Q-learning uses the action-value function $Q(\tilde{\mathbf{s}}, a)$ to evaluate the average future reward return expressed as a function of the current state $\tilde{\mathbf{s}}$ and action a. This section explains a step-by-step implementation of Q-learning algorithm proposed in the CTM framework.

To apply RL in a signalised CTM framework, a definite simulation length is used for periodically observing traffic behaviors within a study time-interval. When the current time slot of CTM reaches the simulation length, the system enters the next *episode*. In practice, episodes can represent the repeatable and non-repeatable traffic phenomena. On one hand, in a repeatable case, we can use RL to tackle a recurrent congestion, e.g. during rush hours, in which traffic behaviours statistically repeat themselves from one day to another. In this case, at the beginning of each episode, our road system modelled by CTM can be reset to the same initial-value cell density settings. On the other hand, in a non-repeatable case, RL can be used to deal with a non-recurrent congestion scenario resulted from unexpected incidences like accidents or road surface maintenance. In this case, our interest is on how RL would allow the signal controller to quickly learn and adapt its strategic decisions upon those unexpected changes. Consequently, the CTM state in the first time slot of next episode is defined in this case as the CTM state in the last time slot of previous episode.

Whether RL is applied in the repeatable or non-repeatable cases, within each episode, the RL-based traffic controller is designed to make a sequence of signal-light decisions. Let the decision epoch t_ω refer to the time instant when decision ω is made, where $\omega = 1, 2, \ldots$ and $t_\omega = t_1, t_2, \ldots$, respectively.

For each episode, the optimisation procedure of Q-learning can be summarised as follows. 1) *System Initialisation*
The number of vehicles in state vector $\mathbf{s}(0)$ can be intialised by (4) at the beginning of an episode to the latest observed state of the system in the previous episode in the non-repeatable case or to a nominal operating point of the system at the considered time period in the repeatable case. In practice, the number of vehicles $\tilde{s}^p(0)$ for all p can be measured from road traffic by counting from the sensors embedded on the road. The action value function $Q(\tilde{\mathbf{s}}, a)$ can be initialised to the latest updated value in the previous episode for the non-repeatable case or to zero for the repeatable case. It should be noted that, different intialisations of $Q(\tilde{\mathbf{s}}, a)$ yield different results, mainly, in terms of the time convergence (the time that the algorithm needs to learn to reach the solution). Let $\omega = 1$.
2) *Action Selection*
At decision ω, with the current state observable at $\tilde{\mathbf{s}}$, the agent (traffic controller) chooses an action $a \in \mathcal{A}(\tilde{\mathbf{s}})$ to control the traffic signal by changing $G^p(t)$ in Section 2.3. The action can be chosen by the ϵ-greedy algorithm [1], where the greedy action is here defined as

$$a = \arg\min_{a\prime} Q(\tilde{\mathbf{s}}, a\prime).$$

According to this algorithm [1], Q-learning chooses the greedy action with probability $1-\epsilon$. And, with probability ϵ, the other actions are randomly selected according to a uniform distribution. In practice, an ϵ is a small positive value representing the explorability of learning algorithm.
3) *Update of System Dynamics*
Calculate the CTM state from time slot $t = t_\omega$ to time slot $t = t_{\omega+1} - 1$. Here, the next state

vector ($\tilde{\mathbf{s}}\prime$) is calculated from the CTM state at time slot $t = t_{\omega+1} - 1$. In this paper, three Q-functions have been compared, namely, the total network delay by considering the accumulative vehicle delays in only the directions facing red light signal, receiving the green light signal, or both. The observed reward $R(\omega)$ can then be correspondingly calculated from

$$R(\omega) = \begin{cases} \sum_{t=t_\omega}^{t_{\omega+1}-1} \Upsilon(t) & \text{in case of total network delay} \\ \sum_{t=t_\omega}^{t_{\omega+1}-1} \Upsilon_{red}(t) & \text{in case of red light delay} \\ \sum_{t=t_\omega}^{t_{\omega+1}-1} \Upsilon_{green}(t) & \text{in case of green light delay.} \end{cases} \tag{10}$$

4) *Update of Action Value Function*
In this paper, the algorithm can learn from its past experiences accumulated in Q-function and the reward in (10) newly gained from the most recent action ω. By following [1], Q-function can be updated as follows

$$Q(\tilde{\mathbf{s}}, a) \leftarrow Q(\tilde{\mathbf{s}}, a) + \alpha[R(\omega) + \gamma \min_{a\prime} Q(\tilde{\mathbf{s}}\prime, a\prime) - Q(\tilde{\mathbf{s}}, a)].$$

Here, $Q(\tilde{\mathbf{s}}\prime, a\prime)$ represents the action value function for the next observable state vector $\tilde{\mathbf{s}}\prime$ and next possible action $a\prime \in \mathcal{A}(\tilde{\mathbf{s}}\prime)$. Practically, $\alpha \in (0, 1]$ is the learning rate and $\gamma \in [0, 1)$ is the discount rate applied to the future expected rewards.
5) *Update of State Variable and Timing Parameter*
Update state $\tilde{\mathbf{s}} \leftarrow \tilde{\mathbf{s}}\prime$. And update $\omega \leftarrow \omega + 1$.
6) *Stopping Condition*
Repeat steps 2)–5) until the end of episode.

4 Results and Discussions

This section is aimed at reporting the findings from our series of experiments. Firstly, the convergence time and corresponding computational complexity of the proposed Q-learning algorithm has been presented. Secondly, three reward functions in (10) have been compared in terms of the achievable minimum total network delay values. Thirdly, with the best choice in the reward value accounting for the vehicle delay in red-light traffic direction, Q-learning performance has been investigated in stationary/non-stationary stochastic loading scenarios. Lastly, the applicability of macroscopic CTM-based solution of the proposed Q-learning algorithm has been tested in microscopic mobility environments using AIMSUN. All the experimental results share the following common parameter settings.

1. System Parameters: As illustrated in Figure 1, suppose that the length of each road approaching the considered intersection is 800 metres and each road is discretised into 10 equal-length cells, i.e. $I = 10$. Each time slot has been set to 5 seconds. Each cell has the capacity $c_i^p(t)$ of 60 passenger car units (pcu) and the maximum flow rate $q_i^p(t)$ of 6.9 pcu/slot. The wave speed coefficient δ_i^p is 0.8. Note that the values of CTM parameters are based on the actual traffic data collection being calibrated for Payathai road in Bangok, Thailand [10].

2. Control Parameters: The length of each episode is 20 minutes or 240 time slots. An action has been chosen every 3 time slots. Note that the longer the action selection is, the more

outdated the decision becomes. The number of quantisation levels f has been set to 3. Practically, three levels are corresponding to the three sensors that are often deployed on the real road configuration. The first sensor at the entry of the road is used for preventing the spill-back of vehicles to upstream neighbourhoods. The second sensor is deployed in the middle of the road for the queue length estimation. The third sensor placed at the stop-line of the road is used for the wasted green prevention in an actuated signal control.

4.1 RL Validation

This paper proposes the newly developed version of the signalised CTM with RL. The validation of the RL in various traffic conditions are reported. The optimal signal timing under static traffic condition with fixed cycle length, namely, best periodic signal solution (BPSS) and the Q-learning solution by using the proposed framework have been compared. Define λ_1 and λ_2 as the average rate of arrival traffic from West to East and North to South, respectively. Consider deterministic demand patterns with $\{\lambda_1, \lambda_2\} = \{8,8\}, \{11,5\}, \{13,3\}, \{15,1\}$ pcu/slot. Note that the other traffic conditions can be achieved by other sets of demand patterns as well, but we have analysed the example of four settings given above. From trial-and-error, the RL parameters are set to $\epsilon = 0.1, \alpha = 0.01, \gamma = 0.005$ within 100 episodes. Theoretically, the learning rate (α) determines how fast the newly acquired information will override the old information. The possible value of α is in the range of $0 < \alpha \leq 1$. The discount factor (γ) determines the importance of future rewards where $0 \leq \gamma < 1$. If $\gamma = 0$, then the agent will be "opportunistic" by only considering current rewards. The parameter ϵ is a small probability, where a larger ϵ is used for a more exploration-oriented design and a smaller ϵ is used for a more exploitation-oriented design [1]. In practice, the parametric tuning for the algorithm is one of the major challenges because in different scenarios, the parameters need to be readjusted. However, the advantage of the effects of Q-learning parameters is the usable range of these parameters are wide. With the flexibility of the Q-learning parameters, the obtained solution of Q-learning can be found without readjusting as discussed in the following section of the performance in stationary/non-stationary stochastic loadings. By using our proposed red light delay (8) as the

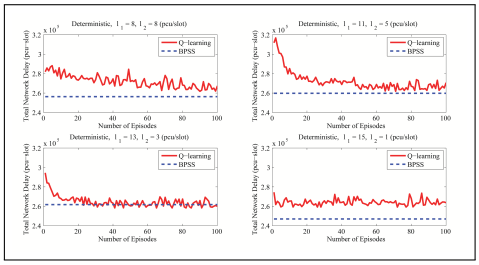

Figure 2: Total network delay from Q-learning vs BPSS

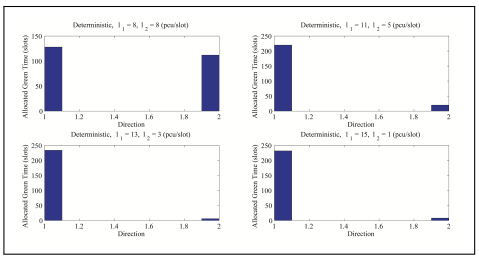

Figure 3: Allocated green time to each direction in last episode of Q-learning

reward function, Figure 2 and Figure 3 illustrate the total network delay and the allocated green time to each direction, respectively. Note that the red light delay used herein has been chosen from the following subsection focusing on the effect of reward functions. Figure 2 shows that the total network delay from Q-learning can be found close to the solution from BPSS in most scenarios. Particularly, when $\{\lambda_1, \lambda_2\} = \{15,1\}$ pcu/slot, Q-learning solution yields unsatisfactory result because of small traffic λ_2. Technically speaking, Q-learning requires the knowledge from its past experiences. But with a small traffic demand, the system cannot offer sufficient experiences to the Q-learning in order to achieve the solution properly. Figure 3 shows the number of time slots allocated to each direction. The result shows that the allocated green time in each

direction is proportional to the incoming traffic demand of that direction.

The computational complexity has been measured in terms of the required amount of memory and the computational time to achieve the final solution. Let the number of elements in the quantised state space be denoted by $|\tilde{\mathcal{S}}|$ and that in the action space be denoted by $|\mathcal{A}|$. Note that the action space $|\mathcal{A}| = P$ where P is the total number of all road network directions. Let k be the total number of the green time pairs in the overall searching space of periodic signal solutions. To search for the BPSS within these k possibilities per each state, the required amount of memory is $O(a^P)$ (approximately 748 kbytes) where a is a constant. However, the amount of memory required for Q-learning is $O(|\tilde{\mathcal{S}}|P)$ (approximately 114 kbytes). The BPSS grows exponentially depending on the number of the green time pairs to be searched whereas the growth of Q-learning depends on the quantised state space and the number of actions. The memory requirement can be saved with respect to the increasing of k. The computational time for the BPSS is $O(a^P)$ (approximately 1222 seconds) whereas the computational time for the Q-learning is $O(|\tilde{\mathcal{S}}|P)$ (approximately 15 seconds). The result shows that the computation of Q-learning for obtaining a control signal is significantly faster than BPSS.

4.2 Effect Of Reward Functions

The procedure to find the traffic signal solution has been illustrated in the RL validation. In this subsection, three different reward functions have been investigated in both symmetric and asymmetric loading patterns. To make the experiments more realistic, the traffic demand is no longer deterministic. In this subsection, the traffic demand is a Poisson process with a constant arrival rate for each direction. For symmetric loadings, both directions have equal approaching demand from $\{1,1\},\{3,3\},...,\{15,15\}$ pcu/slot, respectively. For asymmetric loadings, λ_1 has been set to 13 pcu/slot and λ_2 is varied from $1,2,...,15$ pcu/slot. The results have been obtained with the manually fine-tuned RL parameters $\epsilon = 0.1, \alpha = 0.01, \gamma = 0.005$. As illustrated

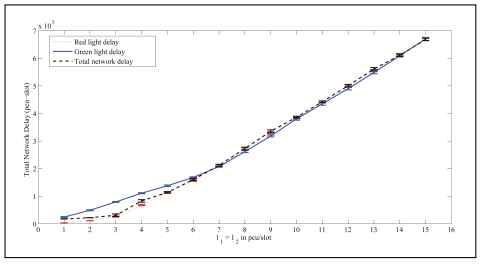

Figure 4: Total network delay from three reward functions on symmetric loadings

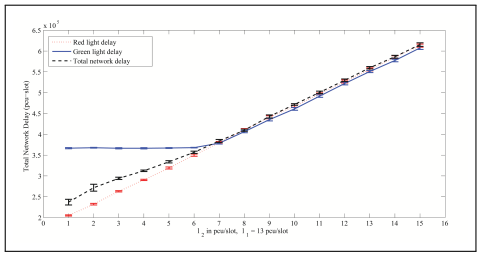

Figure 5: Total network delay from three reward functions on asymmetric loadings

in Figure 4 and Figure 5, with 95% confidence interval of both symmetric and asymmetric loadings, the proposed red light delay as the reward function decreases the total network delay in comparison with the conventional case of total network delay as the reward function and greatly decreases the total network delay in comparison with the case of green light delay as the reward function. The previous statement is valid in the loading region where the summation of overall traffic demand from all directions does not exceed its maximum flow capacity ($\lambda_1 + \lambda_2 \leq 6.9$ pcu/slot). On the contrary, when the summation of overall traffic demand from all directions exceeds the maximum flow capacity ($\lambda_1 + \lambda_2 > 6.9$ pcu/slot), the case of green light delay as the reward function yields slightly low total network delay in comparison with the case using the other two reward functions. Consider the system in the case where the summation of overall traffic demand from all directions does not exceed its maximum flow capacity. In this case, any control strategy can be used because usually there is no congestion of vehicles. In such

scenario, the control strategy is not complicated. However, the system in the case where the summation of overall traffic demand from all directions exceeds the maximum flow capacity, the control strategy has concerned because traffic congestion becomes a severe problem. Therefore, the following discussion will focus on the case of the summation of overall traffic demand from all directions exceeds the maximum flow capacity only.

Mathematical analysis when the summation of overall traffic demand from all directions exceeds the maximum flow capacity

To discuss all the results under the condition when the summation of overall traffic demand from all directions exceeds the maximum flow capacity, define the major flow (minor flow) as the incoming traffic demands that exceed (does not exceed) the capacity. Two types of the road traffic phenomena have been investigated. The experiments are concerned with a major flow conflicted with a minor flow (Ma-Mi condition) and two major flows conflicted with each other (Ma-Ma condition). For the Ma-Mi condition, consider an example demand setting $\{\lambda_1, \lambda_2\} = \{13, 3\}$ pcu/slot. Our experimental results in Figure 6, Figure 7 and Figure 8 show the total network delay in each time slot, the delay of all cells in each direction and the action chosen in each time slot, respectively. All the results in Figure 6, Figure 7 and Figure 8 have been observed at the final episode at the convergence.

With a simplified derivation, our result can be explained by using mathematical analysis as follows. Consider the derivation of accumulative delay of all cells in each direction as used in Figure 6 to Figure 8. From (5) – (6), the accumulative delay of all cells in direction p up to time slot T can be obtained from $\sum_{t=0}^{T}\sum_{i=0}^{I} d_i^p(t) = \sum_{t=0}^{T}\sum_{i=0}^{I}\big(s_i^p(t) - y_{i+1}^p(t)\big)$.

At the asymptote (all the cells in overloaded direction being fully occupied), define $\bar{\Upsilon}_{red}$, ($\bar{\Upsilon}_{green}$) as the asymptotic increasing rate of expected value of the accumulative red (green) light delay. Likewise, define $\bar{\Upsilon}$ as the asymptotic increasing rate of expected value of the accumulative total network delay. The term $y_{i+1}^p(t)$ becomes zero when calculating $\Upsilon_{red}(t)$ and becomes non-zero (6.9 pcu/slot) when calculating $\Upsilon_{green}(t)$. The calculation is therefore given by

$$\bar{\Upsilon}_{red} = \begin{cases} 3 - 0 = 3, & G^1(t) = 1 \\ 13 - 0 = 13, & G^2(t) = 1, \end{cases} \tag{11}$$

$$\bar{\Upsilon}_{green} = \begin{cases} 13 - 6.9 = 6.1, & G^1(t) = 1 \\ \max(3 - 6.9, 0) = 0, & G^2(t) = 1, \end{cases} \tag{12}$$

$$\bar{\Upsilon} = \begin{cases} 3 + 6.1 = 9.1, & G^1(t) = 1 \\ 13 + 0 = 13, & G^2(t) = 1 \end{cases} \tag{13}$$

From (11), if the reward function is $\Upsilon_{red}(t)$, then the minimum total network delay can be achieved by allocating the green light signal to the major flow (λ_1). Likewise, in (12), if the reward function is $\Upsilon_{green}(t)$, then the minimum total network delay can be achieved by allocating the green light signal to the minor flow (λ_2). Using $\Upsilon_{green}(t)$ as the reward function leads to the wasted green scenario (green light allocation to a particular direction without remaining vehicles) as illustrated by the term $max(3 - 6.9, 0)$. However, if $\Upsilon(t)$ is chosen as the reward function, then the minimum total network delay can be achieved by allocating the green light signal to the major flow (λ_1). The total network delay is a bit higher than the case of $\Upsilon_{red}(t)$. To explain why the total network delay from $\Upsilon(t)$ is higher than $\Upsilon_{red}(t)$. There are two concerned effects in using $\Upsilon_{red}(t)$ or $\Upsilon(t)$ as the reward function. One is the indistinguishable effect from $\Upsilon(t)$ where the agent only knows the overall network delay (7). Regardless of whether proper or improper

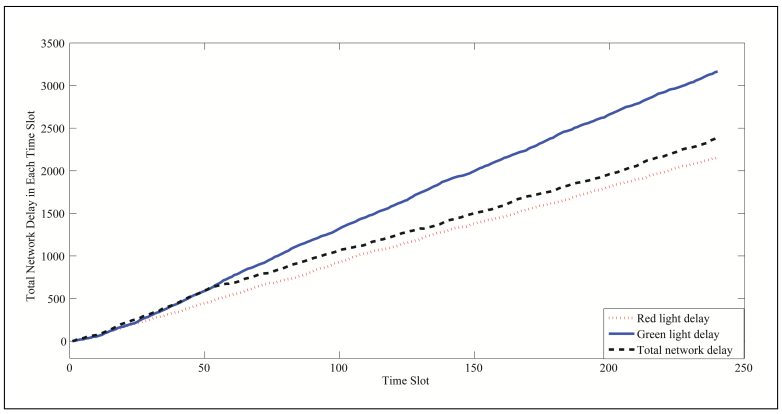

Figure 6: Ma-Mi: Total delay in each time slot

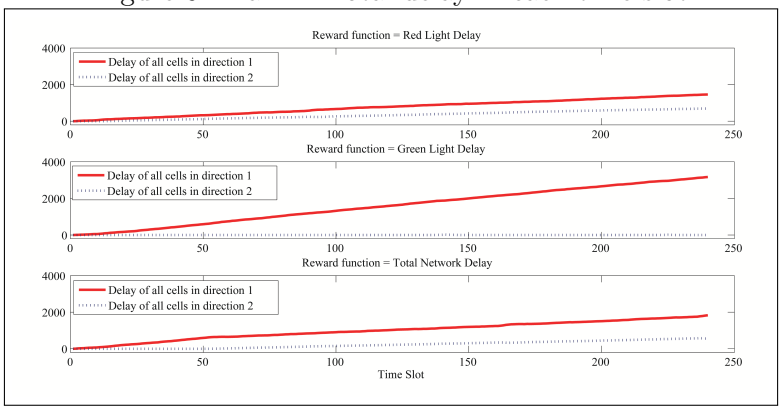

Figure 7: Ma-Mi: Three types of reward functions and its delay in each component

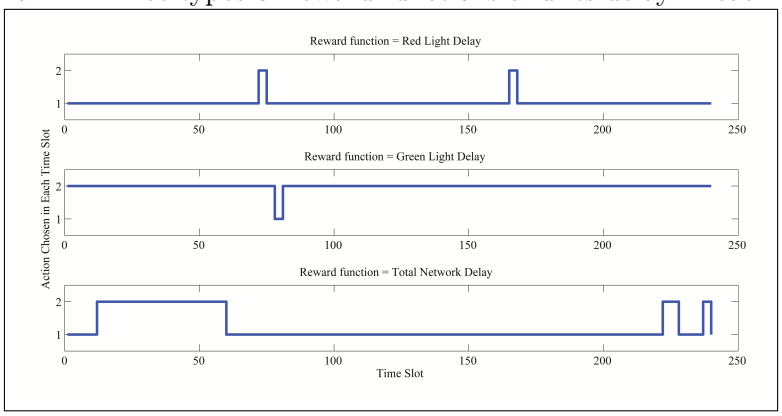

Figure 8: Ma-Mi: Action chosen in each time slot

action has been chosen, the value of reward in terms of total network delay is indifferent due to the summation of all vehicle delays in the system. The indistinguishable effect results in an inaccuracy (an improper action selection) and an inefficiency (an increasing of undesirable total network delay) of the action selection from Q-learning. Another is the timing effect of switched actions. In this case, the more often the action switches, the worse the total network delay is. From the discussion in the Ma-Mi condition, the recommended reward function would be the red light delay ($\Upsilon_{red}(t)$), which gives the lowest total network delay in comparison with the other two reward functions.

Likewise the Ma-Mi condition, the Ma-Ma condition can be calculated using the same ideas

above. For Ma-Ma conditions, the proper management of traffic signal control becomes a major concern. The next recommended traffic signal would preferably remain the same as the current traffic signal to avoid the occurrence of the system loss time. The any reward function can be used because the traffic is jammed. The reduction of total network delay becomes insignificant.

The goal is for RL to minimise the total network delay. Surprisingly, by using the total network delay as a reward function, the results were not necessarily as good as initially expected. Rather, both simulation and mathematical derivation results confirm that using the newly proposed red light delay as the RL reward function gives better performance than using the total network delay as the reward function. Note that a good reward function must be able to allow the algorithm to steer its instantaneous searching directions towards the final goal of minimising the total network delay. But that reward function itself needs not be the objective function i.e. the total network delay. Instead, from our numerical experiments, one should rather opt for using the red-light delay as the reward function so that the effect on future expected total network delay can be reflected within only a few time slots after an action decision has been made. On the contrary, if the total network delay is used as the reward function, then the algorithm eventually cannot find the proper solution.

4.3 Q-Learning Performance In Stationary/Non-Stationary Stochastic Loadings

In the RL validation section, four different traffic demand patterns have been investigated. In fact, such simplification can be relaxed to more realistic case by considering on the random source probabilities. Let the traffic demand be a Poisson process with a constant arrival rate for each direction. From the previous subsection, the red light delay has been chosen as a reward function. The performance of Q-learning in adapting its solution to reach the convergence will be examined. The experiments have been set into two scenarios. Firstly, the stationary test, the change of traffic demand from a deterministic to a Poisson has been illustrated in Figure 9. Secondly, the non-stationary test, in reality, road network capacity changes upon time (early morning, rush hour, etc.) as illustrated in Figure 10. Starting from uncongested traffic condition, the 1^{st} episode until the 100^{th} episode, the traffic demand pattern is $\{\lambda_1, \lambda_2\} = \{6, 6\}$ pcu/slot. And then, the road network becomes congested (jammed) condition, the $101^{st} - 140^{th}$ episodes, traffic demand pattern is therefore changed to $\{\lambda_1, \lambda_2\} = \{13, 3\}$ pcu/slot. The congested condition returns to uncongested condition, the $141^{st} - 180^{th}$ episodes, the traffic demand pattern is $\{\lambda_1, \lambda_2\} = \{6, 6\}$ pcu/slot. Finally, the congested condition happened again, the episodes 181^{st} the traffic demand pattern is $\{\lambda_1, \lambda_2\} = \{11, 5\}$ pcu/slot. The results show the adaptability of

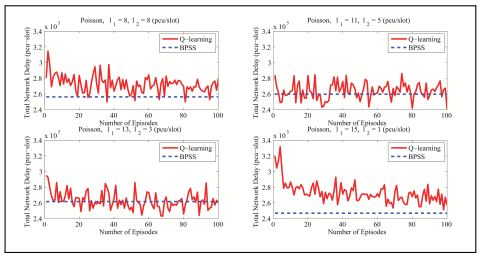

Figure 9: Total network delay from Q-learning with Poisson arrival

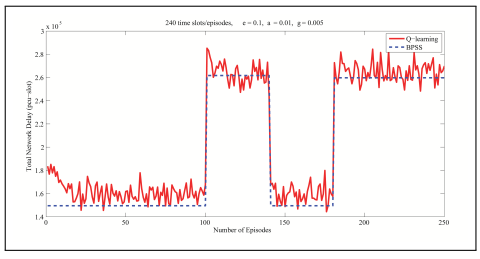

Figure 10: Total network delay obtained from Q-learning with the change of load patterns

Q-learning in reaching the solution close to the obtained solution from the BPSS method in both experiments. The abrupt change of the traffic demand patterns from uncongested to congested conditions have been imposed. However, the Q-learning still performs well in tracking closer to

the BPSS solution. Therefore, with significantly less demanding computational time than BPSS, the Q-learning algorithm can be used in real-time learning-based scenarios.

4.4 Comparison in Practical Case Study

In this subsection, we investigate an isolated intersection located in the middle of Bangkok, Thailand. As illustrated in Figure 11, the investigation area covers the Ratchapruek and the Ratchadaphisek roads where two segments are the arterial. While operating in the rush-hour periods, two roads are fully occupied. In reality, the intersection is controlled by the 3-phase signal timing plan. However, from the measured data, one of these three signal phases can be considered as the minor road segments because there are relatively few vehicles in comparison with the other two directions. So, the signal phases have been simplified to only two signal phases allocated to the Ratchapruek and the Ratchadaphisek road, respectively. The simulation settings have been set to the real data measured from the embedded sensors on each road segment at the intersection. All the considered road segments are 3-lane; the length of each road is 800 metres and each road is equally divided into 10 equal-length cells. The system parameters have been based on the previous subsection. The maximum green time is set to 120 time slots. The control plan of the CTM-based RL obtained from the MATLAB will be applied to the AIMSUN for the study in microscopic levels.

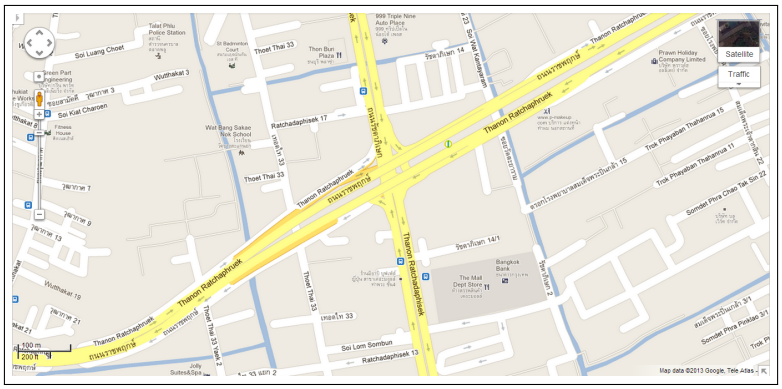

Figure 11: Ratchadaphisek-Ratchapruek intersection

Figure 12 and Figure 13 illustrate the time history of the number of vehicles approaching the Ratchadaphisek and Ratchapruek intersection. For the microscopic simulation settings, a bus

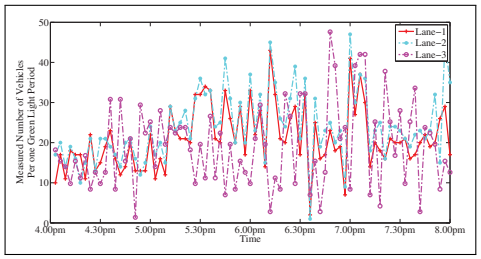

Figure 12: Number of vehicles from the Ratchadaphisek road approaching the intersection

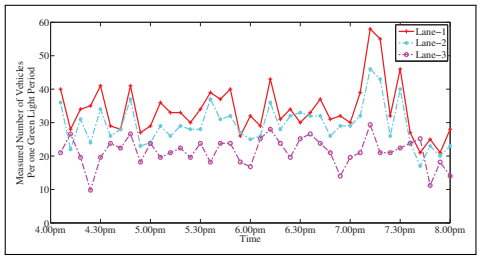

Figure 13: Number of vehicles from the Ratchapruek road approaching the intersection

stop is placed on the Ratchadaphisek road at 200 metres before its stopping line. As illustrated

in Figure 14, each bus waits at the bus stop for picking up the passengers. The 1^{st} lane of the Ratchadaphisek road has been occupied by buses and taxis and the 2^{nd} lane has been affected from those public vehicles. Therefore, the part of the road segment around the bus stop becomes a temporary blockage. In this scenario, the bus stop has been transparently embedded into the CTM model by permanently reducing the average vehicle speed on the bus lane. With the measured data, the simulation testing in the AIMSUN has been set to 4 hours in the evening rush-hour period from 4.00 pm to 8.00 pm. From the recorded data, the average speed of the vehicle is 50 km/h. However, the average speed in the bus lane has been reduced from 50 km/h to 8 km/h and the average speed of the middle lane has been reduced from 50 km/h to 30 km/h, respectively. The results have been obtained on three different signal control strategies which are the actual control obtained from the sensors, the fully actuated control using the embedded sensors deployed along the road segments and the control plan from CTM-based RL obtained from MATLAB. As illustrated in Table. 1, the proposed CTM-based RL reveals that

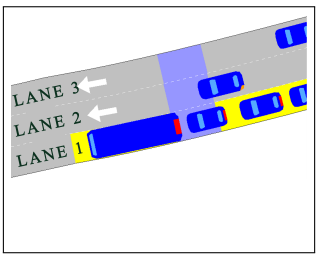

Figure 14: Road network with a bus lane

Table 1: The comparison among three types of traffic signal control

	Actual control	Actuated control	CTM-based RL
average vehicle delay (seconds)	137	97.9	82.7
average vehicle delay reduction (percent)	-	28.54	39.64
time spent for vehicle dissipations (hours)	3.4	3.1	2.6

the reduction of the average delay can be significantly decreased by 39.64%. From the recorded data, the green light status has been changed too often. By that, this control plan is not good as expected because of the system loss time from the frequent signal switching. For the fully actuated control, the maximum-gap distance technique between vehicles has been employed. Note that the actuated control employs vehicle detectors installed around an intersection to change the traffic signals of that intersection. Once vehicle detectors response for actuation, the actuated phase normally starts with a minimal preset green time, and green time phase is automatically extended [13]. The average speed of the vehicles has been reduced from the bus stop. Therefore, the traffic signal has been changed upon their own gap distance and individual speed. As illustrated in the previous subsections, the green light will be opened as long as possible up to the maximum green time. The CTM-based RL tries to avoid the system loss time; therefore, the average delay can be reduced. Moreover, the total time spent for vehicle dissipations has also decreased from 3.4 hours to 2.6 hours (approximately reduced by 22.53%). As illustrated in Table. 2, the scenario has been slightly changed by removing the bus lane from the actual bus stop whereas the traffic arrivals from the other directions are totally unchanged. In such scenario, the average speed of the vehicle is set 50 km/h because the temporal blockage from buses and taxis has been removed. For the case when the removing the bus stop, the proposed CTM-based RL reveals that the reduction of the average delay can be significantly

Table 2: The comparison among three types of traffic signal control without bus lane

	Actual control	Actuated control	CTM-based RL
average vehicle delay (seconds)	89.5	60.3	55.7
average vehicle delay reduction (percent)	-	32.63	37.77
time spent for vehicle dissipations (hours)	2.7	2.5	2.1

decreased by 37.77% and significantly reduced the time spent for vehicle dissipations by 22.22%.

5 Conclusion

A new framework to control the traffic signal lights by applying one of the reinforcement learning tools, namely, the Q-learning has been proposed to seek the best possible solution to control the traffic signals where the network state has been modelled by the signalised cell transmission model. The road traffic condition is mainly focused on the situation when the summation of overall traffic demand from all directions exceeds the maximum flow capacity.

The proposed framework is used to find the best traffic signal strategy. Surprisingly, using the newly proposed red light delay as the RL reward function gives better performance than using the total network delay as the reward function. The results have been reported from the series of experiments which are the RL validation, the effect of reward functions, the RL performance in stationary/non-stationary stochastic loadings and the applicability of the CTM-based solution of the RL algorithm in the microscopic mobility environments using AIMSUN.

The simulation results show that our proposed framework can computationally efficiently find the proper solution for road traffic systems by comparing with the best periodic signal solution (BPSS). The effect of reward functions has also been investigated and the adaptability of the RL algorithm in adjusting its solution with Poisson arrival upon the change of time has also been observed. The results from the macroscopic level show that RL yields the results similar to the BPSS method. For the practical case study conducted by AIMSUN, the proposed CTM-based RL reveals that the reduction of the average delay can be significantly decreased by 39.64% from the actual traffic signal strategy. For the case when the removing the bus stop, the CTM-based RL also has also been reduced the average delay by 37.77%. Therefore, the practical case study from the urbanised isolated intersection can provide substantially impact to the transportation problems.

With the newly proposed reward function applied to an isolated intersection, this paper has reported the results and its applicabilities. The extension of our proposed framework for a road network scale is currently ongoing and the results will be reported in the forthcoming papers.

Acknowledgments

The authors would like to acknowledge support received from Thailand Graduate Institute of Science and Technology (TGIST), associated with National Science and Technology Development Agency (NSTDA), the Special Task Force for Activating Research (STAR) Funding in Wireless Network and Future Internet Research Group, Chulalongkorn University and Associate Professor Dr. Sorawit Narupiti at transport engineering division of Department of Civil Engineering, Chulalongkorn University for technical support in transportation infrastructures and intelligence controls.

Bibliography

[1] R. Sutton, A.G. Barto (1998); *Reinforcement Learning: An Introduction.* The MIT Press, Cambridge, Massachusetts.

[2] B. Abdulhai, L. Kattan (2003); Reinforcement learning: Introduction to theory and potential for transport applications. *Canadian Journal of Civil Engineering*, 30(6), 981–991.

[3] C. Jacob, B. Abdulhai (2005); Integrated traffic corridor control using machine learning. *International Conference on Systems, Man & Cybernetics*, 3460–3465.

[4] D.D. Oliveira et al (2006); Reinforcement learning-based control of traffic lights in non-stationary environments: a case study in a microscopic simulator. *Forth European Workshop on Multi Agent Systems.*

[5] C.F. Daganzo (1995); The cell transmission model part II: Network traffic. *Transportation Research Part B: Methodological*, 29b(2), 79–93.

[6] H.K. Lo et al (2001); Dynamic network traffic control. *Transportation Research Part A: Policy and Practice*, 35(8), 721–744.

[7] M. Maher, O. Feldman (2002); The application of the cell transmission model to the optimisation of signals on signalised roundabouts. *European Transport Conference*, 1–13.

[8] H.K. Lo, A.H.F. Chow (2004); Control strategies for oversaturated traffic. *Journal of Transportation Engineering*, 466–478.

[9] W.H. Lin, C. Wang (2004); An enhanced 0-1 mixed-integer LP formulation for traffic signal control. *IEEE Transactions on Intelligence Transportation Systems*, 5(4): 238–245.

[10] K. Tueprasert, C. Aswakul (2010); Multiclass cell transmission model for heterogeneous mobility in general topology of road network. *Journal of Intelligent Transportation Systems*, 14(2): 68–82.

[11] G. Flotterod, K. Nagel (2005) Some practical extensions to the cell transmission model. *Proceedings of the 8th Internationall IEEE Conference on Intelligent Transportation Systems.*

[12] A. Sadek, N. Basha (2006); Self-learning intelligent agents for dynamic traffic routing on transportation networks. *International Conference on Complex Systems*, 503–518.

[13] N.H. Gartner et al (1995); Development of advanced traffic signal control strategies for Intelligent Transportation Systems : multilevel design, *Transportation Research Record*, 98–105.