Proceedings of the 16th International IEEE Annual Conference on Intelligent Transportation Systems (ITSC 2013), The Hague, The Netherlands, October 6-9, 2013

Q-learning Method for Controlling Traffic Signal Phase Time in a Single Intersection

Sahar Araghi, Abbas Khosravi, Michael Johnstone, and Doug Creighton

Abstract— **This study investigates the optimal setting of green times for traffic lights in an isolated intersection with the purpose of minimizing congestion. A machine learning method forms the backbone of the proposed method. Here, Q-learning is applied for signal light timing to minimize total delay. It is assumed that an intersection behaves similar to an intelligent agent learning to plan green times in each cycle using current traffic information. Compared to previous studies in this field, we expand the state space and innovatively set the reward to the average difference between traffic that enters the intersection and the queue length in the corresponding links. In contrast to previous studies, it is also assumed that the cycle time is variable. The performance of the proposed method is comprehensively compared with two traditional alternatives for controlling traffic lights. Simulation results indicate that the proposed method significantly reduces the total delay in the network when compared to the alternative methods.**

I. Introduction

Increases in the number of vehicles and higher levels of urbanization have emphasized the need for an efficient transportation system. A fully developed city has a limited ability to reconstruct its traffic network, thus, providing real time control is a necessary part of modern traffic control systems.

The main signal control task is to simultaneously increase the intersection capacity, decrease delays and guarantee the safety of people and vehicles. Furthermore, it can reduce fuel usage and emission by decreasing stop and delay times.

Different approaches have been taken when studying high traffic demands in urban cities. Fixed-time traffic light control is one of the earliest methods. In the fixed-time method, a predetermined time for each intersection's traffic light is based on previously surveyed traffic flows for different times of the day and different days of the year. This method has a considerable effect in reducing travelling delay time, but is not based on real traffic information and cannot adapt to any sudden changes in traffic conditions. Therefore, having a more flexible approach is necessary.

SCATS is an intelligent transportation system developed in Sydney and is now used in many parts of Australia and also in other countries. It is a kind of adaptive system, which has detectors for recognizing the amount of traffic in an intersection and extends or terminates green times based on current traffic volumes. SCATS is able to manage traffic according to predetermined programs for each individual intersection corresponding to the time of day or day of week.

Considerable effort has been given to implementing learning methods as intelligent approaches for planning traffic lights. Game theory [1], fuzzy logic [2],[3], neural networks [2],[3] and Q-learning [4], [5], [6], [7], [8] are examples of the learning methods, all of which have had positive effect in reducing traffic congestion and improving traffic flow.

In this paper, Q-learning is applied for controlling signal lights for an isolated intersection. The objective is to reduce the average delay time of the traffic using the intersection. Q-learning is a reinforcement learning method that is able to learn the optimal strategy for signal timing without the need for a model of the system. Q-learning is an important and well-studied reinforcement learning algorithm and it is known that its policy converges to the optimal point [8].

Here, we improve on Abdoos et al.'s [9] method for controlling traffic lights. We use the method proposed in [9] as the base of our learning method. By changing the definition of state spaces, it becomes possible to recognize more cases in comparison to [9]. Recognizing states in more detail is useful for allocating appropriate green time and therefore making the learning more accurate. We also avoid the limitation of fixed cycle time and provide more actions in our Q-learning method. Previous methods consider all actions to have positive rewards, which may cause delays in learning. Here, both negative and positive consequences of an action are considered as rewards. Finally, we implement our proposed method, Abdoos et al.'s method and a fixed-time method and present a comparison. The results illustrate an improvement in reducing the average delay time of traffic entering the intersection during simulation time.

The rest of this paper is organized as follows. Reinforcement learning and Q-learning are discussed in section II. In section III, related work that applies reinforcement learning for controlling traffic signal lights is presented. Our proposed method is presented in section IV. Experiments and discussions are presented in section V and our conclusions are stated in section VI.

II. Reinforcement Learning

Reinforcement learning is a machine learning method useful in unknown environments. In this method, learning progresses based on environmental feedback, named reward or punishment related to the problem, as shown in Fig.1.

Sahar Araghi, Abbas Khosravi, Michael Johnstone, and Doug Creighton are with the Centre for Intelligent Systems Research (CISR), Deakin University, Geelong, Australia (e-mail: {saraghi, abbas.khosravi, michael.johnstone, douglas.creighton}@deakin.edu.au).

978-1-4799-2914-613/$31.00 ©2013 IEEE

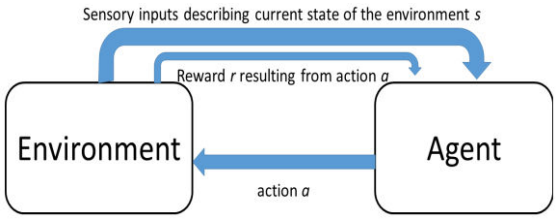

Figure1. Interaction of agent and environment in reinforcement learning.

In reinforcement learning, an agent can learn an optimal policy without being able to predict the impact of its actions, its immediate payoffs or planning ahead. An agent requires a minimal amount of episodic memory for storing the last action taken, the related state in which the last action was selected, the current state, and the payoff received [10].

Q-learning is an incremental reinforcement learning method that does not need a model of the environment and can be used online [11]. The Q-learning algorithm stores the expected reinforcement value associated with each state-action pair, represented by $Q(s_t, a_t)$, in a look-up table. The goal of the Q-learning algorithm is to choose an action that maximizes discounted cumulative rewards over time. It means that there is higher value for short term rewards in comparison to distant future rewards. The model of Q-learning is based on Markov decision process formalism. An agent who is in a state s_t, performs an action a_t, and receives the reward r_t for the related state-action from the environment. The state is changed to s_{t+1} by considering probability transition T. The Q-values stored in the Q-table are updated as below:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\,(r_t + \gamma\ max_{a \in A}\, Q(s_{t+1}, a) - Q(s_t, a_t)\,) \quad (1)$$

where $\alpha\ (0 \le \alpha < 1)$ is the learning rate and $\gamma\ (0 \le \gamma \le 1)$ is the discount factor.

In a particular state, Q-learning helps to choose the next action based on the expected reward. Fig. 2 provides a detailed description of the Q-learning algorithm.

Q-Learning algorithm

Initialize Q(s, a) arbitrarily
Repeat (for each episode):
  Initialize s
  Repeat (for each step of episode):
    Choose a from S using policy derived from Q (e.g. ϵ-greedy)
    Take action a; observe r, s'
    $Q(s,a) \leftarrow Q(s,a) + \alpha\,[r + \gamma\ max_{a'}\, Q(s', a') - Q(s,a)]$
    $s \leftarrow s'$
  Until s is terminal

Figure 2. Q-learning algorithm pseudo code.

III. Reinforcement Learning for Traffic Light Control

In the field of traffic light signal control, one of the most common goals is to improve the efficiency of the controller by maximizing the traffic throughput at an intersection.

For a dynamic view of traffic, a flexible approach to control is reinforcement learning. Wieiring et al. [12] proposed a transition model that estimates waiting time of cars in different states. In [13], a context detection reinforcement learning method is presented. This method creates a partial model of the environment based on demand. Over time, the partial models are improved or new ones are constructed. In [14] and [15], adaptive reinforcement learning to control a model free traffic environment is proposed. An adaptive control of traffic lights is also studied in [8], where a function approximation method is applied as a mapping between states and signal timing.

Jacob et al. [16] used Q-learning as a solution for controlling highway traffic. For isolated intersection control, [12] and [17] consider Q-learning with different objective functions. These studies seek to achieve optimal traffic signal control for an isolated intersection. Although in reality there are multiple intersections in an urban traffic network, here we focus on the optimization of traffic signals for one intersection.

In [9], Abdoos et al. proposed a method for controlling signal lights for each intersection based on Q-learning and have extended that approach to 50 intersections. Each intersection is presented as an agent and the whole system formed as a multi-agent system. They considered the average queue length in approaching links in a fixed cycle as states of Q-learning. The number of permutations of the approaching links form the number of states; for example, k! for an agent with k approaching links. Abdoos et al, considered an intersection with four approaching links and their state space consists of 24 states. Table I shows their state space. In this table, l_i represents the lengths of queue in an approaching link i.

Abdoos et al. considered different phase splits of the cycle time for the actions of Q-learning. Phase split refers to the division of the cycle time into a sequence of green signals for each group of approaching links. In addition, the cycle time is set as a fixed value. They consider minimum green time for each phase and the cycle time is divided to a fixed minimum green time and extension time that can be assigned to each phases. The action space was defined by $<n_{ph}, t_{min}, n_{ex}, h_{ex}>$, and n_{ph}: number of phases, t_{min}: minimum green time for phases (seconds), n_{ex}: number of extensions, h_{ex}: length of each extension (seconds). The efficient cycle length δ represented as equation 2.

$$\delta = n_{ph} \times t_{min} + n_{ex} \times h_{ex} \quad (2)$$

Another important factor for Q-learning is reward. In the related research, reward is defined as inversely proportion of average length of the queues in the approaching links.

Here, we improve on Abdoos's method for each intersection by considering more details for the Q-learning algorithm's parameters.

IV. Proposed Method

In our method, we change parameters used for decision-making when compared with Abdoos's work. In [9] there is no separate state for approaching links with equal lengths of queue. For example, in their list of 24 states there is no

978-1-4799-2914-613/$31.00 ©2013 IEEE

TABLE I . EXAMPLE OF STATE SPACE IN [9].

State number	Link order	State number	Link order
State 1	$l_1 \geq l_2 \geq l_3 \geq l_4$	State 13	$l_2 \geq l_3 \geq l_1 \geq l_4$
State 2	$l_1 \geq l_2 \geq l_4 \geq l_3$	State 14	$l_2 \geq l_4 \geq l_1 \geq l_3$
State 3	$l_1 \geq l_3 \geq l_2 \geq l_4$	State 15	$l_3 \geq l_2 \geq l_1 \geq l_4$
State 4	$l_1 \geq l_4 \geq l_2 \geq l_3$	State 16	$l_4 \geq l_2 \geq l_1 \geq l_3$
State 5	$l_1 \geq l_3 \geq l_4 \geq l_2$	State 17	$l_3 \geq l_4 \geq l_1 \geq l_2$
State 6	$l_1 \geq l_4 \geq l_3 \geq l_2$	State 18	$l_4 \geq l_3 \geq l_1 \geq l_2$
State 7	$l_2 \geq l_1 \geq l_3 \geq l_4$	State 19	$l_2 \geq l_3 \geq l_4 \geq l_1$
State 8	$l_2 \geq l_1 \geq l_4 \geq l_3$	State 20	$l_2 \geq l_4 \geq l_3 \geq l_1$
State 9	$l_3 \geq l_1 \geq l_2 \geq l_4$	State 21	$l_3 \geq l_2 \geq l_4 \geq l_1$
State 10	$l_4 \geq l_1 \geq l_2 \geq l_3$	State 22	$l_4 \geq l_2 \geq l_3 \geq l_1$
State 11	$l_3 \geq l_1 \geq l_4 \geq l_2$	State 23	$l_3 \geq l_4 \geq l_2 \geq l_1$
State 12	$l_4 \geq l_1 \geq l_3 \geq l_2$	State 24	$l_4 \geq l_3 \geq l_2 \geq l_1$

difference between $l_1 > l_2 > l_3 > l_4$ and $l_1 = l_2 = l_3 = l_4$ or $l_1 = l_2 = l_3 > l_4$. However, these cases can cause different situations in the traffic network.

A second point in decision making is to consider the length of queues that form in different states. Let Q_i be the queue length in the i-th lane ($i = 1, \dots, 4$). Consider the case that we have these two situations of queues: $(Q_1: 4, Q_2: 5, Q_3: 2, Q_4: 3)$and $(Q_1: 40, Q_2: 50, Q_3: 20, Q_4: 30)$. The same state is considered for both and similar green times will be chosen. However, it does not look reasonable to assign similar green times for these two cases, as the effect of queue length has been completely ignored. Here we categorize all lengths of queues in three ranges: low, medium and high. Different combinations of these values for all approaching links make different states. For example, an intersection with k approaching links we will have 3^k members in the state space. The state space of our method is larger than the one proposed by Abdoos et al, but it is expected to be more accurate.

Action sets are a combination of green times for each phase. The cycle time is flexible and based on the traffic demand, rather than fixed. For an intersection with four phases, some possible actions are: {13, 13, 13, 13}, {23, 23, 33, 33} or {33, 13, 33, 33} in which each member of the sets are the green time for related defined phase. The numbers are chosen in a way to make the action sets similar to Abdoos's proposed numbers in order to have a more reasonable comparison.

In [9] the reward was inversely proportional to the average length of the queues in the approaching links, normalized to remain between 0 and 1. However, normalizing to remain between 0 and 1 causes each action to receive a positive reward. This can cause delays in learning. We consider the distribution of cars in each cycle. The distribution varies on an hourly and daily basis. For this reason, traffic queue length also fluctuates over time. We aimed to provide a more reasonable reward by considering length of traffic queues based on the number of cars that arrive at the intersection. Both negative and positive rewards are received during learning. The reward is the average difference between the number of cars that entered the intersection in the current cycle and the length of queues in related approaching links at the end of the cycle time. Positive reward is obtained when the number of cars entered to the intersection is more than the final queue length, which means most or all cars entered the intersection. On the other hand, reward is negative when the number of cars that entered the intersection is less than the final queues length, meaning the proposed green time has not been enough for the cars to cross the intersection.

V. EXPERIMENTS AND DISCUSSION

The platform of all our experiments is an intersection with four approaching links and four phases as shown in Fig.3. The simulation is performed using Paramics v6. and the Q-learning algorithms are implemented in Matlab R2011b.

Our experiments are based on an intersection with four approach links. As mentioned in previously, we put all queue lengths in approaching links in one of these three groups: low, medium, or high, hence there are 81 states.

19 fixed actions were considered in Abdoos et al.'s study. It was assumed that the cycle time is fixed and the minimum green time is set to 13 seconds. The possible green times were {13, 23, 33}. All combination of these three numbers are not considered in action list due to limitation of the fixed cycle time (100 seconds) and also limitation on the number of extensions for each phase (2 times and each time 10 seconds). In our experiments, we also consider these three fixed numbers as possible green times, but we extend 19 actions to 81 possible actions by considering all combination of these three numbers and having flexible cycle times. The flexibility of cycle time is useful to set the traffic signal lights based on the traffic conditions. Length of a cycle is variable and is obtained from the sum of green times of all phases plus eight. The value of eight comes from the two second safety time required after each phase to clear the intersection. For instance, the cycle time for the set {13, 13, 23, 13} is 70 and for the set {33, 33, 33, 13} is 120.

The rewards are calculated based on the explanation in the previous section. For both learning methods, our method and Abdoos's method, ε-greedy is applied to select the Q-learning action. To implement [9], we apply the settings reported in previous study. ε is set to 0.1 in our experiments, which means the best action is chosen 90% of times and an action is selected randomly just in 10% of cases. $\gamma = 0.9$ and $\alpha = 0.1$ are the other parameters used during learning for both methods.

Table II shows the demands used for the first experiments in testing the three methods: fixed-time method, Abdoos method and our proposed method Single Agent Q-learning (SAQL). The results are calculated after completing learning for Abdoos and SAQL methods. The same settings in PARAMICS are applied for all three methods. During the first experiment, 4,900 cars are injected into the intersection over 10 hours. The green time for all phases in fixed-time method is set to 23 seconds, which is the average of three proposed green time numbers by Abdoos (13, 23, 33).

Fig.4 represents the average delay time for each method during these 10 hours. The cumulative average delays in each cycle of ten hour simulations for fixed time, Abdoos's, and SAQL methods is 49,129 seconds, 49,806 seconds, and 36,275 respectively. These figures indicate that application

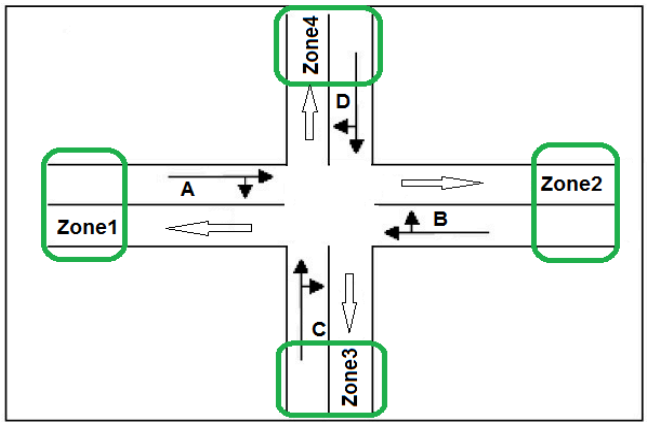

Figure 3. An intersection with four possible phases: A, B, C, D. The cycle time is divided between these four phases. Zones are the areas from which cars entered to the intersection.

of the proposed method improves the traffic control process by 26.2% and 27.2% compared to fixed time and Abdoos's methods respectively.

In the second experiment, we configure an intersection with 2,440 cars in 10 hours. Table III shows the demands in this experiment. As before, the performance of the three methods is quantitatively measured and examined. The same settings are considered for the fixed-time method. Simulations are run for 10 hours. Fig.5 represents a comparison of the average delay time for each method during these 10 hours. Cumulative average delay in this simulation for the fixed time, Abdoos's, and proposed methods are 29,588 seconds, 31,952 seconds, 22,748 seconds. Similar to the previous experiment, again SAQL outperforms the fixed-time and Abdoos's methods by 23.1% and 28.8% respectively. Table IV reviews the results of both experiments.

Although Abdoos's method is useful for many cases presented in [9], we improve its effectiveness for situations in which it may not be suitable. Aforementioned experiments are performed for such cases. In [9], better results for Abdoos et al.'s method in comparison to the fixed method were presented; however our experiments show that better results are related to the traffic demand. During our experiments we considered situations revealing that there are cases in which we can obtain better results through applying fixed-time instead of Abdoos's method. Considering fixed cycle time causes limitation for allocating the most appropriate green time for related traffic demand by reducing the number of possible action from action list. In this work we reduce this limitation by avoiding the limitation of fixed cycle time and extending the action sets in Q-learning.

TABLE II. DEMAND OF CARS FOR EACH ZONE IN FIRST EXPERIMENTS.

	Zone 1	Zone 2	Zone 3	Zone 4	Total
Zone 1		1,000	900	0	1,900
Zone 2	400		0	100	500
Zone 3	0	1,000		300	1,300
Zone 4	500	0	700		1,200
Total	900	2,000	1,600	400	4,990

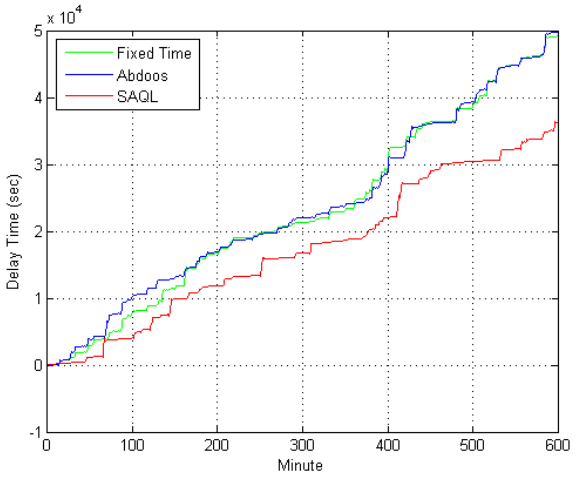

Figure 4. Comparison of the results of fixed-time, Abdoos and SAQL. Cumulative average delay time in seconds during 10 hours (600 minutes) for each method in first experiments is presented.

TABLE III. DEMAND OF CARS FOR EACH ZONE IN SECOND EXPERIMENTS.

	Zone 1	Zone 2	Zone 3	Zone 4	Total
Zone 1		400	200	0	600
Zone 2	400		0	180	580
Zone 3	0	300		320	620
Zone 4	340	0	300		640
Total	740	700	500	500	2,440

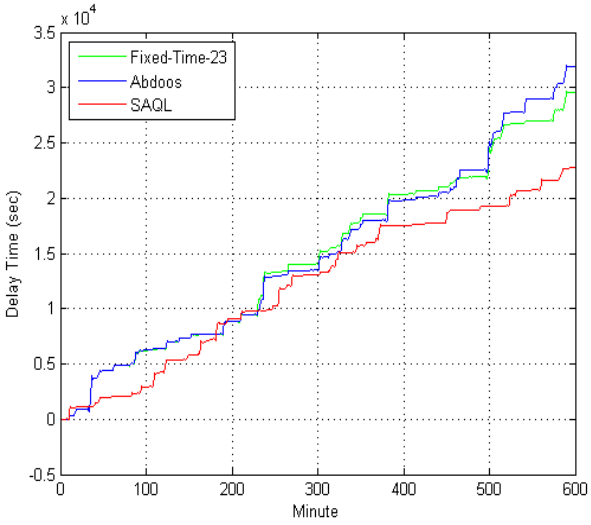

Figure 5. Comparison of the results of fixed-time, Abdoos and SAQL. Cumulative average delay time in seconds during 10 hours (600 minutes) for each method in second experiments is presented.

978-1-4799-2914-613/$31.00 ©2013 IEEE

TABLE IV. REVIEW OF RESULTS FOR TWO DEMAND PROFILES.

Number of Cars	Average Delay Time in Seconds during 10 hours simulation run (600 minutes)		
	Fixed-method	*Abdoos*	*SAQL*
4,900	49,129	49,806	36,275
2,440	29,558	31,952	22,748

VI. CONCLUSION

In this work, we improve on the method proposed by Abdoos et al. [9] for controlling traffic lights. Their method is applied to one intersection and we add additional details in implementing our method. Both methods use Q-learning, which is a promising approach to control traffic lights and supports flexible rather than fixed cycle times. Our improvement is obtained by considering various state-space and reward functions.

The simulation results illustrate that for some situations, which Abdoos's method does not generate appropriate results, SAQL proposes better options for controlling intersection's green time. In future work we will apply this method for a multi-intersection network.

[1] I. Alvarez, A. Poznyak, and A. Malo, "Urban traffic control problem a game theory approach," in *47th IEEE Conference on Decision and Control,*, Cancun, Mexico, 2008, pp. 2168–2172.

[2] P. G. Balaji and D. Srinivasan, "Multi-Agent System in Urban Traffic Signal Control," *IEEE Comput. Intell. Mag.*, vol. 5, no. 4, pp. 43–51, 2010.

[3] Balaji, "Distributed multi-agent based traffic management system," Ph.D dissertation, University of Madras, Singapore, 2011.

[4] P. Chanloha, W. Usaha, J. Chinrungrueng, and C. Aswakul, "Performance Comparison between Queueing Theoretical Optimality and Q-Learning Approach for Intersection Traffic Signal Control," in *4th International Conference on Computational Intelligence, Modelling and Simulation (CIMSiM)*, 2012, pp. 172–177.

[5] B. Abdulhai, R. Pringle, and G. Karakoulas, "Reinforcement Learning for True Adaptive Traffic Signal Control," *J. Transp. Engn.*, vol. 129, no. 3, pp. 278–285, 2003.

[6] I. Arel, C. Liu, T. Urbanik, and A. G. Kohls, "Reinforcement learning-based multi-agent system for network traffic signal control," *IET Intelligent Transportation System*, vol. 4, no. 2, pp. 128–135, 2010.

[7] E. Bingham, "Reinforcement learning in neurofuzzy traffic signal control," *European Journal of Operational Research*, vol. 131, no. 2, pp. 232–241, Jun. 2001.

[8] L. A. Prashanth and S. Bhatnagar, "Reinforcement Learning With Function Approximation for Traffic Signal Control," *IEEE Trans. on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 412–421, 2011.

[9] M. Abdoos, N. Mozayani, and A. L. C. Bazzan, "Traffic light control in non-stationary environments based on multi agent Q-learning," in *14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Washington, DC, USA., 2011, pp. 1580–1585.

[10] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, no. 3–4, pp. 279–292, May 1992.

[11] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

[12] M. Wiering, J. Vreeken, J. van Veenen, and A. Koopman, "Simulation and optimization of traffic in a city," in *IEEE Intelligent Vehicles Symposium*, 2004, pp. 453–458.

[13] B. C. da Silva, E. W. Basso, F. S. Perotto, A. L. C. Bazzan, and P. M. Engel, "Improving reinforcement learning with context detection," in *Proceedings of the 5th international joint conference on Autonomous agents and multiagent systems*, New York, NY, USA, 2006, pp. 810–812.

[14] K. Wen, S. Qu, and Y. Zhang, "A stochastic adaptive control model for isolated intersections," in *IEEE International Conference on Robotics and Biomimetics*, 2007, pp. 2256–2260.

[15] Y. Dai, D. Zhao, and J. Yi, "A comparative study of urban traffic signal control with reinforcement learning and Adaptive Dynamic Programming," in *The International Joint Conference on Neural Networks (IJCNN)*, 2010, pp. 1–7.

[16] C. Jacob and B. Abdulhai, "Integrated traffic corridor control using machine learning," in *IEEE International Conference on Systems, Man and Cybernetics*, 2005, vol. 4, pp. 3460–3465.

[17] S. Lu, X. Liu, and S. Dai, "Incremental multistep Q-learning for adaptive traffic signal control based on delay minimization strategy," in *7th World Congress on Intelligent Control and Automation*, 2008, pp. 2854–2858.