2011 14th International IEEE Conference on
Intelligent Transportation Systems
Washington, DC, USA. October 5-7, 2011

Traffic Light Control in Non-stationary Environments based on Multi Agent Q-learning

Monireh Abdoos , Nasser Mozayani and Ana L. C. Bazzan

Abstract— **In many urban areas where traffic congestion does not have the peak pattern, conventional traffic signal timing methods does not result in an efficient control. One alternative is to let traffic signal controllers learn how to adjust the lights based on the traffic situation. However this creates a classical non-stationary environment since each controller is adapting to the changes caused by other controllers. In multi-agent learning this is likely to be inefficient and computationally challenging, i.e., the efficiency decreases with the increase in the number of agents (controllers). In this paper, we model a relatively large traffic network as a multi-agent system and use techniques from multi-agent reinforcement learning. In particular, Q-learning is employed, where the average queue length in approaching links is used to estimate states. A parametric representation of the action space has made the method extendable to different types of intersection. The simulation results demonstrate that the proposed Q-learning outperformed the fixed time method under different traffic demands.**

I. INTRODUCTION

Traffic control is a challenging issue when it comes to application of computational techniques in the real world. The development of efficient mechanisms for traffic light control is necessary, as the number of vehicles in urban network increases rapidly. The objective of the signal control is to increase intersection capacity, to decrease delays, and at the same time, to guarantee the safety of traffic actors. Moreover, it can reduce fuel usage and reduce emissions.

Signal control is one of the areas involved in the overall effort known as intelligent transportation systems (ITS). ITS can be implemented by some techniques. In the present paper we use multi-agent systems and machine learning to develop a traffic light control mechanism.

For transportation systems, concepts of intelligent agents may be used in different parts of the system such as traffic lights [1], vehicles [2], and pedestrians [3], as well as to model the behavior of traffic system to describe the norm violation and critical situation detection [4].

In order to manage the increasing volume of traffic, traffic lights control by artificial intelligence (AI) techniques are becoming more and more important. Some methods handle control of traffic signals by predefined rule-based system [5], fuzzy rules [6], and centralized techniques [7].

Multi-agent Systems is a subfield of AI that aims to provide principles for construction of complex systems involving multiple agents. Multi-agent systems have gained significant importance because of their ability to model and solve complex real-world problems.

Multi-agent systems provide mechanisms for communication, cooperation, and coordination of the agents. In [8] the individual traffic control is modeled by coordinated intelligent agents. The agents coordinate among themselves based on the information received from each other to control the network. Synchronization of traffic signals has also been studied as coordinated agents in [9]. Cooperative multi-agent system has been proposed to control the signals according to the prediction of traffic volume in neighboring intersections [10].

Regarding use of machine learning methods, in [11] collaborative reinforcement learning has been presented to provide an adaptive traffic control based on traffic pattern observed from vehicle location data. For an overview on applications and challenges regarding multi-agent learning in traffic signal control, the reader is refered to [12].

Traditional reinforcement learning research assumes that the environment is stationary and its dynamics are always fixed. This is not the case regarding a real traffic network as a stationary environment since traffic flow patterns change dynamically over the time.

In the present paper, we use a reinforcement learning approach which is model-free, namely Q-learning (more on this in Section II). Contrary to other works that use this method, here Q-learning is used to control the traffic lights in a large and non-regular (i.e. non grid) network. In summary, a network with 50 junctions arranged in a non grid network is considered.

As it will be seen in the next section, Q-learning does not require a pre-specified model of the environment. Hence, it can be used in dynamic and non-stationary environment. Of course, in this case the mathematical guarantees of convergence no longer hold. However, because the environment is non-stationary, this is not even desirable as one wants the method to re-adapt to the environment when this changes.

Here, each agent is responsible to control the traffic lights in one junction using local information only. Another important characteristic of the present work is that it handles a large state and action space, especially because contrarily to other works we do not make simplifications (e.g. only considering two phases). Here a 4-phase signal timing is considered for each intersection.

The rest of this paper is organized as follows. Reinforcement learning and in particular Q-learning is discussed in Section II. Section III presents some related works that have used reinforcement learning to control traffic lights. The proposed Q-learning method and the network configuration are presented in Section IV. Network configuration and experimental results are respectively given in Section V and VI. Finally the paper is concluded in Section VII.

978-1-4577-2197-7/11/$26.00 ©2011 IEEE

II. Reinforcement learning and Q-learning

Unlike supervised and unsupervised learning, reinforcement learning learns an optimal policy by perceiving states of the environment and receiving information from the environment. Reinforcement learning algorithms optimize environmental feedback by mapping percepts to actions. These are best suited for domains in which an agent has limited or no previous knowledge. The Q-Learning is one particular reinforcement learning algorithm that is model-free. Put simply, this means that an agent has no previous model of how states and actions are related to rewards. The Q-Learning algorithm was proposed by Watkins [13]. As mentioned, it is essentially independent of existing domain knowledge. This property of Q-Learning makes it a favorable choice for unexplored environments.

Classically, Q-Learning is modeled by using a Markov decision process formalism. This means that the agent is in a state s, performs an action a, from which it gets a scalar reward r from the environment. The environment then changes to state s' according to a given probability transition matrix T. The goal of the agent is to choose actions maximizing discounted cumulative rewards over time. Discounted means that short term rewards are weighted more heavily than distant future rewards. The discount factor, $\gamma = [0..1]$, awards greater weight to future reinforcements if set closer to 1. In Q-learning, $Q(s,a)$ is a state-action value representing the expected total discounted return resulting from taking action a in state s and continuing with the optimal policy thereafter.

$$Q(s,a) \leftarrow \beta(r + \gamma V(s'))Q(s,a) \tag{1}$$

$$V(s') \leftarrow \max_a Q(s',a) \tag{2}$$

In Equation 1, β is the learning rate parameter and $V(s')$ is given by Equation 2. The discount factor determines the importance of future rewards. $Q(s,a)$ measures the quality value of the state-action pair and $V(s')$ gives the best Q value for the actions in next state, s'. Q-learning helps to decide upon the next action based on the expected reward of the action taken for a particular state. Successive Q values are built increasing the confidence of the agent as more rewards are acquired for an action. As an agent explores the state space, its estimate of Q improves gradually and Watkins and Dayan have shown that the Q-Learning algorithm converges to an optimal decision policy for a finite Markov decision process [14].

III. Reinforcement learning for traffic light control

Reinforcement learning offers some potentially significant advantages and has become a good solution to control single junction as well as network of traffic signals, in non-stationary and dynamic environments.

Some researchers have tried to use a model-based reinforcement learning method for traffic light control. In [15], transition model has been proposed that estimates waiting times of cars in different states. In [16] Silva *et al* have presented a reinforcement learning with context detection that creates a partial model of the environment on demand. Later, the partial models are improved or new ones are constructed. Adaptive reinforcement learning has been presented to control a model free traffic environment in [17], [18]. Adaptive control of traffic light has also been studied in [19], which uses a function approximation method as a mapping between the states and signal timing.

Some researchers have introduced a reinforcement learning method based on communication between the agents. A collaborative reinforcement learning introduced in [11] has tried to exploit local knowledge of neighboring agents in order to learn signal timings.

In [20] an interaction model based on game theory has been studied. They defined two types of agents for traffic signal control, intersection agents and management agents, and constructed models of two agents. Q-learning was used to build the payoff values in the interaction model. In the interaction model, the renewed Q-values in the distributed reinforcement Q-learning was used to build the payoff values. Therefore, interaction has taken on from the action selection between two agents.

Balaji *et al* have presented a method in which agents try to compute a new phase length based on both local and communicated information [21]. Q values were shared between agents to improve the local observations and create a global view.

In [22], two types of agents have been used to control a five intersection network. Four intersections connected to a central intersection were labeled as outbound intersections. Two types of agents, a central agent and an outbound agent, were employed. An outbound agent that follows the longest queue collaborates with a central agent by providing relative traffic flow values as a local traffic statistic. The relative traffic flow is defined as the total delay of vehicles in a lane divided by the average delay at all lanes in the intersection. The central agent learns a value function driven by its local and neighbors traffic conditions. It incorporates relative traffic flow of its neighbors as a part of its decision-making process.

Some other methods have been using communicated information in state and reward estimation [23], [24], optimization through cooperation based on information fusion technology in a multi-level hierarchical structure [25].

Several researches have tried to realize distributed control that learn optimal signal control over a group of signal by a hierarchical multi-agent system. In [26] a two level hierarchical multi-agent system has been proposed. Local traffic agents are concerned with the optimal performance of their assigned intersection. A second layer has been added as a coordinator that supervises the local agents. Bazzan *et al* [27] have presented a hierarchical reinforcement learning for networks. A number of groups of three traffic light agents each is first composed. These groups are coordinated by a supervisor agents that recommend a joint action.

TABLE I

EXAMPLE OF STATE SPACE

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	$l_4 \geq l_3 \geq l_2 \geq l_1$

It is not common to find a reinforcement learning based method that uses a large network for simulation without any simplification. Most of the works mentioned here have used a grid based network with a small number of states and actions. As mentioned, in the present paper, a Q-learning based method is used, which considers not only a large number of states and actions, but also a non-grid based, non-regular network.

IV. PROPOSED METHOD

In the Q-learning based method used here, the traffic network is considered as a system composed of intelligent agents, where each controls an intersection.

As state estimation, agents use the average queue length in approaching links in a fixed cycle. They then select an action and receive a reward. The state space is discretized by ranking the approaches according to the statistic gathered in the last cycle.

The number of states is equal to the number of permutation of the approaches, i.e., $k!$ for an agent with k approaching links. To see why this is is so, consider the following example. There are 24 states for each junction, since each agent has four approaching links. The state space can be seen in Table I. If l_i represents the i^{th} approaching link for an intersection, $l_1 \geq l_2 \geq l_4 \geq l_3$ shows a state in which the queue length of approaching link l_1 is the longest and l_3 is the shortest. If two approaching links have the same queue length, then they are ranked according to their order in the signal plan.

The actions relate to split of green time, i.e., the action space contains different phase splits of the cycle time. Phase split refers to the division of the cycle time into a sequence of green signals for each group of approaching links. We assumed that the cycle time, δ, is fixed and all junctions use the same value.

It should be noted that for a fixed time controller, all available phases should at least appear once in a cycle. Each phase should have a minimum green time so that a stopped vehicle that receives a green signal has enough time to cross the intersection. The cycle length is divided to a fixed minimum green time, and extension time that can be assigned to different phases.

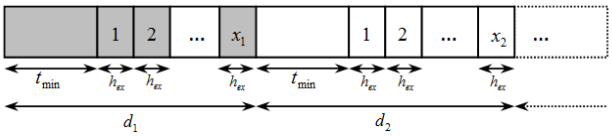

Fig. 1. Phase split parameters used for action definition

Assume that there is n_{ph} phases and minimum green time assigned for each of them is the same and equal to t_{min}. Moreover, there are n_{ex} extensions with fixed length of time, h_{ex}. The actions determine the assignment of n_{ex} extensions to different phases.

For example, assuming a cycle length of 50 seconds, a signal plan containing 3 phases and 10 second as the minimum green time, 3×10 seconds is assigned to the phases and different actions determine the assignment of the remaining 20 seconds to three phases.

The action space is defined by $< n_{ph}, t_{min}, n_{ex}, h_{ex} >$,where: n_{ph}: number of phases t_{min}: minimum green time for phases (seconds) n_{ex}: number of time extensions h_{ex}: length of each extension (seconds) such that the effective cycle length δ is given as in Equation 3.

$$\delta = h_{ex} \times n_{ex} + n_{ph} \times t_{min} \tag{3}$$

The possible green time assignment can be formulated as follows:

$$\sum_{i=1}^{n_{ph}} x_i = n_{ex}, \quad \mathrm{x}_i \in 0, 1, ..., \alpha, \quad 1 \leq \alpha \leq n_{ph} \tag{4}$$

The maximum number of extension is controlled by α. The action space can be reduced by choosing the small value for α. The solutions determine the portion of the cycle time that is assigned to each phase by:

$$d_i = t_{min} + x_i \times h_{ex} \tag{5}$$

In equation 5, d_i is the green time assigned to phase i.

Figure 1 shows the phase split parameter used for the definition of the actions. For example, d_1, the green time assigned to the first phase, includes t_{min} as minimum green time and a number of extension displayed by x_1.

The reward is inversely proportional to the average length of the queues in the approaching links, normalized to remain between 0 and 1.

V. NETWORK CONFIGURATION

The traffic network used in the experiments is shown in Figure 2. It contains 50 junctions and more than 100 links. A four-way intersection is the most common intersection in real world, and therefore it has been used in our approach.

The number of lanes is three for all approaches. During the simulation, new vehicles are generated by uniform distribution over 30 input sections. Time intervals between two consecutive vehicle arrivals at input sections are sampled from a uniform distribution. The network is surrounded by

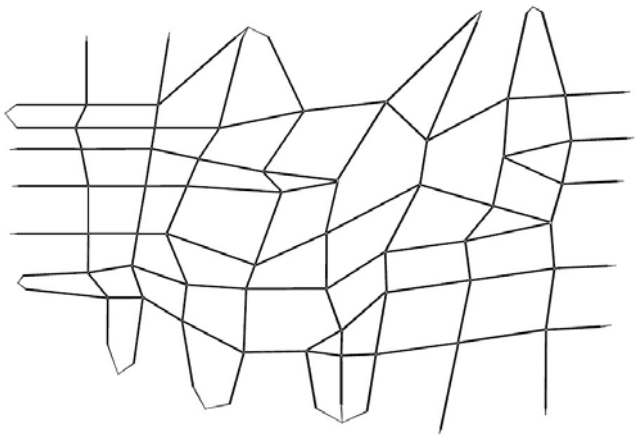

Fig. 2. The network used for the simulation

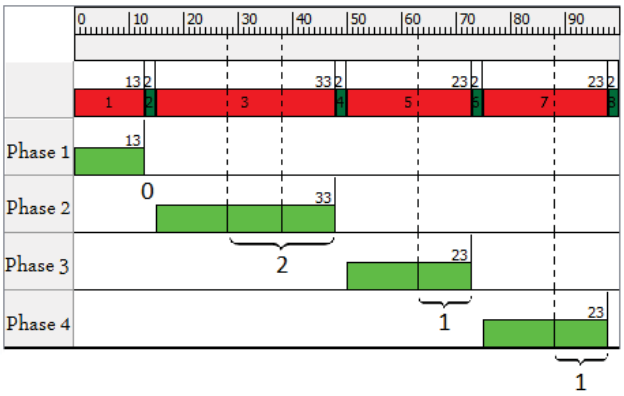

Fig. 3. An example for action corresponds to $[x_1, x_2, x_3, x_4] = [0, 2, 1, 1]$. Two seconds is assigned for all-red interval between two signals

20 centroids. Centroids are points that vehicles enter or exit the network through them.

In this configuration, the number of phases is four and the minimum green time for each phase is set to 13 seconds. Moreover, four 10-second extension intervals are used for signal timing, i.e. $n_{ex} = 4$ and $h_{ex} = 10$. The effective cycle time is computed according to Equation 3 with $\delta = 92 seconds$.

The purpose of all-red interval is to provide a safe transition between two conflicting traffic signal phases. Two seconds is assigned for all-red interval at the end of each signal phase as it has been shown in Figure 3. Since there are four phases, 8 seconds are assigned for all-red time interval. In this case, the total cycle time will be $100 seconds$ for a complete signal plan.

The parameters used in the Q-learning based method are shown in Table II. There are 19 possible solutions for Equation 4 if we assume $\alpha = 2$. For example, $[x_1, x_2, x_3, x_4] = [0, 2, 1, 1]$ is a solution for Equation 4. The corresponding action is (13,33,23,23) as shown in Figure 3.

The action set is as follows:

TABLE II

THE PARAMETERS USED IN THIS PAPER

Parameter	Description	Value
δ	effective cycle length	92
n_{ph}	number of phases	4
t_{min}	minimum green time for each phase	13
n_{ex}	number of extension interval	4
h_{ex}	length of extended time	10
α	maximum number of extension for each phase	2
$\|S\|$	number of states	24
$\|A\|$	number of actions	19

TABLE III

NETWORK CONFIGURATION PARAMETERS

Properties	Value
number of intersections	50
number of links	224
average length of links	847.44m
number of lanes per links	3
maximum speed	50km/h
number of input/output centroid	20
arrival distribution	Uniform
simulation duration	10hour
traffic demand 1	18000 veh/hour
traffic demand 2	27000 veh/hour

$$A = \{a_1(33,33,13,13), a_2(33,13,23,23), ..., a_{19}(23,23,23,23)\} \quad (6)$$

Moreover, we use ε-greedy as a mechanism to select an action in Q-learning with ε fixed at value of 0.9. This means that the best action is selected for a proportion $1-\varepsilon$ of the trials, and a random action is selected (with uniform probability) for a proportion ε.

VI. EXPERIMENTAL RESULTS

The proposed method has been tested on a large network that contains 50 intersections and 112 two-way links using the Aimsun traffic simulator [1]. The system configuration parameters are shown in Table III.

We performed some experiments using Aimsun in order to compare the proposed method with fixed time method that assigns equal green time to each signal group. The experiment aimed to determine the simulated average delay time of the network for different traffic demands. Among different traffic demands we choose two values as medium and high traffic congestion (traffic demands 1 and 2 in Table III).

In these experiments, the average delay time is used for performance evaluation. The results show that the proposed approach has reduced the average delay time in comparison with the fixed time method. Figure 4 shows the result over a traffic demand with 18000 vehicles per hour (demand 1).

The performance of the methods over a traffic demand with 27000 vehicles per hour (demand 2) can be seen in Figure 5. The average value of the delay is shown in Table IV.

[1] http://www.aimsun.com

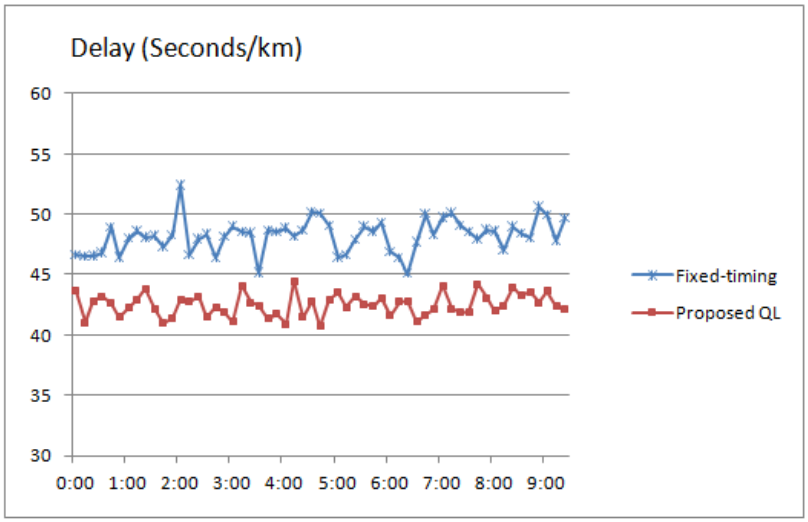

Fig. 4. Comparison between fixed time method and the proposed Q-learning based approach for traffic demand 1

TABLE IV
AVERAGE DELAY OF THE PROPOSED QL AND FIXED TIME METHOD

	Fixed time	Proposed QL
Traffic demand 1	47.977	42.362
Traffic demand 2	113.034	63.687

For the fixed timing, performance begins to drop rapidly when nearly all of the vehicles in some junctions stop. This happens after six hours after the begin of the simulation, for traffic demand 2, as seen in Figure 5. At the end of the simulation, all vehicles stop in the network and there is no vehicle leaving network. Under these conditions, the average delay over all simulated time makes no sense. Therefore, the value reported in Table IV for traffic demand 2 has been computed over the period of the first 6 hours.

Regarding the proposed method, it is possible to see that it could manage to efficiently control the traffic lights in the network under relatively high traffic demand in a better way than the fixed timing method.

VII. CONCLUSIONS

In this work we have shown that Q-learning is a promising approach to control traffic light in a non-stationary environment. A real traffic network is a non-stationary environment because traffic flow patterns are dynamically changed over the time. The proposed method is based on local statistical information gathered in one learning step and tries to learn the best action in different situations. Average queue length in approaching links is used as the statistical information. The proposed method contains a relatively large number of states and actions that can be used in a large network.

The simulation results demonstrated that the proposed Q-learning method outperformed the fixed time method under different traffic demands.

Parametric representation of the action space enables us to define different situations by means of using different values of the parameters. Since the action space is defined by a number of parameters, it can be extended to a larger space or smaller one by changing the parameters. As a result, it can be applied in real traffic networks with a non predictable traffic demand. Moreover, it can be used in different types of intersections with a different number of approaching links.

Future work includes applying the method to other networks, extension of the action space with different parameters to find the proper parameters for a given traffic demand, and examination of impact of different parameters values on learning performance.

VIII. ACKNOWLEDGEMENTS

A. Bazzan and M. Abdoos are partially supported by CNPq. The present research was supported by Research Institute for Information and Communication Technology - ITRC (Tehran, Iran). The first author gratefully acknowledge the support of this institute.

REFERENCES

[1] Z. Liu, "A survey of intelligence methods in urban traffic signal control," *International Journal of Computer Science and Network Security*, vol. 7, no. 7, pp. 105–112, 2007.

[2] J. Adler, G. Setapathy, V. Manikonda, B. Bowles, and B. V.J., "A multi-agent approach to cooperative traffic management and route guidance," *Transportation Research Part B*, vol. 39, pp. 297–318, 2005.

[3] K. Teknomo, "Application of microscopic pedestrian simulation model," *Transportation Research Part F*, vol. 9, pp. 15–27, 2006.

[4] A. Doniec, R. Mandiau, S. Piechowiak, and S. Espié, "A behavioral multi-agent model for road traffic simulation," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 8, pp. 1443–1454, 2008.

[5] V. Hirankitti, J. Krohkaew, and C. Hogger, "A multi-agent approach for intelligent traffic-light control," in *Proc. of the World Congress on Engineering*, vol. 1. London, U.K.: Morgan Kaufmann, 2007.

[6] I. Kosonen, "Multi-agent fuzzy signal control based on real-time simulation," vol. 11, no. 5, pp. 389–403, 2003.

[7] P. Balaji, P. Sachdeva, D. Srinivasan, and T. C.K., "Multi-agent system based urban traffic management." Singapore: IEEE, 2007, pp. 1740–1747.

[8] R. T. v. Katwijk, B. De Schutter, and J. Hellendoorn, "Multi-agent coordinationof traffic control instruments," in *Proc. of the International Conference on Infrastructure Systems 2008: Building Networks for a Brighter Future*, vol. 1, Rotterdam, The Netherlands, November 2008.

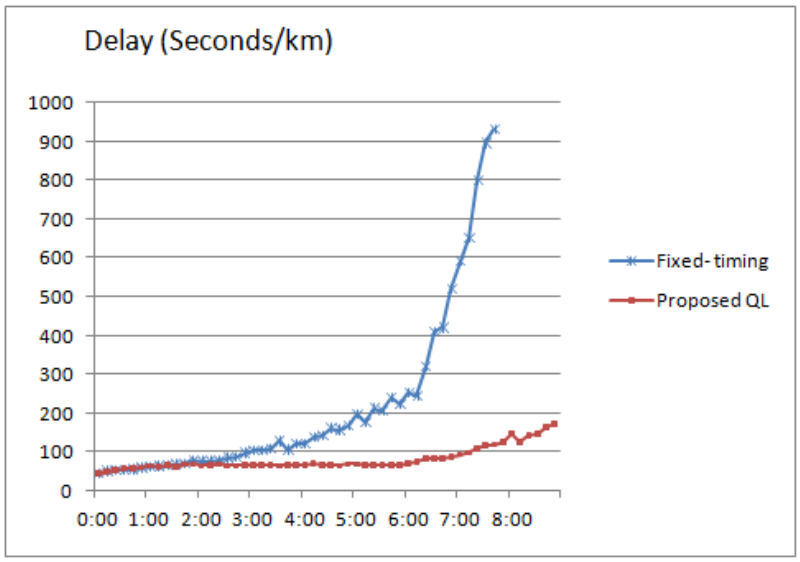

Fig. 5. Comparison between fixed time method and the proposed Q-learning based approach for traffic demand 2

[9] A. L. C. Bazzan, "A distributed approach for coordination of traffic signal agents," *Autonomous Agents and Multiagent Systems*, vol. 10, no. 1, pp. 131–164, March 2005. [Online]. Available: www.inf.ufrgs.br/~bazzan/downloads/jaamas10_2_131_164.pdf.gz

[10] F. Daneshfar, F. Akhlaghian, and F. Mansoori, "Adaptive and cooperative multi-agent fuzzy system architecture," in *Proc. 14th International CSI Computer Conference*. IEEE, 2009, pp. 30–34.

[11] A. Salkham, R. Cunningham, A. Garg, and V. Cahill, "A collaborative reinforcement learning approach to urban traffic control optimization," in *Proc. of International Conference on Web Intelligence and Intelligent Agent Technology*. IEEE, 2008, pp. 560–566.

[12] A. L. C. Bazzan, "Opportunities for multiagent systems and multiagent reinforcement learning in traffic control," *Autonomous Agents and Multiagent Systems*, vol. 18, no. 3, pp. 342–375, June 2009. [Online]. Available: http://www.springerlink.com/content/j1j0817117r8j18r/

[13] C. Watkins, "Learning from delayed rewards," Ph.D. dissertation, University of Cambridge, 1989.

[14] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, 1992.

[15] M. Weiring, "Multi-agent reinforcement learning for traffic light control," in *Proc. of the Seventh International Conference on Machine Learning*, 2000, pp. 1151–1158.

[16] B. C. d. Silva, E. W. Basso, 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, AAMAS, 2006*. Hakodate, Japan: New York, ACM Press, May 2006, pp. 811–812. [Online]. Available: www.inf.ufrgs.br/maslab/pergamus/pubs/Silva+2006.pdf

[17] K. Wen, S. Qu, and Y. Zhang, "A stochastic adaptive control model for isolated intersections," in *Proc. 2007 IEEE International Conference on Robotics and Biomimetics*. Sanya, China: IEEE, 2008, pp. 2256–2260.

[18] Y. Dai, D. Zhao, and J. a. Yi, "A comparative study of urban traffic signal control with reinforcement learning and adaptive dynamic programming," in *Proc. of International Joint Conference on Neuarl Networks*, Barcelona, 2010, pp. 1–7.

[19] L. Prashanth and S. Bhatnagar, "Reinforcement learning with function approximation for traffic signal control," *IEEE Transaction on Intelligent Transportation Systems*, pp. 1–10, 2010.

[20] X. Xinhai and X. Lunhui, "Traffic signal control agent interaction model based on game theory and reinforcement learning," in *Proc. 2009 International Forum on Computer Science-Technology and Applications*. IEEE, 2009, pp. 164–168.

[21] P. Balaji, X. German, and D. Srinivasan, "Urban traffic signal control using reinforcement learning agents," *IET Intelligent Transportation Systems*, vol. 4, no. 3, pp. 177–188, 2010.

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

[23] J. Medina, A. Hajbabaie, and R. Benekohal, "Arterial traffic control using reinforcement learning agents and information from adjacent intersections in the state and reward structure," in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, sept. 2010, pp. 525–530.

[24] W. Yaping and Z. Zheng, "A method of reinforcement learning based automatic traffic signal control," in *Proc. Third International Conference on Measuring Technology and Mechatronics Automation*. IEEE, 2011, pp. 119–122.

[25] Z. Yang, X. Chen, Y. Tang, and J. Sun, "Intelligent cooperation control of urban traffic networks," in *Proc. Fourth International Conference on Machine Learning and Cybernetics*. Guangzhou, China: IEEE, 2005, pp. 1482–1486.

[26] J. France and A. A. Ghorbani, "A multiagent system for optimizing urban traffic," in *Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology*. Washington, DC, USA: IEEE Computer Society, 2003, pp. 411–414.

[27] A. L. C. Bazzan, D. d. Oliveira, and B. C. d. Silva, "Learning in groups of traffic signals," *Engineering Applications of Artificial Intelligence*, vol. 23, pp. 560–568, 2010.