Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 109C (2017) 513–520

Procedia Computer Science

www.elsevier.com/locate/procedia

The 8th International Conference on Ambient Systems, Networks and Technologies (ANT 2017)

Adaptive Traffic Signal Control : Exploring Reward Definition For Reinforcement Learning

Saad Touhbi[a,b,c,*], Mohamed Ait Babram[a,b], Tri Nguyen-Huu[b], Nicolas Marilleau[b], Moulay L. Hbid[a,b], Christophe Cambier[b,c], Serge Stinckwich[b,c,d]

[a] *LMDP/IRD Cadi Ayyad University, Marrakech, Morocco*
[b] *IRD, UMI 209, UMMISCO, IRD France Nord, F-93143, Bondy, France*
[c] *Sorbonne Universites, Univ. Paris 06, UMI 209, UMMISCO, F-75005,Paris,France*
[d] *Universite de Caen Basse-Normandie, Caen, France*

Abstract

As mobility grow in urban cities, traffic congestion become more frequent and troublesome. Traffic signal is one way to decrease traffic congestion in urban areas but needs to be adjusted in order to take into account the stochasticity of traffic. Reinforcement learning (RL) has been the object of investigation of many recent papers as a promising approach to control such a stochastic environment. The goal of this paper is to analyze the feasibility of RL, particularly the use of Q-learning algorithm for adaptive traffic signal control in different traffic dynamics. A RL control was developed for an isolated multi-phase intersection using a microscopic traffic simulator known as Paramics. The novelty of this work consists of its methodology which uses a new generalized state space with different known reward definitions. The results of this study demonstrate the advantage of using RL over fixed signal plan, and yet exhibit different outcomes depending on the reward definitions and different traffic dynamics being considered.

Peer-review under responsibility of the Conference Program Chairs.

Keywords: Reinforcement learning; traffic optimization; traffic light control; Q-learning; urban mobility

1. Introduction

Urban cities are usually covered by a complex traffic network responsible for supporting the daily demands in the area. Unfortunately, the traffic demands are high, dynamic and in constant increase. Infrastructure improvement has been the primary method to serve these demands. However, this is not always possible due to constraints such as financial resources and space. This has led to options considering the improvement of the existing infrastructure, optimizing the utilization of the available infrastructure and lowering the costs of travel time through intelligent transportation systems (ITS). Adaptive traffic signal control is most used since the early seventies [1] and has shown to be

* Corresponding author.
E-mail address: saad.touhbi@edu.uca.ma

Peer-review under responsibility of the Conference Program Chairs.
10.1016/j.procs.2017.05.327

a suited approach to alleviate traffic congestion as opposed to pre-timed and actuated control systems for signalized intersections. Many methods have been developed and investigated in the literature (SCOOT[2] and SCATS[3], PRODYN[4], OPAC[5], UTOPIA[6], RHODES[7]) but these methods require a pre-specified model of the environnement. But due to the stochastic nature of traffic, An approach that can adapt to the changes in traffic and does not need a specified model for a certain environnement would be more simple for the control process. Reinforcement learning[8] on the other hand has the ability to adapt and self-learn from past experiences. Therefore it has more potential to improve service over time through continuous interaction with the environnement. In this paper, we will investigate the use of reinforcement learning, in particular Q-learning on trafic signal control. The study will include previous work with a new representation of state space based on queuing levels. We will also look into the effect of different reward definitions on traffic patterns.

2. Related Work

Traffic signal systems are most used in intersections, especially in urban areas. Generally, traffic signals go through repeated *"cycles"*, each cycle corresponding to one complete rotation through all of the indications provided at the intersection. Each cycle consists of a sequence of N phases. A *"phase"* is a group of traffic flow movement through the intersection (e.g. North to South and South to North); a typical phase has a time interval of passage defined as *"green"* time, followed by a duration of *"yellow"* and then *"red"* (red can signify the passage to the next phase or a halt time for the intersection to be empty; in this case we call it *"All red"*). The duration is determined for each phase either as a fixed plan using the commonly used Webster method[9], or using adaptive allocation of green time for each phase and variable sequences of phases depending on traffic dynamics[2,3,4,5,6,7]. Application of RL to adaptive traffic light control has been proven to be efficient in many papers[10]. SARSA for traffic light control was introduced by Thorpe[11], Abdulhai et al.[12] introduced Q-learning for an isolated intersection. Bingham[13] introduced a neuro-fuzzy traffic signal controller that uses RL for learning the neural network. Olivera et al.[14], proposed an RL method with context detection to solve signal control optimization. These previous studies[10,11,12,13,14] are designed to solve fixed phasing sequence for signal control and have been used on hypothetical intersections to show their feasibility. However, El-Tantawy et al.[15] used several RL algorithms for a comprehensive analysis on the effect of state space, action space, action selection and reward definition on patterns of traffic on a real world intersection, and proved that variable phasing can be more efficient especially in high and dynamic traffic patterns. Brys et al.[16] used variable phasing and alternative reward definition to solve the traffic signal control on hypothetical road network. In this paper we extend the previous work done by El-tantawy et al.[15] by proposing at first a new state space definition that could provide a more standard representation of the state of the intersection. Additionally, this paper provides an analytical study of the effect of reward definition and traffic patterns on the performance of RL in an isolated intersection, which could lead to an in-depth study on traffic variability and its effect on the RL controlled traffic intersections.

3. Proposed System

For our study, we will use the intersection in Downtown Toronto (Front and Bay Street) used by El-Tantawy et al.[15] with two different traffic volumes. Our performance metrics are the average delay experienced by each vehicle, the rate of vehicles passing through the intersection, i.e. throughput and the average queue length.

3.1. Q-learning

Q-learning is a RL technique that is used to estimate an optimal action-selection policy for a given finite state space S and a set of actions A. Q-learning is an off policy (model free) algorithm since the agent attempts to improve a policy with no required knowledge on the system. Performing an action in a specific state provides a reward; the goal of the algorithm is to maximize the long-term reward. This algorithm learns by updating an action-function $Q : S \times A \mapsto \mathbb{R}$ that gives the expected utility of taking an action $a \in A$ on a given state $s \in S$ which eventually return after enough

training the highest value for a each state. Before learning, Q returns an arbitrary value. At iteration j, Q is updated as

$$Q(s_j, a_j) \leftarrow Q(s_j, a_j) + \alpha_j \left(R_{j+1} + \gamma_j \max_{a \in A} Q(s_{j+1}, a) - Q(s_j, a_j) \right) \tag{1}$$

Where R_j is the reward at iteration j when performing action a on a state s, α_j is the learning rate at iteration j defining the level of dependence between past knowledge and its new knowledge, and γ_j is the discount factor defining the level of importance of the next state. Both these latter parameters have a value between zero and one. If the algorithm has a higher learning rate (close to one), then the agent will depend more on the know knowledge acquired for his current action then the past knowledge. As for the discount factor, a value close to one would mean that the resulting state is important and has influence on the performance of the agent. For each iteration, the Q-value of the current action is updated according to the new feedback reward and the next action is chosen, this is stored in a table we call Q-table, which plays the role of the experience database of the agent in question. The learning rate adopted for this paper is

$$\alpha_j = \frac{1}{\ln(v_j(s, a))} \tag{2}$$

where $v_j(s, a)$ is the number of visits to a particular state-action pair. This means a higher learning rate at first, which then decreases and tends towards zero after sufficient exploration. This gradually decreases the effect of future experiences on the Q-value. For the discount factor γ_j, we choose a constant value of 0.5.

3.2. State and Action definition

3.2.1. State definition

The previous studies[15,12] used maximum queue length (or number of vehicles in a queue) on each phase as a variable to present the state space. This has shown to be more powerful than other approaches like using the cumulative delay of vehicles in a phase or the arrival rate. However, if we take into consideration vehicle dimension, this adds a correlation between states in the state space presented by maximum number of vehicles in each lane, e.g if we take two vehicles on a queue of a phase with length of five meters that would give us a queue length of 10 meters, and in another case, we could have a heavy lorry of ten meter, this makes the two cases in the same state knowing that they are totally different when it comes to queue discharge on the phase. Also if we take into consideration the length of a road, we could see that the state space presented by maximum queue length in each lane would not accurately monitor the load on a certain phase. This is why we are defining a new state space on which we use the maximum residual queue (queue length on a lane divided by the lane length). This would help to define states that take into consideration the vehicle's dimension and the lane length.

Let us denote q_k^t the queue length (in meters) on lane k at time t, and l_k the length of lane k (in meters). We define the queuing level lq_k^t on lane k at time t as the queue length on lane k at time t divided by the length of the lane. It reads

$$lq_k^t = \frac{q_k^t}{l_k}. \tag{3}$$

Let us denote N the number of phases during one cycle, and L_i the set of lanes that are active (i.e. showing green light) during phase i. We define the maximum queuing for each phase i at time t as

$$s_i^t = \max_{k \in L_i} lq_k^t. \tag{4}$$

We propose to represent the states by $N + 2$ parameters. The first N components are given by the maximum queuing levels s_i^t for each phase, and the last two components are 1) the index of the current phase, 2) elapsed time of the current phase, i.e. the time between a value *mingreen* that represents the minimum green time set up for the active phase and a value *maxgreen* defining the maximum green time accordingly.

3.2.2. Action definition

The agent (the signalized intersection controlled by RL) has N possible actions[14] that can be selected at each iteration of the decision and learning process. When a new active phase is selected at time t, the next iteration will take place at time $t + mingreen$. If the current phase remains active, then the next iteration will take place at time $t +$ one second. This reflects that when a light turns green, there is a minimum amount of time before it can change; after this amount of time, the state of the system is reevaluated every second. Action at time t is denoted $a_t = i,\ i \in 1, ..., N$, i being the number of the action taken. If an iteration of the learning process occurs at time t, we denote $p(t)$ the previous iteration. If it is the same as the action at previous iteration (i.e. $a_t = a_{p(t)}$), the result of the action is to extend the green time on the current phase i by 1 second. The next iteration of decision making is launched one second afterwards. If not ($a_t = i \neq a_{p(t)}$), the result is to change the phase to the $i - th$ phase after accounting for the yellow and the All red on the current phase, and the minimum green on the next chosen phase. A new iteration of decision making starts after a time of *mingreen*.

3.3. Reward definition

3.3.1. Reward definition 1: Queue length

The reward is defined as the difference of the sum of squared maximum queue lengths between two decision points as defined by Oliveira et al.[14]. It reads

$$R_t = \sum_{i \in N} (\max_{l \in L_i} q_l^t)^2 - \sum_{i \in N} (\max_{l \in L_i} q_l^{p(t)})^2. \tag{5}$$

3.3.2. Reward definition 2: Cummulative delay

The reward is defined as the difference between the total cumulative delays between two decision points[15,16,17]. It reads

$$R_t = \frac{\sum_{i \in N} \sum_{l \in L_i} (\sum_{v \in V_l^t} Cd_v^t - \sum_{v \in V_l^{p(t)}} Cd_v^{p(t)})}{\max(\sum_{v \in V_l^t} Cd_v^t, \sum_{v \in V_l^{p(t)}} Cd_v^{p(t)})}, \tag{6}$$

where V_l^t contains all vehicles on lane l at time t and Cd_v^t is the vehicle cumulative delay, ie the total time spent by the vehicle v in a queue up to time t[15,17]. The cumulative delay of a phase is the summation of the cumulative delay of vehicles traveling in the lane group of this phase.

3.3.3. Reward definition 3: Throughput

This reward is defined as the number of cars that cross the intersection between two decision points[16]. It reads

$$R_t = \sum_{l \in L_i} Vout_{t,p(t)}, \tag{7}$$

where $Vout_{t,p(t)}$ is the number of vehicles that crossed the intersection between t and $p(t)$ on lane l.

3.4. Action Selection Strategy

Convergence of Q-learning agents requires adequate exploration, especially if the environment being controlled is stochastic. Exploration ensures that all areas of interest are visited sufficiently often. In this paper, a *ϵ-greedy* action selection strategy is adopted, where the best action is selected with the probability ϵ and a random action is selected with the probability $1 - \epsilon$. The exploration rate is chosen to be gradually decreasing in order to give the agent more exploration at the beginning and exploitation at the end. For this, we represented the exploration rate as an exponentially decreasing function: $\epsilon = e^{-Ej}$ as proposed in[15] where E is a constant and j the number of iterations. For the testbed intersection, we use a constant $E = 0.01$, which is sufficient for the exploration of our state space.

4. Experimental Setup

We now describe the experimental setup we used in order to illustrate the RL efficiency.

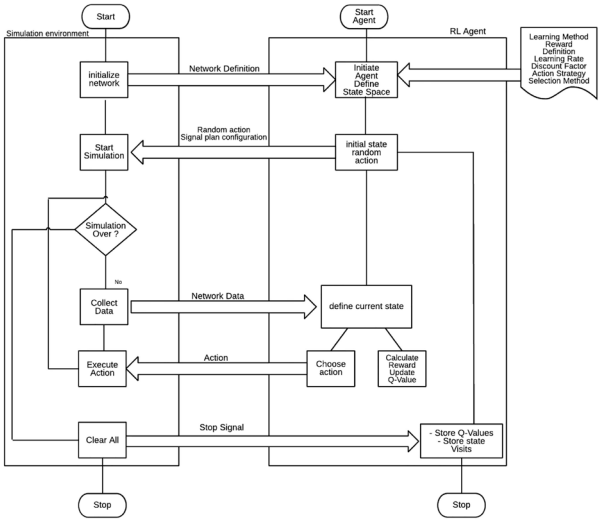


Fig. 1. RL agent interaction with environment.

4.1. RL agent

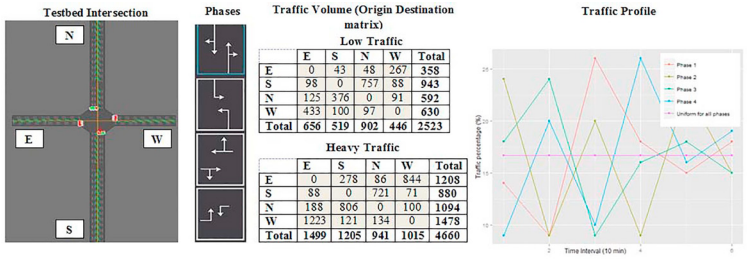
The RL agent implementation for this study was developed with the script language Python as a controller with many options (e.g. regarding reward definition, learning method...etc.) to test different scenarios in interaction with the simulation environment as represented in Fig. 1.

4.2. Testbed Intersection

The program was tested on four-way intersection that was used by El-tantawy et al.[15]. The intersection has three lanes per incoming link/street including a protected left turn lane. This results in four phase intersection used with two different Origin-Destination (OD) matrices[15] that represent a low traffic and heavy traffic volumes, these two combined with to traffic profiles, uniform and variable (the variable profile combines all the variable arrival of vehicles in each phase) (Fig. 2) gives four traffic patterns (low-uniform, low-variable, high-uniform, high-variable). A fixed signal plan was defined for each of the two OD matrices using Webster's method[9] (which results in a cycle length of 120s) . A microscopic simulator called Paramics was used to build the scenarios. Our agent interacts with the simulator using a plugin developed within Paramics with the help of the Application Programming Interface (API) functions (Fig. 1). The experiments consist of 100 one hour simulation runs consisting on continuous learning of the agent and a repeated process of simulation runs. Combined with the three reward definitions defined in section 3.3, this gives in total 16 scenarios (two OD matrices x two traffic profiles x three reward definitions). We analyze disaggregated data consisting of average delay, throughput and queue length, and aggregated data, such as cumulative delay, queue levels, Q-Values, state visits, state-action visits for all simulation runs.

5. Results And Discussions

Fig. 3 shows the average delay of vehicles performance between Q-learning and pre-timed signal plan. The algorithm starts in exploration and therefore we see scenarios with the worst results and after enough learning the outcome becomes more stable and advantageous. In average, the performance improves well. In (Table. 1) we see an improvement of up to 68.5% (compared to the Cummulative Delay Reward) in average delay in low uniform traffic and 49% in average delay in high variable which is bigger than the performance of 48.7% in previous work[15]. The results also show a high throughput, due to the optimal choice by the RL controller to give right of way to the phase with high



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

Heavy Traffic	E	S	N	W	Total
E	0	278	86	844	1208
S	88	0	721	71	880
N	188	806	0	100	1094
W	1223	121	134	0	1478
Total	1499	1205	941	1015	4660

Fig. 2. Simulated Intersection and Traffic dynamics.

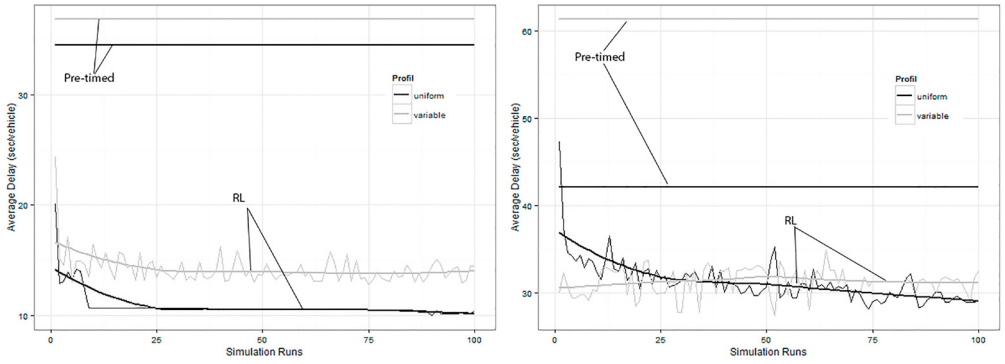


Fig. 3. Comparaison of RL vs. pre-timed resulting average vehicles delay (in seconds) in each simulation run for low traffic (on the left) and high traffic (on the right).

level of staturation. This shows an improvement of 8.1% in throughput compared to 3.4% in the previous study [15]. but this however has increased the average queue length due to the number of vehicle arrival to the intersection.

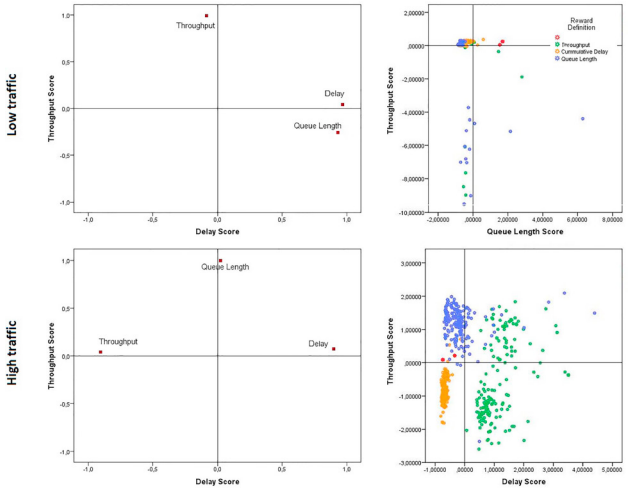


Fig. 4. PCA analysis of reward definition in low (upper row) and high traffic (lower row) scenarios: factorial (left) and perceptual (right) maps.

5.1. Reinforcement learning Effect

In order to better understand the effect of reward definitions on the outcome, we decided to do a Multivariate analysis, in particular a principal component analysis (PCA) on the output results of the scenarios (average delay,

queue length, throughput). The knowledge generated by this analysis, helps classifying the different reward definitions and their outcome on traffic dynamics. A first classification however showed that varying traffic profiles does not affect the performance of the different algorithms, so we will focus on the variation of traffic volume.

5.1.1. Low Traffic

Low traffic was demonstrated in this paper by using the O-D matrix shown in Fig. 2 with a total volume of 2523 vehicles/hour. The results of the PCA used on these scenarios which are approximately 95% representative show that the queue length and the average delay have a strong correlation and queue length is in strong correlation only with throughput (Fig. 4 Factorial Map). So we can rely on two parameters, either delay and throughput or delay and queue length to represent the quality of traffic in this intersection. The choice of the reward is not affecting the performance of the algorithm as shown in the perceptual map in Fig. 4 . However we can see some heterogeneity of queue length reward in the results. This is due to the fact that queue length has limits in penalizing and rewarding the agent. As an example, if the queue length value is equal to zero in the active phase and the queue length value on the other phases is at its limits the agent would not know if he made the correct action by extending the green time of the current phase since the queue length reward would be zero. On the other hand, the cumulative delay reward has a strong homogeneity in results despite the fact that in variable profile the throughput reward produces better results (Table. 1) and this shows how cumulative delay is suitable for the convergence of RL.

Table 1. Detailed results of each scenario

	Traffic Volume	**Traffic profile**	**Average Delay(sec)**	**Throughput (veh)**	**Avg. Queue Length Per Phase (m)**
Queue Length Reward	Low	Uniform	13.9	2446	17.15
		Variable	11.4	2441	18.57
	High	Uniform	54.52	4131	48.32
		Variable	44.36	4293	48.65
Cumulative Delay Reward	Low	Uniform	10.86	2489	18.22
		Variable	14.2	2530	21.48
	High	Uniform	31.22	4544	41
		Variable	31.26	4551	39.68
Throughput Reward	Low	Uniform	13.76	2485	19.1
		Variable	11.92	2482	22.88
	High	Uniform	65.6	3490	38.37
		Variable	141.19	3216	45.8
pre-timed Wesbter's method	Low	Uniform	34.47	2470	33.79
		Variable	36.8	2497	34
	High	Uniform	48.1	4315	44.64
		Variable	61.34	4211	44.36
RL Improvement (%)	Low	Uniform	68.5	0.8	46
		Variable	61.4	1.3	36.8
	High	Uniform	35	5.3	8.1
		Variable	49	8.1	10.5

5.1.2. High Traffic

The OD matrix of high traffic is represented in this paper by a traffic volume of 4660 vehicle/hour (Fig. 2) which is 45% more than the low traffic. The factorial map of high traffic in Fig. 4 shows the same correlations as the factorial map of low traffic (except the negative correlation between delay and throughput), which means we can also select two of these parameters to represent the traffic quality in this intersection. The perceptual map in Fig. 4 shows a different outcome for each reward definition and the choice of the later is crucial to the performance of the algorithm. The cumulative delay reward produces homogeneous results and reduces queue lengths and average delay of vehicles.

This helps the algorithm to stabilize and learn with less exploration. The same holds for the queue length reward, except that it is inferior to the cumulative delay reward due to the example we depicted before. The throughput reward produces heterogeneous results which require more exploration and since throughput reward as defined does not reveal variable numeric rewards with big differences, it could help to give a constant high learning rate to make future knowledge more important.

6. Conclusion

In this paper, a reinforcement learning platform was developed to adapt traffic signal control to the dynamics of traffic pattern which showed robust adaptation and remarkable performance compared to pre-timed signal control (using Webster method) in a real life intersection. Also the new state space performs well in throughput and average delay, particularly for high and variable traffic. The analysis of the reward definitions showed that the performance of a reward function depends on the traffic volumes in the intersection and also the equipment used to monitor the intersection since some indicators need more sophisticated sensors such as cumulative delay which would require video surveillance or GPS equipped vehicles. Queue length showed to be secondary in high traffic but is easier to measure with standard sensors (e.g. loop detectors). As a perspective, it would be interesting to extend the work done in this paper to an intersection with more data available in order to investigate the effect of real traffic profiles and their role on the performance of RL or reward definitions. Also, the effect of different parameters regarding the architecture of intersections needs to analyzed (e.g. number of phases, number of lanes on each road...etc.). On the RL level, investigations should be done by developing/testing other reward models (e.g. delay squared) and their combination in a real world use cases. The coordination of multiple intersections needs to be investigated due to the fact that traffic arrival patterns at an intersection highly depend on the way traffic is controlled at the upstream intersection.

References

1. Robertson, D.. TRANSYT: A Traffic Network Study Tool. *Road Research Laboratory* 1969;**253**:37.
2. Hunt, P.B., Robertson, D.I., Bretherton, R.D., Winton, R.I.. Scoot - a Traffic Responsive Method of Coordinating Signals. *Transport and Road Research Laboratory* 1981;:41.
3. Sims, A.G., Dobinson, K.W.. The Sydney coordinated adaptive traffic (SCAT) system philosophy and benefits. *IEEE Transactions on vehicular technology* 1980;**29**(2):130–137.
4. Henry, J.J., Farges, J.L., TUFFAL, J.. The PRODYN real-time traffic algorithm. *In Proc of the {IFAC} Symposium, Baden-Baden* 1984; :305–310.
5. Gartner, N.H.. OPAC: A Demand Responsive Strategy for Traffic Signal Control. *Transportation Research Record* 1983;**906**(January 1983):75–81.
6. Mauro, Vito and Di Taranto, C.. UTOPIA. *Control, computers, communications in transportation* 1990;.
7. Donati, F., Mauro, G., Roncolini, G., Vallauri, M.. A Hierarchical Decentralised Traffic Light Control System. *IFAC 9th World Congress* 1984;**2**:11G/A—-1.
8. Barto, A.G., Sutton, R.S., Barto, A.G.. *Reinforcement learning: An introduction*; vol. 9 of *A Bradford book*. Bradford Book; 1998.
9. Webster, F.V.. *Traffic signal settings*. Road research technical paper. H.M. Stationery Office; 1958.
10. Bazzan, A.L.C.. Opportunities for multiagent systems and multiagent reinforcement learning in traffic control. *Autonomous Agents and Multi-Agent Systems* 2009;**18**(3):342–375.
11. Thorpe, T.L.. *Vehicle Traffic Light Control Using {SARSA}*. Master's thesis; Computer Science Department, Colorado State University, Fort Collins, Colorado; 1997.
12. Abdulhai, B., Kattan, L.. Reinforcement learning: Introduction to theory and potential for transport applications. *Canadian Journal of Civil Engineering* 2003;**30**(6):981–991.
13. Bingham, E.. Reinforcement Learning in Neurofuzzy Traffic Signal Control. *European Journal of Operational Research* 2001;**131(2)**:232—-241.
14. de Oliveira, D., Bazzan, A.L.C., da Silva, B.C., Basso, E.W., Nunes, L.. Reinforcement Learning based Control of Traffic Lights in Non-stationary Environments: {A} Case Study in a Microscopic Simulator. In: *Proceedings of the 4th European Workshop on Multi-Agent Systems EUMAS'06, Lisbon, Portugal, December 14-15*. 2006, p. 31–42.
15. El-Tantawy, Samah and Abdulhai, B.. Comprehensive Analysis of Reinforcement Learning Methods and Parameters for Adaptive Traffic Signal Control. In: *Transportation Research Board 90th Annual Meeting*. 2011, .
16. Brys, T., Pham, T.T., Taylor, M.E.. Distributed learning and multi-objectivity in traffic light control. *Connection Science* 2014;**26**(1):65–83.
17. Arel, I., Liu, C., Urbanik, T., Kohls, A.. Reinforcement learning-based multi-agent system for network traffic signal control. *IET Intelligent Transport Systems* 2010;**4**(July 2009):128.