Urban traffic signal control using reinforcement learning agents

Article *in* IET Intelligent Transport Systems · October 2010

DOI: 10.1049/iet-its.2009.0096 · Source: IEEE Xplore

3 authors, including:

Balaji Parasumanna Gokulan
Yodlee Infotech Private Limited
13 PUBLICATIONS **410** CITATIONS

SEE PROFILE

D. Srinivasan
National University of Singapore
344 PUBLICATIONS **9,032** CITATIONS

SEE PROFILE

Published in IET Intelligent Transport Systems
Received on 28th October 2009
Revised on 26th February 2010
doi: 10.1049/iet-its.2009.0096

ISSN 1751-956X

Urban traffic signal control using reinforcement learning agents

P.G. Balaji X. German D. Srinivasan

Department of Electrical and Computer Engineering, 4 Engineering Drive 3, National University of Singapore, Singapore 117576, Singapore
E-mail: g0501086@nus.edu.sg

Abstract: This study presents a distributed multi-agent-based traffic signal control for optimising green timing in an urban arterial road network to reduce the total travel time and delay experienced by vehicles. The proposed multi-agent architecture uses traffic data collected by sensors at each intersection, stored historical traffic patterns and data communicated from agents in adjacent intersections to compute green time for a phase. The parameters like weights, threshold values used in computing the green time is fine tuned by online reinforcement learning with an objective to reduce overall delay. PARAMICS software was used as a platform to simulate 29 signalised intersection at Central Business District of Singapore and test the performance of proposed multi-agent traffic signal control for different traffic scenarios. The proposed multi-agent reinforcement learning (RLA) signal control showed significant improvement in mean time delay and speed in comparison to other traffic control system like hierarchical multi-agent system (HMS), cooperative ensemble (CE) and actuated control.

1 Introduction

Traffic control in the urban areas is becoming increasing complex with the exponential growth in vehicle count. Expansion of the road network to accommodate the increased vehicle count is not a socially feasible option and is essential to increase the utilisation of the existing infrastructure through proper regulation of traffic flow. Traffic signals were introduced to control the traffic flow, thereby improving the safety of road users. However, traffic signals create bottleneck for traffic flow in lanes that do not have the right of way during a specific phase and optimisation of signal timings is required to reduce the overall delay experienced by all vehicles at the intersection. Optimisation can be performed in offline (pre-timed) or online (adaptive) manner.

In pre-timed or fixed time signal control, Webster's formula is used to calculate the green and the cycle time offline using traffic data collected from the road network. Pre-timed signal control cannot handle any variation in traffic from the training patterns resulting in increased travel time delay. Adaptive signal controls overcome this limitation by adjusting the timings dynamically with changing traffic patterns. Actuated signal controls adaptively increments/decrements the green time of a phase on detecting the presence/absence of vehicle in a lane. Actuated controls lack the ability to foresee increased traffic flow and bases its decision on instantaneous flow values. Further, it results in higher delay as green time is not held for upstream platoons causing higher percentage of vehicles to be stopped [1].

Various computational intelligence techniques such as hybrid fuzzy genetic algorithm [2], ant colony-based optimisation [3], emotional algorithms [4] and neuro-fuzzy networks [5] calculate the green time required by forecasting the future traffic inflow. First limitation is, a large training data that encompass all the dynamics of the traffic is required for fine tuning parameters of the controller and is difficult to obtain. Second, most of the above controllers were designed for isolated intersection, thereby simplifying the model and reducing its suitability to coordinated interconnected intersections.

SCOOT [6, 7], SCATS [8, 9] and Green Link Determining (GLIDE) [10] are examples of centralised traffic signal

controllers that have been implemented on large-scale networks successfully. However, centralised controllers increase the requirement for extensive communication of information and computational requirement for efficient data mining of information required to compute optimal green time. The limitation can be solved by implementing a distributed multi-agent architecture where larger problem can be divided into smaller sub-problems. Multi-agent system is a group of autonomous agents capable of perceiving the environment and decides its own course of action for achieving a common goal. The agents could achieve this either by cooperation or competition. The communication between agents increases the global view of the agent and increases the coordination. In [11], distributed agent system utilising evolutionary game theory to assign reward or penalty was proposed. The limitation is the necessity to compute pay-off matrix for each state-action pair. In [12], the advantages and disadvantages of multi-agent system have been highlighted and a theoretical model of agent based on estimated traffic state has been proposed. In [13–15], semi-distributed agent architecture based on distributed constraint optimisation, swarm intelligence methods and hybrid intelligent techniques combining fuzzy logic, neural networks and evolutionary computation have been attempted. The limitation is the amount of data to be communicated and conflict of decision among agents. Agent system with reinforcement learning capability has shown to improve the performance significantly [16]; however, tests were conducted on simple road network with less number of intersections. In this paper, a reinforcement learning distributed multi-agent architecture has been proposed and tested on a large urban arterial road network.

The paper is organised into seven sections. Section 2 details the proposed multi-agent architecture. Section 3 describes learning of parameters using reinforcement learning. Section 4 details the performance measures used followed by a brief note in Section 5 on the benchmarks used. Section 5 discusses the simulation platform used and the comparative analysis over the benchmark signal controls. Section 6 summarises the work done in this paper.

2 Proposed agent architecture

The proposed multi-agent system has a distributed architecture with each agent capable of making own decisions without any central supervising agent. Traffic signal at each intersection is controlled by an agent. The agent collects local traffic data collected from induction loop detectors placed near the stop line of incoming and outgoing links connecting the neighbouring intersections. Agents' communicate outgoing traffic information to the neighbouring agents. The structure of individual agent architecture is shown in Fig. 1.

Based on locally collected inputs and communicated information, intersection agents determine green time required for each phase in the next cycle period. Agent possesses local memory to store traffic demand and create a data repository to assess future traffic demand and effectiveness of agent's actions. Agents fine tune and learn the decision model of each intersection by observing the expected utility for each state-action pair and update using online Q-learning.

The traffic demands in the road network are quite uniformly spread and can be characterised by different type of distribution based on the traffic flow information collected from the network. However, vehicles have a large number of route choices and route selected depends on driver behaviour therefore no specific green wave policy can be selected based only on historic traffic flow patterns. Therefore explicit offset settings was not used in this work as synchronisation is achieved through communication of information between agents and learning by visiting each state-action pairs sufficient number of times.

2.1 Traffic input

Traffic data like vehicle occupancy T_{occupied}, the amount of time the vehicle is present on the detector, Q_{length}, length of queue of vehicles at the end of each phase and V_{count},

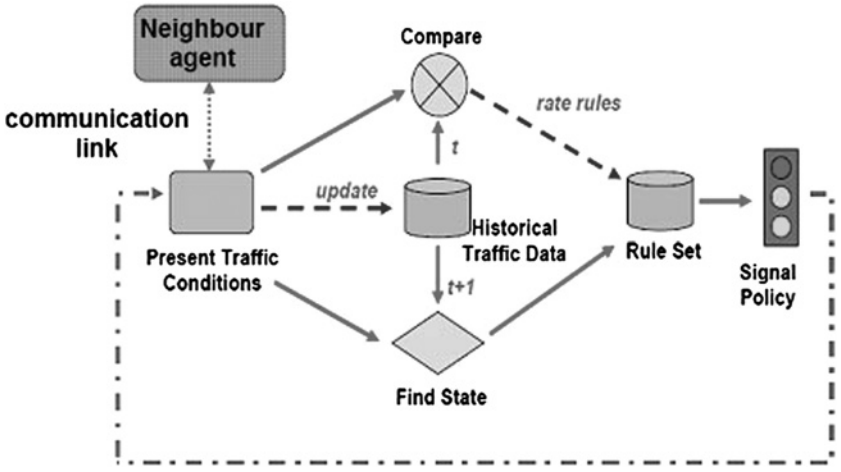

Figure 1 *Proposed agent architecture*

doi: 10.1049/iet-its.2009.0096

number of vehicles crossing a specific intersection during a phase are collected from the incoming and outgoing links of the intersection. Since vehicles in most lanes are free to choose the exit lanes, data from outgoing detectors need to be transmitted to neighbouring intersection agents to enable prediction of incoming traffic. For proper estimation of current traffic state, queue value has to be used along with vehicle occupancy and count, which tends to stagnate during high traffic flow.

Traffic state estimation is performed based on occupancy, queue and vehicle count of maximum congested lanes (1) as averaging across the lane causes improper classification of traffic

$$T_{\text{occupied}} = \max_i(\max_j(T_{\text{occupied}}(a_i l_j))) \tag{1}$$

where $a_i l_j$ is the jth lane for approach a_i.

2.2 Rule base

Agents compute new phase length in a cycle based on the locally observed traffic input and information communicated by neighbouring intersections using a set of rules. The rules specify the required amount of change in green time δ_i for specific traffic condition at the intersection and is calculated as weighted sum of rules output as shown in

$$T_{\text{phase}}^{\text{new}} = T_{\text{phase}}^{\text{old}} + \sum_i \rho_i \delta_i, \quad T_{\min} \le T_{\text{phase}}^{\text{new}} \le T_{\max} \tag{2}$$

where ρ_i is the weight assigned to each rule, δ_i is the output of each rule and i is the number of rules. Upper and lower limits $[T_{\max}, T_{\min}]$ are imposed to avoid indiscriminate increase of green time. For all signals, $T_{\min}$ is fixed at 10 s, however, $T_{\max}$ varies in accordance to the number of phases and total cycle length limited to 120 s. Different rules used by agent for estimation of green time are explained in the following sections. Calculation of ρ_i is explained in detail in Section 3.3.

2.2.1 Occupancy ratio: Agent uses the occupancy ratio (ratio of vehicle occupancy time to green time of the phase) to estimate green time required by vehicles present at the stop line of the intersection. Occupancy is directly related to vehicle density and indicates the current state of the intersection. Based on the speed–flow–density characteristics, ratio of vehicle occupancy (in seconds) to green time of a phase gives an accurate indication of degree of saturation of the network. However, there is no universal best value for the occupancy ratio and varies with level of congestion. An underutilised phase (large green time for low vehicle count) have a low occupancy ratio and increases the delay experienced by vehicles. Based on the occupancy ratio, each agent computes the extension or reduction in green time of phase in progress

$$\delta_1 = \frac{T_{\text{occupied}}}{O_{\text{ccRatio}}} - T_{\text{phase}} \tag{3}$$

2.2.2 Local traffic variations: Second most important influencing factor in the adjustment of green time of a phase is the local variation in traffic condition at consecutive time periods. As the cycle length is dynamically adjusted by agents, ratio of T_{occupied} to T_{phase} at consecutive time periods is used rather than raw vehicle occupancy data

$$\text{load} = (T_{\text{occupied}}/T_{\text{phase}}) \tag{4}$$

Agents decide the required change in green time by comparing the load with $\text{load}_{\text{target}}$ or threshold value. The threshold value is computed as average of load value experienced during previous t cycles, where t represents the number of previous cycles to be considered. The change in green time is computed using

$$\delta_2 = \Delta * T_{\text{phase}} \tag{5}$$

where

$$\Delta = \begin{cases} \max(\text{load}_{\text{new}} - \text{load}_{\text{target}}, 0), & \text{if } \text{load}_{\text{new}} > \text{load}_{\text{old}} \\ \min(\text{load}_{\text{new}} - \text{load}_{\text{target}}, 0)/2, & \text{if } \text{load}_{\text{new}} < \text{load}_{\text{old}} \end{cases}$$

The old value of load is updated with current load value after every time period. To ensure that extension of green time is relatively slower than reduction of green time, a correction term of 1/2 is included in the computation of Δ. A large correction value can cause instability because of shorter phase split.

2.2.3 Neighbourhood advice: Agent's environment is usually affected by the action of neighbouring agents. This necessitates modifying the behaviour of agent based on neighbouring agents communicated information. The neighbouring agents communicate vehicle occupancy and count at outgoing link of its intersection. Data are communicated as a simple broadcast with identification tag to all the neighbouring intersections. Based on the information in the directory facilitator, agents decide to receive the broadcast information. The received data are stored as Advice in the data repository. The communicated data permit forecasting of traffic inflow and accordingly adjust the green time as in

$$\delta_3 = \frac{\text{Advice}_{\text{new}}}{O_{\text{ccRatio}}} - T_{\text{phase}} \tag{6}$$

$$\text{Advice}_{\text{new}} = \frac{\text{Advice}_{\text{mem}} + \text{Advice}_{\text{old}}}{2} \tag{7}$$

After calculation of δ_3, average value of the $\text{Advice}_{\text{mem}}$ received in the current time period and $\text{Advice}_{\text{old}}$ is used to update the repository.

If an approach is congested to such that at least one turn movement is blocked during a phase, then the vehicles count for that movement is set to zero. This situation arises

when the queue from the downstream intersection reaches the current intersection because of queue spillback and leads to deadlock formation. It is not possible for an agent to differentiate between an empty and saturated intersection. However it is possible to differentiate the two scenarios by using a combination of occupancy and count data communicated by the neighbouring agents. If vehicle count on the outgoing lane is null and occupancy is not null, agent can distinguish the lane as congested and blocked because of the queue spillback. Under such circumstances, green time for the phase is kept fixed at a minimum limit of 10 s to allow clearance of vehicles.

3 Reinforcement learning

All agents in the network must be capable of learning the model of intersection controlled based on the assessment of present average traffic condition and previous day traffic condition at the same period. The learning period has to be long enough to allow aggregation of sufficient traffic information to estimate the current traffic state and capture traffic dynamics. The learning period can be found by experimentation and was calculated as 500 s for the specific road network considered for testing. Conventional learning is difficult as the exact desired value is not available and unsupervised learning or a combination of both (reinforcement learning) needs to be employed. Reinforcement learning utilises scalar time delayed reward received from the environment on selecting an action in a specific state to modify the parameters of intersection model. In this paper, $\boldsymbol{Q}$-learning was used to modify the parameters.

3.1 $\boldsymbol{Q}$-learning

$\boldsymbol{Q}$-learning [17] is a reinforcement learning technique that learns the action value function which provides the expected utility of taking an action in a given state and then following a fixed policy thereafter. The utility or reward is received after time delay from the environment and is a scalar quantity that do not exactly specify the action to be taken. Each agent maintains a $\boldsymbol{Q}$-matrix that stores the $\boldsymbol{Q}$-values for each state-action pair and is updated iteratively as shown in (8). The $\boldsymbol{Q}$-value reaches optimum value when all states are visited sufficiently larger number of times.

$$\boldsymbol{Q}(s, a)^* = (1 - \alpha)\boldsymbol{Q}(s, a) + \alpha(r + \gamma \max_i \boldsymbol{Q}(s', a_i)) \quad (8)$$

where $\boldsymbol{Q}(s, a)^*$ is the optimal value, α is the learning rate in the range [0, 1] and γ is the future discount reward. The learning rate and future discount reward are computed through experimentation to be 0.33 and 0.05 as trade-off needs to be made between the rate of convergence and precision.

3.2 Traffic state estimation

Traffic is discretised into different states using queue and flow data. Average queue computed at the end of each phase by (9) can be used for traffic classification. However for proper classification of traffic, flow value needs to be used in conjunction with queue. The values of queue and flow at time period t are taken as $\boldsymbol{Q}_{\text{score}}$ and flow(t). At the start of learning process, there is no history data available. However, at time period $t+1$, the data of $\boldsymbol{Q}_{\text{score}}$ and flow(t) get stored as history value for the previous day namely H_{score} and H_{flow}, respectively. The change in current traffic is computed as the difference between current traffic queue score and queue experienced in the previous day at the next time period $\boldsymbol{Q}_{\text{score}}(t) - H_{\text{score}}(t+1)$ and assigned membership grade that classifies current traffic change into three low, medium and high traffic as shown in Fig. 2

$$\boldsymbol{Q}_{\text{score}} = \frac{\sum \boldsymbol{Q}_{\text{length}}(\text{phase}_i)}{N_{\text{phase}}} \quad (9)$$

The rate of change of traffic δ_x is computed as $(H_{\text{flow}}(t+1) - \text{flow}(t))/\text{flow}(t)$ and $(H_{\text{score}}(t+1) - \boldsymbol{Q}_{\text{score}})/\boldsymbol{Q}_{\text{score}}$ and is used to determine whether the traffic is decreasing, stable or increasing and assigned a membership grade similar to Fig. 2. Combining the rate of change of traffic and current change in traffic, traffic at the intersection is classified into nine possible states and is shown in Table 1. The current traffic state is determined as the output with highest firing level using fuzzy logic.

3.3 Parameter update

Traffic states have been completely defined in the previous section. However, the action space needs to be defined to

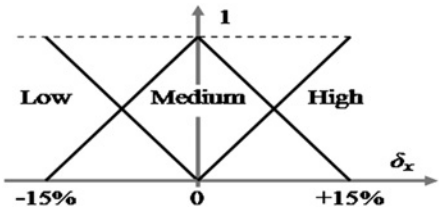

Figure 2 *Structure of fuzzy membership*

Table 1 Possible traffic states

current traffic	low	0	1	2
	medium	3	4	5
	high	6	7	8
		decreasing	stable	increasing
		changes in traffic		

IET Intell. Transp. Syst., 2010, Vol. 4, Iss. 3, pp. 177–188
doi: 10.1049/iet-its.2009.0096

complete $\boldsymbol{Q}$-matrix. The total green time extension can be defined by ρ_i whose value lies in the range between [0, 1] and divided into 12 equal values. Each agent maintains a $\boldsymbol{Q}$-matrix that matches the nine traffic states defined in the previous section to the 12 action values. At the end of each learning period, the agent computes the reward r received from environment after choosing action a_i when in state s_i. The reward value is computed using

$$r^{\text{new}} = \frac{\boldsymbol{Q}_{\text{score}}(t) - H_{\text{score}}(t)}{\boldsymbol{Q}_{\text{score}}(t)} + \eta \frac{\boldsymbol{Q}_{\text{score}}(t) - \boldsymbol{Q}_{\text{score}}(t-1)}{\boldsymbol{Q}_{\text{score}}(t)} \tag{10}$$

where η is in the range [0, 1]. The reward value is positive if the queue is smaller than the historic queue as well as the queue for previous period. Since, the traffic demands vary with time, queue value also varies. Comparison with historic value tracks the traffic pattern over a large period and queue value of previous time period tracks short time variations. Therefore η needs to be kept small so that reward comes from comparison with historic values. Once current state is detected, the appropriate action value is chosen as one with highest $\boldsymbol{Q}$-value. In case of multiple actions having same $\boldsymbol{Q}$-value, one of the action is selected randomly. Greedy action selection strategy was used to increase the exploration to increase the visited state-action pairs.

3.4 Memory update

Once the state-action pair has been found, the memory of the agent is updated. The historic queue score can be updated iteratively in

$$H_{\text{score}}^{\text{new}} = (1-\beta)H_{\text{score}}^{\text{old}} + \beta r \tag{11}$$

The coefficient β decreases from 0.5 to 0.1 for the first few iterations, then stays at 0.1. The same equation as in (11) is used for calculating H_{flow}. The value of β needs to be varied with large steps in the start so that higher preference is given to time delayed rewards than historical value and reduced to lower value so that the learnt values are not forgotten and is determined through experimentation with different values.

Cooperation between the agents is achieved by averaging the $\boldsymbol{Q}$ matrix values between immediate adjacent neighbouring agents each time period. This ensures the improvement in performance as agents learn from experience of the neighbouring agents.

4 Performance measures

The performance of proposed reinforcement learning (RLA) algorithm in a simulated road traffic environment is evaluated based on three parameters namely vehicle count, total mean delay and current mean speed of vehicles inside the road network (Fig. 3).

4.1 Vehicle count

Vehicle count is the total number of vehicles present inside the road network at a given time and is calculated as the

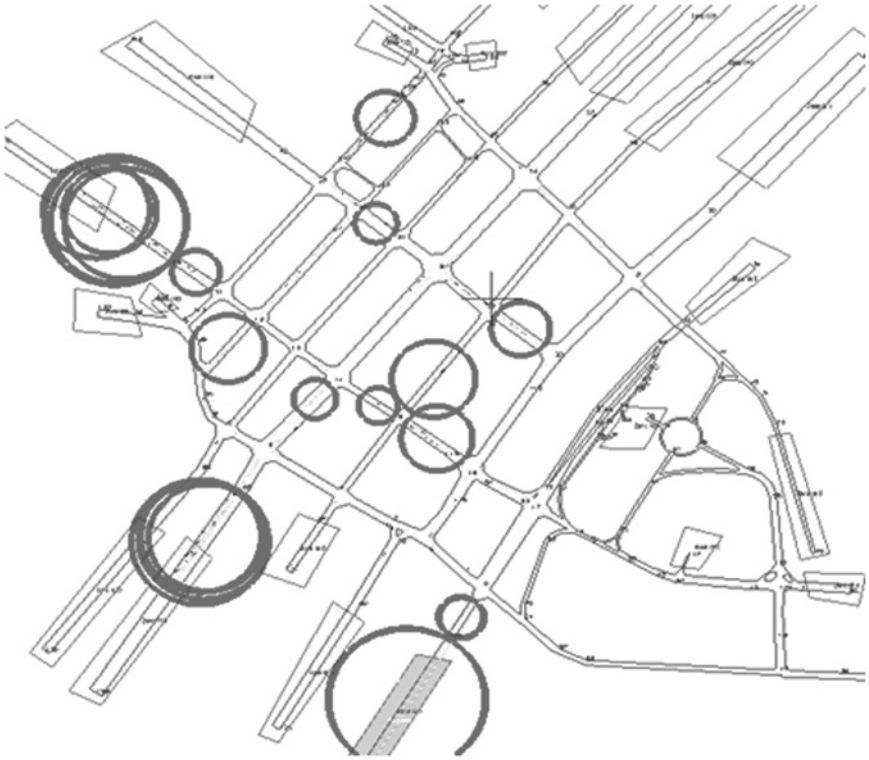

Figure 3 *Simulated road network with indication of prominent hotspots caused because of pre-timed signals*

difference between number of vehicles entering and leaving the network during the estimation period. The vehicle count gives an accurate indication of the congestion level inside the network at a specified period of time.

4.2 Total vehicle mean delay

Total mean delay is the average value of delay experienced by vehicles to reach the destination from starting point of the network and is expressed in seconds. Mean delay is the sum of total stopping time which corresponds to time lost waiting at intersections, and the travel time which depends on speed of vehicles inside the network

$$T_{AD} = \sum_{i=1}^{n} T_D / T_N \tag{12}$$

where T_{AD} is the total average delay, n is the number of intersection, T_D is the delay experienced by vehicles at an intersection and T_N is the total number of vehicles released into the network. Little [18] and Highway Capacity Manual – HCM2000 [19] show the wide acceptance of delay parameter for validating the signal controllers.

4.3 Current vehicle mean speed

For better understanding of the results, current mean speed of the vehicle inside the network is used along with the time delay value. The importance of using current mean speed in validating the signal controller has been highlighted in [20].

5 Benchmarks

It is difficult to find a good benchmark for large-scale traffic signal control problem given the following reasons:

1. Some of the existing algorithms are developed for simplified traffic scenarios and hence not suitable for benchmarking.

2. Commercial traffic signal control programs, which are known to work well, are not easily available because of proprietary reasons.

Hence in all the experiments, GLIDE [10], modified version of SCATS used in Singapore, hierarchical multi-agent system (HMS) [21] developed in and cooperative ensemble (CE) [22] are used as benchmarks. HMS and CE have already been compared with GLIDE and hence simulation plot results are not included to avoid redundancy. HMS is a semi-distributed multi-agent traffic signal control with hierarchical architecture. It consists of three layers of agents with increasing hierarchy and control. The agent at the intersection decides the green time required based on local traffic information and cannot communicate with agents in same hierarchy and uses Webster's method to compute the green time requirement. The zonal agent oversees the functioning of five intersection agents by monitoring the action plan of individual agent and providing directives received from supervising agent. Supervising agent is in the top layer of hierarchy and oversees the functioning of entire system. The zonal agents utilise evolutionary fuzzy algorithm to generate the rule base for control and compute cooperation levels required between agents using neuro-fuzzy system. For detailed description of the HMS, refer to [21].

CE [22] is a distributed multi-agent architecture, where the agents self-organise and form clusters of cooperating agents. The clusters are formed dynamically using graph theoretical method. The teams or clusters cooperate to reduce the overall time delay experienced by the group rather than an individual. Overlap in the cooperative clusters is possible; however, it is limited to avoid excessive computation.

6 Simulation results and discussions

The proposed RLA signal controller was tested on a simulated network of 29 intersections. The simulated network is the highly congested section of the busy Central Business District area in Singapore [23]. The network is simulated using PARAMICS, a microscopic simulation software capable of simulating the driver's behaviour, dynamic re-routing of vehicles and incidents efficiently. The network serves as an ideal test bed because of the geometry and heterogeneity in the classification of links (major and minor roads with varying speed limits).

Four types of simulation were used to evaluate the performance of the proposed RLA signal control. They are namely the typical scenario with morning peak (3 h), typical scenario with morning and evening peaks (24 h), extreme scenario with dual peaks (6 h) and extreme scenario with multiple peaks (24 h). It must be noted that extreme traffic scenarios are hypothetical traffic peaks created to test the reliable control of traffic by the proposed RLA signal control in cyclic repetitive stress conditions. It also serves to showcase the response and settling time of the signal control.

The origin–destination data collected from Land Transport Authority Singapore, is used to recreate the peak traffic conditions. Even though the peak traffic data are pre-fixed, the number of vehicles actually released into the network varies according to the random seeds set before the simulation. Since PARAMICS dynamically adjusts traffic model characteristics like gap acceptance, lane change, merge and so on, the traffic dynamics is different for each simulation run with different random seeds. The PARAMICS model has been validated for the specific data and has been previously used for simulation testing in [5, 20, 22, 23].

6.1 Typical scenario with morning peak (3 h)

The typical scenario with morning peak is used to validate the performance improvement in traffic condition using RLA

doi: 10.1049/iet-its.2009.0096

signal control for short time traffic variations. Twenty simulation runs using different random seeds were carried out for each signal control technique compared. Since variance of the outcome of simulation runs was small, average value was taken as the representation of the outcome.

Fig. 4*a* shows comparison of the time delay experienced by vehicles in road network using different types of traffic signal control. The proposed RLA signal control shows a 15% improvement in delay in comparison to other benchmarks. The improvement in performance can be attributed to the ability of RLA signal control to foresee traffic increase based on the communicated information and adjust the green timing before the traffic arrives at the intersection. During low traffic period, HMS and CE experience higher delays as their actions are based only on locally collected data and thereby cause more vehicles to be retained inside the network. Though under high traffic condition, the decisions are more coordinated, the number of vehicles to be cleared is much larger because of the vehicles retained during low traffic period, thereby increasing the delay than RLA signal control.

Fig. 4*b* shows the comparison of RLA signal control with and without communication of data between agents. In case of no communication, higher delay is experienced during the peak traffic conditions and is almost equivalent to HMS and CE signal control. CE shows higher delay as it is difficult to form clusters in a dynamic environment with continuous change in traffic flow input.

6.2 Typical scenario with morning and afternoon peaks (24 h)

For the typical scenario with morning and evening peaks (24 h), 20 different simulation runs using different random seeds were carried out for each signal control technique. Average value of simulation runs were taken into consideration when evaluating the performance of each control technique.

Fig. 5*a* shows comparison of mean vehicle delay using different signal control techniques for 24-h typical two peak traffic scenario. Although HMS signal control shows higher mean delay during traffic peaks, RLA signal control

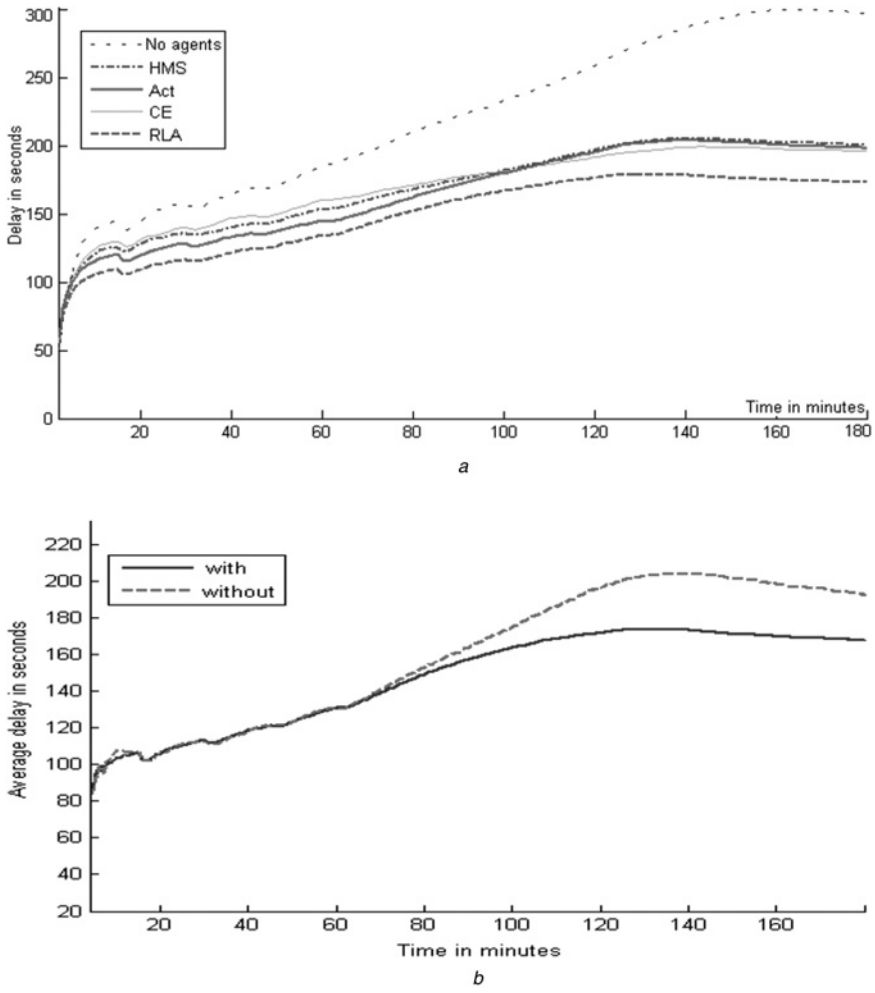

Figure 4 *Three hour single peak traffic scenario*

a Comparison of time delay for 3-h single peak traffic simulation scenario for different agent architectures
b Comparison of proposed RLA architecture with and with communication between agents

has a stable lower delay throughout the simulation period. Fig. 5*b* further indicates a smoother speed transition on using RLA signal control than other controllers.

6.3 Extreme traffic scenarios

Two hypothetical simulation scenarios were designed to test the settling and response time of the signal control algorithms when subjected to repetitive high and low traffic demand. The input demand and the number of vehicles remaining inside the network for 24-h eight peak simulation are shown in Fig. 6*a*. The stress experienced by signal controllers can be seen from the growing values of vehicles retained inside the network. Main reason for the stress can be attributed to vehicle count at the beginning of each peak. When the settling time (time required to bring the vehicle count to non-peak condition) is larger, there is an overlap in the peak traffic build up regions of consecutive peaks causing an increased vehicle count at the start of next peak traffic regions and can be seen from Fig. 6*b*.

The other extreme traffic condition scenario simulated was 6 h two peak traffic condition with higher demand values than 24-h simulation. Fig. 7 shows the mean vehicle delay for short extreme scenario. These simulation scenarios test the limits of the algorithms, as they attempt to stabilise traffic when subjected to repeated peaks. As the HMS algorithm performs better than the CE algorithm under the eight peak extreme scenario [22], results of CE are not included in Fig. 6*b*. RLA algorithm performs better than HMS signal control. HMS signal control produces higher time delay because of the delay in propagation of control signal from supervising agent and absence of local communication between intersection agents. Lower mean delay value of RLA algorithm clearly indicates faster

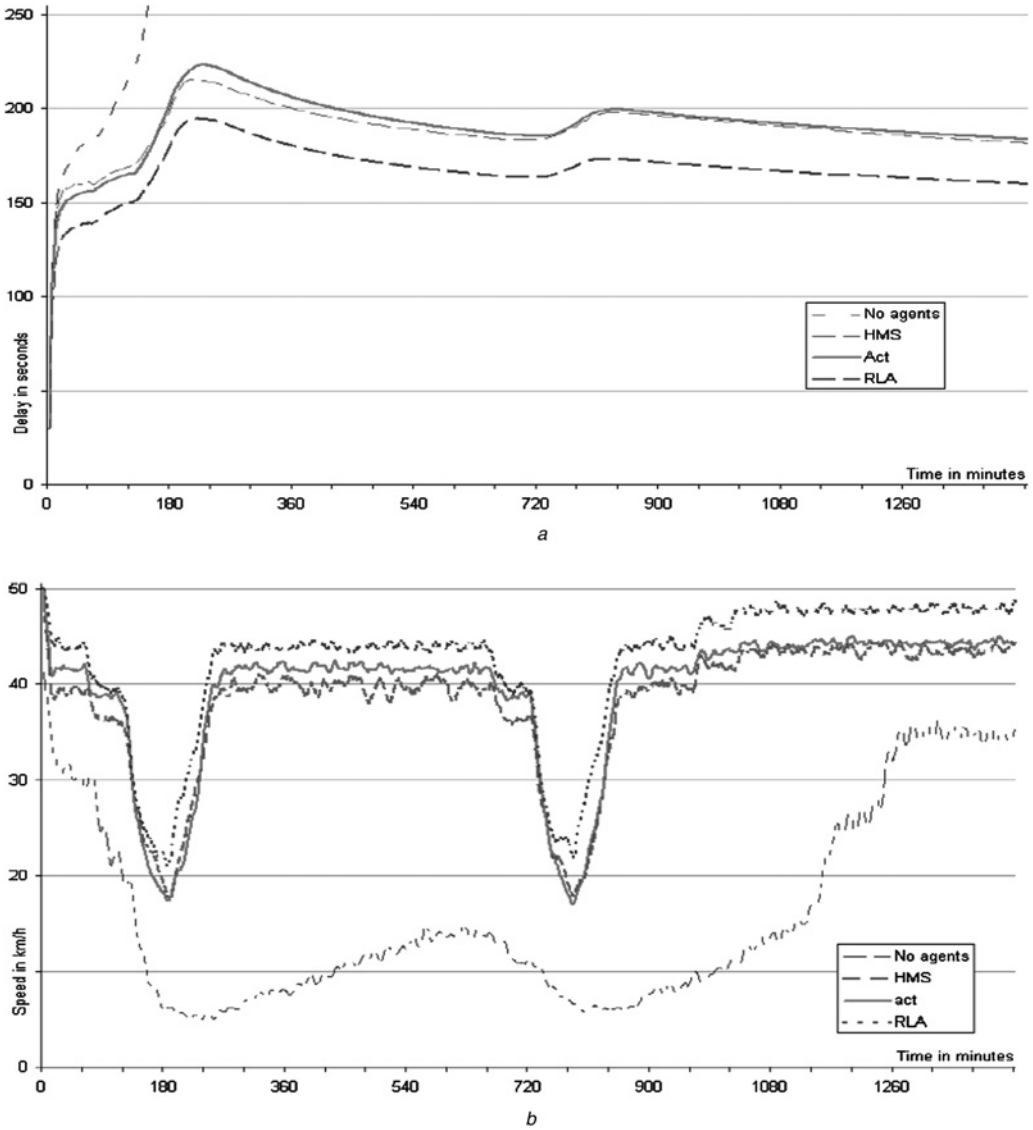

Figure 5 *Twenty-four hour two peak traffic simulation scenario*

a Average travel time delay comparison
b Comparison of current vehicle mean speed

doi: 10.1049/iet-its.2009.0096

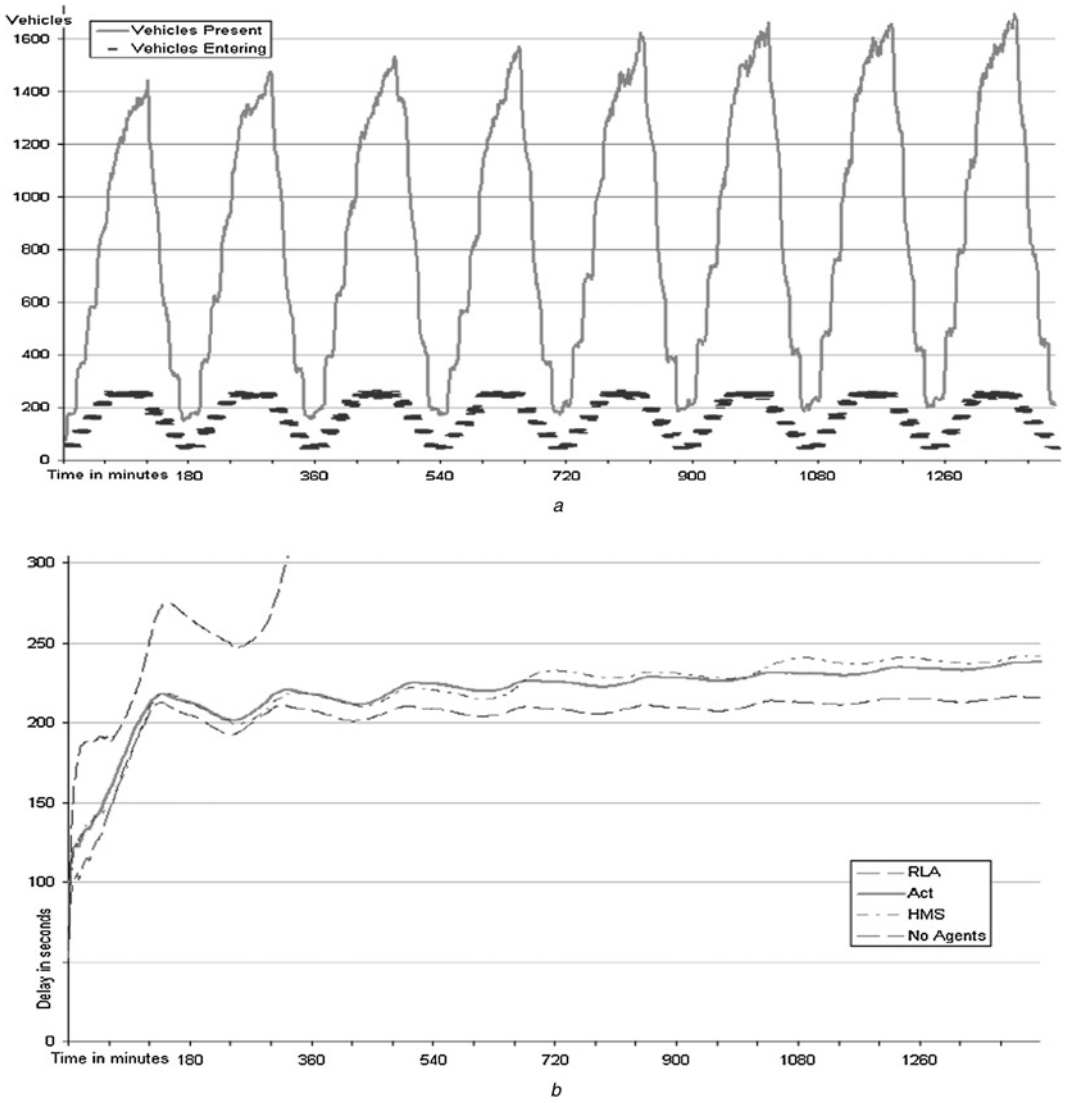

Figure 6 *Twenty-four hour eight peak traffic simulation scenario*

a Traffic demand and count of vehicles present inside the network
b Average travel time delay of vehicles

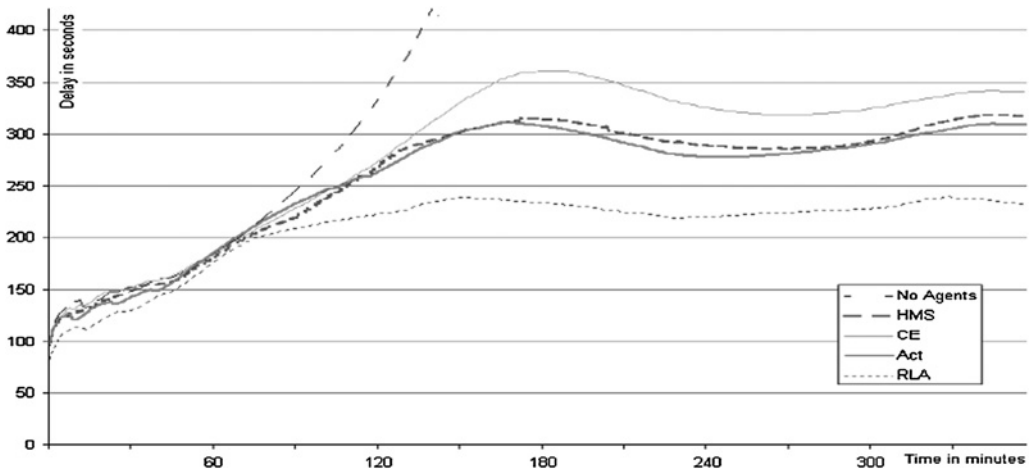

Figure 7 *Mean travel time delay comparison for 6-h two peak traffic scenario*

settling time and better adaptability to variations in the traffic demand over other traffic signal controls.

6.4 Response time and cycle length variation

The response time of the RLA signal control can be best illustrated with frequency of change in the phase length. Fig. 8*a* displays the length of each phase of a cycle for a four-phase intersection in middle of the network controlled by RLA signal control. The links having the right of way during the third phase have the lowest traffic demand and is reflected in the phase timing. The cycle time is lower in the non-peak period and dynamically varies with changing traffic pattern. It is not possible to compare the signal timings across the intersections because of the stochastic nature of traffic input and random seed initialisation. However, this can serve as an indication for the adaptability of the proposed RLA signal control.

6.5 Improvement because of learning

Reinforcement learning vastly improves the performance of the RLA signal control. Fig. 8*b* shows the variation of the

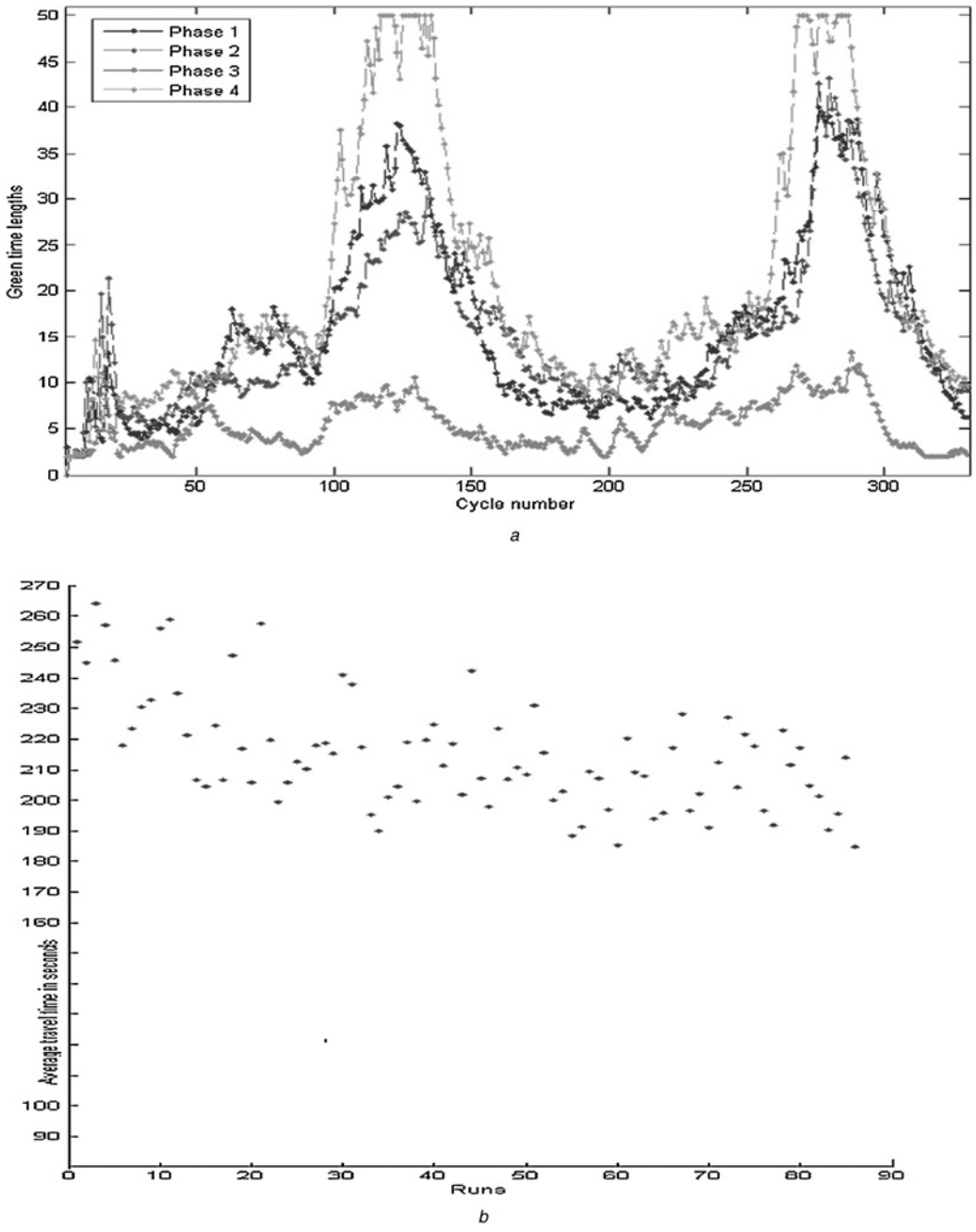

Figure 8 *Green timing and influence of reinforcement learning*

a Change in signal green time settings of an intersection
b Improvement in the average delay experienced due to reinforcement learning

doi: 10.1049/iet-its.2009.0096

Table 2 Worst and best time delay comparison of signal controls

		Pre-timed	HMS	CE	RLA	Act.	GLIDE
one peak	Nv	120	95	89	90	100	–
	V	37	46	44	49	45	–
	d	297	200	191	163	196	–
typical day	Nv	317	286	301	266	295	–
	V	35	43	38	48	44	40
	d	500	182	200	160	184	200
short extreme	Nv	Sat.	216	258	170	215	–
	V	0	35	36	48	42	10
	d	Sat.	315	340	232	309	650
long extreme	Nv	Sat.	250	–	206	205	–
	V	0	33	–	42	37	0
	d	Sat.	242	–	216	238	Sat.

Nv: number of vehicles at the end of the simulation, *V*: speed, *d*: mean delay

average value of mean vehicle delay experienced with each simulation run. After 90 continuous simulation runs, the mean delay value reduced to around 50% from what was at the start of the simulation run. HMS [21] utilises selective back propagation method for learning the parameters of neuro-fuzzy system. The back-propagation method has a limitation of getting into local optimum and therefore increases the time delay.

Table 2 shows the comparison of traffic data obtained using RLA signal control with all other benchmarks including GLIDE which is currently used in Singapore. Simulation model of GLIDE was obtained from [5, 21]. Average values obtained from 20 simulation runs are compared in Table 2. Standard deviation of delay was around 4 and 5–6% variation for vehicle count and speed. Proposed RLA signal control showed 9–15% improvement in performance when compared to other benchmarks in all the 20 simulation runs.

7 Conclusion

The proposed RLA signal control has a fully distributed architecture with agents capable of interacting with each other to effectively compute the optimal value of green time that reduces the overall travel time delay and increases vehicle mean speed. The update of traffic pattern in the repository and shared communication between agents increased the forecasting capability of each agent. This property of the agent effectively reduced the formation of congestion and improved clearance of vehicles at the intersection. Online Q-learning has been adopted to multi-agent scenario and Q-matrix was shared between agents to improve the local observations and create global view. Simulation tests conducted on a virtual traffic network of Central Business District in Singapore for four different traffic scenarios showed almost 15% improvement over the benchmark signal controls. Further improvements because of online reinforcement learning of the parameters have been demonstrated effectively.

8 Acknowledgment

This research work was supported by National University of Singapore under the research grant WBS: R-263-000-425-112.

9 References

[1] KOONCE P.: 'Traffic signal timing manual', US Department of Transportation FHWA-HOP-08-024, Federal Highway Administration, 2008

[2] SANCHEZ J.J., GALAN M., RUBIO E.: 'Genetic algorithms and cellular automata: a new architecture for traffic light cycles optimization'. Proc. Congress on Evolutionary Computation, 19–23 June 2004, Piscataway, NJ, USA, 2004, pp. 1668–1674

[3] HOAR R., PENNER J., JACOB C.: 'Evolutionary swarm traffic: if ant roads had traffic lights'. Proc. 2002 World Congress on Computational Intelligence – WCCI'02, 12–17 May 2002, Piscataway, NJ, USA, 2002, pp. 1910–1915

[4] ISHIHARA H., FUKUDA T.: 'Traffic signal networks simulator using emotional algorithm with individuality'. Proc. IEEE Intelligent Transportation Systems, 25–29 August, 2001, Oakland, CA, USA, pp. 1034–1039

[5] SRINIVASAN D., CHOY M.C., CHEU R.L.: 'Neural networks for real-time traffic signal control', *IEEE Trans. Intell. Transp. Syst.*, 2006, **7**, pp. 261–272

[6] HUNT P.B., ROBERTSON D.I., BRETHERTON R.D., WINTON R.I.: 'SCOOT – a traffic responsive method of coordinating signals' (United Kingdom, 1981)

[7] PECK C., GORTON P.T.W., LIREN D.: 'Application of SCOOT in developing countries'. Third Int. Conf. on Road Traffic Control, 1–3 May 1990, London, England, pp. 104–109

[8] SIMS A.G., DOBINSON K.W.: 'The Sydney Coordinated Adaptive Traffic (SCAT) system philosophy and benefits', *IEEE Trans. Veh. Technol.*, 1980, **t-29**, pp. 130–137

[9] LOWRIE P.R.: 'The Sydney Coordinated Adaptive Traffic System-principles, methodology, algorithms'. Int. Conf. on

Road Traffic Signalling, 30 March–1 April 1982, London, UK, pp. 67–70

[10] KEONG C.K.: 'The GLIDE system – Singapore's urban traffic control system', *Transp. Rev., Transnatl. Transdiscipl. J.*, 1993, **13**, pp. 295–305

[11] BAZZAN A.L.C.: 'A distributed approach for coordination of traffic signal agents', *Auton. Agents Multi-Agent Syst.*, 2005, **10**, pp. 131–64

[12] ROOZEMOND D.A.: 'Using intelligent agents for pro-active, real-time urban intersection control', *Eur. J. Oper. Res.*, 2001, **131**, pp. 293–301

[13] MIZUNO K., NISHIHARA S.: 'Distributed constraint satisfaction for urban traffic signal control'. Second Int. Conf. on Knowledge Science, Engineering and Management. KSEM 2007, 28–30 November 2007, Berlin, Germany, 2007, pp. 73–84

[14] DE OLIVEIRA D., BAZZAN A.L.C.: 'Traffic lights control with adaptive group formation based on swarm intelligence'. Ant Colony Optimization and Swarm Intelligence. Proc. Fifth Int. Workshop, ANTS 2006, 4–7 September 2006, Berlin, Germany, pp. 520–521

[15] CHOY M.C., CHEU R.L., SRINIVASAN D., LOGI F.: 'Real-time coordinated signal control through use of agents with online reinforcement learning'. Transportation Research Board Meeting (82nd), Washington, DC, 2003, pp. 64–75

[16] CAMPONOGARA E., KRAUS JR. W.: 'Distributed learning agents in urban traffic control', *Prog. Artif. Intell.*, 2003, **2902**, pp. 324–335

[17] WATKINS C., DAYAN P.: 'Technical note: ***Q***-learning', *Mach. Learn.*, 1992, **8**, pp. 279–292

[18] LITTLE J.D.C.: 'A proof for the queuing formula: $L = \{\text{lambda}\}\ W$', *Oper. Res.*, 1961, **9**, pp. 383–387

[19] 'Highway capacity manual – HCM2000' (Transportation Research Board, National Research Council, 2000)

[20] BALAJI P.G., SRINIVASAN D., CHEN-KHONG T.: 'Coordination in distributed multi-agent system using type-2 fuzzy decision systems'. IEEE 16th Int. Conf. on Fuzzy Systems (FUZZ-IEEE), 1–6 June 2008, Piscataway, NJ, USA, pp. 2291–2298

[21] CHOY M.C., SRINIVASAN D., CHEU R.L.: 'Neural networks for continuous online learning and control', *IEEE Trans. Neural Netw.*, 2006, **17**, pp. 1511–1531

[22] SRINIVASAN D., CHOY M.: 'Distributed problem solving using evolutionary learning in multi-agent systems', *Adv. Evol. Comput. Syst. Des.*, 2007, **66**, pp. 211–227

[23] CHOY M.C., SRINIVASAN D., CHEU R.L.: 'Cooperative, hybrid agent architecture for real-time traffic signal control', *IEEE Trans. Syst. Man Cybern. A (Syst. Hum.)*, 2003, **33**, pp. 597–607

doi: 10.1049/iet-its.2009.0096