Design and Evaluation of Impact of Traffic Light Priority for Trucks on Traffic Flow

FINAL REPORT

METRANS Project USC 2-1b

By

Petros Ioannou (Principal Investigator)
University of Southern California
Electrical Engineering – Systems, EEB 200B
Los Angeles, CA 900089-2562

June 15, 2015

Disclaimer

Abstract

Current traffic light control systems treat all vehicles the same. Trucks however have different dynamics than passenger vehicles. They take a longer distance to stop, have lower acceleration rates, have bigger turning rates that cause bigger traffic disturbances consume more fuel and pollute more. These dynamic differences create delays at intersections that affect the travel time of all vehicles involved and may have a negative effect on the environment if not taken into account by traffic control systems.

In this report we consider the problem of taking into account the presence of trucks in controlling the traffic lights at intersections with the objectives of minimizing delays for all vehicles involved and reducing pollution. The problem was motivated from the observation that in many cases it is better for all vehicles involved to extend the green cycle in order to have a heavy truck cross instead of forcing it to stop and restart again. The system is similar to the bus priority system that currently operates in many cities except that in the case of trucks the objective involves benefits for all vehicles than just the trucks involved. We proposed two different controllers for the truck signal priority problem: a neural network-based controller and one based on integrated priority strategies. The first controller is based on the use of a neural network approach to model the vehicle delays by distinguishing between different classes of vehicle and the use of optimization to reduce the vehicle delays by properly controlling the lights. The controller is adaptive since the delay prediction model is updated once new data is obtained from the infrastructure. The second controller is similar to the bus priority traffic light approach and uses a combined passive and active strategy in order to minimize delays. The similarity with bus priority is that it gives priority to trucks in certain situations if such an action benefits the overall system. In the case of bus priority the objective is to give priority to busses without optimizing for the delays of all vehicles involved. Both controllers have been evaluated using a microscopic traffic flow model for a single intersection and a network of intersections.

Table of Contents

List of Tables

List of Figures

Acknowledgments

We would like to thank METRANS for funding this project.

1 Introduction

Trucks have a detrimental impact on traffic flows, especially at intersections, because of their slow dynamics, large size and high emissions [1]. The time for a heavy truck to respond to a traffic light, accelerate and cross the intersection is much higher than that of normal passenger cars [2]. However, today's traffic lights do not take into account the presence of trucks but instead treat them as other vehicles for traffic light control purposes. With new sensor, communication and GPS technologies, a traffic light at an intersection could be informed of the approach of different vehicles and their characteristics (class, position, dynamics, speed etc.) [3-5]. The traffic signal controller could take into account the differences of vehicle dynamics between trucks and passenger cars in an effort to achieve better performances in terms of reducing traffic delays for all vehicles. For example under certain conditions it may be beneficial to all vehicles involved to give priority to certain vehicles, such as trucks, that take longer to decelerate and accelerate. The approach is similar to bus priority that is currently in use in many cities except that the objectives are different. In the case of busses the objective is to minimize the travel time or maximize the passenger throughput, whereas in the case of truck priority is to give priority to trucks in cases that all other vehicles will benefit too. Recent efforts with traffic signal control and priority systems, especially bus priority systems that give priority signal to buses [5-9] and adaptive signal control systems that adaptively respond to changes in traffic patterns [10-21], while giving priority to some special classes of vehicles.

There are two main priority strategies for control of signalized intersections: passive priority and active priority. Passive priority does not require an active communication between vehicles and signal controller and is implemented based on past knowledge of traffic flows and patterns, such as traffic volumes, approaching speeds, vehicle composition of every direction and turns. The passive priority system gives longer green time to priority directions [6][7]. Active priority requires the detection of approaching trucks and the subsequent priority request-response bidirectional communication between vehicles and signal controllers [5][8]. Reference [9] proposes a framework of integrating passive priority and active priority together to realize bus priority.

Adaptive Traffic Control Systems (ATCS) are traffic management systems, which adjust the timing of traffic signals to adapt to changing traffic patterns and ease traffic congestion. In such systems, a performance index (PI), e.g., overall delay, number of stops, queue lengths, fuel consumption or a combination of these parameters, is minimized [10]. ATCS optimize traffic flow on arterial networks with multiple signalized intersections. SCOOT [11] and SCAT [12] are prominent and well established ATCS. Systems such as MOTION [13] and BALANCE [14] are good examples of ATCS. Improved traffic modeling techniques developed in the recent years together with the increase in computation power encourage developments of more sophisticated ATCS. SCOOT minimizes the average queues by adjusting the signal timings and continuously measuring traffic volumes. The potential timing plans are evaluated heuristically to adjust the signal timings. Both SCOOT and SCAT suffer from inefficient handling of saturated conditions due to inadequate real-time adaptability [15]. LHOVRA[16,17], OPAC (Optimized Policies for Adaptive Control)[18][19], and RHODES (Real-time Hierarchical Optimizing Distributed Effective System)[20] are other examples of traffic light controllers. LHOVRA can support limited priority functions in isolated intersections based on road vehicle detectors. The OPAC system could give priority to certain vehicles, such as emergency vehicles, if they are operating on restricted lanes. RHODES implements traffic signal control with a MPC (Model Predictive Control) methodology and the priority is realized by giving weights to different vehicles. Traffic-responsive urban control (TUC) is another adaptive control strategy [21]. Based on a store-and-forward modeling of the urban network traffic and using linear-quadratic regulation theory, the design of TUC leads to a multivariate regulator for traffic-responsive, coordinated network-wide signal control that is also particularly suitable for saturated traffic conditions. Real-time decisions in TUC cannot be taken more frequently than at the maximum employed signal cycle. The strategy will need to be redesigned in the case of modifications and expansions of the controlled network. TUC was compared with a fixed-time signal control and shown to lead to reduction in total waiting time and total travel time in the system. Many optimization and intelligent control algorithms have been used in these traffic light control systems, including fuzzy logic [22], neural network [23], cell transmission model [24][25], dynamic programming [26], Genetic Algorithm (GA) [27][28] and Q-learning [29][30]. The optimization techniques [26-30] involve a search for the optimal signal sequence using traffic flow information and can be used for truck priority systems.

In a large network, the isolated traffic light control of each intersection without taking into account the traffic situation in the network and how other traffic light intersections are controlled may lead to unnecessary congestion in the network. Therefore, in order to optimize the traffic flows effectively, we need to consider the traffic situation and state of the adjacent intersections for each individual intersection. The dependency of traffic volumes at each intersection on its neighbors makes it difficult to set the signal timings for a large traffic network with multiple intersections. An interesting approach to deal with this traffic signal control problem is to use a distributed control technique involving multiple agents. The goal of a multi-agent control system is to reduce the traffic congestion for multiple intersections simultaneously. For effective traffic signal control, such controllers need to adapt themselves continuously. De Oliveira and Camponogara [31] proposed a network of distributed agents to control linear dynamic systems which are put together by interconnecting linear subsystems with local input constraints. The framework decomposes the optimization problem obtained from the model predictive control approach into a network of coupled and small sub-problems to be solved by the agent network. Each agent senses and controls the variables of its intersection, while communicating with agents in the neighborhood to obtain variables and coordinate their actions. The proposed approach achieved performance comparable to the TUC system. A real-time traffic controller is proposed in [32] using a distributed network of agents. The online learning and update process for each agent is improved by designing a stochastic cooperative parameter update algorithm. In another study [33], a collaborative reinforcement learning (RL) algorithm is employed using a local adaptive round robin phase switching model at each intersection. Each intersection collaborates with adjacent agents in order to learn appropriate phase timings. In [34], a multi-agent RL was designed to optimize traffic signals at multiple intersections. The RL systems are trained using the waiting times for vehicles and different settings of the traffic signals. The results presented in [35] show that the proposed algorithm outperforms non-adaptive traffic light control systems..

In this report, we extend the concept of bus priority and adaptive signal control system techniques to traffic light control with truck priority. In the case of bus priority, the system objective is to reduce delays of buses at signalized intersections irrespective of the traffic in opposite directions whereas the truck priority system is motivated by the objective of reducing the overall traffic delay and environmental impact. In contrast to bus priority, the truck priority faces the following challenges: 1) The arrival frequency of trucks is much higher than that of buses especially in the neighborhoods of cargo ports, warehouses, markets, etc.; 2) The trucks do not have relatively fixed schedules as buses. Therefore, the truck arrival time for one intersection cannot be accurately predicted until it enters the controlled road network and its route has been assigned; 3) The priority level of trucks is lower than that of buses, allowing the flexibility to consider the impact to traffic in other directions.

With the advancement of drayage optimization and real time routing as well as connected vehicle technologies the accuracy of truck arrival times at intersections will improve.

We propose a truck priority system that integrates passive priority and active priority request-response strategies to give passing priority to trucks without causing additional delays to traffic in other directions. Passive priority strategy provides an optimized baseline signal for active priority strategy with a simulation-based optimization algorithm using detected or historical traffic information such as vehicle flow volumes, speeds and compositions. On the other hand, the active priority strategy responds to real time priority requests from approaching trucks and decides to grant or refuse the requests based on traffic conditions. By timing the traffic signals to give priority to trucks when trucks are present, we can achieve two benefits. First, the trucks will clear the intersection faster without having to make frequent stops that introduce additional delays due to their stopping and acceleration time. Second, the less decelerations/accelerations a truck goes through the less fuel it consumes and the less pollution it generates. Such traffic light priority may also have a beneficial effect on the travel time of passenger vehicles due to elimination of delays caused by trucks stopping and going. Our analysis of the proposed systems reveals these benefits and possible tradeoffs.

This report is organized as follow. Section 2 introduces the differences of characteristics between trucks and passenger cars. Section 3 proposes the design of our truck priority systems at signalized intersections. Section 4 shows the evaluation results of proposed systems on a practical road network. The conclusions are given in Section 5.

2 Vehicle Characteristics at Traffic Lights

In traffic engineering, the allowable lengths of vehicles can be significantly different. The length of a passenger car is usually about 20 ft while the truck length is from 35 to 80 ft [1], i.e., the length of a truck is about 1.5 – 4 times the length of a passenger car. Table I shows the acceleration rates of a typical passenger car and truck [2]. As shown in the table, the acceleration of a typical truck is much lower than that of typical passenger car. Therefore, a truck requires a longer time to resume its full speed after stopping and more deceleration distance to stop before a red light at intersections in comparison to other vehicles with higher deceleration and acceleration capabilities, like cars (as shown in Fig. 1). The traffic delay generated by truck stops is much larger than the delay of same number of car stops due to slow dynamics of trucks.

Table 1: Acceleration rates of typical car and truck [2]

Speed Range (mph)	**Acceleration Rates (ft/sec^2)**	
	Passenger Car	***Typical Truck***
0 – 20	7.5	1.6
20 – 30	6.5	1.3
30 – 40	5.9	0.7
40 – 50	5.2	0.7
50 – 60	4.6	0.3

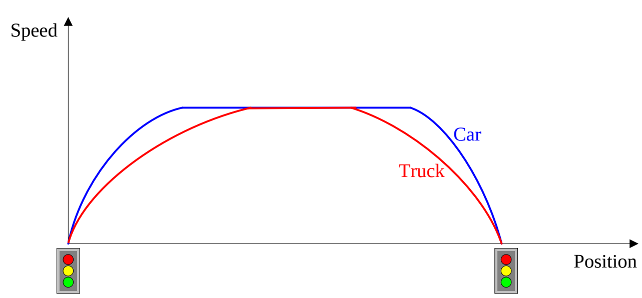


Figure 1: Comparison of Truck and Car Dynamics

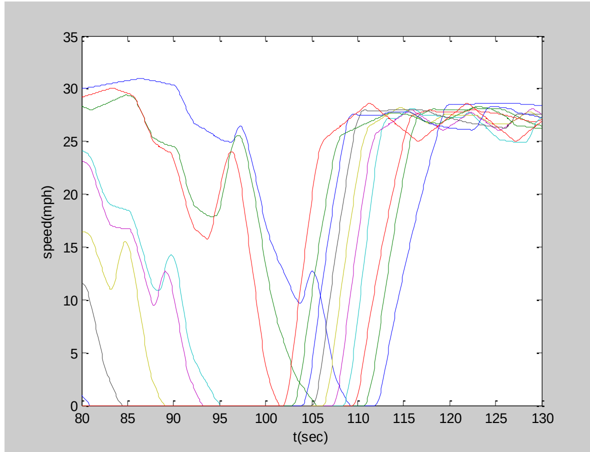


Figure 2: All car flow at an intersection

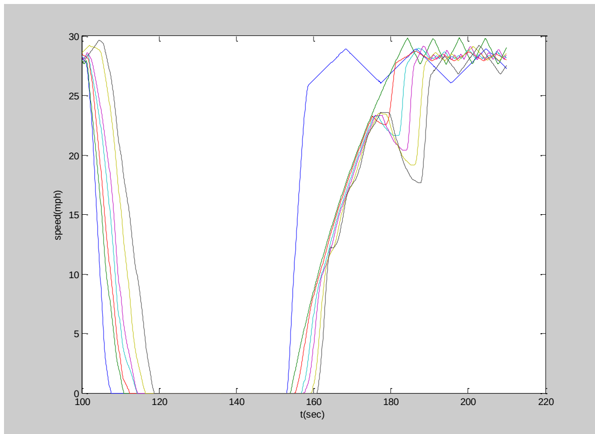


Figure 3: Flow with a truck at an intersection

If cars stopped behind a truck at a red traffic light, their acceleration and speed when the traffic light becomes green, is dictated by the slow dynamics of the truck. Their lower speed will increase the delay that is the time it takes to clear the intersection. By giving priority passing to the leading truck we can reduce the entire traffic delay not only for the truck but also for the following passenger cars. In general, by reducing the number of stops of trucks at traffic lights not only reduces the traffic delays for all cars involved but also has a positive impact on the environment since trucks produce much more air pollution than normal cars especially during stop and go traffic. Fig. 2 and Fig. 3 show two groups of speed profiles of an all-car queue and a queue with a truck (second position) before at a traffic signal respectively.

The headway of same type of cars in a waiting queue can be expressed by the following model presented in [36],

$$T_n = \sum_{i=1} \Delta_i + nh \tag{1}$$

where T_n is the green time required to move a queue of *n* cars through the intersection in sec;

Δ_i is the incremental headway of *i*th car, that is the additional headway due to driver reaction to green signal and vehicle acceleration. $\Delta_i = 0$ when $i \geq 5$, sec ;

h is the saturation headway, that is the average headway that would be achieved by a saturated and stable moving queue of vehicles passing through the intersection if the signal was always green, sec.

n is the number of cars.

However, equation (1) will not be accurate due to the presence of a truck in the queue since the truck has an impact on the headways of the following vehicles due to its slow dynamics. Considering the above simulation results, we conclude that avoiding a truck appearing in the front of a stopping queue before an intersection is an efficient way to reduce travel delays for all vehicles involved.

3 Proposed Truck Signal Priority Systems

In this report we design, analyze and evaluate two different approaches for traffic light control with truck priority. The first method is based on the use of a neural network system to predict delays and an optimization method to minimize these delays by generating the appropriate traffic light signal sequence. The second method combines a passive with an active approach and uses real time simulations together with an optimization technique to generate the signal sequence. The two methods are presented in the following subsections and tested and evaluated in section 4.

3.1 Approach 1: Neural Network Based System

3.1.1 System Architecture

We propose a multi-agent distributed traffic signal controller, which incorporates the truck data into the optimization problem. We first develop the delay predictor model. This neural network based model predicts very short-term delays of all the vehicles in the network based on the information of the cars and trucks and also information obtained from neighboring signals. In the next step, we develop an algorithm to optimize the traffic delay (predicted by the neural network). This algorithm optimizes the next transition time of traffic signals that minimizes the delay for each intersection by considering the state of the adjacent intersections and, therefore, minimizes the overall delay of the traffic network. Fig. 4 shows the block diagram of the control scheme. A neural network is used to estimate the states and delays associated with the traffic lights at each intersection by taking into account different classes of vehicles. The estimated or predicted delays are used by an optimization algorithm to generate the control strategy for timing the traffic lights in order to minimize the delays.

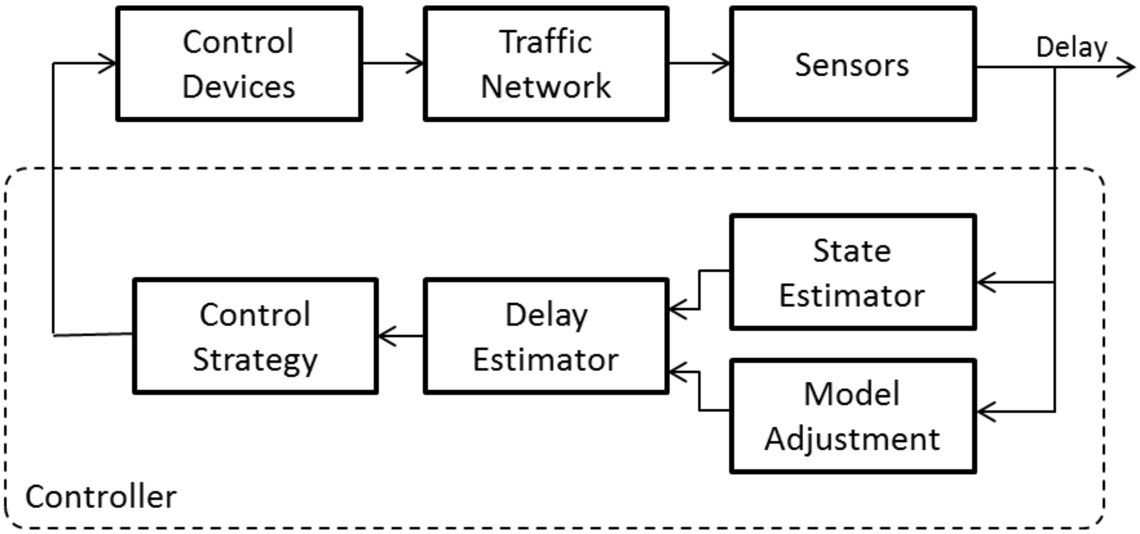


Figure 4: Block Diagram of the control scheme.

The details of the approach are provided in the following subsections.

3.1.2 Traffic Network Model

We first start with the single intersection, and expand the model to multiple intersections. The delay is defined as the average of differences between the actual travel time and the free flow travel time with no stops or slowing (at maximum allowed speed) for all vehicles in the network. Delay is then predicted using a neural network model. For a single intersection we do not consider the state of the adjacent intersections, and the controller is the result of a convex optimization problem. Fig. 5 illustrates a single intersection. By feeding the information (average speed and number) of trucks and cars as two separate sets of inputs we guarantee that the extra delay contributed by trucks is considered in the model.

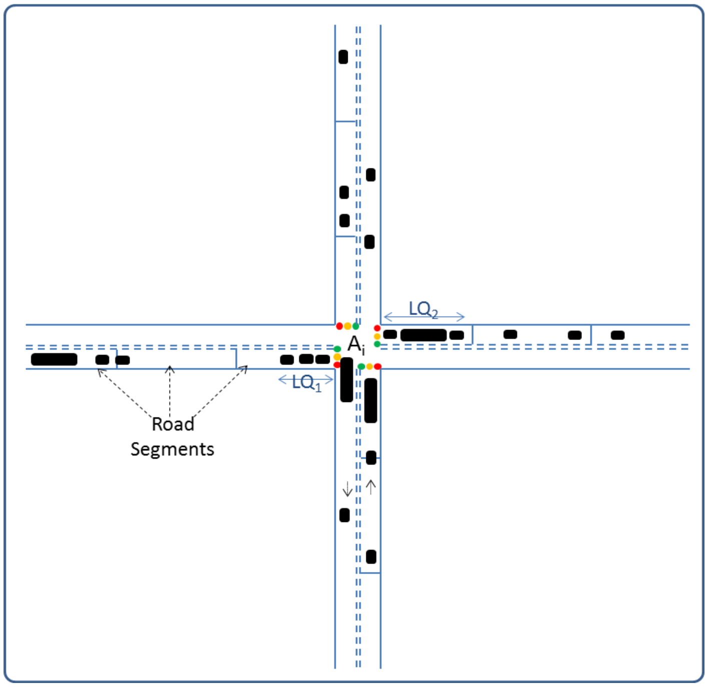


Figure 5: Schematic showing a single intersection

The subproblem of controlling a traffic signal in an intersection is handled by an agent. For each intersection we use a single layer neural network (NN) to predict the average delay of all vehicles. Each road link is divided into a number of segments (Fig. 6), number and average speed of trucks and cars for each road segment, the length of queues at each link, current status of traffic lights as well as the future state of the traffic lights are fed into the neural network model. The inputs to the NN are as follows:

NTj: Number of trucks in segment *j*

VTj: average speed of trucks in segment *j* (km/h)

NCj: Number of cars in segment *j*

VCj: average speed of cars in segment *j* (km/h)

LQi: length of queue at link I (m)

Si: current state of traffic signal (green/red)

S': future state of the traffic signals (transition time of signals in seconds)

Fig. 6 shows the structure of the neural network used to model the time delay:

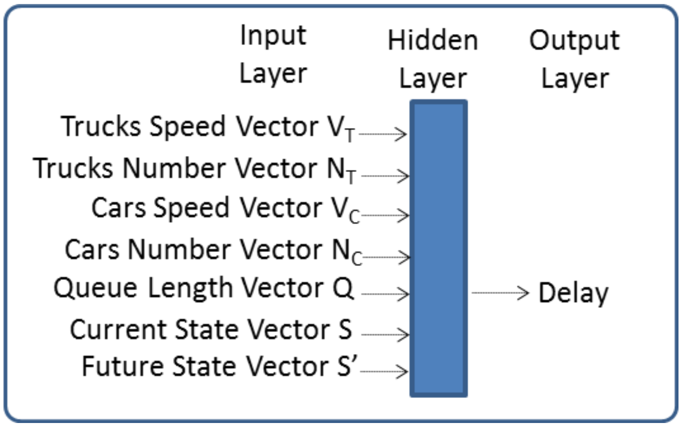


Figure 6: Structure of the neural network model to predict the overall delay of the network based on the vehicles information, current state of the traffic signals and future state of the traffic signals.

The delay is modeled as:

$$D_d^k = f^k(S^k, R_d^k) = \sum_{j=1}^{N} a_j^k y\left({\boldsymbol{w}_j^k}^T S\right) \quad (2)$$

where D_d^k denotes the delay of all the vehicles for the next time period d at time step k, S^k denotes the input vector at time step k defined as $S^k = [V_T, N_T, V_C, N_C, Q, S]$, and R_d^k is the future transitions of signal at time step k for the same time period. N denotes the number of hidden units in the Neural Network, $\boldsymbol{w}^k = [\boldsymbol{a}_1^k, \dots, \boldsymbol{a}_N^k, \boldsymbol{w}_1^k, \dots, \boldsymbol{w}_N^k]$ is a $1 \times N_w$ vector which consists of the weights of the NN, $S = [S^k, R_d^k]$ is the input vector, and $y(v) = (1 + e^{-v})^{-1}$ is the logistic function. The weights $\boldsymbol{w}_k$ are updated online by backpropagation using the gradient descent method [19]:

$$w_i^{k+1} = -\nabla J^{k+1} + \alpha^k w_i^k \quad , \quad i = 1, \dots, N_w \quad (3)$$

where J^{k+1} is the new performance surface and α^k is the dynamic learning rate. Traffic light control relies on finding the arguments $R_d^{k*} = \mathrm{argmin} D_d^k$ which minimizes the delay with respect to constraints on minimum and maximum green light cycles. We describe the solution methodology in the next section after introducing the multi-intersection scheme.

Expanding the single-intersection model involves introducing new inputs to the delay model. As explained in the introduction, to control each signal efficiently, we need information from adjacent signals in addition to the local traffic data. Figure 7 illustrates the NN with the new parameters.

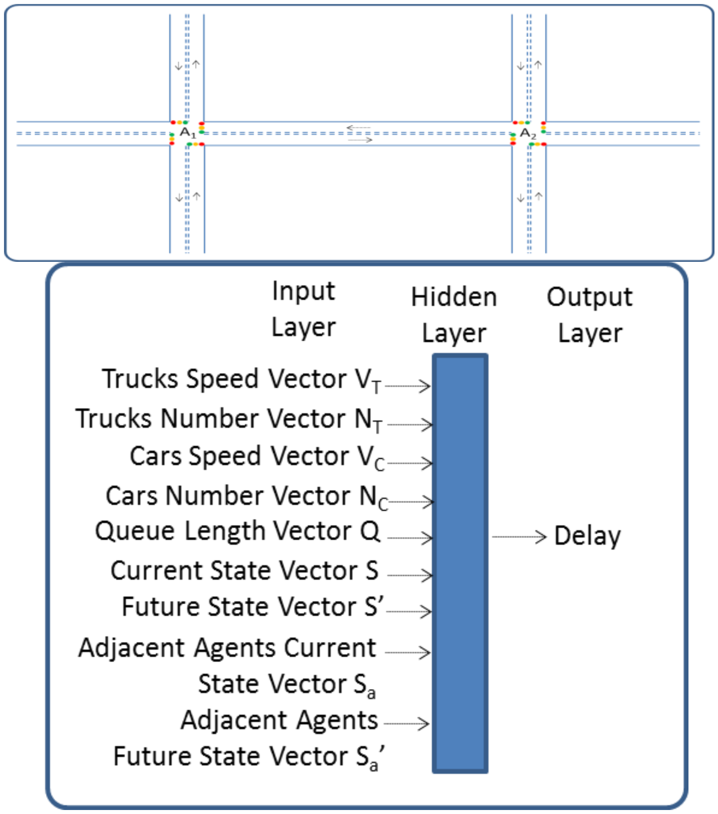


Figure 7: Schematic and neural network structure for predicting delay in the multi-intersection model. Information from adjacent intersections are fed into the network in addition to the local traffic data.

The training of the NN is similar to the single intersection. After the model is trained, a prediction of the model can be made using:

$$D_{i,d}^k = f_i^k\left(S_i^k, S_{i,a}^k, R_{i,d}^k, R_{i,a}^k\right) = \sum_{j=1}^{N} a_{ij}^k y\left(\mathbf{w}_{ij}^{k\,T} S\right) \tag{4}$$

where the index i denotes the intersection and the subscript a means the variable belongs to the adjacent intersections.

3.1.3 Controller Design and Control Algorithm

At each time step, agents decide about the next few transitions of the controlled signal. The decision involves finding the argument to minimize the sum of delays of all intersections simultaneously:

$$J = \min_S \sum_{i=1}^{I} f_i^k\left(S_i^k, S_{i,a}^k, R_{i,d}^k, R_{i,a}^k\right)$$

$$s.t.\;\; v_i \le g_i \le u_i \tag{5}$$

where v_i and u_i denote the minimum and maximum green/red time, g_i denotes the green time for each route, and *I* is the total number of intersections. Since $R_{i,d}$ is the information vector obtained from infrastructure and known to the controller at the decision making time we can rewrite equation (5) as,

$$J = \min_S \sum_{i=1}^{I} f_i(S_i, S_i^a)$$

$$s.t.\;\; v_i \le g_i \le u_i \tag{6}$$

Since the neural network model is a sum of sigmoidal functions, we have:

$$f_i(S_i, S_i^a) = \sum_{j=1}^{N} a_{ij} y(w^T S) \tag{7}$$

where $S \triangleq \begin{bmatrix} S_1 \\ \vdots \\ S_I \end{bmatrix}$ is the state vector of the whole network, and $y(v) = (1 + e^{-v})^{-1}$ is the logistic function. Note that $S_i \triangleq [s_{i1} \quad \cdots \quad s_{iN_s}]^T$. By defining $g_{ij}(.) \triangleq a_{ij} y(.)$ the optimization problem becomes:

$$J = \min_S \sum_{i=1}^{I} \sum_{j=1}^{N} g_{ij}\left({w_{ij}}^T S\right)$$

$$s.t.\;\; l_{ij} \le s_{ij} \le u_{ij} \tag{8}$$

where *I* denotes the total number of network agents, and *N* is the number of hidden units for each agent. At each time step the optimum set of signal transition times is obtained by solving the above optimization problem. Equation (8) is a linear constrained general nonlinear optimization problem which consists of a sum of sigmoidal functions. The functions are not separable in the current form, and thus, solving it is not a trivial task. By using the following linear transformation we obtain a set of separable functions.

$$Z = WS \tag{9}$$

where $Z \triangleq [z_{11} \quad \cdots \quad z_{1N} \quad \ldots \quad z_{I1} \quad \cdots \quad z_{IN}]^T$ denotes the new state vector, and $W \triangleq [w_{11} \quad \cdots \quad w_{1N} \quad \ldots \quad w_{I1} \quad \cdots \quad w_{IN}]^T$ is a $IN \times IN_s$ matrix, where N_s is the number of states for each agent. Then the problem becomes:

$$J = \min_{S} \sum_{i=1}^{I} \sum_{j=1}^{N} g_{ij}(z_{ij})$$

$$s.t. \;\; l'_{ij} \leq z_{ij} \leq u'_{ij} \tag{10}$$

which is a separable nonlinear optimization. The final set of optimization problems reduces to:

$$J_{ij} = \min_{S} g_{ij}(z_{ij})$$

$$s.t. \;\; l'_{ij} \leq z_{ij} \leq u'_{ij} \;,\; i = 1, \ldots, I \,,\; j = 1, \ldots, N \tag{11}$$

where l'_{ij}, u'_{ij} are the new sets of constraints obtained by the linear transformation. Note that $g_{ij}(z_{ij}) = a_{ij} y(z_{ij})$, and, depending on the sign of a_{ij}, $g_{ij}(z_{ij}) = a_{ij} y(z_{ij})$ is a strictly increasing or strictly decreasing function, given $a_{ij} > 0$ or $a_{ij} < 0$ respectively, since $y(v) = (1 + e^{-v})^{-1}$ is a strictly increasing function. Therefore,

$$z^*_{ij} = \begin{cases} l'_{ij} & a_{ij} > 0 \\ u'_{ij} & a_{ij} < 0 \end{cases} , \begin{array}{l} i = 1, \ldots, I \\ j = 1, \ldots, N \end{array} \tag{12}$$

where z^*_{ij} is the optimum transformed state vector. The actual state vector is obtained by,

$$S^* = W^{-1} Z^* \tag{13}$$

In order to have a unique and invertible transformation the number of states (future transitions of signals) must be equal to the number of hidden units for each agent. In this case, W is a square matrix with non-zero elements on the diagonal. As discussed in the results section the accuracy of the delay prediction improves as the number of hidden units increases. However, there is a trade-off between the accuracy of prediction and the processing time. Our simulation results discussed in the results section show that the number of hidden units between 5 and 11 is a good choice in terms of both accuracy and processing time.

The control procedure of the whole network consists of the following steps:

Step 1: At time step k, generate the control vector S^k by solving the optimization problem (8).

Step 2: modify the control vector by taking a weighted average of the current and previous decisions:

$$S^{k*} = \sum_{i=1}^{p} \beta_i S^{k-i} \ , \ 0 < \beta_p < \cdots < \beta_1 < 1 \tag{14}$$

Step 3: apply the modified control vector S^{k*} to the signals in the network.

Step 4: update the weights of the model (4).

Step 5: return to step 1 at time step k+1.

By applying the above control algorithm, at each time step, the future transitions of the signals are updated by progression of time. Fig. 8 shows an example of signal timing progression generated by simulation:

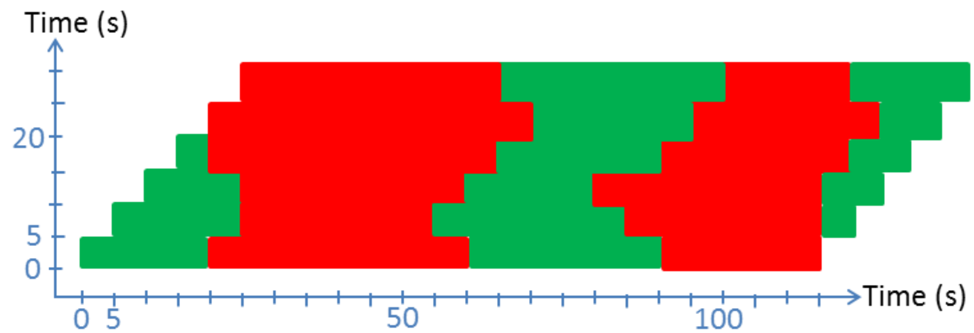


Figure 8: Sample results for traffic light transitions at a junction using the above control strategy.

The horizontal axis of Figure 8 shows the planned transitions for the next 2 minutes which are made at time 0 based on the traffic state observed at the moment. The vertical axis shows the times at which the 2 minutes ahead predictions are updated. For example at time step 5 (vertical axis), based on the developments in the traffic situation, the transition plan is updated which is illustrated on the second horizontal row. The adjustments at each decision time are made by taking a weighted average of the 3 previous decisions to prevent abrupt changes to the plan.

3.2 Approach 2: Integrated Truck Priority Strategies System

3.2.1 System Architecture

There are two main priority strategies at signalized intersections: passive priority and active priority [5] [8]. Passive priority does not require an active communication between vehicles and signal controller and is implemented based on past knowledge of traffic flows and patterns, such as traffic volumes, approaching speeds, vehicle composition in all directions and turns. The passive priority operation is realized by giving longer green time to the directions that have larger volumes or more trucks in order to improve traffic condition, reduce total delay and the number of stops as well as environmental emissions. Active priority requires the detection of approaching trucks and the subsequent priority request-response bidirectional communication between trucks and signal controllers. Fig. 9 gives an overview of the proposed truck priority system. The system architecture has the following modules:

- Traffic flow observer; This module works in the network level to observe the state of the controlled road network for priority modules with detection techniques and active communication between vehicles and signal controllers. It provides traffic flow information for two priority modules. This module provides the traffic flow information for passive priority module (link traffic flow volumes, compositions, etc.) and detects the priority requests and queue lengths at the controlled intersections for active priority module.

- Passive priority module; This module works in the multi-intersection level. It generates optimized baseline signal plan for the active priority module using the observed traffic flow information.

- Active priority module; The module works in the intersection level under the assumption that there is a continuous communication between trucks and signal controllers. This module receives the priority requests from approaching trucks then makes priority decisions based on the baseline signal and real time traffic conditions.

- Signal generation module; outputs operation signal timing based on the baseline signal and priority decisions.

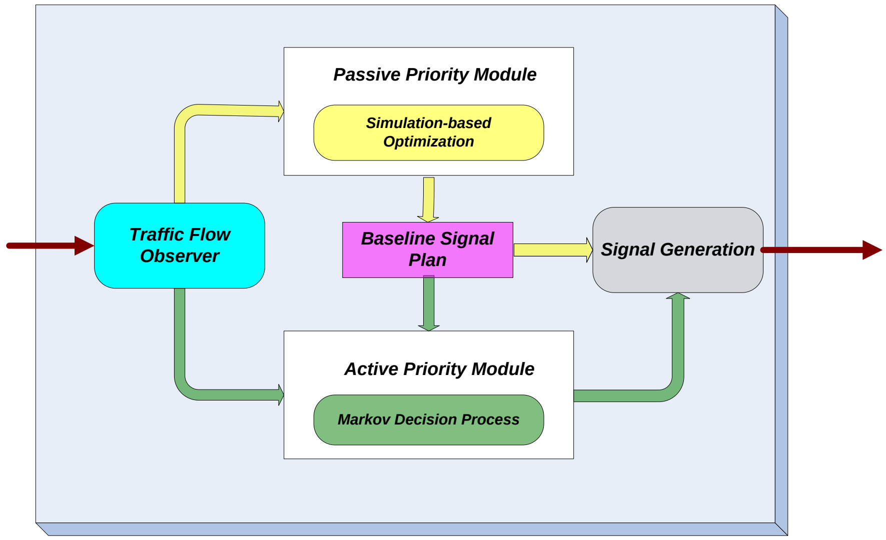


Figure 9: Architecture of proposed truck signal priority system.

3.2.2 Passive Priority Module Design

The passive priority module provides an optimized baseline signal to improve given performance criteria. MPC (Model Predictive Control) approaches have been applied in many practical systems such as RHODES and OPAC [16-20]. A MPC based intersection signal optimization approach needs a traffic flow model that makes predictions of future traffic state, an optimization technique to find optimal signals for intersections under required constraints, and performance criteria to improve [16][18][20]. However, it faces many difficulties. First, the traffic flow model is hard to formulate accurately because the traffic on a road network having signalized intersections is nonlinear and time variant. Moreover, the performance criteria such as traffic delay, number of stops, or environmental impact are not explicit functions of traffic flows, making performance predictions difficult. As a result, the control performance is limited by traffic model formulation and non-explicit objective functions.

We propose a simulation based control approach to find the optimal baseline signals for active priority module based on continuously measuring road traffic flows as shown in Fig. 10. In this approach, the traffic simulator is used to predict future traffic states and performance criteria instead of mathematical models used in MPC approaches. This approach has several advantages: 1) the traffic simulator can predict traffic flows accurately in a fast forward manner given estimated traffic demands; 2) the performance costs are easily obtained from the simulation results. However, this simulation-based approach faces a time complexity problem because of the required traffic simulation time and becomes computationally intractable when the size of the controlled road network increases. In order to handle the scalability and computation problem the controlled road network is divided into a group of subnetworks of much lower complexity and a multi-agent simulation based control technique is used to solve the optimal problem.

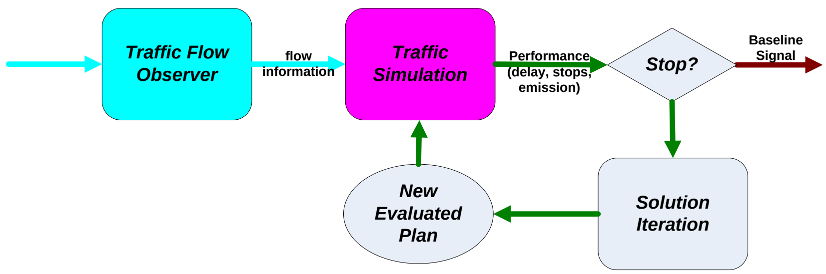


Figure 10: Simulation-based passive priority

Consider a road network with n agents. Let X_t^i $i=1,...,n$ denote the local traffic state of subnetwork i at time step t. Then $X_t := (X_t^1, X_t^2, ..., X_t^n)$ denotes the local states of all agents that is also the global state of the road network at time t. Let U_t^i, $i=1,...,n$, denote the control input of control station i, i.e., the signal settings of all signalized intersections in the agent area. $U_t := (U_t^1, U_t^2, ..., U_t^n)$ denotes the collection of all control inputs at time t. During the control process, each agent controller i observes the global network state X_t by combining its local state and states of other agents, as well as the one-step delayed control inputs U_{t-1} of all agents. Then each local agent controller i generates the optimal control input U_t^i based on the state prediction generated by the traffic simulator.

The agent traffic states from t to $t + p$ are predicted using the traffic simulator and the optimal control input is generated by optimizing the cost function of performance criteria where p is the prediction time step. In summary, the problem is formulated as follows:

$$X_{t+1}^i = f^i\left(X_t^i, U_t^i, U_{t-1}, W_t\right) \tag{15}$$

$$U_t^i = \arg\min_{U_t^i} c^i\left(X_{t:t+p}^i, U_{t:t+p}^i\right) \tag{16}$$

and

$$c^i\left(X^i_{t:t+p}, U^i_{t:t+p}\right) = \omega^T P^i_{t:t+p} \tag{17}$$

where f^i is the traffic simulator function of agent i; W_t denotes the simulation disturbance at time step t; $c(.)$ is the optimization cost function; ω is the weight of performance criteria and P^i is the simulated performance of agent i.

The proposed method can accept any quantifiable performance measures that could be generated by simulation results. In this report we use the sum of 1) Average delay of all vehicles/cars/trucks (with weights one) and 2) Average stop frequency of all vehicles/cars/trucks (with weights ten) as the cost function value to improve.

Another problem is to select an optimization algorithm that searches new evaluated signal input and monitors the convergence of the optimization process to solve problem (16). A wide class of search algorithms could be applied to determine the optimal signal input. However, the gradient-based algorithms are not suitable here because the simulation module can be seen as a black-box function since the explicit objective function and its derivative are unavailable. There are several kinds of algorithms that can be used for this problem.

(1) Trajectory search family algorithms: Typical methods include pattern search method [37][38], simulated annealing, Tabu search, hill climbing method, etc. These algorithms find a satisfactory solution by iteratively using a local or neighborhood search procedure that moves from one potential solution to an improved neighbor solution. These algorithms are derivative-free and easy to implement even for complex problems. However, they may be attracted by local optima and therefore can not give exact optimal solution in some cases.

(2) Population-based family algorithms: Typical methods include Genetic Algorithm (GA), Genetic programming, Evolutionary programming, Ant colony algorithm, bee colony algorithm, etc. These algorithms do not get stuck at local optima but they can not guarantee that they will find the global optimum. Many papers show the applications of these algorithms in traffic control and optimization. [27] and [28] proposed two typical transit priority systems using GA. However, they need a large number of evaluations and searching rounds for one converging procedure. For example the minimum execution time is 4.29 hours even when using a parallel structure GA algorithm for an 86-intersection network [27] and the total number of evaluations is more than

4000 for a 12-intersection road network [28]. Therefore, evolutionary algorithms may not have a good performance if the number of evaluations or the required searching time is limited.

(3) Q-learning algorithms such as [29][30]. Q-learning algorithm generates new actions based on observation of old action performances and finally learns an optimal mapping between system states and control actions. Q-learning method is powerful when the traffic demand and network flows are static but not useful for dynamic traffic flows.

Initialization at *k* = 0:

1) Obtain the initial solution x_0;
2) Select algorithm parameters:
   M, $\lambda \in [1, \infty)$, $\theta \in (0,1)$, and ε.

For each time step *k*, do:

1) Receive the latest optimal control inputs from other agents in simulation;
2) Determine new possible search direction $d = \left(d^1, d^2, .., d^n\right)$ with GA algorithm;
3) Run simulation to evaluate the performance of candidate solution with received inputs from other agents, get $f(x_k)$;
4) For each search direction $i = 1, 2, ..., n$:

   Compute directional search step $s_k^i = \Delta_k d_k^i$;

   Evaluate $f(x_k + s_k^i)$ if $x_k + s_k^i$ is a feasible solution;

   End for

5) Select $s_k = \arg\min_{s_k^i} f(x_k + s_k^i)$;
6) If $f(x_k + s_k) \geq f(x_k)$

   Update step size $\Delta_{k+1} = \theta \Delta_k$ and $x_{k+1} = x_k$

   Else

   Update $\Delta_{k+1} = \lambda \Delta_k$ and $x_{k+1} = x_k + s_k$;

   Send x_{k+1} to other agents;

7) If $\Delta_{k+1} \leq \varepsilon$ or $k = M$, stop algorithm, return x_{k+1};

End for

Figure 11: Simulation-based optimization algorithm

Based on the above methods, we apply a simulation based search algorithm in Fig. 11 to find the optimal input U_t^i . The algorithm is implemented by adjusting current solution to some directions and then evaluating new solutions via simulations. The directions include adding/reducing green ratio of a phase, adding/reducing signal cycle, adding/reducing signal offsets, etc. A pattern search algorithm searches feasible decreasing directions and a GA algorithm generates new possible feasible directions based on the found feasible directions with better performance. If such feasible directions exist, the algorithm moves to new solutions in those directions and repeats the search process with increased step size. At the same time the agent will broadcast a message of its latest optimal input to other agents whenever a better solution exists for this agent. Otherwise, it repeats the search from current solution with decreased step size until reaching the objective solution when the step size is less than a predefined threshold value or the iteration number achieves an upper bound *M*. Since the objective function is nonlinear and non-convex, the global optimal solution is not guaranteed. The parameters in the algorithm are: the prediction time step p = 20 minutes; M = 10, $\lambda = 2$, θ=0.5, and $\varepsilon = 2$.

3.2.3 Active Priority Module Design

Fig. 12 illustrates the procedure of the active priority module that has the following stages:

(1) Priority Request and Arrival Time Prediction

When a truck approaches an intersection, it reports its arrival by sending information that includes type, length, speed, etc. to the signal controller. After the signal controller receives the request, it predicts the arrival time of the truck. This arrival time is an important factor used to determine whether the truck needs a priority action. If the truck could report its dynamics continuously via active communications before it arrives at the decision position the controller could update the arrival time of the truck continuously and make a more accurate decision. If the traffic light is green when the truck arrives then it does not need a priority passing.

(2) Priority Action Determination

If the truck satisfies the requirements of a priority passing, the next stage is to determine which priority action is needed. The possible priority actions may be green extension, early green or phase insertion. This determination depends on the truck arrival time and the future traffic light state on the truck arrival time. For example, a priority request occurred during phase 1 and this phase 1 is also the phase for the approaching truck to pass the intersection. But the arrival time of the truck is outside the green interval of phase 1. In this case, the controller needs to extend the green interval for the current phase 1 (case 2 in Fig. 13) in order to give priority for the truck. Similarly, if phase 2 is the phase for the truck however the arrival time is ahead of the starting time of that phase's green state, then the controller needs to start the green state earlier than planned in order to give priority passing (as case 3 in Fig. 13). As shown in Fig. 13 giving priority for trucks in one phase or direction has a negative impact on traffic flows in the next phase. Therefore, both the green extension time and earlier green time should be limited within a threshold time. This minimum threshold time is determined by the time required for a truck to travel from the priority request position to the intersection stop line. In addition, the threshold time is within the value of 10% cycle time which can be altered to support priority operations. [38]. For these reasons in this report the threshold time is chosen to be five seconds since the distances between the priority request positions to the intersection stop line are about 30-50 meters in the evaluation traffic model. This threshold time is enough for a truck to travel the distance with normal speed and does not have negative impact on traffic flows of non-prioritized phases.

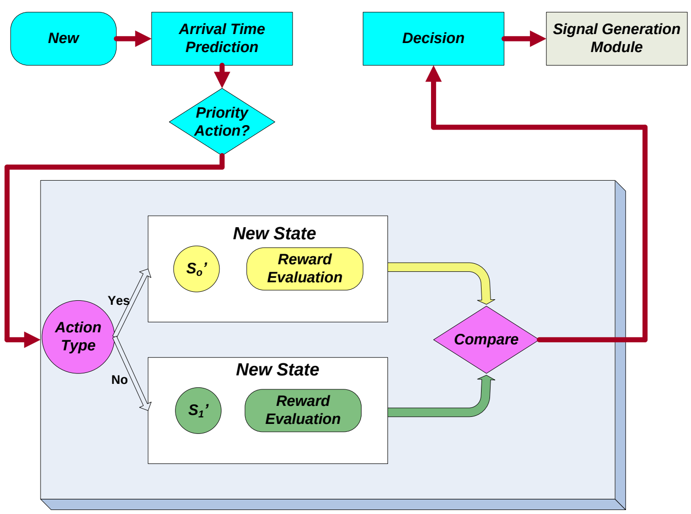
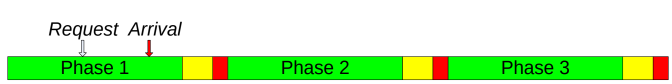


Figure 12: Active Priority Module



Case 1: No Action

Case 2: Green Extension Action

Case 3: Early Green Action

Case 4: Phase Insertion Action

Figure 13: Priority action examples

(3) Action Evaluation

We formulate the active priority decision problem as a Markov decision problem since it is observable, stochastic, and sequential [39]. The state s in the decision is the set of waiting queues of the intersection.

$$s = (\theta_1, \theta_2, ..., \theta_n) \tag{18}$$

where θ_i is the vehicle queue in ith direction, and n is the total number of possible directions of the intersection;

Assume that the priority requests from r approaching trucks need to be processed. The priority decision vector can be written as:

$$a = \{a_1, a_2, a_3, ..., a_r\} \tag{19}$$

where a_i is the decision of ith priority request.

$$a_i = \begin{cases} 1, & \text{give the priority} \\ 0, & \text{refuse the priority} \end{cases} \tag{20}$$

The state transition model is the system model of predicting future traffic state based on current state and priority action decision. The state transition model is:

$$\dot{\theta}(s,a,t) = \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \cdots \\ \dot{\theta}_n \end{pmatrix} = f_{in}(t \mid s) - f_{out}(t \mid s,a) \tag{21}$$

where *n* is the number of directions, $f_{in}(t \mid s)$ is the vector of upstream input traffic flows that can be collected via road detectors at time *t*. $f_{out}(t \mid s,a)$ is the vector of downstream traffic flows, i.e., the traffic flows leaving the intersection at time *t* when the traffic state *s* and priority action *a* are given.

The control objective is to reduce the delay and number of stops as a result of the priority decision which can be measured by the integral of waiting queues of all directions from current state *s* of time *t* to new state *s'* of time *t* + *T* where *T* is the time length of action *a*. This reward function considers the traffic flows from all directions and is given as:

$$R_a(s,s') = \frac{1}{T}\sum_i \int_t^{t+T} w_i \theta_i(s,a,\tau) d\tau \tag{22}$$

where $R_a(s,s')$ is the total reward from current state *s* to next state *s*', w_i is the weight of *i*th direction depending on the number of trucks in the queue.

Then the optimal decision iteration is:

$$\pi(s) := \arg\min_a R_a(s,s') \tag{23}$$

where $\pi(s)$ is the optimal decision at state *s*, the decision is to select the action *a* that minimizes the reward from current state *s* to next state *s*'.

(4) Output Decision

After evaluating the impact of priority response action, it will return a decision about granting priority to a truck or not. If the priority request is declined, then the following signal phases will not be changed. Otherwise, the signal phases will be changed to permit the truck to cross the intersection.

4 Evaluation Results

4.1 Evaluation Environment

This section presents the evaluation results of the proposed signal control systems on a road network. The proposed road network is adjacent to the Long Beach port. It is circled by Pacific Coast Hwy, N Wilmington Blvd, W Anaheim St and N Avalon Blvd and consists of more than 100 intersections in total 15 of which are signalized (see Fig. 14). The 15 intersections are controlled by 15 signal controllers in neural based approach 1 and controlled by three agents and each agent controls five intersections in system of approach 2. A microscopic traffic simulator of the selected road network has been implemented in VISSIM (see Fig. 15).

The priority control algorithms are implemented in MATLAB/C++ and integrated with the simulation environment via COM (Component Object Model) interface [40].

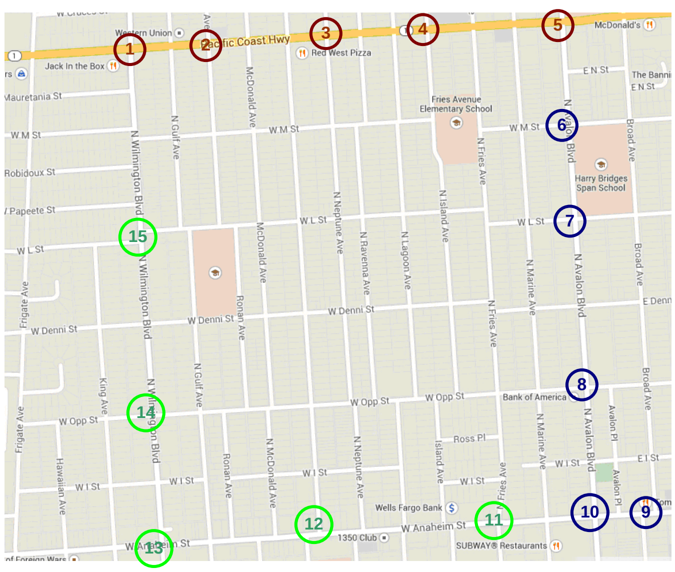


Figure 14: Selected road network

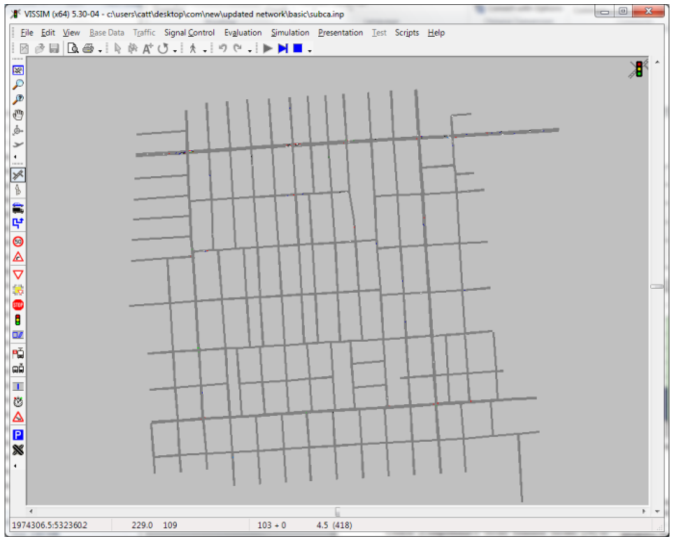


Figure 15: Traffic simulator of selected road network

4.2. Training of Neural Network Delay Predictor

First we evaluate the delay prediction model obtained by training the neural network. The training is done by using the MATLAB Neural Network Toolbox. Fig. 16 shows the prediction results for the multi-intersection test network. The Mean Squared Error (MSE) of delay predictions initially improves by increasing the number of hidden layers and nodes. On the other hand increasing the number of nodes beyond a certain point causes over-training and increases the MSE. Another important practical issue is the processing time needed to train the NN which is proportional to the size of the neural network. Table 2 shows the trade-off between MSE versus processing time for our test network. Based on the results shown in this table we choose a single layer network with 7 nodes. The MSE of the predictions are acceptable for this network and adding further layers or nodes to the model increases the processing time without significantly improving the error of delay prediction.

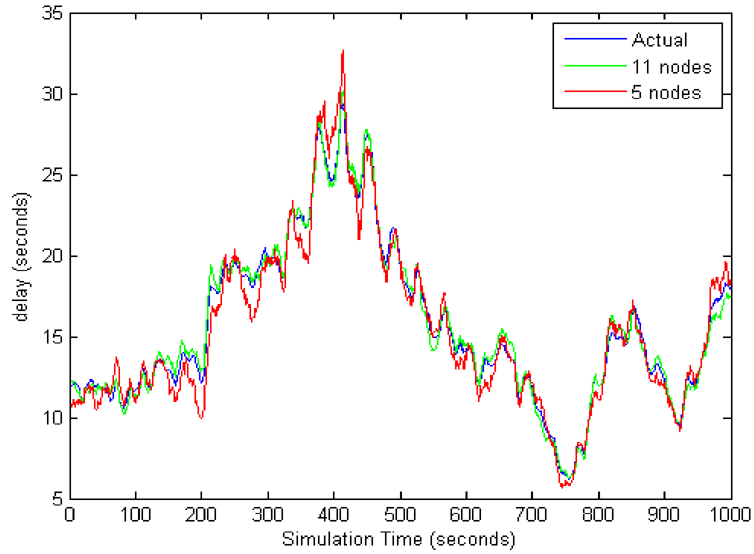


Figure 16: Performance of the delay prediction model for different NN size. For 11-node network the MSE is 2.2% while for a 5-node network the MSE increases to 7.5%.

Table 2: Mean Square Error and Processing Time for Neural Network

NN Structure	MSE (%)	Process Time (s)
Single-layer w/ 3 nodes	11.3	0.70
Single-layer w/ 5 nodes	7.5	0.98
Single-layer w/ 7 nodes	5.0	1.37
Single-layer w/ 9 nodes	3.4	1.92
Single-layer w/ 11 nodes	2.2	2.69
Single-layer w/ 13 nodes	2.2	3.76
2-layer w/ 3 nodes	10.4	1.50
2-layer w/ 5 nodes	6.9	2.10
2-layer w/ 7 nodes	4.6	2.94
2-layer w/ 9 nodes	3.1	4.12
2-layer w/ 11 nodes	2.1	5.76
2-layer w/ 13 nodes	1.8	8.07

We evaluate the performance of the proposed model with regard to the existence of interconnections between agents. In other words, we compare two modes; one with the use of communication between intersections, and one without . Fig. 17 shows the inputs used to feed the neural network in each case.

We also compare these two systems with the performance of the widely deployed actuated traffic signal controller, where inductive sensors are used to detect vehicles approaching the intersection. Our simulation results indicate the importance of communication between the intersections in the performance of the controller. Fig. 18-b) shows that the performance of the traffic signal controller is enhanced by 12% by means of communications between adjacent intersections and shows about 26% improvement compared to the actuated traffic signal controller. Also note that using a network-wise controller with intercommunications between the agents, the delay of the vehicles tends to be smoother compared to the delay of an actuated signal controller. Figure 18-a) confirms that making the agents communicate with each other helps reducing the prediction error.

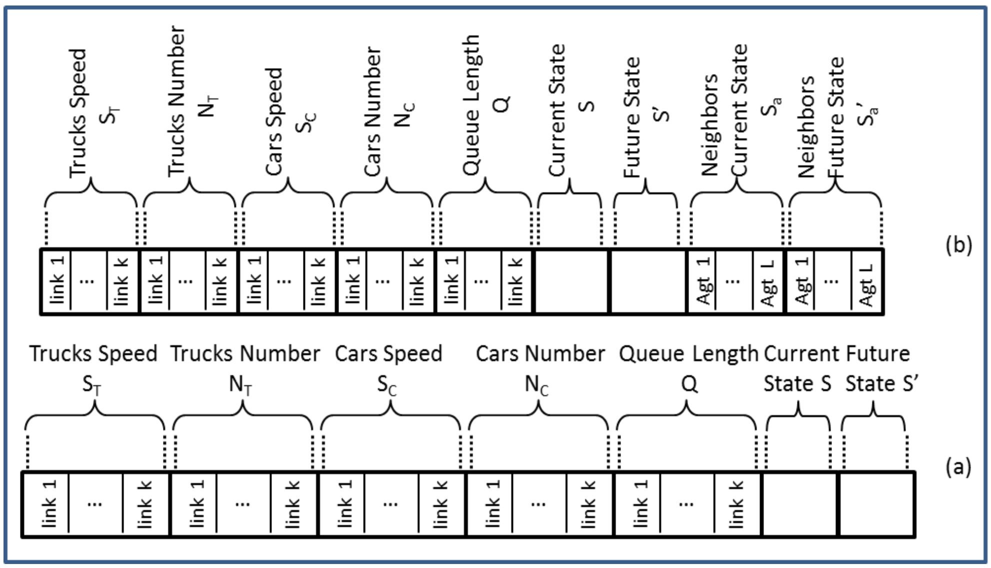


Figure 17: The input vectors for (a) the special case which agents acts independently, and (b) the general case which agents consider the effects of neighboring intersections.

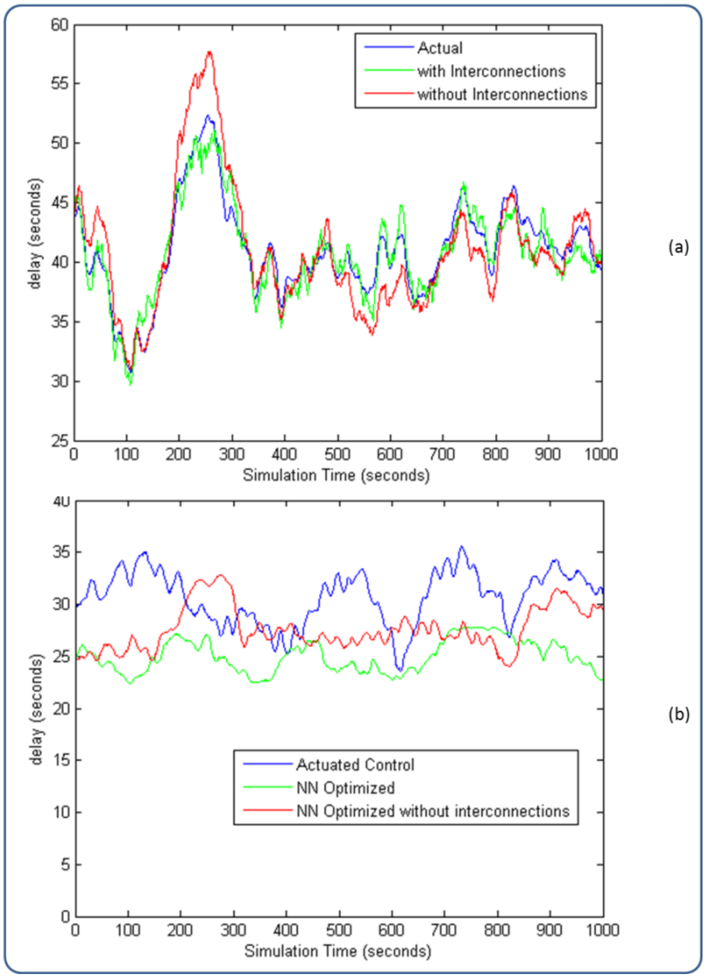


Figure 18: Comparison of two cases; with communication between links, and without communications. (a) the MSE of the prediction model in the case of no-interconnections increases to 11%, where the MSE for the general case is near 5%. (b) the average vehicle d delays obtained by actuated traffic signal controller is 31 seconds, whereas the average delay for the locally optimized controller (without interconnections) is 26 seconds, and the delay for the general case (network-wise optimized) is 23 seconds.

4.3 Comparison of Proposed Systems

We compared our proposed algorithms when the traffic flow in the road network is the average daily flow. Tables 3-5 summarize the evaluation results of different controllers when the truck ratio is 3%, 10%, and 20% of the overall flow respectively. As shown in the tables, both proposed controllers improve the network performance including delay and vehicle stops as well as environmental impact compared to the fixed time control that is the commonly used controller. Controller 2 provides less delay and number of stops for all vehicles compared to controller 1 in all three demands but controller 2 gives shorter truck delay for 3% and 20% truck demands and less number of truck stops for all three demands.

We use the CMEM (Comprehensive Model Emission Model) [41] vehicle emission model to calculate fuel consumption and emissions in order to compare the average vehicle emissions and fuel consumption. As shown in the tables, emissions such as HC, CO, NOx and fuel consumption have been improved especially for trucks with the proposed truck priority system for both approaches. The results also indicate that giving priority to trucks could result in reduced emissions and fuel savings for all vehicles. Controller 1 has a better performance on reducing fuel consumption for trucks while controller 2 has a better performance in reducing the fuel consumption of all vehicles on the average. Moreover, controller 2 generates less air pollution emissions such as CO2 and NOx even though the difference is small and could be viewed as being within the modeling error. In summary, both controllers generate a better performance that benefits all vehicles involved and has a positive impact on the environment with truck priority than without. .

Table 3: Road network results (3% Truck)

	Fixed Time	**Proposed Controller 1**		**Proposed Controller 2**	
		W/out Priority	***W/ Priority***	***W/out Priority***	***W/ Priority***
Avg. Delay/Veh (sec)	85.4	67.2	59.2	51.5	49.3
Avg. Delay/Car (sec)	85.1	67.3	59.5	52.2	49.1
Avg. Delay/Truck (sec)	88.1	65.2	52.1	63.3	55.5
Avg. Stops/Veh	3.84	2.91	2.73	2.76	2.73
Avg. Stops/Car	3.93	2.98	2.77	2.77	2.74
Avg. Stops/Truck	3.8	2.61	1.88	2.50	2.49
Fuel Trucks (g/km)	452.0	336.0	316.8	362.2	354.7
Fuel cars (g/km)	137.8	110.1	106.2	95.6	93.2
Fuel all veh. (g/km)	163.6	132.7	127.2	117.3	115.0
CO2 Emis. All (g/km)	427.9	347.1	333.0	325.5	316.8
NOx Emis. All (g/km)	1.01	0.82	0.78	0.80	0.76

Table 4: Road network results (10% Truck)

	Fixed Time	Proposed Controller 1		Proposed Controller 2	
		W/out Priority	***W/ Priority***	***W/out Priority***	***W/ Priority***
Avg. Delay/Veh (sec)	89.0	70.0	67.3	52.7	49.3
Avg. Delay/Car (sec)	88.7	70.2	67.6	51.6	48.2
Avg. Delay/Truck (sec)	91.8	68.0	64.1	62.7	59.3
Avg. Stops/Veh	4	2.95	2.71	2.72	2.67
Avg. Stops/Car	4.09	3	2.76	2.70	2.65
Avg. Stops/Truck	3.9	2.65	2.04	2.85	2.82
Fuel Trucks (g/km)	470.9	350.1	330.0	377.3	369.5
Fuel cars (g/km)	143.6	114.7	110.7	99.6	97.1
Fuel all veh. (g/km)	170.5	138.3	132.6	122.2	119.8
CO2 Emis. All (g/km)	445.8	361.6	346.9	339.1	330.1
NOx Emis. All (g/km)	1.06	0.86	0.82	0.84	0.80

Table 5: Road network results (20% Truck)

	Fixed Time	Proposed Controller 1		Proposed Controller 2	
		W/out Priority	***W/ Priority***	***W/out Priority***	***W/ Priority***
Avg. Delay/Veh (sec)	93.4	73.5	59.9	53.8	50.3
Avg. Delay/Car (sec)	93.1	73.7	60.3	51.8	48.8
Avg. Delay/Truck (sec)	96.3	71.4	56.6	62.5	56.8
Avg. Stops/Veh	4.22	3.10	2.82	2.73	2.65
Avg. Stops/Car	4.31	3.15	2.87	2.68	2.66
Avg. Stops/Truck	3.96	2.55	2.13	2.95	2.62
Fuel Trucks (g/km)	494.4	367.6	346.5	396.1	387.9
Fuel cars (g/km)	150.7	120.4	116.2	104.5	101.9
Fuel all veh. (g/km)	179.0	145.2	139.2	128.3	125.7
CO2 Emis. All (g/km)	468.0	379.6	364.2	356.0	346.6
NOx Emis. All (g/km)	1.11	0.90	0.86	0.88	0.84

The two controllers have the following differences.

- Controller 1 works on the intersection level to make decision while controller 2 works on both the network and intersection levels.
- Controller 1 does not need an online simulator to compute real time control input after the training of the NN delay predictor. However, controller 2 needs an online simulator to find the baseline signal for active module when the traffic demands change. As a result the controller will face a time complexity problem when the controlled road network is large scaled. The training of the neural network however may have to be repeated occasionally in order to capture changes in the dynamics of the system.
- Controller 1 could deal with more traffic input scenarios including oversaturated traffic flows. The performance of controller 2 under oversaturated traffic flows will be limited due to the fact that the simulation time will become significantly longer when the number of vehicles in the network is increased.

5 Conclusion

In this report, we proposed two truck traffic light priority systems whose performance is demonstrated using a microscopic simulation model of an actual road network. The first system uses a neural network approach to predict the average delay of vehicles by taking into account different classes of vehicles. The predicted delays are fed into an optimizer which generates the optimum signal timing Each intersection has its dedicated delay prediction agent which communicates with agents from other nearby intersections of the network in order to share their delay predictions and current and future states of the traffic lights. The approach of the second system integrates the advantages of passive and active control strategies to achieve better network performance by improving traffic delays, reducing the number of stops and emissions in comparison to no-priority and passive priority strategies. It uses a simulation-based approach to generate the predicted states used by an optimization strategy to generate the signal timing at each intersection. Both approaches are evaluated using a microscopic traffic simulation model of a road and their performance improvements by applying truck priority rules have been demonstrated.

References

[1] N. J. Garber, L. A. Hoel, Traffic & Highway Engineering, 4th edition, Cengage Learning, June 2008, ch. 3.

[2] Traffic Engineering Handbook, 5th Edition, Institute of Transportation Engineers, Washington DC, 2000, ch. 3.

[3] N. Saunier, T. Sayed, C. Lim, "A prototype system for truck signal priority using video sensors," 2009 Annual Conference and Exhibition of the Transportation Association of Canada - Transportation in a Climate of Change, pp. 1-16, Oct. 2009.

[4] J. Bonneson, D. Middleton, K. Zimmerman, H. Charara, M. Abbas, "Intelligent detection-control system for rural signalized intersections", Aug. 2002.

[5] C. Liao, G.A. Davis, Bus signal priority based on GPS and wireless communications phase 1 – simulation study, Department of Civil Engineering, University of Minnesota, July 2006.

[6] W. Ma, X.Yang, "A passive transit signal priority approach for bus rapid transit system," 2007 IEEE Conference on Intelligent Transportation Systems, pp. 413-418.

[7] F. Li, D. H. Wang, J. Wang, and S. Jin, "An approach of transit passive priority with transit phase overlapped at intersection of arterial signal progression," 2008 IEEE Conference on Intelligent Transportation Systems, pp.729-733.

[8] N. Hounsell and B. Shrestha, "A new approach for co-operative bus priority at traffic signals", IEEE Transaction on Intelligent Transportation Systems, vol. 13, no. 1, pp. 6-14, 2012.

[9] W. Mang, X. Yang, "Design and Evaluation of an Adaptive Bus Signal Priority System Based on Wireless Sensor Network", 2008 IEEE Conference on Intelligent Transportation Systems, pp. 1073-1077.

[10] T. Pohlmann, B. Friedrich, T. U. Braunschweig, and A. C. T. Model, "Online Control of Signalized Networks using the Cell Transmission Model," 2010.

[11] P. B. Hunt, D. I. Robertson, R. D. Bretherton, and R. I. Winton, "SCOOT-a traffic responsive method of coordinating signals," 1981.

[12] A. G. Sims and K. W. Dobinson, "The Sydney coordinated adaptive traffic (SCAT) system philosophy and benefits," Veh. Technol. IEEE Trans., vol. 29, no. 2, pp. 130–137, 1980.

[13] C. Bielefeldt and F. Busch, "MOTION-a new on-line traffic signal network control system motion," 1994.

[14] B. Friedrich and H. Keller, "Balance-a method for integrated traffic adaptive and vehicle actuated signal control," in Proc. of the 7th IFAC Symposium, 1994, pp. 24–26.

[15] R. Jayakrishnan, S. P. Mattingly, and M. G. McNally, "Performance study of SCOOT traffic control system with non-ideal detectorization: field operational test in the city of Anaheim," in 80th Annual Meeting of the Transportation Research Board, 2001.

[16] A. Peterson and T. Bergh, Report 1991:51E "LHOVRA A traffic signal control strategy for isolated junctions", Swedish National Road Administration.

[17] P. Kronborg, F. Davidsson, and J. Edholm, "Self Optimising Signal Control", Transport Research Institute, Sweden, 1993.

[18] L. C. Liao, "A review of the optimized policies for adaptive control strategy (OPAC)", California PATH working paper, UCB-ITS-PWP-98-9.

[19] Farhad Pooran, "OPAC adaptive engine Pinellas County Deployment", Baltimore Regional Traffic Signal Forum, 2011.

[20] P. Mirchandani, L. Head, "A real-time traffic signal control system: architecture, algorithms, and analysis", Transportation Research Part C, pp. 415-32, 2001.

[21] C. Diakaki, M. Papageorgiou, and K. Aboudolas, "A multivariable regulator approach to traffic-responsive network-wide signal control," Control Eng. Pract., vol. 10, no. 2, pp. 183–195, 2002.

[22] Y. Tian, Z. Li, D. Zhou, J. Song, D. Xiao, "Interactive signal control for over-saturated arterial intersections using fuzzy logic," 2008 IEEE Conference on Intelligent Transportation Systems, pp. 1067-1072.

[23] X. Kuang, L. Xu, “Real-time traffic signal intelligent control with transit-priority,” Journal of Software, vol. 7, no. 8, pp. 1738-1743, 2012.
[24] T. Pohlmann, B. Friedrich, “Online control of signalized networks using the cell transmission model,” 2010 IEEE Conference on Intelligent Transportation Systems, pp. 1-6.
[25] S. Timotheou, C. G. Panayiotou, M. M. Polycarpou, “Towards distributed online cooperative traffic signal control using the cell transmission model,” 2013 IEEE Conference on Intelligent Transportation Systems, pp. 1737-1742.
[26] T. Li, D. Zhao, J. Yi, “Adaptive dynamic programming for multi-intersections traffic signal intelligent control,” 2008 IEEE Conference on Intelligent Transportation Systems, pp. 286-291.
[27] M. Mesbah, M. Sarvi, and G. Currie, “Optimization of transit priority in the transportation network using a genetic algorithm”, IEEE Transaction on Intelligent Transportation Systems, vol. 12, no. 3, pp. 908-919, 2011.
[28] J. Stevanovic, A. Stevanovic, P. Martin, and T. Bauer, “Stochastic optimization of traffic control and transit priority settings in VISSIM”, Transportation Research Part C, 16, pp. 332-349, 2008.
[29] S. EI-Tantawy, B. Abdulhai, “Multi-Agent Reinforcement Learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC),” 2012 IEEE Conference on Intelligent Transportation Systems, pp. 319-326.
[30] S. Araghi, A. Khosravi, M. Johnstone, and D. Creighton, “Q-learning method for controlling traffic signal phase time in a single intersection,” 2013 IEEE Conference on Intelligent Transportation Systems, pp. 1261-1265.
[31] L. B. de Oliveira and E. Camponogara, “Multi-agent model predictive control of signaling split in urban traffic networks,” Transp. Res. Part C Emerg. Technol., vol. 18, no. 1, pp. 120–139, 2010.
[32] D. Srinivasan and M. C. Choy, “Cooperative multi-agent system for coordinated traffic signal control,” in IEE Proceedings-Intelligent Transport Systems, 2006, vol. 153, no. 1, pp. 41–50.
[33] A. ad Salkham, R. Cunningham, A. Garg, and V. Cahill, “A collaborative reinforcement learning approach to urban traffic control optimization,” in Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 02, 2008, pp. 560–566.
[34] M. Wiering, “Multi-agent reinforcement learning for traffic light control,” 2000.
[35] I. Arel, C. Liu, T. Urbanik, and A. G. Kohls, “Reinforcement learning-based multi-agent system for network traffic signal control,” Intell. Transp. Syst. IET, vol. 4, no. 2, pp. 128–135, 2010.
[36] R. Hooke, T.A. Jeeves. “"Direct search" solution of numerical and statistical problems,” Journal of the Association for Computing Machinery (ACM) , vol. 8, no. 2, pp. 212–229, 1961.
[37] R. M. Lewis, V. Torczon, “Pattern search algorithms for bound constrained minimization,” SIAM Journal on Optimization, vol. 9, no. 4, pp. 1082-1099, 1999.
[38] Y. Li, P. Koonce, M. Li, K. Zhou, Y. Li, S. Beaird, W.B. Zhang, L. Hegen, K. Hu, A. Skabardonis, Z. S. Sun, “Transit signal priority research tools”, Technical report, California Partners for Advanced Transit and Highways (PATH), May, 2008.
[39] H.S. Chang, M. C. Fu, J. Hu, S. I. Marcus, Simulation-based Algorithms for Markov Decision Processes, Springer, March 2007.
[40] VISSIM – COM User Manual, PTV Vision, 2011.
[41] M. Barth, G. Scora, and T. Younglove, “Estimating emissions and fuel consumption for different levels of freeway congestion”, Transportation Research Record: Journal of the Transportation Research Board, no. 1664, pp. 47-57, 1999.