An adaptive model for traffic flow optimisation in dynamic environments

M.V. Rahul* and Rajashree Shettar

Department of Computer Science,
Rashtreeya Vidyalaya College of Engineering,
Bangalore, Karnataka, India
Email: rahulvenkk@gmail.com
Email: rajashreeshettar@rvce.edu.in
*Corresponding author

K.N. Subramanya

Department of Industrial Engineering and Management,
Rashtreeya Vidyalaya College of Engineering,
Bangalore, Karnataka, India
Email: subramanyakn@rvce.edu.in

Abstract: Formulating the solution as an optimisation problem has proven to be effective in developing solutions to many real world problems. We generally obtain the best possible solution using these methods. In this work, the traffic scheduling problem has been formulated as a waiting time minimisation problem, and appropriate cost functions have been developed, in pursuit of finding the optimal solution. A first-in, first-out queuing model is used, with the vehicles arriving in a Poisson process, and the service time being exponentially distributed. The key feature of this model is that it adapts to varying service and arrival rates of the lanes. These rates are forecast using a neural network model, and appear in the objective function. It was observed that the use of the neural network greatly improved the robustness of the model. Although the model has been developed for a four lane, two way architecture, it can be generalised to any architecture. Results have been analysed by comparing the proposed method to a proportional time distribution. It is shown that the proposed model performs relatively well, when there is rapid variation in the arrival and service rates.

Keywords: queuing model; isolated intersections; neural network; interior point optimisation; traffic scheduling; Poisson process; four lane intersection.

Reference to this paper should be made as follows: Rahul, M.V., Shettar, R. and Subramanya, K.N. (2019) 'An adaptive model for traffic flow optimisation in dynamic environments', *Int. J. Computational Science and Engineering*, Vol. 19, No. 1, pp.25–35.

Biographical notes: M.V. Rahul is currently an undergraduate student at the Rashtreeya Vidyalaya College of Engineering, Bangalore, India. He is pursuing a Computer Science and Engineering degree. His research interests include computer vision, machine learning, and optimisation techniques.

Rajashree Shettar is a Professor in the Computer Science Department, and Associate Dean for Post Graduate Studies at the Rashtreeya Vidyalaya College of Engineering, Bangalore, India. Her research interests include data mining, web mining and information retrieval.

K.N. Subramanya is the Principal of Rashtreeya Vidyalaya College of Engineering, Bangalore, India. He is also a Professor at the Industrial Engineering and Management Department. His research interests include supply chain and logistic management, information system design, decision sciences, applied ergonomics, systems engineering and strategic management.

1 Introduction

In congested urban centres, traffic congestion is a rising problem. In addition to the delay caused in travel time, it adds to pollution and results in inefficient use of fuel. In most traffic signals, the cycle time is equally distributed among the lanes, and is hard-coded in the circuitry. This is highly inefficient, and leads to congestion or traffic build up in the lanes. Also, the variation in the service and arrival rates is not taken into account. In this work, we propose an approach that finds the optimal cycle time distribution that takes into account the variation in arrival and service rates of the lanes. Many approaches to traffic scheduling, that make use of game theory, fuzzy logic, etc., have been proposed by researchers, in the past.

Adaptive green interval algorithms that consider variable cycle times in each iteration have been depicted in Aljaafreh et al. (2014), Lee et al. (2007). In Aljaafreh et al. (2014), it is shown that algorithms that consider variations in arrival and service rates perform better with increasing variance in input parameters, as compared to baseline techniques.

Fuzzy logic controllers have also proven to be effective, owing to computational efficiency, and their resemblance to a manned traffic controller system. In Ge (2014), a two stage fuzzy control method has been proposed for four lane architecture. The method takes priorities into consideration, by using a parameter called urgency degree. Fuzzy systems, do not find the optimal solution, but give a good approximation to the optimal solution, when such a solution is difficult to find by conventional techniques. Fuzzy inference and implications are generally used to find the solution. In addition, such systems do not directly account for varying arrival and service rates, but such variances are taken into account by the implications. In this work, it is shown that an optimal solution can be found using a queuing model that directly accounts for variation in arrival and service rates.

The arrival and service rates can be estimated using vehicle counters as shown in Taghvaeeyan and Rajamani (2014), where wireless magnetic devices are used to estimate the vehicle count. Other ways of obtaining the vehicle count are by using computer vision approaches, as done in Tourani and Shahbahrami (2015). Here, edge detection and frame differencing are used to detect the vehicles, and hence obtain the count.

In De Schutter (1999) it was shown that a reliable and optimised light switching sequence can be found, considering the dynamic changes in queue lengths. It is shown that both linear and nonlinear variants of the objective function can be used. The nonlinear variant can be optimised using nonlinear programming. A disadvantage of this approach was that it did not consider varying arrival and service rates.

In recent times, reinforcement learning and deep convolutional networks have been used to model the isolated intersection, as seen in Genders and Razavi (2016), Koltovska and Bombol (2014). In Genders and Razavi (2016), the convolutional network learns/gets reinforced by a reward function, to choose which lanes to activate at a particular time. The authors provide an end to end solution that can be implemented in practice. In Toulni and Nsiri (2018), a cluster-based routing protocol using the road traffic information is proposed, and results are shown on vehicular ad hoc networks (VANETs). The method obtains a solution that is optimal, for the entire traffic network. Similarly, in Boussoufa-Lahlah et al. (2018), an optimal-path selection algorithm for VANETs is proposed, which finds the optimal path for a vehicle under consideration, but the method does not adapt to the changing vehicular conditions in real-time, nor does it minimise the waiting time of all vehicles jointly. In this work, however, we model the problem as a waiting time optimisation problem, in order to find the best solution.

It has been shown in Gu et al. (2017) and Fan et al. (2018), that time series forecasting, is an effective method to model dynamic systems. We therefore employ a neural network to forecast the arrival and service rates of the lanes. Hence the model is continuously learning, and evolving to new changes.

In Liu et al. (2018), it is shown that load balancing in multi agent systems can be achieved using multiple linear regression analysis. In contrast, we use a time series model that achieves load balancing, in addition to an optimal cycle time distribution. Our model is developed for a four lane, two way architecture as shown in Figure 1, but can be adapted to any multi-lane, multi-way architecture.

2 Proposed method

The architecture of the traffic signal intersection that is considered is shown in Figure 1. At the start of each cycle, the cycle time is distributed among the lanes, after which, the green signals of the lanes are activated in the order lane 1, lane 2, lane 3, and lane 4. When the green signal is given for lane *i*, vehicles from lane *i* can go straight, left or right. Only vehicles from one lane can be serviced during the green time of that lane. The algorithm proposed has been developed for a four lane intersection, with only one lane being serviced at a given time. It can be modified to work for any kind of intersection, having three or two lanes, etc. Also the algorithm can be adapted to work for an architecture in which more than one lane is serviced at a given time.

Let the number of vehicles present in the lane i be x_i at the start, and the time to be allocated at the start of the cycle, for lane i be t_i, then the function to be optimised is designed in the following manner. Each lane has a cost associated with it. Cost is defined in general as the product of the number of vehicles and the time those vehicles wait. For a particular distribution of time between the lanes, a cost is obtained. The task here is to find the best such time distribution, at the start of each cycle. Let t_1, t_2, t_3, t_4 denote the time distribution. The waiting time needs to be minimised, in each cycle, by finding the best set of t_i's.

Let r_i denote the arrival rate for lane i, which is the number of vehicles that arrive per unit time. Also, let s_i denote the service rate of lane i, which is the number of vehicles that are serviced per unit time. These rates can change after each cycle, and it is assumed that they remain constant for the duration of one cycle. It is also assumed that the vehicles arrive in a Poisson process, similar to Jha and Shukla (2016), which causes the inter-arrival times of vehicles to be exponentially distributed. Essentially, the first vehicle that arrives waits for a time,

$$K_1 = \left(\frac{r_i t_i - 1}{r_i} \right) \tag{1}$$

The second vehicle that arrives waits for a time,

$$K_2 = \left(\frac{r_i t_i - 1}{r_i} \right) \quad (2)$$

Similarly the n^{th} vehicle to arrive waits for a time,

$$K_n = \left(\frac{r_i t_i - n}{r_i} \right) \quad (3)$$

Hence, if y is the total cost for the vehicles that arrive in time t_i for lane i then, f: $(r_i, t_i) \rightarrow y$, where

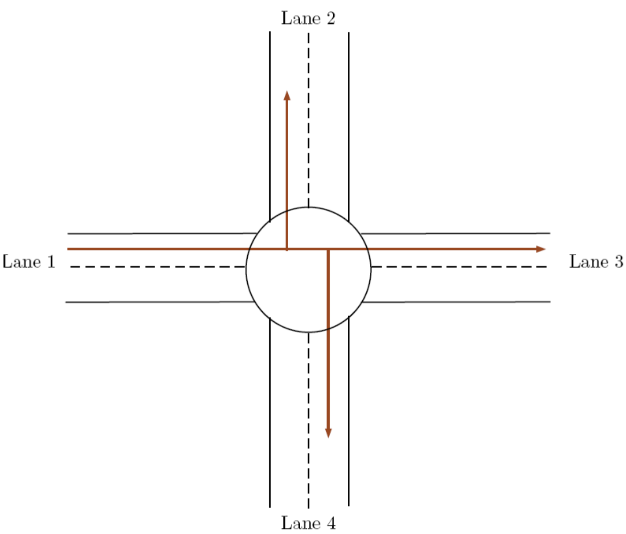
$$y = f(r_i, t_i) = \sum_{c=1}^{r_i t_i} K_c \quad (4)$$

Using the formula for the sum of an arithmetic progression,

$$f(r_i, t_i) = \frac{t_i (r_i t_i - 1)}{2} \quad (5)$$

Assuming that the service time is exponentially distributed, by a similar proof it can be shown that the vehicles that are serviced in time t_i of lane i contribute a cost equal to $f(s_i, t_i)$. To find the cost for a particular lane during the green time, three cases need to be considered.

Figure 1 An isolated intersection (see online version for colours)



2.1 Case 1: $s_i t_i < I_i$

Here, s_i is the service rate for lane i and I_i is the vehicle count of lane i at the start of its green time. In this case, the vehicles that arrive during the green time for lane i, do not get serviced. Effectively $(I_i - s_i t_i)$ vehicles remain unserviced and these contribute a total cost of $(I_i - s_i t_i)t_i$. Adding up the costs contributed by the serviced and unserviced vehicles, the total cost of lane i for case 1, during the green time is given by,

$$G_i^1 = (I_i - s_i t_i) t_i + f(s_i, t_i) \quad (6)$$

where G denotes cost incurred during the green time.

2.2 Case 2: $I_i < s_i t_i < I_i + r_i t_i$

This case is depicted in Figure 2. Here the thick horizontal box represents the number of vehicles in that lane, in green state. At the start of the green time (at time t_0), I_i number of vehicles are present in lane i. At the end of the green time, a maximum of $I_i + r_i t_i$ vehicles can be present in that lane. As the servicing capacity, $s_i t_i > I_i$, I_i vehicles are serviced in $\frac{I_i}{s_i}$ time. These vehicles do not get serviced at once, they need to wait to get serviced and this contributes a cost of $f\left(s_i, \frac{I_i}{s_i}\right)$. The cost is denoted by,

$$G_i^2(0, s) = f\left(s_i, \frac{I_i}{s_i}\right) \quad (7)$$

where 0 indicates time t_0, and s indicates the cost incurred by waiting for getting serviced. Similarly,

$$G_i^2(0, r) = 0 \quad (8)$$

where 0 indicates time t_0 and r indicates the cost incurred due to waiting, while the vehicles arrive. $G_i^2(0, r) = 0$ because the I_i vehicles do not arrive during the green time of lane i. While the I_i vehicles are serviced, $\left(\frac{I_i}{s_i}\right) r_i = I_i \rho_i$ vehicles arrive. Assume that these $I_i \rho_i$ vehicles can be serviced in the time remaining. This implies, $\left(t_i - \frac{I_i}{s_i}\right) s_i \geq I_i \rho_i$. This has been shown in Figure 2. During the time these vehicles are serviced, $\left(\frac{I_i r_i}{s_i^2}\right) r_i = I_i \rho_i^2$ vehicles arrive. Assume that these vehicles can be serviced as well. To derive the edge case, assume that starting at time t_{n-1}, $I_i \rho_i^{n_i - 1}$ vehicles arrive. These vehicles can be serviced. While these $I_i \rho_i^{n_i - 1}$ are serviced, $I_i \rho_i^{n_i}$ vehicles arrive. Here, assume that all these $I_i \rho_i^{n_i}$ cannot be serviced. The time remaining after the $I_i \rho_i^{n_i - 1}$ vehicles are serviced is,

$$time_{\alpha_i} = \left(t_i - \frac{I_i}{s_i} \left(\frac{\rho_i^n - 1}{\rho_i - 1} \right) \right) \quad (9)$$

During this time, $\alpha_i = time_{\alpha_i} * s_i$ vehicles can be serviced. The remaining $\beta_i = I_i \rho^{n_i} - \alpha_i$ vehicles remain unserviced. In addition, $\gamma_i = time_{\alpha_i} * r_i$ vehicles arrive when the α_i vehicles are being serviced. These γ_i vehicles also remain unserviced. It can be observed that the $I_i \rho_i^k$ vehicles that are present at time $t_k(0 < k < ni)$ contribute a cost of,

$$G_i^2(k, s) = f\left(s_i, \left(\frac{I_i}{s_i}\right) \rho_i^k\right) \quad (10)$$

and

$$G_i^2(k, r) = f\left(r_i, \left(\frac{I_i}{s_i}\right)\rho_i^{k-1}\right) \tag{11}$$

Also, as discussed earlier, it can be shown that,

$$G_i^2(0, s) = f\left(s_i, \frac{I_i}{s_i}\right) \tag{12}$$

and

$$G_i^2(0, r) = 0 \tag{13}$$

In addition,

$$G_i^2(n_i, r) = f\left(r_i, \left(\frac{I_i}{s_i}\right)\rho_i^{n_i-1}\right) \tag{14}$$

and

$$G_i^2(n_i, s) = f(s_i, time_{\alpha_i}) \tag{15}$$

The β_i vehicles wait for an additional $time_{\alpha_i}$. Therefore,

$$cost_{\beta_i} = \beta_i * time_{\alpha_i} \tag{16}$$

and the vehicles that arrive in $time_{\alpha_i}$ contribute a cost of

$$cost_{\alpha_i} = f(r_i, time_{\alpha_i}) \tag{17}$$

Hence, the total cost,

$$G_i^2 = \sum_{k=0}^{n_i}\left(G_i^2(k, r) + G_i^2(k, s) + cost_{\beta_i} + cost_{\alpha_i}\right) \tag{18}$$

where the value of n_i (35) calculated using,

$$\alpha_i \le I_i\rho_i^{n_i} \tag{19}$$

The number of vehicles remaining in the lane i after its green time is given by

$$rem_i = \alpha_i + \beta_i \tag{20}$$

In Tables 1, 2, 3, 4, R_i^2 denotes the cost of lane i when it is in red state for case 2 and $R_i^2(k)$ denotes the cost of lane i when lane k is in green state.

2.3 Case 3: $s_i t_i \ge I_i + r_i t_i$

This case is depicted in Figure 3. Here the thick horizontal box represents the number of vehicles in that lane in green state. At the start of the green time (at time t_0), Ii number of vehicles are present in lane i, At the end of the green time, a maximum of $I_i + r_i t_i$ vehicles can be present in that lane. As the servicing capacity, $s_i t_i \ge I_i + r_i t_i$, I_i vehicles are serviced in $\frac{I_i}{s_i}$ time. These vehicles do not get serviced at once, they need to wait to get serviced and this contributes a cost of $f\left(s_i, \frac{I_i}{s_i}\right)$. This cost is denoted by,

$$G_i^3(0, s) = f\left(s_i, \frac{I_i}{s_i}\right) \tag{21}$$

where 0 indicates time t_0, and s indicates the cost incurred by waiting for getting serviced. Similarly,

$$G_i^3(0, r) = 0 \tag{22}$$

where 0 indicates t_0 and r indicates the cost incurred due to waiting while the vehicles arrive. $G_i^3(0, r) = 0$ because the I_i vehicles do not arrive during the green time of lane i. While the I_i vehicles are serviced, $\left(\frac{I_i}{s_i}\right)r_i = I_i\rho_i$ vehicles arrive. Assume that these $I_i\rho_i$ vehicles can be serviced in the time remaining. This implies, $\left(t_i - \frac{I_i}{s_i}\right)s_i \ge I_i\rho_i$. This has been shown in Figure 3. During the time these vehicles are serviced, $\left(\frac{I_i r_i}{s_i^2}\right)r_i = I_i\rho_i^2$ vehicles arrive. Assume that these vehicles can be serviced as well. To derive the edge case, assume that starting at time $t_n - 1$, $I_i\rho_i^{n_i-1}$ vehicles arrive. Also assume that $I_i\rho_i^{n_i} \le 1$. Due to this, less than 1 vehicle arrives when the $I_i\rho_i^{n_i-1}$ vehicles are being serviced. Therefore any vehicle that arrives after time t_n is serviced immediately and does not contribute any cost. It can be observed that the $I_i\rho_i^k$ vehicles that are present at time $t_k (0 < k < n_i)$ contribute a cost of

$$G_i^3(k, s) = f\left(s_i, \left(\frac{I_i}{s_i}\right)\rho_i^k\right) \tag{23}$$

and

$$G_i^3(k, r) = f\left(r_i, \left(\frac{I_i}{s_i}\right)\rho_i^{k-1}\right) \tag{24}$$

Also, as discussed earlier, it can be shown that,

$$G_i^3(0, s) = f\left(s_i, \frac{I_i}{s_i}\right) \tag{25}$$

and

$$G_i^3(0, r) = 0 \tag{26}$$

In addition,

$$G_i^3(n, s) = 0 \tag{27}$$

and

$$G_i^3(n, r) = 0 \tag{28}$$

Hence, the total cost,

$$G_i^3 = \sum_{k=0}^{n_i}\left(G_i^2(k, r) + G_i^2(k, s)\right) \tag{29}$$

where the value of n_i (35) is calculated using,

$$I_i \rho_i^{n_i} \leq 1 \tag{30}$$

Here, the number of vehicles remaining in the lane after its green time, $rem_i = 0$. In Tables 1, 2, 3, 4, R_i^3 denotes the cost of lane i when it is in red state for case 3 and $R_i^3(k)$ denotes the cost of lane i when lane k is in green state.

2.4 Putting all the costs together

Table 1 shows the values of the cost of lane 1 when it is in red state. It also shows the total costs of lane 1 for the three cases. Similarly, Tables 2, 3 and 4 show the values of cost for lanes 2, 3 and 4 respectively. In these tables, while R_i^j denotes the cost of lane i when it is in red state for case j, $R_i^j(k)$ denotes the cost of lane i when lane k is in green state and lane i is in red state. These costs have been depicted in Table 1, Table 2, Table 3 and Table 4. For any lane i, the total cost for it is given by,

$$C_i = \begin{cases} C_i^1 & s_i t_i \leq I_i \\ C_i^2 & x_i < s_i t_i < I_i + r_i t_i \\ C_i^3 & s_i t_i \geq I_i + r_i t_i \end{cases} \tag{31}$$

and if m is the total cost, the final optimisation function, Z: $(t_1, t_2, t_3, t_4) \rightarrow m$, is given by,

$$Z(t_1, t_2, t_3, t_4) = \sum_{i=1}^{4} C_i \tag{32}$$

$$\delta_i = \left(t_i - \frac{I_i}{s_i} - \frac{I_i r_i}{s_i^2} - \ldots - \frac{I_i r_i^{n_i - 1}}{s_i^{n_i}} \right) s_i$$

$$time_{\alpha_i} = t_i - \frac{I_i}{s_i} \left(\frac{\rho_i^n - 1}{\rho_i - 1} \right)$$

$$\alpha_i = time_{\alpha_i} * s_i$$

$$\beta_i = I_i \rho_i^{n_i} - \alpha$$

$$\gamma_i = time_{\alpha_i} * r_i$$

Here, $\alpha_i <= I_i \rho_i^{n_i}$.

Figure 2 Case 2

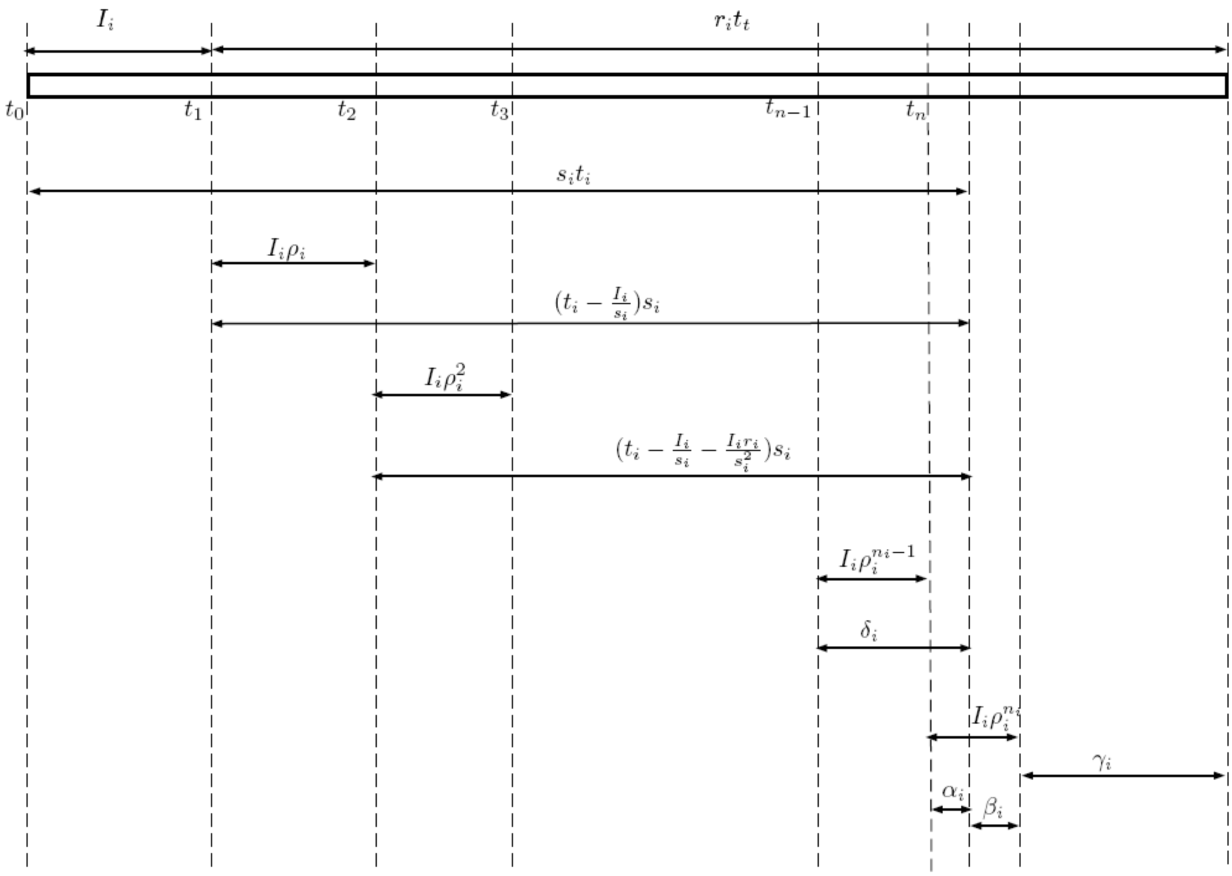


Figure 3 Case 3

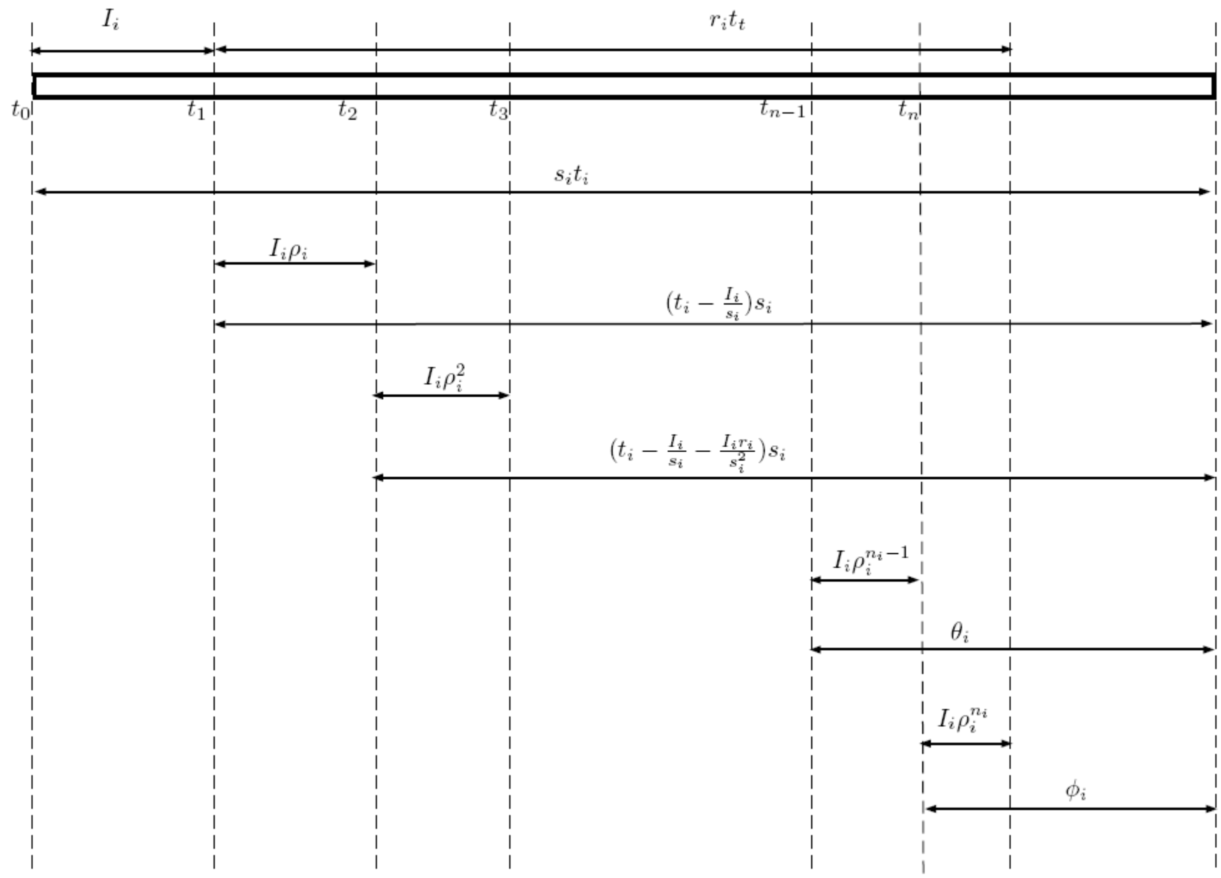


Table 1 Lane 1

Case-1 ($s_1 t_1 \leq x_1$)		*Case-2* ($x_1 < s_1 t_1 < x_1 + r_1 t_1$)		*Case-3* ($s_1 t_1 \geq x_1 + r_1 t_1$)	
Parameter	*Value*	*Parameter*	*Value*	*Parameter*	*Value*
$R_1^1(1)$	$(x_1 - s_1 t_1 + r_1 t_1) t_2 + f(r_1, t_2)$	$R_1^2(1)$	$rem_i * t_2 + f(r_1, t_2)$	$R_1^3(1)$	$f(r_1, t_2)$
$R_1^1(2)$	$(x_1 - s_1 t_1 + r_1 t_1 + r_1 t_2) t_3 + f(r_1, t_3)$	$R_1^2(2)$	$(rem_i + r_1 t_2) t_3 + f(r_1, t_3)$	$R_1^3(2)$	$(r_1 t_2) t_3 + f(r_1, t_3)$
$R_1^1(3)$	$(x_1 - s_1 t_1 + r_1 t_1 + r_1 t_2 + r_1 t_3) t_4 + f(r_1, t_4)$	$R_1^2(3)$	$(rem_i + r_1 t_2 + r_1 t_3) t_4 + f(r_1, t_4)$	$R_1^3(3)$	$(r_1 t_2 + r_1 t_3) t_4 + f(r_1, t_4)$
R_1^1	$R_1^1(1) + R_1^1(2) + R_1^1(3)$	R_1^2	$R_1^2(1) + R_1^2(2) + R_1^2(3)$	R_1^3	$R_1^3(1) + R_1^3(2) + R_1^3(3)$
C_1^1	$G_1^1 + R_1^1$	C_1^2	$G_1^2 + R_1^2$	C_1^3	$G_1^3 + R_1^3$

Table 2 Lane 2

Case-1 ($s_2 t_2 \leq I_2$)		*Case-2* ($I_2 < s_2 t_2 < I_2 + r_2 t_2$)		*Case-3* ($s_2 t_2 \geq I_2 + r_2 t_2$)	
Parameter	*Value*	*Parameter*	*Value*	*Parameter*	*Value*
$R_2^1(0)$	$x_2 t_1 + f(r_2, t_1)$	$R_2^2(0)$	$x_2 t_1 + f(r_2, t_1)$	$R_2^3(0)$	$x_2 t_1 + f(r_2, t_1)$
$R_2^1(2)$	$(I_2 - s_2 t_2 + r_2 t_2) t_3 + f(r_2, t_3)$	$R_2^2(2)$	$(rem_i) t_3 + f(r_2, t_3)$	$R_2^3(2)$	$f(r_2, t_3)$
$R_2^1(3)$	$(I_2 - s_2 t_2 + r_2 t_2 + r_2 t_2) t_4 + f(r_2, t_4)$	$R_2^2(3)$	$(rem_i + r_2 t_3) t_4 + f(r_2, t_4)$	$R_2^3(3)$	$(r_2 t_3) t_4 + f(r_2, t_4)$
R_2^1	$R_2^1(0) + R_2^1(2) + R_2^1(3)$	R_2^2	$R_2^2(0) + R_2^2(2) + R_2^2(3)$	R_2^3	$R_2^3(0) + R_2^3(2) + R_2^3(3)$
C_2^1	$G_2^1 + R_2^1$	C_2^2	$G_2^2 + R_2^2$	C_2^3	$G_2^3 + R_2^3$

Table 3 Lane 3

Case-1 ($s_3t_3 \le I_3$)		*Case-2* ($I_3 < s_3t_3 < I_3 + r_3t_3$)		*Case-3* ($s_3t_3 \ge I_3 + r_3t_3$)	
Parameter	*Value*	*Parameter*	*Value*	*Parameter*	*Value*
$R_3^1(0)$	$x_3t_1 + f(r_3, t_1)$	$R_3^2(0)$	$x_3t_1 + f(r_3, t_1)$	$R_3^3(0)$	$x_3t_1 + f(r_3, t_1)$
$R_3^1(1)$	$(x_3 + r_3t_1)t_2 + f(r_3, t_2)$	$R_3^2(1)$	$(x_3 + r_3t_1)t_2 + f(r_3, t_2)$	$R_3^3(1)$	$(x_3 + r_3t_1)t_2 + f(r_3, t_2)$
$R_3^1(3)$	$(I_3 - s_3t_3 + r_3t_3)t_4 + f(r_3, t_4)$	$R_3^2(3)$	$(rem_i)t_4 + f(r_3, t_4)$	$R_3^3(3)$	$f(r_3, t_4)$
R_3^1	$R_3^1(0) + R_3^1(1) + R_1^1(3)$	R_3^2	$R_3^2(0) + R_3^2(1) + R_3^2(3)$	R_3^3	$R_3^3(0) + R_3^3(1) + R_3^3(3)$
C_3^1	$G_3^1 + R_3^1$	C_3^2	$G_3^2 + R_3^2$	C_3^3	$G_3^3 + R_3^3$

Table 4 Lane 4

Case-1 ($s_4t_4 \le I_4$)		*Case-2* ($I_3 < s_3t_3 < I_3 + r_3t_3$)		*Case-3* ($s_3t_3 \ge I_3 + r_3t_3$)	
Parameter	*Value*	*Parameter*	*Value*	*Parameter*	*Value*
$R_4^1(0)$	$x_4t_1 + f(r_4, t_1)$	$R_4^2(0)$	$x_4t_1 + f(r_4, t_1)$	$R_4^3(0)$	$x_4t_1 + f(r_4, t_1)$
$R_4^1(1)$	$(x_4 + r_4t_1)t_2 + f(r_4, t_2)$	$R_4^2(1)$	$(x_4 + r_4t_1)t_2 + f(r_4, t_2)$	$R_4^3(1)$	$(x_4 + r_4t_1)t_2 + f(r_4, t_2)$
$R_4^1(2)$	$(x_4 + r_4t_1 + r_4t_2)t_3 + f(r_4, t_3)$	$R_4^2(2)$	$(x_4 - r_4t_1 + r_4t_2)t_3 + f(r_4, t_3)$	$R_4^3(3)$	$(x_4 - r_4t_1 + r_4t_2)t_3 + f(r_4, t_3)$
R_4^1	$R_4^1(0) + R_4^1(1) + R_4^1(2)$	R_4^2	$R_4^2(0) + R_4^2(1) + R_4^2(2)$	R_4^3	$R_4^3(0) + R_4^3(1) + R_4^3(2)$
C_4^1	$G_4^1 + R_4^1$	C_4^2	$G_4^2 + R_4^2$	C_4^3	$G_4^3 + R_4^3$

Here, $I_i \rho_i^{n_i} <= 1$

$$\theta_i = \left(t_i - \frac{I_i}{s_i} - \frac{I_i r_i}{s_i^2} - \ldots - \frac{I_i r_i^{n_i - 2}}{s_i^{n_i - 1}} \right) s_i$$

$$\phi_i = \left(t_i - \frac{I_i}{s_i} - \frac{I_i r_i}{s_i^2} - \ldots - \frac{I_i r_i^{n_i - 1}}{s_i^{n_i}} \right) s_i$$

subject to the constraints,

$$\sum_{i=1}^{4} t_i = CT \tag{33}$$

$$t_i >= MT \tag{34}$$

where CT is the cycle time and MT is the minimum time to be allocated to each lane. Also, using (19) and (30), it can be shown that,

$$n_i = \begin{cases} \left\lceil \dfrac{\log\left(\dfrac{(\rho_i - 1) s_i t_i + I_i}{I_i (2 + \rho_i)} \right)}{\log(\rho_i)} \right\rceil & x_i < s_i t_i < I_i + r_i t_i \\ \left\lceil \dfrac{-\log(I_i)}{\log(\rho_i)} \right\rceil & s_i t_i \ge I_i + r_i t_i \end{cases} \tag{35}$$

2.5 Estimating arrival and service rate values

In a practical scenario, the arrival and service rates for each lane are not available at the start of the cycle. We assume that the true rates of the lanes, for a particular cycle, are available only after the cycle is finished. Hence, the rates must be forecast using the values from the previous iterations. For doing so nonlinear autoregressive neural networks, N_i^r and N_i^s are trained for each lane, *i*. Here *r* signifies the neural network trained for arrival rates, and *s* signifies the neural network trained for service rates. We therefore get eight neural networks in total. To ensure that the process of estimation is not computationally expensive, each neural network has a single hidden layer, with ten neurons. We use sigmoid activation functions for the hidden layer. The architecture of the neural networks is shown in Figure 4.

We now describe the process of training the neural network, N_i^v, where $0 < i \le 4$, and $v \in \{r, s\}$. Let us denote v_i^m, as the rate for lane *i*, for the m^{th} iteration. For the first four iterations in lane *i*, we use v_i^{m-1}, as the rate for the m^{th} iteration, v_i^1, being a random number in the range [0, 5].

In order to estimate the rates in the subsequent iterations, the following method is adopted. At each iteration *m*, where $m > 4$, the neural network, N_i^v is trained. Let us denote the set of true rates, $V_i^m = (v_i^x : 0 < x < m-1\}$. These rates are available at the start of the m^{th} iteration.

Let $A_i^j = \{v_i^k : j-4 \le k < j\}$, and $B_i^j = (A_i^j, v_i^j)$, where A_i^j, is the input to the network, and v_i^j, is the label/ground truth. Using these values, we create a dataset, denoted by D_i^m, such that $D_i^m = \{B_i^j : q < j < m\}$, which is used to train the network N_i^v, at iteration *m*. Hence, at iteration *m*, $N_i^v : A_i^m \to \hat{v}_i^m$(as shown in Figure 4), where $\hat{v}_i^m$ is the predicted rate. The value *q* is adjusted so that the training set size does not exceed a threshold, which we set as 1,000. The Levenberg-Marquardt method, as described in Mor (1978), was used for optimisation of the cost function.

Figure 4 Neural network architecture

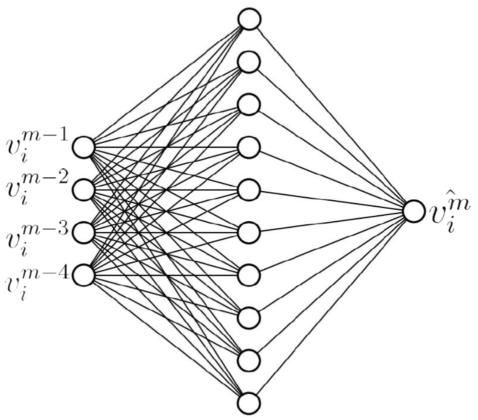


Table 5 Neural network vs. weighted averaging

Test case number	*Data*	*Number of iterations, iter*	*Average error*	
			Using weighted averaging	*Using neural network*
1	$2rand^{(2,2.5)} + rand(0, 1) + sin(m)$	50	0.99	0.71
2	$2rand^{(2,2.5)} + rand(0, 1)$	50	0.33	0.304
3	$2rand^{(2,2.5)} + rand(0, 1) + (m/10)^2$	50	1.15	0.67

2.5.1 *Evaluation of the neural network*

We use a weighted averaging method as a baseline for evaluating the neural network. Essentially, the baseline method obtains:

$$\hat{v}_i^m = v_i^{m-1} + 0.5 * v_i^{m-2} + 0.25 * v_i^{m-3} + 0.125 * v_i^{m-4}.$$

Similar to our proposed method, we use random values for v_i^m, in the first four iterations.

In Table 5, we show that the neural network, achieves lesser average error than the baseline method, particularly in test case 1 and test case 3, where v_i^m shows large fluctuations. The function $rand(a, b)$, outputs a random value in the range $[a, b]$. Figure 5 shows the arrival rate vs. iteration plots for test case 1, where it can be seen that the baseline method fails to capture the pattern in the time series.

3 Results

The proposed approach was tested by simulating a traffic signal using MATLAB code. In a more realistic scenario, however, the proposed technique could be modified, to use per person delay instead of per vehicle delay, as done in Jiao et al. (2015). The interior point optimisation function of fmincon tool of MATLAB (2017) has been used for optimisation. This method is described in Byrd et al. (2000).

Figure 5 Neural network results, (a) moving average (b) NN prediction (see online version for colours)

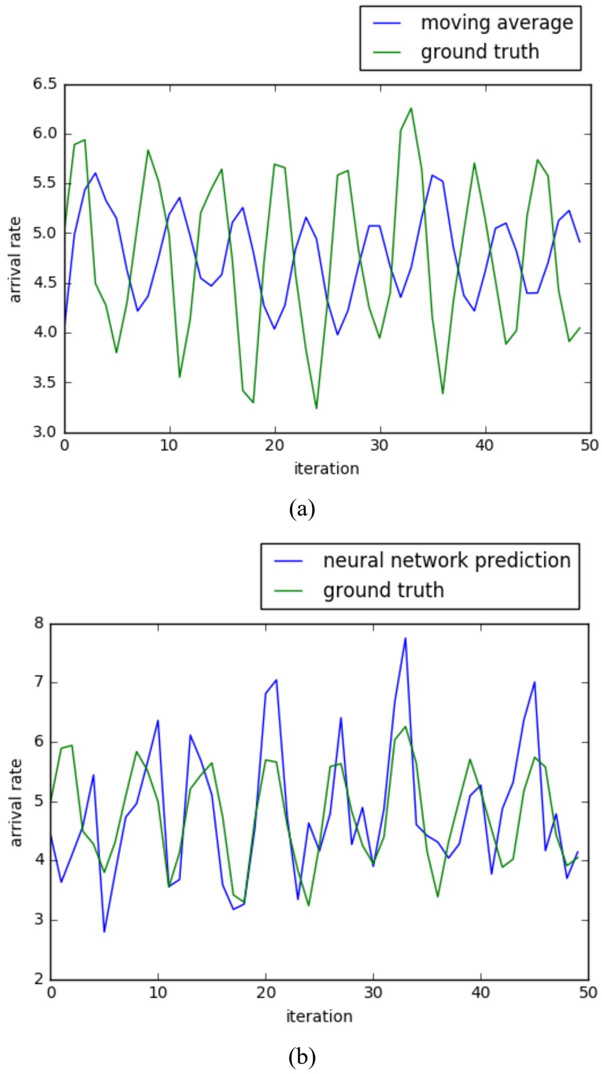


Figure 6 An isolated intersection (see online version for colours)

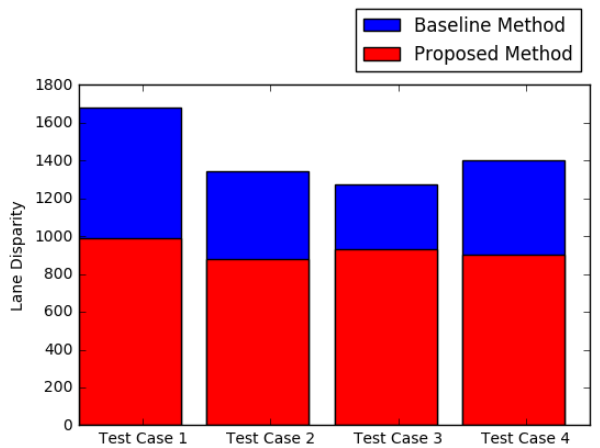
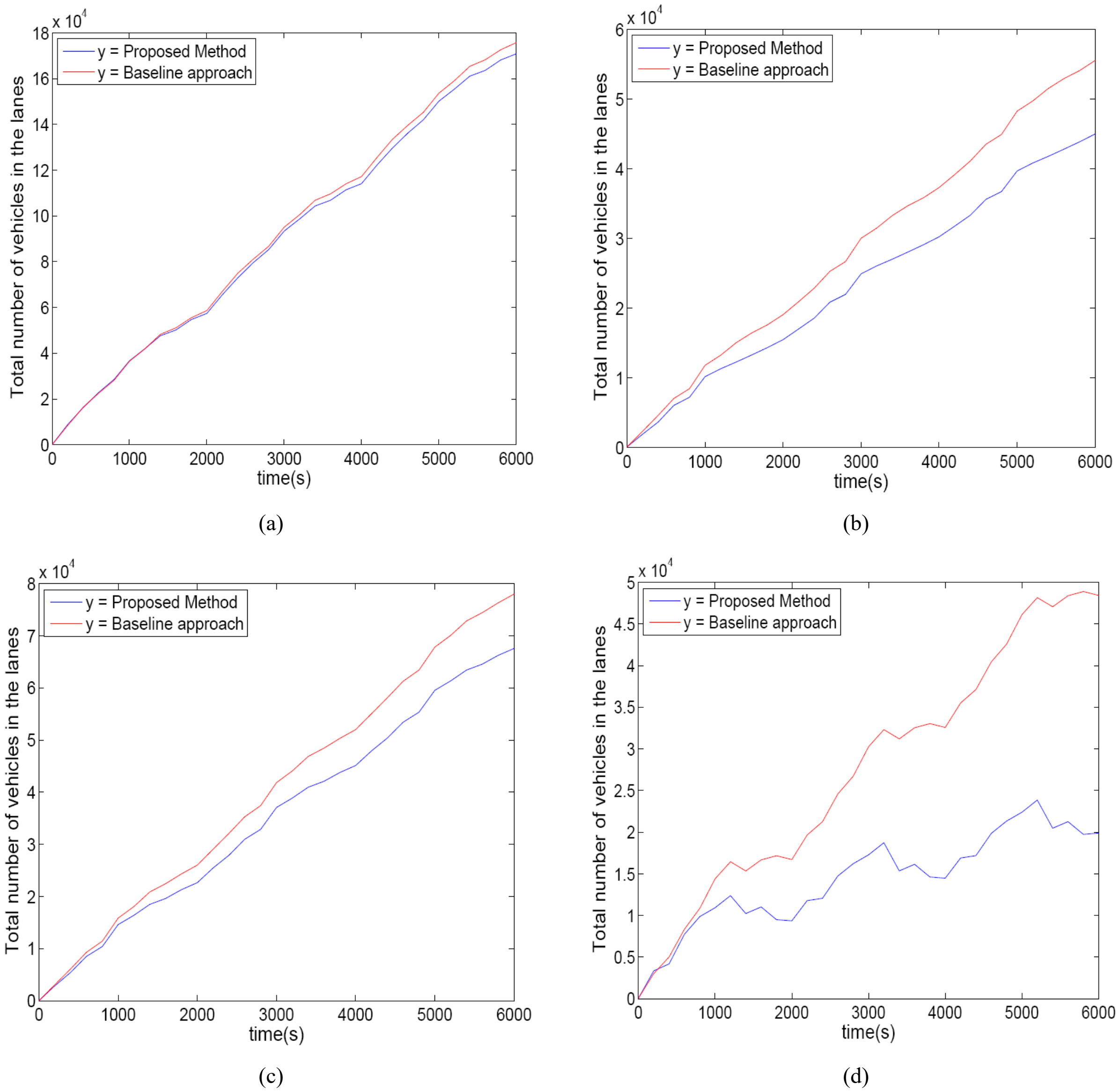


Table 6 Results: case 1 has $r_i^m > s_i^m$, case 2 has $r_i^m < s$, case 3 has $r_i^m \approx s_i^m$, and case 4 does not impose any constraint

Test case number	*Data*		*Total number of vehicles after 30 iterations*	
	r_i^m	s_i^m	*Baseline method*	*Proposed method*
1	$r_i^m = 2^{rand(2,2.5)} + rand(0, 1) + rand(2.5, 3) * sin(m)$	$s_i^m = r_i^m + rand(1, 2)$	175,738	170,816
2	$r_i^m = 2^{rand(2,2.5)} + rand(0, 1) + rand(2.5, 3) * sin(m)$	$s_i^m = \max\left(0, r_i^m - rand(1, 2)\right)$	55,540	44,986
3	$r_i^m = 2^{rand(2,2.5)} + rand(0, 1) + rand(2.5, 3) * sin(m)$	$s_i^m = r_i^m + rand(-0.3, 0.3)$	77,995	67,581
4	$r_i^m = 2^{rand(2,2.5)} + rand(0, 1) + rand(2.5, 3) * sin(m)$	$s_i^m = 2^{rand(2,2.5)} + rand(0, 1) + rand(2.5, 3) * sin(m)$	48,437	19,899

Figure 7 Plots of results, (a) test case 1 (b) test case 2 (c) test case 3 (d) test case 4 (see online version for colours)



The results are shown in Table 6, in which the proposed method is analysed for different relationships between the arrival and service rates. In particular, we evaluate our method on cases where the arrival rates are greater than, less than, comparable to the service rate, with random noise being added in each test scenario. In addition, results are shown for the case in which no constraint is imposed on the arrival and service rates. The proposed model is contrasted against a baseline approach, which divides the available cycle time among the lanes, in proportion to the number of vehicles that were present in the lanes at the start of each cycle. As shown in the table, the proposed approach easily outperforms the baseline approach in all cases, as it finds the optimal time distribution. Table 7 shows the initial lane configuration, for each of the test cases. Plots for the test cases in Table 6 are shown in Figure 7. It can be observed that the proposed method consistently maintains lower number of total vehicles, irrespective of the relationship between the arrival and service rates.

In order to evaluate the performance of our method on load balancing, we use the following method. Let us denote the number of vehicles present in lane *i*, at the end of the m^{th} iteration to be x_i^m. The lane disparity after *m* iterations is $LD_m = std(\{x_i^m : 1 < i <= 4\})$, where the function *std* finds the standard deviation of a set of values. In Figure 6, it can be observed that the proposed method obtains a lower lane disparity than the baseline method, on all the four test cases, which shows that the method implicitly balances the load among the lanes.

Also, the model developed still requires to be tested on a real traffic signal, where the arrival and service rates are obtained using computer vision techniques as done in Hsieh et al. (2006).

Table 7 Lane initialisation for the experiments in Table 6

Test case number	1	2	3	4
$[x_1, x_2, x_3, x_4]$	[5, 18, 19, 31]	[5, 80, 10, 7]	[5, 8, 9, 3]	[10, 3, 4, 36]

4 Conclusions

This paper presents an approach to solve the traffic scheduling problem, and finds the optimal distribution of time for a cycle, by minimising the total waiting time. In addition, it was established that the relationship between the arrival rate and the service rate, decides the form of the objective function. A robust model was developed, by using a neural network to forecast the arrival and service rates, as a time series. In effect, the model continuously adapts to changing conditions. It was found that the model performs particularly well, when the service and arrival rates vary rapidly from cycle to cycle. Also, the model can adapt in real-time, as it finds a new optimal time distribution for each cycle.

References

Aljaafreh, A., Al-oudat, N. and Saleh, M. (2014) 'Adaptive traffic-signal control using discrete event simulation model', *International Journal of Computer Applications*, Vol. 101, No. 12, pp.7–12.

Boussoufa-Lahlah, S., Semchedine, F. and Bouallouche-Medjkoune, L. and Farhi, N. (2018) 'PSCAR: a proactiveoptimal-path selection with coordinator agents assisted routing for vehicular ad hoc networks', *International Journal of High Performance Computing and Networking*, Vol. 11, No. 2, pp.129–144.

Byrd, R.H., Gilbert, J.C. and Nocedal, J. (2000) 'A trust region method based on interior point techniques for nonlinear programming', *TEM Journal*, Vol. 89, No. 1, pp.149–185.

De Schutter, B. (1999) 'Optimal traffic light control for a single intersection', *Proceedings of the American Control Conference*, Vol. 3, pp.2195–2199.

Fan, W., Zhang, H., Li, K.C., Zhang, S., Marino, M.D. and Jiang, H (2018) 'An efficient algorithm for modeling and dynamic prediction of network traffic', *International Journal of Computational Science and Engineering*, Vol. 16, No. 3, pp.311–320.

Ge, Y. (2014) 'A two-stage fuzzy logic control method of traffic signal based on traffic urgency degree', *Modelling and Simulation in Engineering*, pp.41–46, Hindawi Publishing, Article ID: 694185.

Genders, W. and Razavi, S. (2016) *Using a Deep Reinforcement Learning Agent for Traffic Signal Control* [online] https://arxiv.org/pdf/1611.01142.pdf (accessed 7 December 2017).

Gu, L., Guo, H. and Liu, X. (2017) 'Fuzzy time series forecasting based on information granule and neural network', *International Journal of Computational Science and Engineering*, Vol. 15, Nos. 1–2, pp.146–152.

Hsieh, J., Yu, S., Chen, Y. and Hu, W. (2006) 'Automatic traffic surveillance system for vehicle tracking and classification', *IEEE Transactions on Intelligent Transportation Systems*, Vol. 7, No. 2, pp.175–187.

Jha, M. and Shukla, S. (2016) *Design of Fuzzy Logic Traffic Controller for Isolated Intersections with Emergency Vehicle Priority System using MATLAB Simulation* [online] https://arxiv.org/pdf/1405.0936.pdf (accessed 5 December 2017).

Jiao, P., Li, Z., Liu, M., Li, D. and Li, Y. (2015) 'Real-time traffic signal optimization model based on average delay time per person', *Advances in Mechanical Engineering*, Vol. 7, No. 10, pp.1687814015613500.

Koltovska, D. and Bombol, K. (2014) 'Intelligent agent based traffic signal control on isolated intersections', *TEM Journal*, Vol. 3, No. 3, pp.216–222.

Lee, J., Abdulhai, B., Shalaby, A. and Chung, E. (2007) 'Real-time optimization for adaptive traffic signal control using genetic algorithms', *Journal of Intelligent Transportation Systems*, Vol. 9, No. 3, pp.111–122.

Liu, D., Xiao, X. and Zeng, X. (2018) 'Load balancing algorithm based on multiple linear regression analysis in multi-agent', *International Journal of Computational Science and Engineering*, Vol. 16, No. 3, pp.234–241.

Mor, J.J. (1978) 'The Levenberg-Marquardt algorithm: implementation and theory', in *Numerical Analysis*, pp.105–116, Springer, Berlin, Heidelberg.

Taghvaeeyan, S. and Rajamani, R. (2014) 'Portable roadside sensors for vehicle counting, classification, and speed measurement', *International Journal of Control*, Vol. 15, No. 1, pp.73–83.

Toulni, H. and Nsiri, B. (2018) 'Cluster-based routing protocol using traffic information', *International Journal of High Performance Computing and Networking*, Vol. 11, No. 2, pp.108–116.

Tourani, A. and Shahbahrami, A. (2015) 'Vehicle counting method based on digital image processing algorithms', Paper presented at the *2nd International Conference on Pattern Recognition and Image Analysis* (*IPRIA*), 11–12 March, Rasht, Iran.