Automating Traffic Signals

Shubham Luhadia
Roll No:11010176
Indian Institute of Technology
Guwahati

Soumak Datta
Roll No:11010180
Indian Institute of Technology
Guwahati

Simrat Singh Chhabra
Roll No:11010165
Indian Institute of Technology
Guwahati

ABSTRACT

With growing urbanization, a problem that has garnered much attention all over the world is *Traffic Congestion*. Most of the existing solutions require either huge amounts of time and money or efforts of highly trained personnels with domain knowledge. In this, a real-time, adaptive, software-based approach implementing *Reinforcement Learning* seems as the most useful solution. Using this technique Traffic Signals could be automated so that they adapt to the traffic, in real-time, for easing congestion problems. We look at this solution in detail, implement it for a traffic model and compare it with the constant-timed traffic signal approach. Also, we extend it further to include varying traffic trends and attempt to adapt our intelligent signal to respond to these trends gracefully.

Keywords

Adaptive Traffic Signals, Intelligent Driver Model, Reinforcement Learning, Q-Learning, Simulated Annealing

1. INTRODUCTION

The increased Vehicular Queueing that occurs on road networks leading to longer waiting periods is called as Traffic Congestion[1]. This leads to not only delays and loss of potentially fruitful human hours but also pollution, fuel wastage and as multiple cases suggest, road rage. The solutions offered mostly include Road Infrastructure Enhancement, increasing reliance on public transport (like buses, car pools), constructing alternatives (metro). But all these require lots of time and financial capital, as well as, trained professionals with domain expertise.

So, at this juncture, Adaptive Traffic Signals are a much better option. By making the traffic lights sensitive to the traffic demands, more intelligent decisions can be made leading to less inequities in the waiting periods experienced by different drivers.

One of the foremost challenges that this solution faces is that for feasible implementation, often assumptions and idealizations need to be considered that take them far from the real picture[3]. This happens in order to accommodate the need for a model that would effectively define the Problem Environment. But such a model description requires an experienced, trained personnel. Moreover, it is also possible that traffic patterns might not be clear in all places. Along with this, the model has to satisfy the demand of generality so that separate models are not required for each particular situation.

What we actually need is a software-based approach that does not require a pre-specified model but determines the inherent associations in the environment through its experiences in real time. *Reinforcement Learning* is one such domain of Artificial Intelligence that provides these benefits. Also, it has the major advantage of being unsupervised - it does not require a plethora of training examples to gain insight into the inherent connections but instead utilizes hit-and-trial mechanisms. It is adaptive in the sense that it is capable of responding to even environment conditions changing dynamically[1]. We will look into Q-learning, the Reinforcement Learning algorithm considered by us.

2. RELATED WORK

In order to get a baseline on the work done in this field, we went through the related work. In 2003, Abdulhai et. al. [1] proposed using the technique of Q-Learning for the purpose of training truly adaptive traffic signals. they were among the first to recognize the strengths of Reinforcement Learning(RL) for such problems which put forward a difficulty in modelling. They model the traffic such that the inter arrival times follow the Poisson[2] distribution. There model works well but it has not been tested on a trusted traffic model. Moreover, they don't attempt to study the effects of traffic following time-varying distributions.

In 2006, Oliveira et. al. [3] put forward implementation and analysis of different algorithms under RL for tackling the problem of Adaptive Traffic Control. This basically stemmed from the idea that in case of a non-stationary environment like traffic, multiple partial models could be used where each model prioritizes one kind of flow in the complex interaction of multiple flows. But this not only increases the complexity of the implementation but also the converging time and its responsiveness to traffic changes.

Other studies [2] have been found to use Q-Learning where the dependence on the behaviour of the driver is considered important. But that subverts the actual reason for the choice of Q-Learning as an algorithm - to avoid modelling idealisations.

Although majority of the works in this field focus on RL, there have been a few cases where the application of an RL algorithm has failed to give satisfactory results and people have tried out other methods like linear regression [5]. However, one of the main reasons for its failure is the improper

[1] http://en.wikipedia.org/wiki/Traffic_congestion

[2] http://mathworld.wolfram.com/PoissonDistribution.html

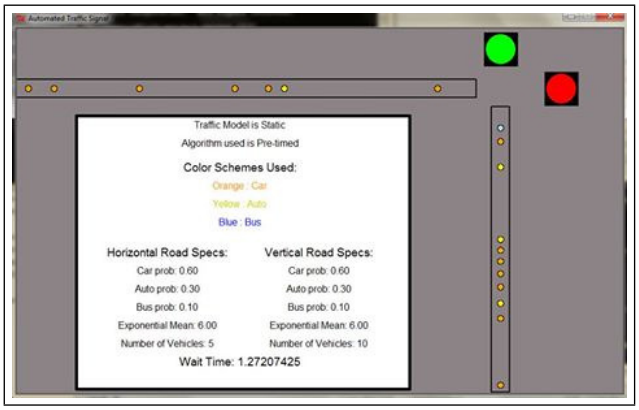

Figure 1: GUI of our Q-Learning Implementation

choice of optimization function, which showcases the sensitivity of such algorithms to the parameter that is to be optimized.

Our approach to alleviating the problem of traffic congestion relies on the implementation of Q-Learning approach(fast and adaptive based on the primer specified by Harmon et. al. [4]), on a realistic traffic model (See Figure 1). Section 3 discusses our traffic model, Section 4 presents an overview on the Q-Learning technique and Section 5 extends its implementation to the purview of Traffic control, Section 6 chalks out the performance of our implementation and finally Section 7 concludes the report.

3. TRAFFIC MODEL

Before applying any adaptive methods to the traffic signals, an appropriate model for the traffic intersecting at a road junction needs to be chosen. For this purpose, we use the Intelligent Driver Model[6] which had been developed in 2000 by Treiber et al. It is based on actual data collected from several German freeways. This model shows the behavior of individual vehicles on a particular road as per the formula :

$$\frac{dv_\alpha}{dt} = a\left(1 - \left(\frac{v_\alpha}{v_0}\right)^4 - \left(\frac{s^*(v_\alpha, \Delta v_\alpha)}{s_\alpha}\right)^2\right) \quad (1)$$

where

$$s^*(v_\alpha, \Delta v_\alpha) = s_0 + v_\alpha T + \frac{v_\alpha \Delta v_\alpha}{2\sqrt{ab}}$$

In (1), v_α denotes velocity of vehicle α, Δv_α is its velocity difference(approach rate) to the preceding vehicle and s_α is the gap to the preceding vehicle. The constant T (set to 1.6 secs) denotes the safe time headway and the Jam distance (s_0) is set to 2 m .

We have considered 3 different types of vehicles (Car, Auto and Bus) for efficient modelling of a real-time traffic scenario. The values of the other parameters in (1) are shown in Table 1 for different vehicles.

Our traffic model can be used for modelling any number of roads intersecting at a traffic junction and all these roads can have different traffic distribution. However, for simplicity and initial study, we will consider only 2 one-way one-lane roads intersecting at a road junction through this paper. For traffic on a road, we will consider 2 different types of distribution:

Vehicle	Desired velocity(v_0) m/s	Maximum acceleration(a) m/s^2	Desired deceleration(b) m/s^2
Car	33.33	0.73	1.67
Auto	20	0.73	1.67
Bus	15	0.73	1.67

Table 1: Parameters for Different Vehicles

- **Static Distribution:** In this distribution, the probability of a vehicle being car,auto or bus is fixed and their values are read from a specification file. The interarrival time for the different vehicles is chosen as an exponential distribution with a given mean, also read from the specification file.
- **Time-varying Distribution:** Here, the probability of a vehicle being car,auto or bus is not fixed but varies as sine functions of time (with sum of the probabilities being 1). The interarrival time is also modelled similarly, with the mean varying as sine functions of time. This distribution is complex to analyze but is more closer to how traffic varies throughout the day.

4. Q - LEARNING

Reinforcement Learning(RL), paradigm stemming from Dynamic Programming, calls for an agent to be just presented with the goal. Then with no further information about the environment, the agent follows trial-and-error tribulations to gather its own information about the complex relationships that might exist in the environment. It uses the feedback(reinforcement) provided by the environment to determine the contribution of its chosen actions in approaching goal.

On of the most important RL techniques is Q-Learning. Its key elements[1] are:

- **Markovian Process:** The agent receives the state, $\boldsymbol{s}$ at time t and together with any action, it is sufficient to determine the next state. No past history of states needed to transition to next state.
- **Q-Values:** For any state, the action $\boldsymbol{a}$ to be selected is chosen based on Q-values i.e. estimates of value of $(\boldsymbol{s,a})$ to achieve the goal. Initially all the Q-values are null, and are gradually filled as the agent progresses. A gradual shift is engineered from *exploration*(choose actions more randomly) to *exploitation*(choosing action with best estimates) as the Q-values start converging.
- **Reinforcement:** On application of action $\boldsymbol{a}$ on state $\boldsymbol{s}$, the environment returns a reward/penalty $\boldsymbol{r(s,a)}$. Ultimate aim is to optimize this feedback.
- **Update Rule:** The Q-Values are updated using the equation (2):

$$\Delta = \alpha_{s,a}\Big\{r(s,a) + \gamma_t \cdot OPT[Q_{t-1}(s',a')] - Q_{t-1}(s,a)\Big\} \quad (2)$$

where

$$Q_t(s,a) = Q_{t-1}(s,a) + \Delta$$

In (2), Δ is the increment to $Q_{t-1}(s,a)$ to get updated Q-Value, $Q_t(s,a)$. $\alpha_{s,a}$ is the training rate, annealed from

high to a low value for Q-Values to converge, ***r(s,a)*** is the reinforcement for taking action $\boldsymbol{a}$ in state $\boldsymbol{s}$, γ_t is the discount rate to bound the estimate, and $\mathbf{OPT}[Q_{t-1}(s', a')]$ refers to the optimum (**MAX** for rewards and **MIN** for penalties) path found till time t-1 from state ***s'*** (consequence of taking $\boldsymbol{a}$ in $\boldsymbol{s}$).

Finally, in the exploitation phase, on converging of the Q-Values, the actions with optimum reinforcements are chosen to get a sequence of actions taking to the goal.

5. Q-LEARNING FOR TRAFFIC SIGNALS

For implementing Q-learning on our traffic model (Section 3), we adopted the following approach:

5.1 States

We considered the ordered tuple of the number of cars waiting on each road at any time as a state. For ease in processing, the number of cars were divided into mutually exclusive ranges. For example, in our 2-roads model, if road 1 has 18 cars and road 2 has 25, the state is $\boldsymbol{s}$ = (1,2) [in case of modulo 10 ranges].

5.2 Actions

In any state the agent can decide what the signal should be on each road and for how long it should remain so (before again taking an action). For example, if the signal is (1,0) i.e. Green for road 1 and Red for road 2 and the time it would remain so is 5 seconds, then, our action is, $\boldsymbol{a}$ = {(1,0),5} .

5.3 Transition Function

Given state $\boldsymbol{s}$ and action $\boldsymbol{a}$, the next state can be determined by giving the appropriate signal as mentioned in $\boldsymbol{a}$ for duration (also mentioned in $\boldsymbol{a}$) to our traffic model which varies the car speeds accordingly, and accounting for the waiting cars at the end of the period to get the new state ***s'***.

5.4 Q-Values

Initially all the $Q_0(s, a)$ values are given a low value of 0. For maneuvering the transition from Exploration to Exploitation, we combine the Soft-Max[3] principle with Simulated Annealing[4]. The temperature $\boldsymbol{T}$ follows a geometric schedule i.e. $T_{new} = \beta \cdot T_{old}$ where β is kept at 0.995. First an ϵ is chosen as,

$$\epsilon = \frac{e^{-1/T}}{e^{-1/T_{max}}} \tag{3}$$

this is used to decide whether to choose from one of the already explored actions or explore a new one with probability ϵ. In case of an already explored one, the soft-max principle is employed whereby the action is chosen with probability p(s,a),

$$p(s, a) = \frac{e^{\frac{-Q[s,a]}{T}}}{\sum_a e^{\frac{-Q[s,a]}{T}}} \tag{4}$$

As the value of $\boldsymbol{T}$ decreases, $e^{-1/T}$ decreases and hence ϵ decreases. Also, p(s,a) of lower Q[s,a] would be higher. So exploitation is preferred over exploitation.

[3] `http://artint.info/html/ArtInt_266.html`
[4] `http://artint.info/html/ArtInt_89.html`

5.5 Reinforcement

We consider the sum of cumulative waiting period of all cars on a road, over all roads and this is divided by the duration specified by the action. This is our reinforcement. Since we want waiting period to be minimized, so it is in fact a penalty(this explains the negative sign in 4). So that frequent Green/Red swaps(i.e. extremely short duration signal changes) are not given preference, hence the division by action duration is done. Also, the average is not the linear mean but power mean, i.e. the waiting period of a car is squared rather than incrementing by 1 so as to deter long queues.

5.6 Update Rule

α is annealed as ϵ so that convergence is facilitated. $\mathbf{OPT} = \mathbf{MIN}$ as the reinforcements are penalties. Discount rater, γ_t is fixed at 0.8.

5.7 Accommodating Time-varying Distribution

This implementation of Q-learning is extended by us to even incorporate the Time-varying traffic distribution. For this, we monitor the variance of the Total delay (power averaged) as the agent progresses. When the variane remains below a threshold for consecutive 12 episodes (of 5 min each), the the agent is said to be in steady state. If in this steady state, the variance starts increasing (for 12 consecutive episodes), then the traffic distribution is assumed to have changed and the temperature $\boldsymbol{T}$ is increased to further explore the possible actions that might again bring back the steady state.

6. PERFORMANCE EVALUATION

We evaluated the performance of our Automated Traffic Signal implementation under various scenarios. We use Total waiting time (power-aged) is chosen as our performance metric so that longer a vehicle waits more is its contribution to the metric. Also, ultimate aim is to minimize the Total waiting time.

1. We run our Q-learning algorithm on the static traffic model with 2 roads having different distributions (mean inter arrival times as 5 and 10 sec) and compare it with the basic pre-timed case (first with both roads having equal weights, 15sec and then with both having unequal weights 30 and 15 sec respectively so that the road 1 gets more Green signal time as compared to road 2 in the second case). Q learning works much better than the pretimed(equal) approach. As of pretimed(unequal) approach, as the ratio of the two weights of the roads are equal to the ratio of means of inter arrival time, it works really well and our Q-learning approach, after stabilizing, performs, at least as good as it.(See Figures 2,3)

2. We test which of the two approaches, segregating no of vehicles into ranges on modulo-10 basis(0-9:0, 10-19:1 etc) or power-of-2 basis(1:0, 2-3:1, 4-7:2, 8-15:3 etc). Averaged over 5 runs, power-of-two works considerably better than modulo-10 approach.(See Figure 4)

3. With both roads having exactly same distributions, we run the Q-learning and The Pretimed(equal) algorithms. In the exploration phase, former makes certain bad decisions resulting in a little large total delay

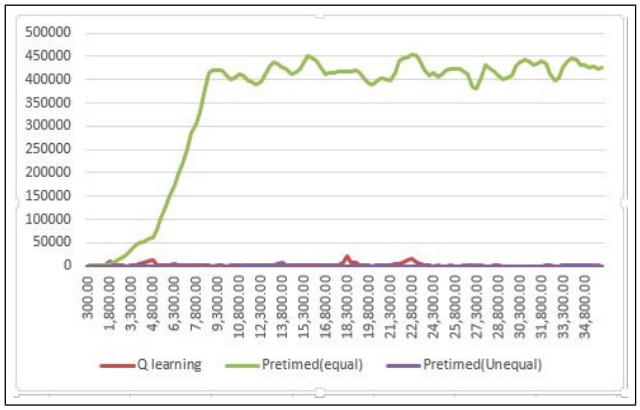

Figure 2: Evaluation 1: Q-learning, Pretimed(equal) and Pretimed(Unequal)

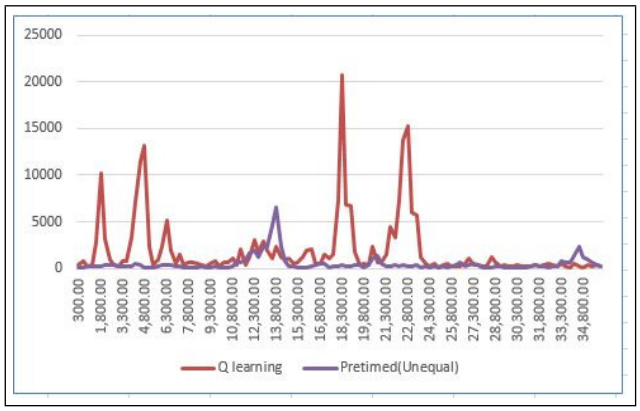

Figure 3: Evaluation 1: Q-learning and Pretimed(Unequal)

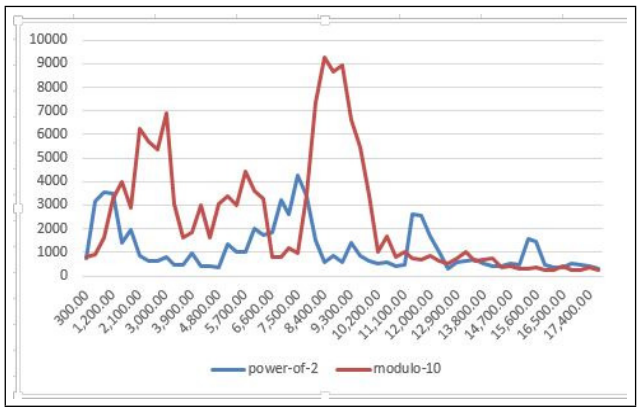

Figure 4: Evaluation 2: state segregation by power-of-2 and modulo-10

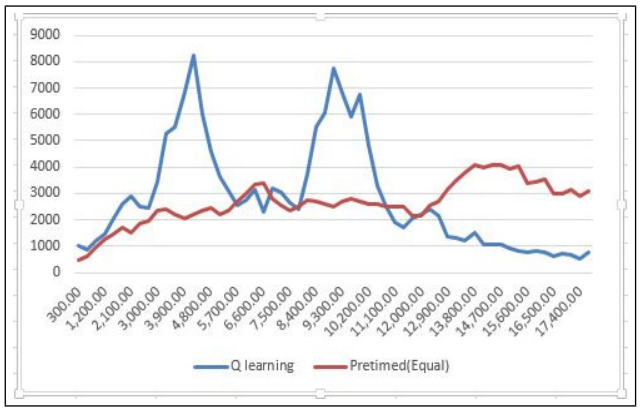

Figure 5: Evaluation 3: Q-learning and Pretimed(Equal)

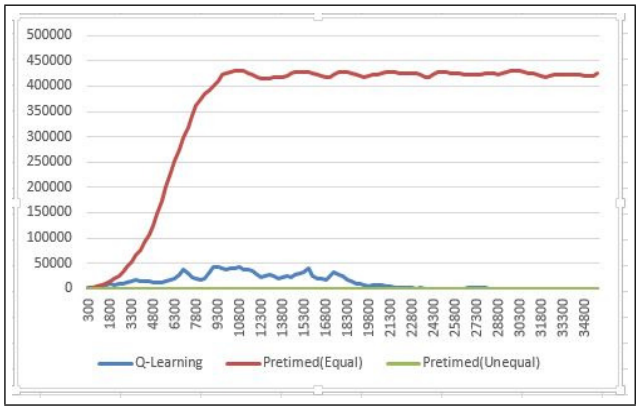

Figure 6: Evaluation 4: Q-learning, Pretimed(equal) and Pretimed(Unequal)

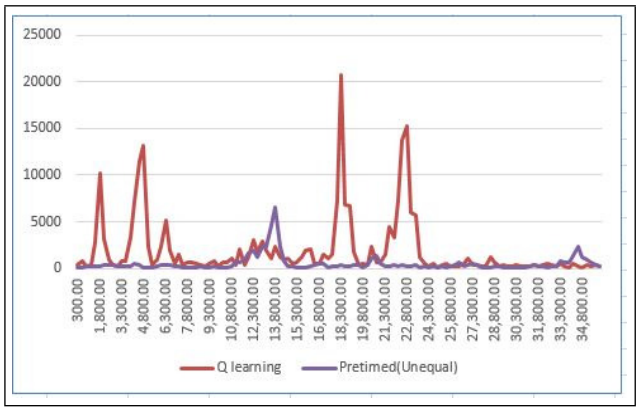

Figure 7: Evaluation 4: Q-learning and Pretimed(Unequal)

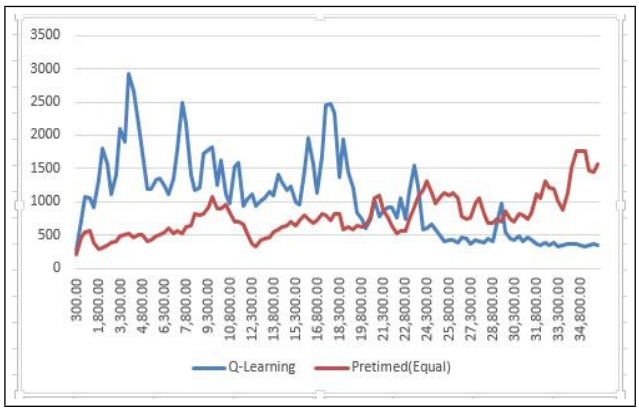

Figure 8: Evaluation 5: Q-learning and Pretimed(equal)

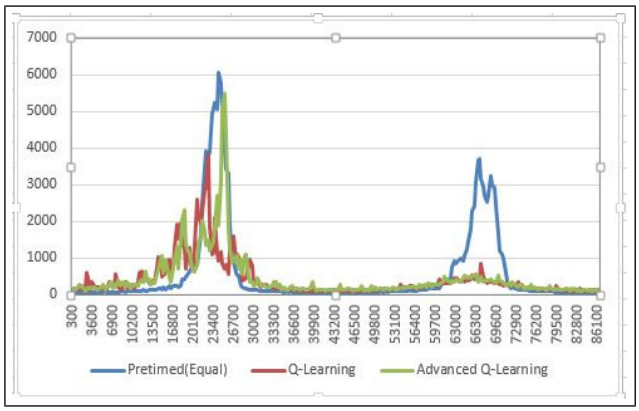

Figure 9: Evaluation 6: Pretimed(Equal), Q-learning and Advanced Q-learning

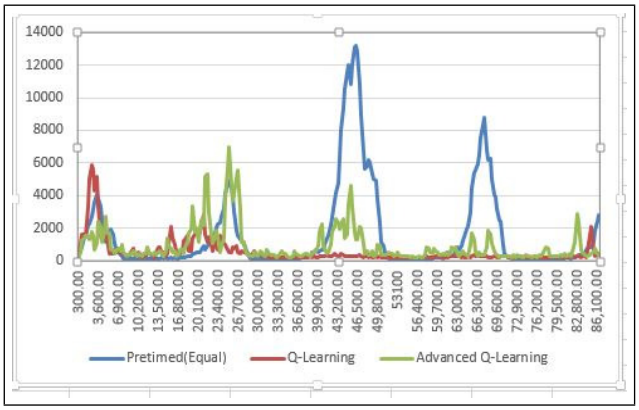

Figure 10: Evaluation 7: Q-learning, Pretimed(equal) and Pretimed(Unequal)

(higher no. of peaks due to averaging over 20 runs) but it stabilizes at a faily low value, while latter starts better but keeps on aggravating the situation, ending much worse. (See Figure 5)

4. We repeat the same experiment as 1 above but now average it over 20 tests so as to remove any bias. Again the same results are obtained.(See Figures 6,7) Notice the larger peaks in Figure 6 are due to different positions of peaks in different tests.

5. Keeping the same distribution on both the roads, we make the vehicle composition different. While probabilities of a new vehicle being a car or an auto rickshaw or a bus is 0.6, 0.3 0.1 respectively, on road 1, it is 0.1, 0.3, 0.6 on road 2. With different vehicles having different lengths, and optimal speeds, this affects the whole process. The time required to stabilize increases although over the long course (See Figure 8), it works better (averaged over 20 runs).

6. We use the Time-varying distribution with both the roads being in phase and test the Pretimed(equal), Q-Learning and advanced Q-Learning approached over it. We observe that while Q-learning and Advanced Q-Learning behave almost the same, the Pretimed(Equal) approach also works pretty good except at peaks of the sine-curve where it responds weakly.(See Figure 9)

7. This time the Time-varying distribution is used with both the roads out of phase. Pretimed(equal), Q-Learning and advanced Q-Learning are tested. Q-Learning performs the best, Pretimed(Equal) responds with great peaks in Total Waiting times. This also shows that Advanced Q-Learning performs, although much better than pretimed approach but less than Q-learning when the distribution changes gradually. (See Figure 10)

7. CONCLUSION

We looked at Automating Traffic Signals using the Q-leaning Algorithm, which falls under Reinforcement Learning in Artificial Intelligence paradigm. We developed a comprehensive Traffic Model and tested our implementation of the Q-Learning algorithm against the Pretimed approach. We found out that our approach, after reaching its exploitation phase, performs at least as well as former and much better in most of the cases. In case of changing traffic(evolving goal) conditions, Q-Learning far outperforms the basic approach. We also implemented an Advanced Q-Learning algorithm to deal with Time-varying traffic distribution. But when the variation in traffic is gradual, Q-Learning works better than the Advanced version.

Possible future work could include adding more generalizations to the Traffic Model considered, so that it could be brought as close to real world as possible. Secondly, more granularity could be considered in terms of states so that better accuracy can be achieved. Other algorithms under the ambit of RL can also be tried. Using Neural Networks[1] or other constructs for reducing the memory requirements of the Algorithm is also a viable future work. And finally, this could be extended to include not only a single traffic signal but a network of signals for an integrated adaptive approach.

8. REFERENCES

[1] B. Abdulhai, R. Pringle, and G. J. Karakoulas. Reinforcement learning for true adaptive traffic signal control. *Journal of Transportation Engineering*, 129(3):278–285, 2003.

[2] T. Chu. Real-time reinforcement learning in traffic signal system.

[3] D. de Oliveira, A. L. Bazzan, B. C. da Silva, E. W. Basso, L. Nunes, R. Rossetti, E. de Oliveira, R. da Silva, and L. Lamb. Reinforcement learning based control of traffic lights in non-stationary environments: A case study in a microscopic simulator. In *EUMAS*, 2006.

[4] M. E. Harmon and S. S. Harmon. Reinforcement learning: A tutorial. *WL/AAFC, WPAFB Ohio*, 45433, 1996.

[5] M. C. Thomas Davids and L. Knepper. Training intelligent stoplights.

[6] M. Treiber, A. Hennecke, and D. Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, 62(2):1805, 2000.