Multi-Agent Learning Systems for Traffic Control

Sayambhu Sen

M.Tech (Systems Engineering) SR-14351

Outline

Introduction

- Reduction of Traffic congestion is essential for a rapidly developing city like Bangalore.
- Traffic Signal Control (TSC) is essential to reduce the average delay experienced by commuters.
- Multi-Agent Reinforcement Learning (MARL) is used for solving the TSC problem.

Defining the MDP problem

- We model our system as a Markov Decision Process(MDP).
- Each individual traffic signal at each junction is modelled as an independent agent.

- A state s^j for a given junction j is given as a vector of dimension $L+1$, where L denotes the number of incoming lanes in that junction.
- The *ith* component of the state vector , $q_i^j, i \in \{1, 2, ...L\}$ denotes the queue length of the traffic in the *ith* lane of that junction.
- The last component q_{L+1}^j denotes, the index of the phase that has been set to green in the round robin (RR) schedule of the traffic controller.

- Thus, the state space of the entire system can be modelled as $S = \bigtimes_{j=1}^{N} S^j$
- In order to further reduce the state space, we discretize the queue lengths and the actions as follows

$$q_i^j(t) = \begin{cases} 0, & \text{if } {q_i^j}' < D1 \\ 1, & \text{if } D1 \leq {q_i^j}' < D2 \\ 2, & \text{if } D2 \leq {q_i^j}' \end{cases} \tag{1}$$

- The action space of each agent is discretized as follows: $low = 10$ seconds, $medium = 20$ seconds, $high = 30$ seconds

- The cost for choosing a certain action at a particular state for a particular agent at a junction is given by:

$$c_j(t) = \frac{1}{|N_j|} \sum_{k \in N_j} \sum_{i=1}^{L_k} q_i^k(t+1) \tag{2}$$

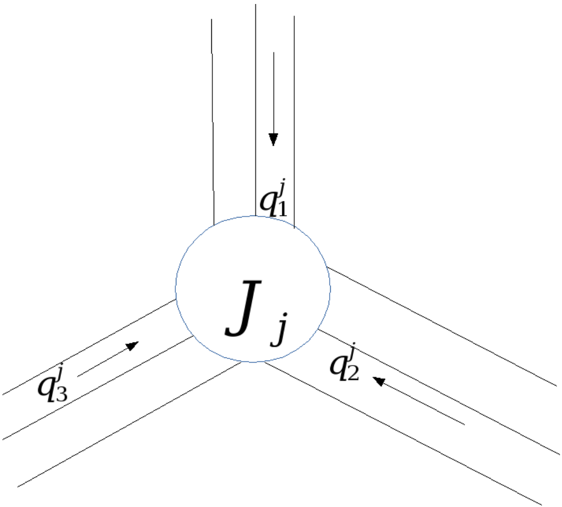

Figure 1: A simple 3 junction road network considered with the junction captioned with Jj.

Q Learning Algorithm

- The Q learning algorithm for the single agent system.

$$Q_{t+1}(s,a) = Q_t(s,a) + \gamma(t)(c(t) + \alpha \min_{b \in A} Q_t(s',b) - Q_t(s,a)) \tag{3}$$

- The Q Learning algorithm for the Multi-Agent system. (*Prabhuchandran K.J. et. al*)

$$Q^j_{t+1}(s^j,a^j) = Q^j_t(s^j,a^j) + \gamma(t)(c_j(t) + \alpha \min_{b \in A} Q^j_t(s^{j\prime},b) - Q^j_t(s^j,a^j)) \tag{4}$$

- The step sizes $\gamma(t), t \geq 0$ should satisfy the requirement that $\gamma(t) > 0, \forall t$ and that

$$\sum_t \gamma(t) = \infty, \sum_t \gamma^2(t) < \infty \tag{5}$$

- In order to explore, we use the $\epsilon - greedy$ method, or the UCB method given by,

$$a = \arg\max_{c \in A} -Q_t^j(s^j, c) + \sqrt{\frac{lnR_{s^j}(t)}{R_{s^j,c}(t)}} \tag{6}$$

A simple three junction network modelling

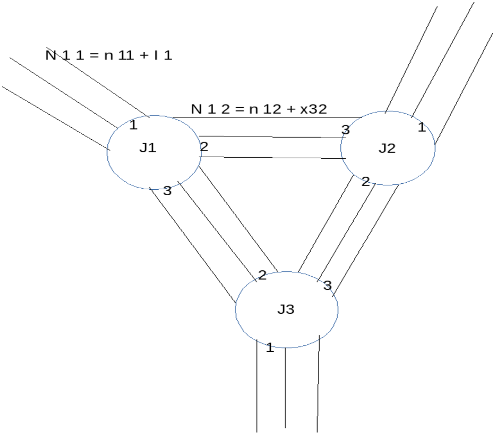

Figure 2: A simple 3 junction road network considered with each of the junctions captioned with Ji. 2 of the roads are named , one connected to outside of the network and another connected to the inside

- At the outer junctions, cars are coming in at a poisson rate.
- $$lj = r_j t$$
  with
  $$p(r_j t) = \frac{\lambda^{r_j t} e^{-\lambda}}{(r_j t)!}$$
- $$x_j^i = \int_{t_j^i}^{\sum_k t_j^k} \sum_k I[Phase_j^k \, is \, on] I[N_j^k > 0] dt$$

- Thus,

$$E(N_1^1) = n_1^1 + r_1(t_1^2 + t_1^3 + t_1^1) - P(N_1^1 > 0)vt_1^1$$

  over a time period of $t_1^1 + t_1^2 + t_1^3$

- And

$$E(N_1^2) = n_1^2 + \alpha_2^{13} P(N_2^1 > 0)vt_2^1 + \alpha_2^{23} P(N_2^2 > 0)vt_2^2 - P(N_1^2 > 0)vt_1^2$$

  over a time period of $t_2^1 + t_2^2 + t_2^3$

Basic Stochastic Approximation scheme

The basic stochastic approximation Lemma depends on 4 assumptions:

A1 The map $h : \mathbb{R}^d \to \mathbb{R}^d$ is Lipschitz , i.e.,

$$\|h(x) - h(y)\| \leq L\|x - y\|$$

for some $0 < L < \infty$

A2 Stepsizes $\{a(n)\}$ are positive scalars satisfying,

$$\sum_n a(n) = \infty, \sum_n a(n)^2 < \infty$$

A3 $\{M_n\}$ is a martingale difference sequence with respect to the increasing family of σ fields $\mathscr{F}_n \triangleq \sigma(x_m, M_m, m \le n) = \sigma(x_0, M_1, ..., M_n), n \ge 0$, i.e.

$$E[M_{n+1}|\mathscr{F}_n] = 0, a.s., n \ge 0$$

and $\{M_n\}$ are square integrable.

$$E[\|M_{n+1}\|^2|\mathscr{F}_n] \le K(1 + \|x_n\|^2), a.s., n \ge 0$$

for some constant $K > 0$

A4 The iterates remain bounded, i.e.

$$\sup_n \|x_n\| < \infty, a.s.$$

- Only (*Borkar et. al*) when all these 4 assumptions are satisfied can we say the following iterative equation:

$$x_{n+1} = x_n + a(n)[h(x_n) + M_{n+1}], n \geq 0,$$

will track the o.d.e.

$$\dot{x}(t) = h(x(t)), t \geq 0$$

Value iteration

- The basic value iteration method as a vector method is given by:

$$F\bar{J} = \bar{c} + \alpha P\bar{J} \tag{7}$$

where

$$P = \begin{bmatrix} p11 & p12 & \cdots \\ p21 & p22 & \cdots \\ \cdots & & \end{bmatrix}$$

and

$$\bar{c} = \begin{bmatrix} \sum_{j\in S} p_{1j}c_{1j}(t) \\ \sum_{j\in S} p_{2j}c_{2j}(t) \\ \cdots \end{bmatrix}$$

Asynchronous Value Iteration

- The Asynchronous version of the value iteration method says that,

$$J_{k+1}(i) = \begin{cases} (TJ_k)(i), & \text{if } i = i_k \\ J_k(i), & \text{otherwise} \end{cases} \tag{8}$$

  where

$$(TJ)(i) = \min_{u \in U(i)} \sum_{j=0}^{n} p_{ij}(u)(g(i,u,j) + \alpha J(j))$$

  for $\alpha < 1$

- It can be proven that this method will converge as long as all states are visited infinite times. (*Bertsekas et. al. NDP*)

Stochastic approximation modification

- A slightly different stochastic approximation method,

$$x_i := x_i + \alpha(F_i(x) - x_i + w_i) \tag{9}$$

  where $x = \{x_1, x_2, \cdots, x_n\} \in \mathbb{R}^n$ and $F = \{F_1, F_2, \cdots, F_n\}$ are mappings from $\mathbb{R}^n to \mathbb{R}$ and w_i is a small random noise term. This algorithm is also seen to converge (*Tsitsiklis et.al.*)

- A slight modification to the Lipscitz assumption is given by

$$\|F(x) - x^\star\|_v \leq \beta \|x - x^\star\|_v \tag{10}$$

Q learning convergence proof

- For , value iteration, the T operator is given by

$$T_i(V) = \min_{u \in U(i)} E(c_{iu}) + \alpha \sum_{j \in S} p_{ij}(u) V_j \tag{11}$$

- The Q learning method based on a modification of the Bellman equation $V^\star = T(V^\star)$
- Let $P = (i,u) | i \in S, u \in U(i)$ be the set of all state-action pairs and let n be it′s cardinality.

- Let after t iterations, the vector $Q(t) \in \mathbb{R}^n$, with components $Q_{iu}(t), (i,u) \in P$ be updated according to the formula,

$$Q_{iu}(t+1) = Q_{iu}(t) + \alpha_{iu}(t)[c_{iu} + \beta \min_{v \in U(s(i,u))} Q_{s(i,u),v}(t) - Q_{iu}(t)] \tag{12}$$

- We now argue that this equation has the form, 9 . Let F be the mapping defined from $\mathbb{R}^n$ onto itself with components F_{iu} defined by

$$F_{iu}(Q) = E[c_{iu}] + \beta E[\min_{v \in U(s(i,u))} Q_{s(i,u),v}] \tag{13}$$

and $E[\min_{v \in U(s(i,u))} Q_{s(i,u),v}] = \sum_{j \in S} p_{ij}(u) \min_{v \in U(j)} Q_{jv}$

- In view of 13, 12 can be written as

$$Q_{iu}(t+1) = Q_{iu}(t) + \alpha(F_{iu}(Q(t))) - Q_{iu}(t) + w_{iu}(t) \quad (14)$$

where

$$w_{iu}(t) = c_{iu} - E(c_{iu}) + \min_{v \in U(s(i,u))} Q(s(i,u),v)(t)$$

$$-E(\min_{v \in U(s(i,u))} Q_{(s(i,u),v)}(t) | \mathscr{F}(t)) \quad (15)$$

- The expectation in the expression $E(\min_{v \in U(s(i,u))} Q_{(s(i,u),v)}(t) | \mathscr{F}(t))$ is with respect to $s(i,u)$.

- The vector form of $F(\bar{Q})$, where n_s is the number of states and n_a is the number of actions, can be written as :

$$\begin{bmatrix} F(Q_{1,a_1}(t+1)) \\ F(Q_{1,a_2}(t+1)) \\ \cdots \\ F(Q_{n_s,a_{n_a}}(t+1)) \end{bmatrix} = \begin{bmatrix} E_{s(1,a_1)}c_{1,a_1}(t) \\ E_{s(1,a_2)}(t) \\ \cdots \\ E_{s(n_s,a_{n_a})}(t) \end{bmatrix} + \beta \begin{bmatrix} E_{s(1,a_1)}[\min_{v\in U(s(1,a_1)} \\ E_{s(1,a_2)}[\min_{v\in U(s(1,a_2)} \\ \cdots \\ E_{s(n_s,a_{n_a})}[\min_{v\in U(s(n_s,a_{n_a}} \end{bmatrix} \tag{16}$$

- This can be writen as

$$\begin{bmatrix} F(Q_{1,a_1}(t+1)) \\ F(Q_{1,a_2}(t+1)) \\ \cdots \\ F(Q_{n_s,a_{n_a}}(t+1)) \end{bmatrix} = \begin{bmatrix} E_{s(1,a_1)}c_{1,a_1}(t) \\ E_{s(1,a_2)}c_{1,a_2}(t) \\ \cdots \\ E_{s(n_s,a_{n_a})}c_{n_s,a_{n_a}}(t) \end{bmatrix} + \beta P \begin{bmatrix} \min_{v\in U(1)} Q_{1,v}(t) \\ \min_{v\in U(2)} Q_{2,v}(t) \\ \cdots \\ \min_{v\in U(n_s)} Q_{n_s,v}(t) \end{bmatrix}$$

where

$$P = \begin{bmatrix} P_{11}(a_1)P_{12}(a_1)\cdots P_{1n_s}(a_1) \\ P_{11}(a_2)P_{12}(a_2)\cdots P_{1n_s}(a_2) \\ \cdots \end{bmatrix}$$

- Taking [3] conditional variance, on both sides of 15, we find that

$$E[\|w_{iu}(t)\|^2|\mathscr{F}(t)] \leq Var(c_{iu}) + \max_{j\in S}\max_{v\in U(j)} Q^2_{jv}(t) \qquad (17)$$

- For [3] discounted problems, $\beta < 1$, 13, yields,

$$|F_{iu}(Q) - F_{iu}(Q')| \leq \beta \max_{j\in S, v\in U(j)} |Q_{jv} - Q'jv|, \forall Q, Q' \qquad (18)$$

Multi-agent Q learning proof

- First we frame the multi-agent Q learning problem as a vector update for a single state for the entire system.
- Then, we try to frame the problem as a large vector update over all states.
- Finally, we will state the problem as an asynchronous update of the large vector and show that it also satisfies our criteria for convergence.

Single State update for the system

- We can think of our system as a network of nodes connected to each other. Thus, our system can be represented as a graph $G = (V, E, A)$.
- Each update of each agent depends on cost at other junctions.
- The state update equation of the entire system can be written as :

$$\begin{bmatrix} Q_1^{t+1}(s^1, a^1) \\ Q_2^{t+1}(s^2, a^2) \\ \dots \\ Q_N^{t+1}(s^N, a^N) \end{bmatrix} = D^{-1}A \begin{bmatrix} c_{junction1}(t) \\ c_{junction2}(t) \\ \dots \\ c_{junction_N}(t) \end{bmatrix} + \beta \begin{bmatrix} \min_{b^1} Q_1^t(s^{1'}, b^1) \\ \min_{b^2} Q_2^t(s^{2'}, b^2) \\ \dots \\ \min_{b^N} Q_N^t(s^{N'}, b^N) \end{bmatrix} \tag{19}$$

Update over all agents, states, actions

- We define an analogous set named $P1 = (i, s_i, a_i), i \in J, s_i \in S_i, a_i \in A_i$ with cardinality $n1$.
- Thus, the large vector is of the form $Q(t) \in \mathbb{R}^{n1}$ where $Q_{i,s_i,a_i}(t)$ update is of the form, 4.
- This can also be brought to the form of 9 by adding and subtracting the expectation term.
- Thus, we can write,

$$Q_{j,s_j,a_j}(t+1) = Q_{j,s_j,a_j}(t) + \gamma(t)(F_{j,s_j,a_j}(Q) - Q_{j,s_j,a_j}(t) + w_{j,s_j,a_j}(t)) \tag{20}$$

where,

$$F_{j,s_j,a_j}(Q) = E[c_{j,s_j,a_j}] + \beta E[\min_{v \in U(s(j,s_j,a_j))} Q_{s(j,s_j,a_j),v}] \tag{21}$$

and here,

$$E[\min_{v \in U(s(j,s_j,a_j))} Q_{s(j,s_j,a_j),v}] = \sum_{s_j{}' \in S_j} P_{j,s_j,s_j{}'}(u_j) \min_{v \in U(s_j{}')} Q_{j,s_j{}',v} \tag{22}$$

- Also, the term w_{j,s_j,a_j} can be written as,

$$w_{j,s_j,a_j} = c_{j,s_j,a_j} + \beta \min_{v \in U(s(j,s_j,a_j))} Q_{s(j,s_j,a_j),v} \\ -(E[c_{j,s_j,a_j}] + \beta E[\min_{v \in U(s(j,s_j,a_j))} Q_{s(j,s_j,a_j),v} | \mathscr{F}(t)]) \tag{23}$$

where

$$\mathscr{F}(t) = \{Q_{j,s_j,a_j}(0), c_{j,s_j,a_j}(t) \forall j \in J, \forall s_j \in S_j, \forall a_j \in A_j \forall t \in T\}$$

Vector update forms

- The vector update for $F(\bar{Q})$ can be written in 2 ways:

$$\begin{bmatrix} F(Q^1_{t+1}(1_1, a^1_1)) \\ F(Q^2_{t+1}(2_1, a^2_1)) \\ \cdots \\ F(Q^N_{t+1}(N_1, a^N_1)) \\ F(Q^1_{t+1}(1_1, a^1_2)) \\ F(Q^2_{t+1}(2_1, a^1_2)) \\ \cdots \end{bmatrix} = \begin{bmatrix} D^{-1} 0 \cdots \\ 0 D^{-1} \cdots \\ \cdots \end{bmatrix} \begin{bmatrix} A 0 \cdots \\ 0 A \cdots \\ \cdots \end{bmatrix} \begin{bmatrix} E_{s_1 -> s_1'}[c_{junction_1}(t)] \\ E_{s_2 -> s_2'}[c_{junction_2}(t)] \\ \cdots \end{bmatrix} \tag{24}$$

$$+\beta \begin{bmatrix} E[\min_{v \in U(s(1_1, a^1_1))} Q_{s(1_1, a^1_1), v}(t)] \\ E[\min_{v \in U(s(2_1, a^2_1))} Q_{s(2_1, a^2_1), v}(t)] \\ \cdots \end{bmatrix}$$

- It can also be written in the form of :

$$\begin{bmatrix} F(Q^1_{t+1}(1_1, a^1_1)) \\ F(Q^2_{t+1}(2_1, a^2_1)) \\ \cdots \\ F(Q^N_{t+1}(N_1, a^N_1)) \\ F(Q^1_{t+1}(1_1, a^1_2)) \\ F(Q^2_{t+1}(2_1, a^1_2)) \\ \cdots \end{bmatrix} = \begin{bmatrix} E[c_{agent_1}(t)] \\ E[c_{agent_2}(t)] \\ \cdots \end{bmatrix} \tag{25}$$

$$+\beta \begin{bmatrix} E[\min_{v \in U(s(1_1, a^1_1))} Q_{s(1_1, a^1_1), v}(t)] \\ E[\min_{v \in U(s(2_1, a^2_1))} Q_{s(2_1, a^2_1), v}(t)] \\ \cdots \end{bmatrix}$$

- The last Expectation term can be written as:

$$\begin{bmatrix} E[\min_{v\in U(s(1_1,a_1^1))} Q_{s(1_1,a_1^1),v}(t)] \\ E[\min_{v\in U(s(2_1,a_1^2))} Q_{s(2_1,a_1^2),v}(t)] \\ \cdots \end{bmatrix}$$

$$= \begin{bmatrix} p_{1_1,1_1}(a_1^1), 0, \cdots, p_{1_1,1_2}(a_1^1), 0, \cdots \\ 0, p_{2_1,2_1}(a_1^2), \cdots, 0, p_{2_1,2_1}(a_1^2), \cdots \end{bmatrix} \begin{bmatrix} \min_{v\in U(s(1_1,a_1^1))} Q_t^1(1_1, v) \\ \min_{v\in U(s(2_1,a_1^2))} Q_t^2(2_1, v) \\ \cdots \\ \min_{v\in U(s(1_1,a_1^1))} Q_t^1(1_2, v) \\ \cdots \end{bmatrix}$$

Vector Update Lipschitz

- Thus, the vector update equation can be written in the form of :
$$\bar{Q}^{t+1} = \bar{Q}^t + \gamma(\bar{F}(\bar{Q}^t) - \bar{Q}^t + \bar{w}^t) \tag{26}$$
where,
$$\bar{F}(\bar{Q}^t) = E(\bar{c}^t) + \beta \bar{P}(\bar{Q}'^{t}_{n_s}) \tag{27}$$
Thus,
$$F(\bar{Q}) - F(\bar{Q}') = \beta P(\bar{Q}_{n_s} - \bar{Q}'_{n_s})$$
- Taking ∞ norm on both sides, we get,
$$\|\bar{F}(\bar{Q}) - \bar{F}(\bar{Q}')\|_\infty \leq \beta \|\bar{Q} - \bar{Q}'\|_\infty \tag{28}$$
since,
$$\|P\|_\infty \leq 1$$
- This, shows that the F, vector operator is Lipschitz, and hence, the operation is a contraction since $\alpha < 1$.
- Now, for the Asynchronous update.

Asynchronous Update

- Now, the update term will be over a certain component of a state with a certain action and we will show that the Lipschitz property still holds.
- For a certain agent i, at a state s_i, with action a_i,

$$
\begin{aligned}
&\bar{F}(\bar{Q})_{i,s_i,a_i} - \bar{F}(\bar{Q}')_{i,s_i,a_i} \\
&= \beta(E[\min_b Q_i(s_i{}', b) - \min_{\tilde{b}} Q_i{}'(s_i{}'', \tilde{b})]) \\
\leq \beta E[&\min_{\tilde{b}, s.t. \tilde{b} = \arg\min Q_i{}'(s_i{}'', \tilde{b})} (Q_i(s_i{}', \tilde{b}) - Q_i{}'(s_i{}'', \tilde{b}))] \\
&\leq \beta \max_{i, s_i{}'', \tilde{b} \in U(s_i{}'')} [Q_i(s_i{}'', \tilde{b}) - Q_i{}'(s_i{}'', \tilde{b})]
\end{aligned}
\tag{29}
$$

- This is similiar to the form used in 18

Stability of iterates

- Also, We know that the expectation of the noise term is zero.
- As for the square of the noise term, similiar to earlier, taking conditional variance on both sides,

$$E[\|w_{j,s_j,a_j}\|^2|\mathscr{F}(t)] \leq Var(c_{j,s_j,a_j}) + \max_{j\in J}\max_{s_j\in S_j}\max_{v\in U(s_j)} Q^2_{j,s_j,v}(t) \tag{30}$$

- Now, in order to prove the stability of the iterates (i.e. the Q values themselves), we will proceed according to the method described in [5].
- In order to show the stability of iterates, we will have to show 2 things
- For the equation

$$x_{n+1} = x_n + a(n)[h(x_n) + M_{n+1}], n \geq 0,$$

A1

$$lim_{r->\infty} h_r(x) = h_\infty(x), x \in \mathbb{R}^n \tag{31}$$

where

$$h_r(x) = \frac{h(rx)}{r}$$

A2

$$E[\|M(n+1)\|^2|\mathscr{F}_n] \leq C_0(1+\|X(n)\|^2), n \geq 0 \tag{32}$$

- When A1 and A2 are both satisfied, along with the condition that the step sizes be tapering like described above, then we can say that the iterates are stable.
- A2 is already satisfied as described above in (30).
- For A1,

$$\bar{F}_r(\bar{Q})_{i,s_i,a_i} = \frac{\bar{F}(\bar{Q})_{i,s_i,a_i}}{r}$$

- Also,

$$\begin{aligned} h_r(Q) &= \frac{F(rQ) - rQ}{r} \\ &= \frac{F(rQ)}{r} - Q \end{aligned}$$

- Thus,

$$\frac{F_{i,s_i,a_i}(rQ)}{r} = \frac{E[c_{i,s_i,a_i}]}{r} + \sum_{s_i' \in S_i} P_{i,s_i,s_i'}(a_i) \min_{v \in U(s_i')} Q_{i,s_i',v}$$

- Thus,

$$lim_{r->\infty} F_{r_{i,s_i,a_i}}(Q) = \sum_{s_i' \in S_i} P_{i,s_i,s_i'}(a_i) \min_{v \in U(s_i')} Q_{i,s_i',v}$$

- Hence,

$$h_\infty(Q) = \beta \sum_{s_i' \in S_i} P_{i,s_i,s_i'}(a_i) \min_{v \in U(s_i')} Q_{i,s_i',v} - Q_{i,s_i,a_i}$$

- Here $h_\infty(Q)$ is also of the form $F_\infty(Q) - Q$
- where $F_\infty(Q)$ is a contraction wrt $\|.\|_\infty$ and thus, the asymptotic stability of the unique equilibrium point of the corresponding ODE is guaranteed.
- Thus, the assumption A4 is also satisfied.

References

1 Neuro Dynamic Programming, *Dmitri P. Bertsekas, John N. Tsitsiklis,* MIT, Athena Scientific, Belmont, Massachusetts

2 Stochastic Approximation: The O.D.E. approach, *Vivek S. Borkar, TIFR, Mumbai; Sameer N. Jalnapurkar, IISc, Bangalore*

3 Asynchronous Stochastic Approximation and Q-Learning *John N. Tsitsiklis,* 1994, Kluwer Academic Publishers, Boston

4 Multi-agent Reinforcement Learning for Traffic signal Control, *Prabuchandran K.J. , Hemanth Kumar A.N , Shalabh Bhatnagar* IISc, Bangalore, 2014 IEEE 17th International Conference on Intelligent Transportation Systems (ITSC) October 8-11, 2014. Qingdao, China

5 THE O.D.E. METHOD FOR CONVERGENCE OF STOCHASTIC APPROXIMATION AND REINFORCEMENT LEARNING. *Vivek S. Borkar, Sean P. Meyn* SIAM J.CONTROL OPTIM. Vol.38 No.2, pp. 447-469