Deep Reinforcement Learning for Intelligent Transportation Systems

Anonymous Author(s)
Affiliation
Address
email

Abstract

Intelligent Transportation Systems (ITSs) are envisioned to play a critical role in improving traffic flow and reducing congestion, which is a pervasive issue impacting urban areas around the globe. Rapidly advancing vehicular communication and edge cloud computation technologies provide key enablers for smart traffic management. However, operating viable real-time actuation mechanisms on a practically relevant scale involves formidable challenges, e.g., Markov Decision Processes (MDP) and conventional Reinforcement Learning (RL) techniques suffer from poor scalability due to state space explosion. Motivated by these issues, we explore the potential for Deep Q-Networks (DQN) to optimize traffic light control policies. As an initial benchmark, we establish that the DQN algorithms yield the "thresholding" policy in a single-intersection. Next, we examine the scalability properties of DQN algorithms and their performance in a linear network topology with several intersections along a main artery. We demonstrate that DQN algorithms produce intelligent behavior, such as the emergence of "greenwave" patterns, reflecting their ability to learn favorable traffic light actuations.

1 Introduction

Emerging Intelligent Transportation Systems (ITSs) [1–6] are expected to play an instrumental role in improving traffic flow, thus optimizing fuel efficiency, reducing delays and enhancing the overall driving experience. Today traffic congestion is an exceedingly complex and vexing issue faced by metropolitan areas around the world. In particular, street intersections in dense urban traffic zones (e.g., Times Square in Manhattan) can act as severe bottlenecks.

Current traffic light control policies typically involve preprogrammed cycles that may be optimized based on historical data and adapted according to daily patterns. The options for adaptation to real-time conditions, e.g. through detection wires in the pavement, tend to be fairly rudimentary. Evolving vehicular communication technologies offer a crucial capability to obtain more fine-grained knowledge of the positions and speeds of vehicles. Such comprehensive real-time information can be leveraged, in conjunction with edge cloud computation, for significantly improving traffic flow through more agile traffic light control policies, or in the longer term, via direct actuation instructions for fully automated driving scenarios [7]. While the potential benefits are immense, so are the technical challenges that evidently arise in solving such real-time actuation problems on an unprecedented scale in terms of intrinsic complexity, geographic range, and number of objects involved.

Under suitable assumptions, the problem of optimal dynamic traffic light control may be formulated as a Markov decision process (MDP) [8–10]. The MDP framework provides a rigorous notion of optimality along with a basis for computational techniques such as value iteration, policy iteration or linear programming. However, an MDP formulation involves strong model assumptions, which

Submitted to 32nd Conference on Neural Information Processing Systems (NIPS 2018). Do not distribute.

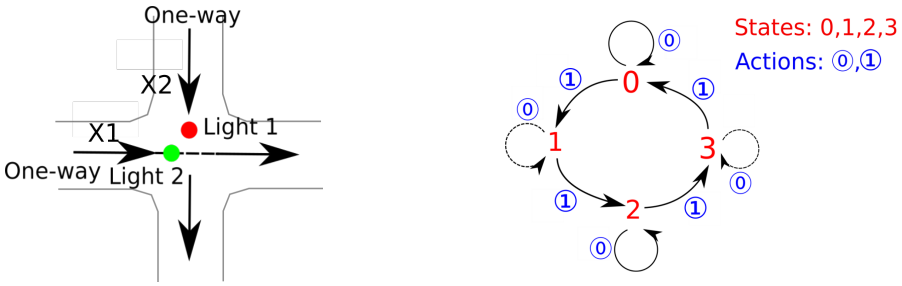

Figure 1: One intersection with two traffic flows (left) where X_1 and X_2 are the queue lengths, and the state transition diagram (right).

may not always be satisfied in reality, and knowledge of relevant system parameters, which may not be readily available. Owing to these issues, an MDP approach tends to be vulnerable to model mis-specification and inaccurate parameter estimation. Moreover, in terms of computational aspects, an MDP approach suffers from the curse of dimensionality, resulting in excessively large state spaces in realistic problem instances and exceedingly slow convergence.

Reinforcement Learning (RL) techniques, such as Q-learning, overcome some of these limitations [11–14] and have been previously considered in the context of optimal dynamic traffic light control [15–18]. However, conventional RL techniques are still prone to prohibitively large state spaces and extremely sluggish convergence, implying poor scalability beyond a single-intersection scenario.

Motivated by the above issues, we explore in the present paper the potential for deep learning algorithms, particularly Deep Q-Networks (DQN) [19], to optimize real-time traffic light control policies in large-scale transportation systems. As an initial validation benchmark, we analyze a single-intersection scenario and corroborate that the DQN algorithms match the provably optimal performance achieved by an MDP approach and exhibit a similar threshold structure. Next, we consider a linear network topology with several intersections to examine the scalability properties of DQN algorithms and their performance in the presence of highly complex interactions created by the flow of vehicles along the main artery. As mentioned above, the use of an MDP approach or standard RL techniques involves an excessive computational burden in these scenarios; hence the optimal achievable performance cannot be easily quantified. As a relevant qualitative feature, we demonstrate that DQN algorithms produce intelligent behavior, such as the emergence of "greenwave" patterns [20, 21], even though such structural features are not explicitly prescribed in the optimization process. This emergent intelligence confirms the capability of the DQN algorithms to learn favorable structural properties solely from observations.

The remainder of the paper is organized as follows. In Section 2, we present a detailed model description and problem statement. In Section 3, we provide a specification of the DQN algorithms for a single intersection as well as a linear network with several intersections. Section 4 discusses the computational experiments conducted to evaluate the performance of the proposed DQN algorithms and illustrate the emergence of "greenwave" patterns. In Section 5, we conclude with a few brief remarks and some suggestions for further research.

2 Model Description and Problem Statement

We model the road intersections and formulate our optimization problem. For the sake of transparency, we consider an admittedly stylized model that only aims to capture the most essential features that govern the dynamics of contending traffic flows at road intersections. We throughout adopt a discrete-time formulation to simplify the description and allow direct application of MDP techniques for comparison, but the methods and results naturally extend to continuous-time operation.

2.1 Single Road Intersection

As mentioned earlier, we start with a single-intersection scenario to facilitate the validation of the DQN algorithms by comparing it with the MDP approach. We consider the simplest meaningful setup with two intersecting unidirectional traffic flows as schematically depicted in the left side of Fig. 1. The state $S(t)$ of the system at the beginning of time slot t may be described by the three-tuple

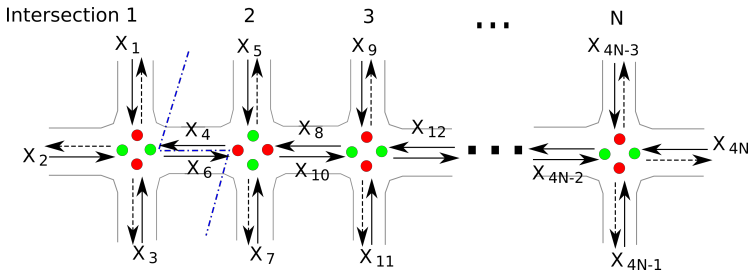

Figure 2: Linear bidirectional road network.

$(X_1(t), X_2(t); Y(t))$, with $X_i(t)$ denoting the number of vehicles of traffic flow i waiting to cross the intersection and $Y(t) \in \{0, 1, 2, 3\}$ indicating the configuration of the traffic lights:

- "0": green light for direction 1 and hence red light for direction 2;
- "1": yellow light for direction 1 and hence red light for direction 2;
- "2": green light for direction 2 and hence red light for direction 1;
- "3": yellow light for direction 2 and hence red light for direction 1.

Each configuration k can either simply be continued in the next time slot or must otherwise be switched to the natural subsequent configuration $(k + 1) \mod 4$. This is determined by the action $A(t)$ selected at the end of time slot t, which is represented by a binary variable as follows: "0" for continue, and "1" for switch:

$$Y(t + 1) = (Y(t) + A(t)) \mod 4. \tag{1}$$

These rules give rise to a strictly cyclic control sequence as illustrated in the right side of Fig. 1.

The evolution of the queue state over time is governed by the recursion

$$(X_1(t + 1), X_2(t + 1)) = (X_1(t) + C_1(t) - D_1(t), X_2(t) + C_2(t) - D_2(t)), \tag{2}$$

with $C_i(t)$ denoting the number of vehicles of traffic flow i appearing at the intersection during time slot t and $D_i(t)$ denoting the number of departing vehicles of traffic flow i crossing the intersection during time slot t. While not essential for our analysis, we make the simplifying assumption that if one of the two traffic flows is granted the green light, then exactly one waiting vehicle of that traffic flow, if any, will cross the intersection during that time slot, i.e.,

$$D_1(t) = \min\{1, X_1(t)\} \text{ if } Y(t) = 0; D_2(t) = \min\{1, X_2(t)\} \text{ if } Y(t) = 2; \tag{3}$$

and $D_1(t) = 0$ if $Y(t) \neq 0$ and $D_2(t) = 0$ if $Y(t) \neq 2$.

2.2 Linear Road Topology

To examine the performance and scalability properties of the DQN algorithms in more complex large-scale scenarios, we will consider a linear road topology. Specifically, we investigate a linear network topology with N intersections and bidirectional traffic flows, representing a main artery with cross streets as schematically depicted in Fig 2. We do not account for any traffic flows making left or right turns, but the analysis could easily be generalized to accommodate that. The state $S(t)$ of the system at the beginning of time slot t may be described by a $(5N)$-tuple $(X_{n1}(t), X_{n2}(t), X_{n3}(t), X_{n4}(t); Y_n(t))_{n=1,\ldots,N}$, with directions 1 and 2 corresponding to the east-west direction of the main artery and the north-south direction of the cross streets, and thus

$$Y_n(t + 1) = (Y_n(t) + A_n(t)) \mod 4, \tag{4}$$

with $A_n(t)$ denoting the action selected for the n-th intersection at the end of time slot t.

The evolution of the various queue states is governed by the recursion

$$X_{ni}(t + 1) = X_{ni}(t) + C_{ni}(t) - D_{ni}(t), \tag{5}$$

with $C_{ni}(t)$ denoting the number of vehicles in direction i appearing at the n th intersection during time slot t and $D_{ni}(t)$ denoting the number of vehicles crossing the n-th intersection in direction i

during time slot t, $i = 1, \ldots, 4$, $n = 1, \ldots, N$. While $C_{11}(t)$, $C_{N3}(t)$, and $C_{n2}(t)$, $C_{n4}(t)$, $n = 1, \ldots, N$, correspond to vehicles approaching the intersection from the external environment, we have $C_{n+1,1}(t+u) = D_{n1}(t)$ and $C_{n3}(t+u) = D_{n+1,3}(t)$, $n = 1, \ldots, N-1$. This reflects that the vehicles crossing the n-th intersection in eastern direction during time slot t appear at the $(n+1)$-th intersection u time slots later; likewise, vehicles passing through the $(n+1)$-th intersection in western direction during time slot t arrive at the n-th intersection u time slots later. In this manner, the vehicles that travel along the main artery create highly complex interactions among the various intersections, which present additional challenges in optimizing the control policy.

Note that

$$D_{n1}(t) = \min\{1, X_{n1}(t)\}, D_{n3}(t) = \min\{1, X_{n3}(t)\}, \text{ if } Y_n(t) = 0, \tag{6}$$

$D_{n1}(t), D_{n3}(t) = 0$ if $Y_n(t) \neq 0$, and similarly for $D_{n2}(t)$ and $D_{n4}(t)$ depending on whether $Y_n(t) = 2$ or not.

2.3 Optimization Goal

We assume that the "congestion cost" in time slot t may be expressed as a function $F(X(t))$ of the queue state, with $X(t) = (X_1(t), X_2(t))$ in the single-intersection scenario and $X(t) = (X_{n1}(t), X_{n2}(t), X_{n3}(t), X_{n4}(t))_{n=1,\ldots,N}$ in the linear topology with N intersections. The goal is to find a dynamic control policy which selects actions over time so as to minimize the long-term expected discounted cost $\mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^t F(X(t))\right]$, with $\gamma \in (0, 1)$ representing a discount factor.

3 Algorithm Design

We provide a detailed specification of the DQN algorithms for the scenarios of a single intersection or a linear topology with several intersections as described in the previous section.

First of all, let $Q(s, a)$ be the maximum achievable expected discounted reward (or minimum negative congestion cost in our context) under the optimal policy starting from state $s = (X; Y)$ when action a is taken. The $Q(s, a)$ values satisfy the equations

$$Q(s,a) = r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s, s'; a) \max_{a' \in \mathcal{A}} Q(s', a') = r(s,a) + \gamma \mathbb{E}\left[\max_{a' \in \mathcal{A}} Q(s', a')\right], \tag{7}$$

with $r(s, a) = F(X)$ denoting the congestion cost in queue state X, and $p(s, s'; a)$ denoting the transition probability from state s to state s' when action a is taken. Observe that the values $V(s) = \max_{a' \in \mathscr{A}} Q(s, a')$ satisfy the Bellman optimality equations

$$V(s) = \max_{a' \in \mathcal{A}} \{r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s, s'; a) V(s')\} = \max_{a' \in \mathcal{A}} \{r(s,a) + \gamma \mathbb{E}[V(s')]\}. \tag{8}$$

The system state S serves as the input for both the target network and the evaluate network in the DQN algorithms, with $S = (X_1, X_2; Y)$ in the single-intersection scenario and $S = (X_{n1}, X_{n2}, X_{n3}, X_{n4}; Y_n)_{n=1,\ldots,N}$ in the linear topology with N intersections. Equation (7) provides the basis for deriving the target Q-values at each time step, while the Q-learning update for the neural network approximator in the i-th iteration is calculated based on

$$\text{Loss}(\theta_i) = \mathbb{E}_{s,a,r,s' \sim memory}\left[\left(r + \gamma \max\left(q\_target(s', a'; \theta_i')\right) - q\_eval(s, a; \theta_i)\right)^2\right], \tag{9}$$

where r is reward (negative cost) in the current step, s' and a' are the state and action in the next step, θ_i are parameters of the evaluate Q-network in the i-th iteration and θ_i' are parameters of the target Q-network with delayed update following the evaluate network.

The DQN algorithms sample from and train on data collected in memory. The online samples are stored in memory for further learning. A warm-up period of k_0 time steps is applied before the learning operations are initiated. The evaluate network is updated with the AdamOptimizer [22] gradient-descent and ϵ-greedy policy, whereas the update of the target network is slightly later.

Algorithm 1 DQN for single intersection or linear road topology with N intersections

1: Initialize queue and control states: either $X_1, X_2 = 0; Y = 0$ [single intersection] or $X_{n1}, X_{n2}, X_{n3}, X_{n4} = 0; Y_n = 0$ for all $n = 1, \ldots, N$ [linear topology];
2: For steps $k = 1, \ldots, K$ do:
3: $s = [X_1, X_2; Y]$ [single intersection] or $s = [X_{11}, \ldots, X_{N4}; Y_1, \ldots Y_N]$ [linear topology];
4: Select $a^* = \arg\max_{a \in \mathcal{A}} q\_eval(s; a)$, using eval_net to evaluate the Q-value for each action;
5: Generate random variables C_1, C_2 [single intersection] or C_{11}, C_{N3} and C_{n2}, C_{n4} for all $n = 1, \ldots, N$;
6: Given a^*, determine new queue and control states X_1', X_2', Y' according to Eq. (1)-(3) [single intersection] or $X_{11}', \ldots, X_{N4}', Y_1', \ldots, Y_N'$ according to Eq. (4)-(6) [linear topology];
7: $r = -((X_1')^2 + (X_2')^2)$ [single intersection] or $r = -\sum_{n=1}^{N} \sum_{i=1}^{4} (X_{ni}')^2$ [linear topology];
8: $s' = [X_1', X_2'; Y']$ [single intersection] or $s' = [X_{11}', \ldots, X_{N4}'; Y_1', \ldots, Y_N']$ [linear topology];
9: Store transition $[s, a^*, r, s']$ in memory;
10: Perform learning operation if $k > k_0$:
11: Sample a minibatch of samples from memory;
12: Update target network: $\theta_i' = \theta_i$;
13: Calculate the target Q-value: $q\_target(s, a^*) = r + \gamma \max_{a' \in \mathcal{A}} q\_target(s', a')$;
14: Update evaluate network with gradient descent (using AdamOptimizer) and ϵ-greedy policy: Loss$(\theta_i) = \mathbb{E}[(q_target - q_eval)^2]$.

Based on the above outline, we provide the specification of the DQN algorithm for the single-intersection scenario Fig. 1 and a linear topology with N intersections in Fig. 2. It is worth observing that even in the latter case we adopt a 'single-agent' DQN algorithm which has access to the global state of the network, as opposed to the 'multiple-agent' method with one agent for each individual intersection as considered in [17, 23]. While the single-agent approach involves a larger state space, it allows more intelligent control and coordination on a global level, which manifests itself for example in the emergence of greenwave patterns as we will demonstrate in the next section.

4 Performance Evaluation

We present simulations to evaluate the performance of the DQN algorithm in Alg. 1, and in particular illustrate the emergence of "greenwave" patterns in linear topology networks.

4.1 Single Road Intersection

As an initial validation benchmark, we first consider a single-intersection scenario as described in Subsection 2.1. The reason for considering this toy scenario is that the state space is sufficiently small for the optimal policy to be computed using the baseline MDP approach. We assume the numbers of arriving vehicles of both traffic flows in each time step as represented by the random variables C_1 and C_2 to be independent and Bernoulli distributed with parameter $p = 1/4$. We use a quadratic congestion cost function $F(X_1, X_2) = X_1^2 + X_2^2$ and a discount factor $\gamma = 0.99$.

Inspection of the results in Fig. 3 shows that the DQN policy as obtained using Alg. 1 coincides with the optimal MDP policy. In particular, it matches the optimal performance and exhibits a similar threshold structure. This structural property was also reported in [24] for a strongly related two-queue dynamic optimization problem (with switch-over costs rather than switch-over times).

4.2 Linear Road Topology

We now turn to the scenario in Fig. 2. This is a more challenging scenario which serves to examine the scalability properties of our algorithm and its performance in the presence of highly complex interactions arising from the flow of vehicles along the main east-west arterial road.

Assume the numbers of externally arriving vehicles in eastern and western directions in each time step, represented by the random variables C_{11} and C_{N3}, to be independent and Bernoulli distributed

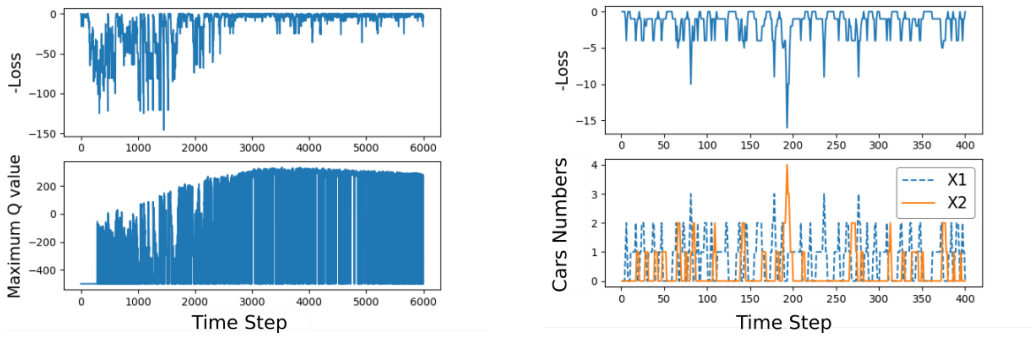

Figure 3: Learning curve for DQN policy (left) and thresholding property of DQN policy (right).

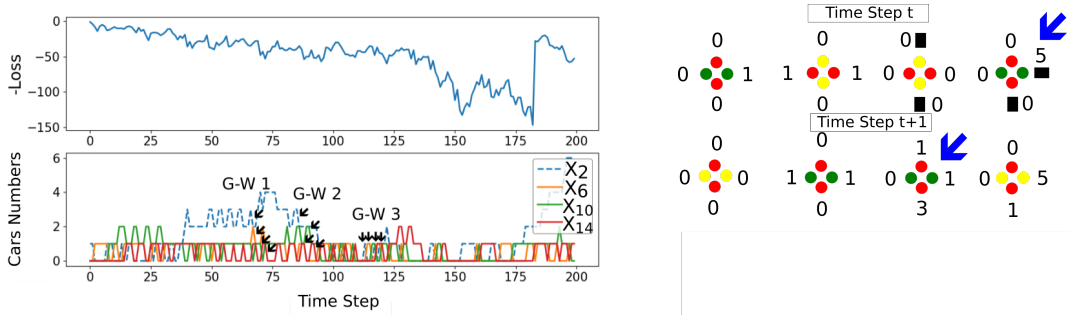

Figure 4: Learning curve for the linear network of size 4×1 (left) and Greenwave traffic lights (right). "Greenwave" is abbreviated to be "G-W". In the right, numbers indicate the number of vehicles waiting on each road. Black rectangles indicate incoming vehicles from peripheral roads.

with parameter $p_1 = 1/4$. The numbers of arriving vehicles in southern and northern directions on each of the N cross streets in each time step, represented by the random variables C_{n2} and C_{n4}, $n = 1, \ldots, N$, are also independent and Bernoulli distributed with parameter $p_2 = 1/8$. We use a quadratic congestion cost function $F(X) = \sum_{n=1}^{N} \sum_{i=1}^{4} X_{ni}^2$ and a discount factor $\gamma = 0.99$. In simulations, the evaluate and target networks used in Alg. 1 have both 4 fully-connected layers of size $200, 100, 40$ and 2, respectively. We use ReLu as activation functions and squared difference loss.

The use of an MDP approach is computationally infeasible in this case due to the state space explosion, and hence the degree of optimality of our algorithm cannot be assessed in a quantitative manner. Instead we have therefore examined qualitative features to validate the intelligent behavior of our algorithm and evaluate its performance merit. In particular, we observed the emergence of "greenwave" patterns as shown in Fig. 4, even though such structural features are not explicitly prescribed in the optimization process. Specifically, the "greenwave" phenomenon is reflected as consecutive reduction of car numbers in each road. This emergent intelligence confirms the capability of our algorithm to learn favorable structural properties solely from observations.

5 Conclusion

We have explored the scope for Deep Q-Networks (DQN) to optimize real-time traffic light control policies in emerging large-scale Intelligent Transportation Systems. As an initial benchmark, we established that DQN algorithms deliver the optimal performance achieved by an MDP approach in a single-intersection scenario. We subsequently evaluated the scalability properties of DQN algorithms in a linear topology with several intersections, and demonstrated the emergence of intelligent behavior such as "greenwave" patterns, confirming their ability to learn desirable structural features. In future research we intend to investigate locality properties and analyze how these can be exploited in the design of distributed coordination schemes for wide-scale deployment scenarios.

References

[1] MM Vazifeh, P Santi, G Resta, SH Strogatz, and C Ratti, "Addressing the minimum fleet problem in on-demand urban mobility," *Nature*, vol. 557, no. 7706, pp. 534, 2018.

[2] Ming Zhu, Xiao-Yang Liu, Feilong Tang, Meikang Qiu, Ruimin Shen, Wei Wennie Shu, and Min-You Wu, "Public vehicles for future urban transportation.," *IEEE Trans. Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3344–3353, 2016.

[3] Ming Zhu, Xiao-Yang Liu, and Xiaodong Wang, "An online ride-sharing path-planning strategy for public vehicle systems," *IEEE Transactions on Intelligent Transportation Systems*, 2018.

[4] Muhammad Alam, Joaquim Ferreira, and José Fonseca, "Introduction to intelligent transportation systems," in *Intelligent Transportation Systems*. Springer, 2016, pp. 1–17.

[5] Yisheng Lv, Yanjie Duan, Wenwen Kang, Zhengxi Li, Fei-Yue Wang, et al., "Traffic flow prediction with big data: A deep learning approach.," *IEEE Trans. Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.

[6] Rajeshwari Sundar, Santhoshs Hebbar, and Varaprasad Golla, "Implementing intelligent traffic control system for congestion control, ambulance clearance, and stolen vehicle detection," *IEEE Sensors Journal*, vol. 15, no. 2, pp. 1109–1113, 2015.

[7] Liang Qi, MengChu Zhou, and WenJing Luan, "A two-level traffic light control strategy for preventing incident-based urban traffic congestion," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 13–24, 2018.

[8] Simona Onori, Lorenzo Serrao, and Giorgio Rizzoni, "Dynamic programming," in *Hybrid Electric Vehicles*, pp. 41–49. Springer, 2016.

[9] Martin L Puterman, *Markov decision processes: discrete stochastic dynamic programming*, John Wiley & Sons, 2014.

[10] Sheldon M Ross, *Introduction to stochastic dynamic programming*, Academic Press, 2014.

[11] Andrew Gehret Barto, Steven J Bradtke, and Satinder P Singh, *Real-time learning and control using asynchronous dynamic programming*, University of Massachusetts at Amherst, Department of Computer and Information Science, 1991.

[12] Andrew W Moore and Christopher G Atkeson, "Prioritized sweeping: Reinforcement learning with less data and less time," *Machine Learning*, vol. 13, no. 1, pp. 103–130, 1993.

[13] Richard S Sutton, "Learning to predict by the methods of temporal differences," *Machine Learning*, vol. 3, no. 1, pp. 9–44, 1988.

[14] Christopher JCH Watkins and Peter Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[15] Elise van der Pol, Frans A Oliehoek, T Bosse, and B Bredeweg, "Video demo: Deep reinforcement learning for coordination in traffic light control," *AmsterdamVrije Universiteit, Department of Computer Sciences*, 2016.

[16] Elise Van der Pol and Frans A Oliehoek, "Coordinated deep reinforcement learners for traffic light control," *Proceedings of Learning, Inference and Control of Multi-Agent Systems (at NIPS 2016)*, 2016.

[17] MA Wiering, "Multi-agent reinforcement learning for traffic light control," in *Proceedings of the Seventeenth International Conference Machine Learning (ICML)*, 2000, pp. 1151–1158.

[18] Hua Wei, Guanjie Zheng, Huaxiu Yao, and Zhenhui Li, "Intellilight: A reinforcement learning approach for intelligent traffic light control," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 2496–2505.

[19] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529, 2015.

[20] Stefan Lämmer and Dirk Helbing, "Self-control of traffic lights and vehicle flows in urban road networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 4, pp. 04019, 2008.

[21] Stefan Lämmer and Dirk Helbing, "Self-stabilizing decentralized signal control of realistic, saturated network traffic," Santa Fe Institute, 2010.

[22] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[23] Sergey Satunin and Eduard Babkin, "A multi-agent approach to intelligent transportation systems modeling with combinatorial auctions," *Expert Systems with Applications*, vol. 41, no. 15, pp. 6622–6633, 2014.

[24] Micha Hofri and Keith W Ross, "On the optimal control of two queues with server setup times and its analysis," *SIAM Journal on Computing*, vol. 16, no. 2, pp. 399–420, 1987.