Deep Reinforcement Learning based Intelligent Traffic Control

Arpan Nookala
Department of Electronics Engineering
Sardar Patel Institute of Technology
Mumbai, India
ganesharpan.nookala@spit.ac.in

Eeshaan Asodekar
Department of Electronics Engineering
Sardar Patel Institute of Technology
Mumbai, India
eeshaan.asodekar@spit.ac.in

Aryan Solanki
Department of Electronics Engineering
Sardar Patel Institute of Technology
Mumbai, India
aryan.solanki@spit.ac.in

Narendra Bhagat
Department of Electronics Engineering
Sardar Patel Institute of Technology
Mumbai, India
narendra_bhagat@spit.ac.in

Deepak Karia
Department of Electronics Engineering
Sardar Patel Institute of Technology
Mumbai, India
deepak_karia@spit.ac.in

2023 IEEE Region 10 Symposium (TENSYMP) | 978-1-6654-8258-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/TENSYMP55890.2023.10223639

Abstract—**The development of Intelligent Traffic Signal Control (ITSC) systems is crucial for enhancing traffic flow and mitigating congestion, which is a widespread problem in urban areas globally. Presently, RADAR or inductive loop-based intelligent systems are used in metropolises of developed countries, but the large investment and infrastructure requirements rule out their widespread application. This paper explores a nascent Deep Reinforcement Learning (DRL) approach to the Traffic Signal Control (TSC) problem, as opposed to classical optimization or rule-based approaches of the past. To address the challenges that limit past RL approaches, the study leverages the Deep Deterministic Policy Gradient (DDPG) algorithm to optimize traffic light control policies. The proposed DRL approach shows intelligent behavior and reduces the average delay time and congestion when compared to the traditional RL, past DRL, and fixed-time signal approaches. A comparative analysis of the reward functions is also presented, which reveals insights into the variance of performance.**

Index Terms—**Machine Learning, Deep Reinforcement Learning, Markov Decision Process, Deep Q-Network, Actor-Critic framework, Traffic Signal Control**

I. Introduction

Increasing traffic jams, delays, and erratic travel times, especially in metropolitan cities are the results of an explosion in the global automobile population. These phenomena especially hurt non-oil-producing nations, costing millions of dollars worth of fuel to go waste.

Currently, to ensure the best possible flow of traffic, traffic engineers construct signal designs and manually adjust the appropriate settings for various traffic scenarios. However, it can be challenging to manually develop signal plans for a variety of traffic conditions, particularly when there are traffic irregularities like accidents, road construction, and traffic barriers. These legacy systems are hamstrung as they aim to manage traffic flow using predetermined timed traffic signals, which in essence is trying to solve a dynamic problem using a static approach without consideration of real-world limitations and suboptimality.

By dynamically modifying the signal timing plan in accordance with current traffic circumstances, intelligent traffic signal control (ITSC) [1] was put forth as an effective method to optimize traffic flow across junctions. Several early ITSC studies have focused on actuated traffic signal control (TSC) systems at intersections [2]–[4]. After the system is installed, the signal controllers use the real-time traffic data obtained to match traffic patterns with predefined rules and responsively manage the signals [5]. However, they do not take into account the long-term traffic patterns leading to suboptimal results. Besides, they are not capable of handling exceptional scenarios since they are not capable of responding to novel scenarios.

Intelligent approaches [6]–[9] to the traffic control problem have been proposed to subdue the limitations of legacy systems and traditional mathematical modeling approaches. The goal of this research work is to explore the development of traffic control systems that go beyond automation, making them adaptive by providing the system the capability to decide on a variable signal time based on instantaneous traffic flow, without the need for hard coding or human intervention.

In recent years, approaches based on artificial intelligence techniques have received research interest as a way to increase the effectiveness of TSC. Artificial intelligence algorithms that use reinforcement learning (RL) have gained scientific attention and are being used more frequently to solve the TSC optimization problem [10]–[12]. RL's ability to learn on its own, not being at the mercy of large and reliable supervised learning datasets and online adaptive learning makes it a potent method for solving the TSC problem.

The application of RL techniques to the TSC optimization problem is significantly facilitated by a number of recent technical developments. Due to the recent advancements in mobile communication technologies, signal controllers now have access to a wider range of traffic features, including vehicle speed, position, fuel consumption, emissions, etc. Availability of these features combined with higher computing power greatly improves RL TSC systems' abilities to learn the traffic environment and identify the optimal solutions for given traffic conditions, in a shorter time span.

Deep Reinforcement Learning (DRL) is an emerging domain within Machine Learning. It combines the function approximation strength of Deep Learning (DL) with the Reinforcement Learning (RL) paradigm's ability to represent and model real-world optimal decision problems. Notable breakthroughs that DRL algorithms have achieved include a superhuman competence in strategy games such as Go [13], [14] and Dota 2 [15], in both cases defeating world champions. This ability to deal with real-world decision phenomena which are inherently embedded with suboptimality has been applied to finance [16], robotics, and even to autonomous driving systems.

The main attributes of DRL which make it a promising candidate for applying to the TSC optimization problem are as follows. First, adaptation to real-time traffic conditions is possible for DRL agents, even if they deviate from the norm due to some unexpected disturbances. Second, DRL can be applied as a model-free approach that has lower complexity and computational requirements. Lastly, DRL is not limited by the curse of dimensionality as traditional RL is. Moreover, DRL has the ability to deal with continuous state spaces, which are most real-life optimization problems, including TSC.

The contributions of the work are threefold. First, the DQN (Deep Q-Network) algorithm is used to build single and multiple-agent intelligent traffic control systems, to prove the merit of applying DRL to this complex optimization problem over currently used FST (Fixed Signal Time) approach. Second, experiments are conducted on the proposed Deep Deterministic Policy Gradient (DDPG) approach, and the results are juxtaposed alongside the computationally simpler DQN algorithm. The proposed and baseline approaches are tested on a single intersection, double intersection, and a grid of intersections and prove that the DRL paradigm is a promising solution to the vexing problem of TSC for optimal traffic flow. Third, exploration of the local optimization versus the global optimization question is conducted by testing a multi-agent reinforcement learning setup on a grid of intersections. Additionally, the effect of choosing a modified reward function on the performance of the DRL TSC systems is also presented.

The remainder of the paper is organized as follows. Following the related work in Section II, Section III describes the model, the environment setting, the problem statement, and the DQN and DDPG algorithms used for single, double, and grid intersection traffic control. Section IV, presents the results obtained from the simulation experiments of the proposed DDPG algorithm alongside the baseline DQN approach and the round-robin FST technique which is currently deployed in TSCs. Section V gives the concluding remarks and proposes the future scope of work.

II. Related Works

This section reviews the past intelligent approaches that have been used by researchers to tackle the TSC optimal traffic flow problem.

The fuzzy Logic approach has the ability to deal with and derive a decision from imprecise, ambiguous, and unclear data, given a set of fuzzy rules, making it an attractive approach for TSC. One of the earliest applications of fuzzy logic in TSC was the development of a fuzzy traffic signal controller (FTSC) by S. Chiu [17]. The FTSC used fuzzy logic to control the signal timings based on real-time traffic conditions, improving the overall traffic flow and reducing congestion compared to traditional TSC methods. Following this numerous studies developed a fuzzy logic-based TSC system that used real-time traffic data to dynamically adjust the signal timings, improving traffic flow and reducing congestion. However, fuzzy logic-based TSC systems have limitations. One of the main limitations is the need for fuzzy logic-based TSC systems to be designed by experts with extensive traffic engineering knowledge. This dependence on expert knowledge can lead to subjective bias and inconsistencies in the decision-making process. Secondly, Fuzzy logic systems are based on a predefined set of rules, which precludes them from handling novel or dynamic situations. Moreover, they are not suitable for controlling large-scale traffic signal control systems, given their need for significant processing power and memory.

Genetic Algorithms (GA) are a form of evolutionary computation wherein the approach improves with each generation. The application of GA to TSC involves formulating the problem as a search for optimal signal timings, where the objective function is defined in terms of various traffic performance parameters, such as average delay, average travel time, etc. Several studies have investigated the use of GA for TSC. In a study by Ceylan et. al. [18], a GA-based approach was proposed for optimizing the signal timings which significantly reduced the average delay and travel time compared to the traditional fixed-time control method. Further studies extended this approach to a grid of intersections. Although it shows promising performance, the GA approach to TSC also has significant deficiencies. First, defining a fitness function that accurately represents the optimization objectives is a challenging task. Second, as GA is a stochastic optimization method, which means that the performance is influenced by the initial population of candidate solutions. Also, its ability to scale is limited due to the increase in computational complexity with the increase in the size of the implementation.

Dynamic Programming (DP) is a mathematical optimization technique that has been applied to various fields, including TSC. DP has been used to optimize the timing of traffic signals by considering traffic flow parameters like waiting time. More recent research by Li et. al. [19] using adaptive dynamic programming outperformed traditional methods and previous approaches in terms of reducing total waiting time and improving traffic efficiency. However, computational complexity is one of the main limitations of applying DP in TSC. DP algorithms typically require solving high-dimensional and complex optimization problems, which can result in significant computational overhead. This becomes especially problematic when DP is applied to large-scale TSC networks. Another limitation of DP in TSC is its sensitivity to model assumptions. In real-world TSC scenarios, traffic patterns are often complex and difficult to model accurately. As a result, DP algorithms may produce suboptimal solutions if the underlying models are not accurate.

Reinforcement learning (RL) is a subset of machine learning that allows an agent to learn by repeated interactions with the system, learning to make optimal decisions. In TSC, RL has been used to optimize traffic signal timing, considering various traffic flow parameters. Arel et. al. [20] explore the use of RL for TSC in a network of intersections. The authors use a multi-agent system to manage the signal timings at each intersection in the network. They implement a Q-Learning algorithm, a type of RL algorithm, to learn the optimal signal timings in real-time based on the traffic conditions. A reduction in the average waiting time for vehicles, decrease in the number of stops, and an overall improvement in the traffic flow, over the previous intelligent approaches, was shown by the results. The authors also consider the limitations of the proposed system. They point out that the RL algorithm used in the system assumes a stationary traffic environment and may not perform well in dynamic scenarios. The system also requires high computational power and large amounts of data to function effectively. Furthermore, the models are prone to slow convergence and punishingly large state spaces. The DRL approach subdues these critical limitations.

III. Methods

The following section details the formulation of the deep reinforcement learning paradigm applied to traffic signal control at a single, double, and grid of intersections. It also gives a distilled description of the three DRL algorithms used in this work.

A. Environment Setting

The model for the TSC optimization problem is formulated such that it mirrors the essential features that orchestrate the traffic flow mechanics at an intersection. The methods and results of this discrete-time formulation, used for lucid description, can be extended to continuous-time operation by MDP techniques. Thus, all analyses performed and insights extracted in this work apply to complex approaches using computer vision techniques, and real-time implementation.

1) Single Intersection: Considering a framework with two unidirectional traffic flows at an intersection. This framework is formulated in the Reinforcement Learning framework as follows:

- State $S(t)$ = $(A_1(t), A_2(t), Z(t))$
  An information vector which encapsulates the number of vehicles from traffic flow direction d, who want to cross over, denoted by $A_d(t)$; and $Z(t)$ which indicates the current status of the traffic signal.
- Signal States $Z(t) \in \{0, 1, 2, 3\}$
  All signals have four states as described by the table I
- Action $a(t)$
  At time step t_i, the signal configuration can either be the same as the previous time step t_{i-1}, or be switched to the next configuration as schematically represented in Fig. 1. This binary decision is made by the action $a(t_{i-1})$ as chosen by the agent, which then updates the signal configuration for the next time step by (1).

$$Z(t_{i+1}) = (Z(t_i) + a(t_i)) \quad mod\, 4 \tag{1}$$

TABLE I
Traffic Signal States

State	Direction 1 Signal State	Direction 2 Signal State
0	red	green
1	red	yellow
2	green	red
3	yellow	red

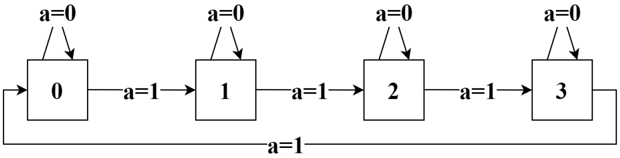

Fig. 1. Signal state mechanism

- Reward r
  The reward signal is defined to be a function of the vehicles waiting, $A_d(t)$, and is designed with the intention to reduce the number of vehicles at the intersection, in an effort of reducing delay and aggregate waiting time. The reward function used is given by (2).

$$r = -((A_1(t))^2 + (A_2(t))^2) \tag{2}$$

  Experiments were also conducted wherein exponent power was varied, and interesting results were obtained.
- Policy $\pi(s)$
  The traffic signal control policy, at a state s, is the probability distribution of the actions the agent can choose to take in s.
- Q-value $Q_\pi(s, a(t))$
  By following the policy π, $Q_\pi(s, a(t))$ is the expected reward for taking an action $a(t)$, while in state s.
- Traffic flow mechanism
  The signal configurations as controlled by the actions of the agent give rise to a cyclic iteration of the configurations. The inflow and outflow of traffic from one timestep to the next ie. the state is updated by (3).

$$\begin{aligned}(A_1(t_{i+1}), A_2(t_{i+1})) =&(A_1(t_i) + I_1(t_i) - O_1(t_i),\\ &A_2(t_i) + I_2(t_i) - O_2(t_i))\end{aligned} \tag{3}$$

  Where $I_d(t)$ and $O_d(t)$ denote the vehicle count incoming and outgoing from the road d of the intersection, respectively.

This framework, III-A1 can be easily extended to a framework with two intersections, and a series of contiguous intersections.

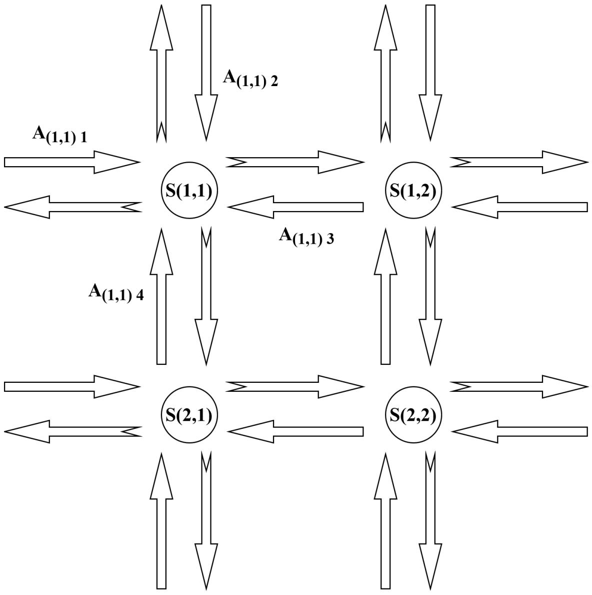

Fig. 2. Grid of Intersections configuration

2) Grid of Intersections: A Two-dimensional $N1$x$N2$ array framework of contiguous bidirectional intersections was also chosen to increase the complexity of the system and to test the scalability of the DRL algorithms to larger setups that mimic real-world scenarios. The representation of a 2x2 grid setup is shown in Fig. 2. This setup's formulation is extrapolated from the single intersection configuration III-A1, with each intersection denoted by (i, j), where $(i, j) \in \{(1, 1), ..., (N1, N2)\}$

- State $S(t) = (A_{(i,j)1}(t), A_{(i,j)2}(t), A_{(i,j)3}(t), A_{(i,j)4}(t), Z_{(i,j)}(t))$
  An information vector that encapsulates the number of vehicles of traffic flowing into and out of the intersection (i, j) from directions $1 - 4$ and $Z_{(i,j)}(t)$ which indicates its current state of the traffic signal at the $(i, j)^{th}$ intersection.
- Signal Configurations $Z_{(i,j)}(t) \in \{0, 1, 2, 3\}$
  The signal configurations remain the same and are extended to all the (i, j) signal intersections.
- Action $a(t)$
  At any given time step t_i, the signal configuration can either be the same as the previous time step t_{i-1}, or be switched to the next configuration, for each of the (i, j) intersections.

  $$Z_{(i,j)}(t_{i+1}) = (Z_{(i,j)}(t_i) + a_{(i,j)}(t_i)) \quad mod\, 4 \qquad (4)$$

  Equation (4) gives the signal configuration updation mechanism, Where $a_{(i,j)}(t_i)$ denotes the action chosen for intersection (i, j) at time step i.
- Reward r
  The reward signal is also an extension of the single intersection configuration given by (2). The only difference is that the cumulative sum is given to the agent when there is a single agent controlling all the intersections. In the case of a multi-agent setup, with one agent for each intersection, each agent is given its individual $r_{(i,j)}$.
- Traffic flow mechanism
  For each intersection, the inflow and outflow of traffic from one timestep to the next follow (5).

  $$A_{(i,j)d}(t_{i+1}) = A_{(i,j)d}(t_i) + I_{(i,j)d}(t_i) - O_{(i,j)d}(t_i) \quad (5)$$

  Where $d \in \{1, 2, 3, 4\}$ denotes the specific road of intersection (i, j) from where the new car is incoming.

Note that this grid setup exponentially increases the complexity of the optimization problem, due to the fact that at any given point a car exiting from one intersection is arriving at the next intersection, until it reaches the edge of the grid, when it exits the grid. Moreover, multiple intersections' traffic flow needs to be optimized simultaneously.

B. DRL Algorithms

This section elucidates the two DRL algorithms implemented to solve the traffic signal control problem.

1) Deep Q-Network (DQN): The fixed Q-target function underlies the DQN algorithm. The architecture is made up of the Q network, the target network, and the experience replay block. In essence, the current state is given as input to a deep neural network that then estimates the Q values. After this, the loss is computed by comparing the estimated Q values with the right-hand side of the Bellman equation (6).

$$\underbrace{\text{New}Q(s,a)}_{\text{New Q-Value}} = Q(s,a) + \underset{\text{New Q-Value}}{\alpha}[\underbrace{R(s,a)}_{\text{Reward}} + \underset{\text{Discount rate}}{\gamma} \overbrace{\max Q'(s',a')}^{\substack{\text{Maximum predicted reward, given} \\ \text{new state and all possible actions}}} - Q(s,a)] \qquad (6)$$

$$Loss = (r + \gamma max_{a'} Q(s', a', \theta') - Q(s, a, \theta)) \qquad (7)$$

This loss (7) is then used to update the weights of the network using stochastic gradient descent and backpropagation.

2) Deep Deterministic Policy Gradient (DDPG): It is an algorithm designed to solve continuous control problems, the core idea of DDPG is to maintain two neural networks: the actor network that learns the policy function and the critic network that learns the value function. The actor network produces an action given a state, and the critic network evaluates the quality of the action given the state. The actor network is trained by maximizing the expected return of the policy. The critic network is trained using the temporal difference (TD) error, which is the difference between the predicted value and the actual value obtained from the environment. The TD error is used to update the critic network parameters using the Bellman equation. The actor network is updated using the policy gradient theorem, which is derived from the deterministic policy gradient algorithm. The gradient of the expected return with respect to the actor-network parameters is computed using the chain rule (8)

$$\nabla_{\theta^\mu} J \approx \mathbb{E} s_t \sim \rho^\beta, a_t \\ \sim \mu[\nabla \theta^\mu \mu(s_t|\theta^\mu) \nabla_a Q(s_t, a|\theta^Q)|_{s_t, a_t=\mu(s_t)}] \tag{8}$$

where $Q(s,a)$ is the estimated value of the current state-action pair obtained from the critic network, and $\mu(s|\theta^\mu)$ is the policy function obtained from the actor network. The DDPG algorithm has been shown to be effective in solving a wide range of continuous control tasks, like robotics.

IV. Results and Discussion

This section showcases the results obtained and a comparative analysis of the two algorithms used viz. DQN and DDPG, with the FST used as a baseline. In particular, the juxtaposition of a more computationally intense algorithm DDPG, as compared to DQN, and the corresponding difference in performance, as the intersection configurations are varied, is presented. Analysis of the variation in the performance of the models with modifications to the reward function has also been presented.

A. Single and Double Intersection Performance

TABLE II
Single and Double Intersection

	Single Intersection		Double Intersection	
Model	**FST**	**DQN**	**FST**	**DQN**
Average queue length	3.539	0.577	3.48875	2.5
Average waiting time	3.073	0.897	6.9775	5.114

Table II compares the average delay time and the average queue length achieved by the FST-30 at single and double intersections. The FST-30 has been specifically chosen as a benchmark as an increase or decrease in the signal period from the 30-second mark significantly degraded the performance. The DQN agent reduces the average queue length by 83.6% and the average waiting time by 67% respectively for the single intersection. For the double intersection, a corresponding improvement of 28.3% and 26% was seen. This is proof enough to indicate that there is a significant drop in the waiting time because of the DRL agent, as compared to the legacy FST approach, because unlike the latter, the former accounts for real-time traffic density and can dynamically adapt. Thereby proving the utility of DRL-based ITSC systems over baseline FST.

B. Grid of intersections

TABLE III
Grid of Intersections Performance - Average waiting time

	Model		
Grid Size	**DQN**	**DDPG (proposed)**	**DQN - multi agent**
1x1	20.406	16.0133	19.793
2x2	11.8716	9.1	11.7016
3x3	17.0688	15.666	16.3451
4x4	18.95	17.76	18.739

With the real-time implementation in mind, experiments were conducted on the grid of intersection framework, which not only increases the randomness of the system but also exponentially adds to the complexity that the model has to digest. From table III a 21.52%, 23.34%, 8.21%, and 6.31% reduction in the delay by the proposed DDPG approach, as compared to the simpler DQN approach [11], [21], [22], is observed for the respective grid sizes. Experiments with a DQN approach wherein each intersection of the grid was an individual agent specifically optimizing that intersection [23], were also conducted. It can be observed from table III that an improvement in the range of 2% to 5% of reduction in the delay time, was achieved by this multi-agent DQN model over the baseline DQN model.

However, it is important to note that because of the limited computational power available, training of the agents on the grid of intersections framework was limited to a certain number of time steps. Thus, the gradual reduction and plateauing of the performance of the DDPG agent compared to the DQN agent with the increase in the size of the grid can be attributed to this factor.

C. Reward function analysis

TABLE IV
Effect of reward function

	Exponent of reward function				
Grid size	**0.5**	**1**	**1.5**	**2**	**2.5**
2x2	8.1466	7.833	7.953	6.553	8.4066
3x3	8.086	8.58	7.78	8.633	7.8133

A brief excursion toward finding the optimal reward function led to us studying the variation in the performance of the DRL models when the exponent of the reward function (2) is changed. Results shown in table IV are obtained by varying the reward term exponent for a DDPG agent model and the significant variation in the average waiting time shows that it can materially affect the performance of the model; thereby requiring further investigation.

V. Conclusion

This work explores the application of the Deep Deterministic Policy Gradient (DDPG) and Deep-Q-Network (DQN) algorithm to optimize traffic flow over a single, double, and grid of intersections of different sizes. To prove the promise of the application of Deep Reinforcement Learning (DRL) algorithms to intelligent traffic management systems, this work showcases the superiority of the DQN agent over the Fixed Signal Time (FST) method, which is still in use in the majority of the world. This superiority in performance is owed to the ability of DRL algorithms to respond dynamically to nuanced situations and be trained on real-time data. Subsequently, the application of the proposed DDPG approach outperformed the vanilla DQN approach and the multi-agent DQN approach. These experimental findings on the computationally demanding grid of intersections environment show promise for the real-world application of DRL algorithms to the domain of intelligent traffic management systems, given the similarity of this framework to the real world in terms of complexity.

In future research, the authors aim to conceptualize and implement a novel real-time DRL based intelligent traffic management system. The system would consist of a camera module, followed by a computer vision algorithm counting the vehicles, which will then be relayed to the DRL algorithm, and the agent would decide the next state of the signal. Additionally, to optimize the aforementioned system, investigation of various other traffic parameters and reward functions is needed, to select the best set of features to send to the agents.

References

[1] M. B. Younes and A. Boukerche, "An efficient dynamic traffic light scheduling algorithm considering emergency vehicles for intelligent transportation systems," *Wirel. Netw.*, vol. 24, no. 7, p. 2451–2463, oct 2018. [Online]. Available: https://doi.org/10.1007/s11276-017-1482-5

[2] P. B. Hunt, D. I. Robertson, R. D. Bretherton, and R. I. Winton, "Scoot-a traffic responsive method of coordinating signals." Transport and Road Research Laboratory (TRRL), 1981.

[3] A. G. Sims and K. W. Dobinson, "The sydney coordinated adaptive traffic (scat) system philosophy and benefits," *IEEE Transactions on Vehicular Technology*, vol. 29, pp. 130–137, 1980.

[4] P. Mirchandani and F.-Y. Wang, "Rhodes to intelligent transportation systems," *IEEE Intelligent Systems*, vol. 20, no. 1, pp. 10–15, 2005.

[5] M. B. Younes and A. F. M. Boukerche, "Intelligent traffic light controlling algorithms using vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 65, pp. 5887–5899, 2016.

[6] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.

[7] R. Sundar, S. Hebbar, and V. 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.

[8] M. Zhu, X.-Y. Liu, and X. Wang, "Joint transportation and charging scheduling in public vehicle systems—a game theoretic approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 8, pp. 2407–2419, 2018.

[9] P. Mannion, J. Duggan, and E. Howley, *An Experimental Review of Reinforcement Learning Algorithms for Adaptive Traffic Signal Control.* Cham: Springer International Publishing, 2016, pp. 47–66.

[10] S. El-Tantawy, B. Abdulhai, and H. Abdelgawad, "Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (marlin-atsc): Methodology and large-scale application on downtown toronto," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1140–1150, 2013.

[11] W. Liu, G. Qin, Y. He, and F. Jiang, "Distributed cooperative reinforcement learning-based traffic signal control that integrates v2x networks' dynamic clustering," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 10, pp. 8667–8681, 2017.

[12] H. Wei, N. Xu, H. Zhang, G. Zheng, X. Zang, C. Chen, W. Zhang, Y. Zhu, K. Xu, and Z. Li, "Colight: Learning network-level cooperation for traffic signal control," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, ser. CIKM '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1913–1922. [Online]. Available: https://doi.org/10.1145/3357384.3357902

[13] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan 2016. [Online]. Available: https://doi.org/10.1038/nature16961

[14] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. P. Lillicrap, K. Simonyan, and D. Hassabis, "Mastering chess and shogi by self-play with a general reinforcement learning algorithm," *CoRR*, vol. abs/1712.01815, 2017. [Online]. Available: http://arxiv.org/abs/1712.01815

[15] OpenAI, C. Berner, G. Brockman, B. Chan, V. Cheung, P. Debiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, R. Józefowicz, S. Gray, C. Olsson, J. Pachocki, M. Petrov, H. P. de Oliveira Pinto, J. Raiman, T. Salimans, J. Schlatter, J. Schneider, S. Sidor, I. Sutskever, J. Tang, F. Wolski, and S. Zhang, "Dota 2 with large scale deep reinforcement learning," 2019. [Online]. Available: https://arxiv.org/abs/1912.06680

[16] E. Asodekar, A. Nookala, S. Ayre, and A. V. Nimkar, "Deep reinforcement learning for automated stock trading: Inclusion of short selling," in *Foundations of Intelligent Systems*, M. Ceci, S. Flesca, E. Masciari, G. Manco, and Z. W. Raś, Eds. Cham: Springer International Publishing, 2022, pp. 187–197.

[17] S. Chiu, "Adaptive traffic signal control using fuzzy logic," in *Proceedings of the Intelligent Vehicles '92 Symposium*, 1992, pp. 98–107.

[18] H. Ceylan and M. G. Bell, "Traffic signal timing optimisation based on genetic algorithm approach, including drivers' routing," *Transportation Research Part B: Methodological*, vol. 38, no. 4, pp. 329–342, 2004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0191261503000158

[19] T. Li, D. Zhao, and J. Yi, "Adaptive dynamic programming for multi-intersections traffic signal intelligent control," in *2008 11th International IEEE Conference on Intelligent Transportation Systems*, 2008, pp. 286–291.

[20] I. Arel, C. Liu, T. Urbanik, and A. G. Kohls, "Reinforcement learning-based multi-agent system for network traffic signal control," *IET Intelligent Transport Systems*, vol. 4, no. 2, pp. 128–135, 2010.

[21] J. Zeng, J. Hu, and Y. Zhang, "Training reinforcement learning agent for traffic signal control under different traffic conditions," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 4248–4254.

[22] S. Park, E. Han, S. Park, H. Jeong, and I. Yun, "Deep q-network-based traffic signal control models," *Plos one*, vol. 16, no. 9, p. e0256405, 2021.

[23] U. P. Tewari, V. Bidawatka, V. Raveendran, V. Sudhakaran, S. K. Shreeshail, and J. P. Kulkarni, "Intelligent coordination among multiple traffic intersections using multi-agent reinforcement learning," *ArXiv*, vol. abs/1912.03851, 2020.