Reinforcement Learning Algorithm on Traffic Light Control

Ooi Wei Hong *School of Computing*
Asia Pacific University of Technology & Innovation (APU) Technology Park Malaysia
57000 Kuala Lumpur, Malaysia
TP065329@mail.apu.edu.my

Lee Boon Jie
School of Computing
Asia Pacific University of Technology & Innovation (APU) Technology Park Malaysia
57000 Kuala Lumpur, Malaysia
TP065421@mail.apu.edu.my

Lee Qi Wen
School of Computing
Asia Pacific University of Technology & Innovation (APU) Technology Park Malaysia
57000 Kuala Lumpur, Malaysia
TP065363@mail.apu.edu.my

Lee Jun Wei
School of Computing
Asia Pacific University of Technology & Innovation (APU) Technology Park Malaysia
57000 Kuala Lumpur, Malaysia
TP072623@mail.apu.edu.my

Dr.Adeline Sneha J
Senior Lecturer/School of Computing
Asia Pacific University of Technology & Innovation (APU) Technology Park Malaysia
57000 Kuala Lumpur, Malaysia
adeline.john@ apu.edu.my

Dr.Kamalanathan Shanmugam
Senior Lecturer/School of Technology
Asia Pacific University of Technology & Innovation (APU) Technology Park Malaysia
57000 Kuala Lumpur, Malaysia
kamalanathan@apu.edu.my

Juhairi Aris Muhamad Shuhili
School of Computing
Asia Pacific University of Technology & Innovation (APU) Technology Park Malaysia
57000 Kuala Lumpur, Malaysia
juhairi.shuhili@apu.edu.my

Abstract*— The efficiency and efficacy of genetic algorithms (GA) and reinforcement learning (RL) algorithms in traffic signal control are examined in this study. The study looks into the benefits and drawbacks of RL and GA in the setting of bad traffic circumstances that increase fuel consumption and trip time. While GA investigates a larger solution area for optimizing traffic light control schemes, RL demonstrates flexibility through self-learning in diverse contexts. The work summaries the body of research on reinforcement learning's optimization for traffic signal management and shows how it may cut down on average travel time, delays, and stops. The presentation includes a thorough comparison of several approaches, such as particle swarm optimization and fuzzy logic. In order to overcome obstacles and enhance performance overall, hybridization with Deep Deterministic Policy Gradient and adjustments to GA are suggested as future paths and improvements. The hardware and software specifications, as well as the process used to apply RL to traffic light management, are described in the materials and techniques section. We address policy gradient algorithms, exploration/exploitation tactics, and the parameters of the algorithm. The effects of changing settings on average wait times and collisions are shown by the results. According to the study's findings, RL greatly enhances traffic flow and cuts typical wait times from 300 to 20 seconds, especially when parameters are optimized. Future studies can investigate how to combine RL with other algorithms for the best possible traffic management in practical situations.

Keywords*— *Reinforcement Learning, Genetic Algorithm, Traffic light control, Q-Learning

I. INTRODUCTION

This journal represents a deep analysis of Reinforcement Learning Algorithm and Genetic Algorithm, for its efficiency and effectiveness in traffic light control. Most vehicles waste time on the road and burn more fuel due to poor traffic conditions. Most vehicles waste time on the road and burn more fuel due to poor traffic conditions (Chian & Kamsin, 2023). The literature review investigates the strengths and weaknesses of Reinforcement Learning and Genetic Algorithms in Traffic Signal Control, as well as the optimization of Reinforcement Learning on traffic light control. Reinforcement Learning (RL) has great potential to tackle these limitations in the recommender systems (Shuhrat, Ramachandran, & Salam, 2021). The journal also did a clear comparison between different algorithms and Reinforcement Learning / Genetic Algorithms. It also discusses the future direction and improvements of these algorithms on traffic light control. Additionally, the journal outlines the materials and method required for the implementation of algorithms in the context of traffic light control.

II. LITERATURE REVIEW

A. *Strengths and Weaknesses of Reinforcement Learning and Genetic Algorithms in Traffic Signal Control*

The contribution of Reinforcement Learning and Genetic Algorithm to traffic signal control is significant, because of the unique strengths and weaknesses. Models like RL-TSC1, RL-TSC2 and RL-TSC3, outperform in adapting to different traffic flows and consistently improving its own decision-making (Patrick Mannion, Jim Duggan, and Enda Howley, 2016). The reason why Reinforcement Learning outperform fixed-queue timer systems is that the adaptability of Reinforcement Learning to variable environments through self-learning and real-time conditions. However, the implementation was not easy as the need for expertise in both traffic engineering and computational intelligence, high cost of human resource and computational intensity for large-scale networks, and sensitivity to parameter settings (Zhide Li， Kaiquan Chen, 2018).

On the other hand, Genetic Algorithm offers a wider solution space for finding global optimum solutions in complex traffic scenarios (Hua Wei Guanjie Zheng, Vikash Gayah, Zhenhui Li, 2021). This algorithm excels in trying different solution in order to find the best optimized traffic light control plans, avoiding the risk of suboptimal choices. Nonetheless, the weaknesses of this algorithm arise in terms of computational intensity and limitation of real-time applicability, scalability issues especially in large-scale urban cities traffic networks with complex infrastructure (Hua Wei Guanjie Zheng, Vikash Gayah, Zhenhui Li, 2021).

B. *Optimization of Reinforcement Learning on traffic light control*

Reinforcement Learning (RL) is a subset of Machine Learning, autonomously learns optimal behaviour by exploring actions through trial and error to maximize rewards (Reinforcement learning, 2023). Unlike supervised learning, reinforcement learning operates without predefined answers, making it suitable for automated systems. Despite its strengths, reinforcement learning algorithms have limitations, prompting research for optimization. Studies aim to improve reinforcement learning 's performance, specifically addressing congestion reduction, enhanced vehicle throughput, and overall traffic efficiency to optimize traffic flow at intersections.

Several studies, including one by Jaun et al. (2021), showcase significant improvements in traffic signal control through RL-based signal optimization models (Jaun, et al., 2021). Multi-Agent Reinforcement Learning (MARL) systems are being investigated for enhancing coordination across intersections. The results demonstrate substantial reductions in delay, stops, and average travel time, particularly during peak hours.

Moreover, Kumar, Nistala Venkata Kameshwer, & Vijay K. (2023) propose a model using RL to classify vehicles and assign different weights, effectively reducing average waiting time by up to 91.7% (Kumar, Nistala Venkata Kameshwer, & Vijay K., 2023). In addition, Xiaoyuan, Xusheng, Guiling, & Zhu (2019) leverage Deep Reinforcement Learning (DRL) for traffic light control, introducing a 3DQN model that outperforms fixed-time signals by over 20% in average waiting time reduction (Xiaoyuan, Xusheng, Guiling, & Zhu, 2019). The study underscores the effectiveness of DRL techniques in optimizing traffic flow and reducing congestion.

Furthermore, Zibo, Tongchao, Wenxing, Fengyao, & Liguo (2021) provide a comprehensive review of traffic signal control, highlighting RL as a promising alternative to traditional methods (Zibo, Tongchao, Wenxing, Fengyao, & Liguo, 2021). The literature emphasizes challenges in RL applications, such as efficient state representation and reward formulation, and concludes that DRL-based approaches generally outperform traditional methods. The simulation results affirm the superiority of DRL over Fixed Signal Timing (FST) control schemes in various traffic modes (P1–P3). DRL consistently reduces waiting times and queue lengths, demonstrating its effectiveness in optimizing traffic signal control at intersections.

In conclusion, the fusion of optimization techniques with RL holds promise for enhancing traffic flow at signalized intersections. The document suggests nuanced strategies, including fine-tuning reward structures, incorporating sophisticated decision-making models like the 3DQN, and adopting principles from MARL. These insights provide a robust foundation for future research and practical implementation of adaptive, efficient, and responsive traffic signal control systems, mitigating congestion and enhancing traffic flow in urban environments.

C. *Comparison with Different Methodologies*

To gain a deeper comprehension of traffic light regulation using reinforcement learning, we have focused on Reinforcement Learning (RL) and Genetic Algorithms (GAs) in the context of traffic signal control, exploring a variety of computational approaches for intelligent traffic control systems (Aradi, 2020). To shed light on these systems' distinctive tactics for optimizing urban traffic flow, the paper contrasts them with other methodologies like fuzzy logic, particle swarm optimization (PSO), and deterministic algorithms (Alam, 2013); (Dezani, 2014); (Silva, 2022).

In comparison with Fuzzy Logic, RL is highlighted for its adaptability to dynamic scenarios through learning optimal control policies, while Fuzzy Logic employs a rule-based system emulating human decision-making processes (Aradi, 2020); (Alam, 2013). The choice between RL and Fuzzy Logic depends on specific traffic scenario requirements.

The integration of GAs with High-Level Petri net models is discussed as a distinctive approach, emphasizing its effectiveness in optimizing traffic flow by considering alternative routes and real-time adjustments (Dezani, 2014). This approach contrasts with a fuzzy logic-based strategy, showcasing the diversity of methodologies in traffic signal control.

PSO is distinguished in the comparison for its ease of use and efficiency in optimizing traffic signal systems, leading to considerable decreases in average waiting (Silva, 2022). GAs's distinct search strategy is compared to PSO; both approaches have benefits and need more investigation to allow for thorough comparisons.

In adaptive traffic signal control, deterministic algorithms are compared with reinforcement learning (RL); in large data volume scenarios, the former outperforms (Abdulhai, 2003)The deterministic algorithm performs better because of its comprehension of the dynamics of the traffic system, which emphasizes its ability to adjust to changing traffic demands.

Finally, the comparison highlights the ever-changing field of intelligent traffic control systems and emphasizes the range of computational approaches that are at one's disposal (Aradi, 2020); (Alam, 2013); (Dezani, 2014); (Silva, 2022); (Abdulhai, 2003). The study underscores how customized approaches based on unique traffic situations and preferences are required to manage complexity in transportation networks. It is believed that more investigation and real-world comparisons are necessary to fully comprehend the advantages and disadvantages of each approach for intelligent traffic control systems.

D. Future directions and improvements

Through three experimental investigations that combine their benefits, it investigates the direction and advancement of genetic algorithm (GAs) and reinforcement learning (RL) in the future. (Bonarini, 2019). The main goal is to overcome the illusory generalization limit – one of the limits of GAs in tandom environments. In order to comprehend its interaction with GAs and create a hybridized procedural technique, the study explores the advanced Deep Deterministic Policy Gradient (DDPG) in deep reinforcement learning. (Bonarini, 2019)

Bonarini (2019) has done experiments that showcase a novel hybrid process that integrates the advantage of reinforcement learning approaches with genetic algorithm. This methodology introduces Policy Feedback as an off-policy learning method. This study encourages creativity in procedural procedures and tackles difficulties in stochastic conditions. The evaluation covers a wide range of industries, including facility planning, scheduling and operation management, and it demonstrates how GAs may be successfully integrated with other optimization strategies to produce better performance measures. (Bonarini, 2019))Vijay Chahar er al. (2021) offer a comprehensive review of generic algorithm modifications, weighing the benefits and drawbacks of each. The study highlights how GAs should be strategically integrated with other optimization techniques, providing a road map for overcoming challenges and improving overall performance indicators. (Vijay Chahar, 2020)

The paper highlights the uses of RL and Deep RL frameworks in fluid dynamics while providing insights into these models in parallel. In their discussion of problems and possible routes for integrating Deep RL with active flow control (AFC), Vingnon et al. (2023) give the fluid mechanics community a toolkit for dealing with practical issues. (C Vignon, 2023)

While acknowledging the drawbacks of GAs, such as high processing costs and intricate parameter settings, Aymeric Vie(2020) offers several possible remedies, including GPU, parallel and quantum computing and inventive representation strategies. The favourable prognosis presents novel opportunities for the application of GAs in the future. (Aymeric Vi´e, 2020)

The combination of these contributions broadens our comprehension of the intricate relationships between RL, GAs and their hybridizations, stimulating original thought and promoting more study of adaptive systems and optimization. The flexibility of GAs in stochastic environments and the necessity of an all-encompassing approach to optimization issues are highlighted in the review's conclusion. (Bonarini, 2019) All things considered, the literature presents a comprehensive picture of the changing scene, addressing issues and pointing forth possible directions for future development.

III. Materials and Method

A. Materials

i. Software Requirements

For this study paper, PyCharm is used to run the provided source code. Pycharm is an advanced Integrated Development Environment (IDE) created especially for Python developers. Python was chosen as the programming language for the provided source code due to its simple syntax and ease of reading. Applications like the reinforcement learning technique for traffic light control may be developed and maintained more quickly according to its clear and simple code structure. It also boasts comprehensive library standards, offering a variety of modules and packages. The provided source code has been installed and imported into three libraries which are Pygame, Scipy and Numpy. Pygame is used to develop a graphical user interface to visualize the reinforcement learning-based traffic light controller. Scipy is essential to solving the mathematical and scientific difficulties involved in applying reinforcement learning algorithms to optimize the traffic light control system. Additionally, Numpy is also used to perform a variety of mathematical operations on arrays, which improves the computational capabilities of the reinforcement learning traffic light control system. The reinforcement learning algorithm is executed in the Windows 11 operating system.

ii. Hardware Requirements

The hardware that has been used for executing the given reinforcement learning algorithm on traffic light control in this paper is the NVIDIA GeForce RTX2050 for the Graphic Processing Unit (GPU). The reinforcement learning algorithm itself does not heavily rely on GPU acceleration in the source code. Intel(R) Core (TM) i5-13420H is the central processing unit (CPU) employed in this research work. The capabilities of the CPU may have an impact on the reinforcement learning algorithm's performance. The computations involved in the reinforcement learning method include value function calculations, reward prediction, q-value updates, and more. The CPU's processing power determines how quickly those calculations are completed. The computer's reinforcement learning method makes use of 8 GB of RAM. It involves learning rate and discount factor updates, q-value storage, and computations that require memory.

B. Methodology

Reinforcement learning is a computational approach geared towards achieving specific goals by allowing a computer system to interact with an unfamiliar and dynamic environment. With the use of this learning paradigm, the computer can decide for itself to maximise the total reward linked to a particular task. Interestingly, reinforcement learning functions both without direct human intervention and without explicit programming for task completion. An overview of a situation involving reinforcement learning is shown in the diagram below.

The main goal of reinforcement learning is to teach an agent how to carry out a task in an environment whose properties are unknown. In this scenario, the environment provides the agent with observations and rewards, while the agent transmits actions back to the environment. In relation to the overall task aim, the reward functions as a metre to indicate the effectiveness of an action.

Two essential parts of the agent are a learning algorithm and a policy.

1) Policy

The policy is a mapping process that chooses activities according to environmental observations. Usually, the policy is a function approximator, like a deep neural network, with programmable parameters.

2) Learning Algorithm

Based on the actions, observations, and incentives that are obtained, the learning algorithm continuously modifies the policy's parameters. The learning algorithm's main objective is to repeatedly improve the policy to find the ideal set of parameters that will maximise the total reward earned throughout the task.

Reinforcement learning basically involves the agent learning its best behaviour through a sequence of environment-trial-and-error interactions without direct human intervention.

To illustrate, consider the task of parking a vehicle using an automated driving system as an example. The objective here is for the vehicle's computer (the agent) to autonomously park the vehicle correctly. The controller generates steering, braking, and acceleration orders (actions) by utilising readings from several sensors (observations), such as cameras, accelerometers, gyroscopes, GPS receivers, and lidar. The actuators in charge of driving the car receive these action commands after that. Except the agent, every factor in this scenario is part of the environment: wind, road conditions, vehicle dynamics, and other variables.

The computer learns by parking the car repeatedly using a trial-and-error method. To facilitate this learning process, a signal is given that denotes failure (reward) if the vehicle does not reach the intended position and orientation and success when it does. The computer modifies its action-selection strategy (policy) after every trial to maximise the reward (learning algorithm). The computer keeps going through this iterative process until it figures out the best course of action for properly parking the car.

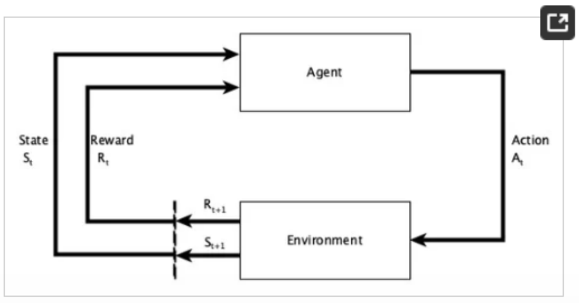


Figure 1: Reinforcement Learning Block Diagram

An agent is a decision-maker who examines the environment and acts in accordance with its true condition (Kolat, 2023). Every activity is assessed based on its calibre. Depending on whether an activity is desired or undesirable, the agent is either rewarded or penalised. Markov decision process (MDP), described in terms of <S, A, R, P>, is the basis for reinforcement learning (RL):

S: Set of observable states.

A: Set of possible agents' actions.

R: Set of gathered rewards based on the quality of the action.

P: The policy is to decide which action is selected at a given state.

In reinforcement learning (RL), a well-crafted rewarding strategy has a significant impact on how well the agent's neural network (NN) is tuned to produce the intended behaviour. By modifying a network's responses, this kind of learning seeks to enable it to function correctly in untested contexts outside of the training set.

IV. Algorithm Implementation

A. Purpose

A branch of machine learning called reinforcement learning (RL) aims to direct an agent to perform in a way that maximizes cumulative rewards within a given environment. In discrete time steps, the agent engages with the environment by obtaining observations (st) from the state space (S), acting (at) in the action space (A), and earning scalar rewards (rt) in response to its activities. An action-to-observation policy (π) governs the behavior of the agent. The agent's learning of a policy that eventually results in the largest cumulative reward is the aim.

B. Parameters

The exploration/exploitation issue is highlighted in the context of reinforcement learning, highlighting the necessity of striking a balance between the exploitation of learnt knowledge and the discovery of novel alternatives. A few discrete action exploration strategies are covered, such as the well-known ε-greedy exploration, with an emphasis on the progressive shift over time from exploration to exploitation. Turning the conversation to continuous actions, the criticism focuses on the implicit investigation in stochastic policies, specifically the supposition that actions have a Gaussian distribution. Problems including noisy trajectories and their detrimental effects on robotic control are emphasized, and a manual variance adaptation approach is suggested to reduce unwanted behavior. Finally, a thorough evaluation of existing methods that use Gaussian action-perturbing exploration in continuous action spaces is done. (Thomas Rückstieß, 2010)

When it comes to policy gradient algorithms, the main issue that has to be addressed is the significant variation in gradient estimates, which causes a delayed convergence. This problem results from repetitive sampling from a probabilistic strategy at each time-step, which introduces noise into the gradient estimate. Parameter-based exploration is a suggested substitute that modifies policy parameters directly instead of affecting the actions that follow. In the segment preceding every episode, a particular approach that uses finite differences for gradient estimation is introduced. Benefits of this strategy include reduced gradient computations, action consistency, noise-free trajectories, and flexibility. (Thomas Rückstieß, 2010)

Within the field of Stochastic Gradient Descent (SGD) techniques, the step-size parameter () is typically chosen by hand, however automated options have been investigated. Theoretical factors, including conditions derived from stochastic approximation theory, provide little help in practice and frequently lead to learning that is considered to be overly sluggish. Conventional selections such as α_t = 1/t, which work well with tabular Monte Carlo techniques, don't work well with Temporal Difference (TD) techniques, nonstationary issues, or situations requiring function approximation. Because they need O(d^2) step-size parameters, recursive least-squares approaches are unfeasible for big function approximation problems, despite being optimum for linear methods. Setting α = 1 would remove sample error in the tabular context after one target, but a slower learning rate is usually preferred. For example, it would take around 10 encounters for α = 1/10 to converge to the mean goal. As a general guideline, one should choose α = TE / (x>x ⇥ ⇤₁) for linear function approximation, where x is a randomly selected feature vector from the same distribution as the SGD input vectors. When feature vectors have lengths that are comparatively constant, this rule performs well. Though the idea of experiences with a state becomes less obvious, an analogous method is still required to achieve similar behavior to linear approximation in general function approximation, where experiences with a state lack a clear description. (Richard S. Sutton, n.d.)

V. Results and Discussion

A. Results

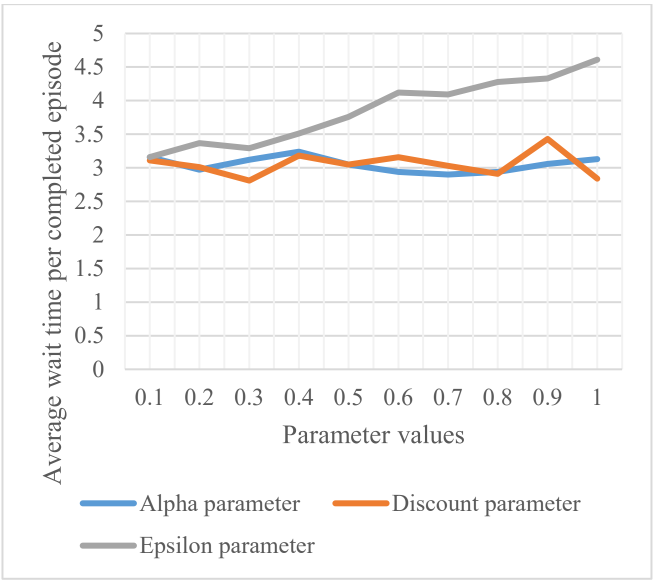


Figure 22: Results of different parameters (Average wait time per completed episode)

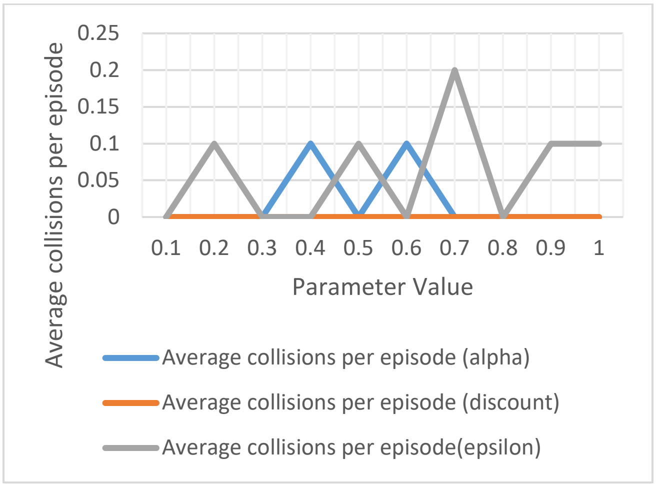
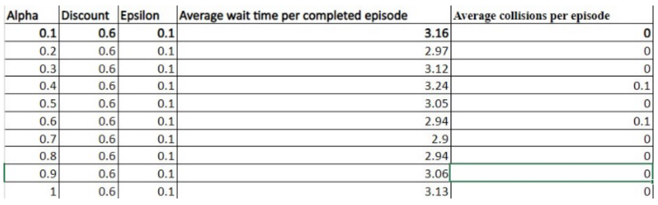
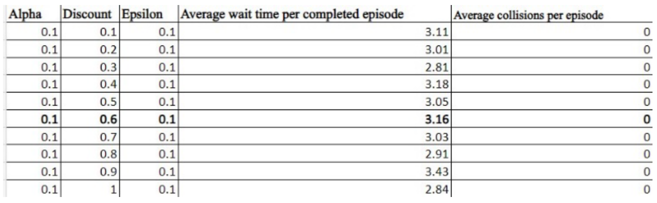


Figure 33: Results of different parameters (Average collisions per episode)



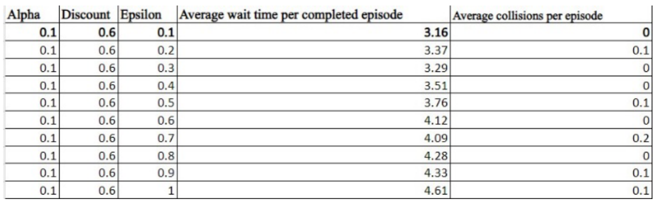
Alpha	Discount	Epsilon	Average wait time per completed episode	Average collisions per episode
0.1	**0.6**	**0.1**	**3.16**	**0**
0.2	0.6	0.1	2.97	0
0.3	0.6	0.1	3.12	0
0.4	0.6	0.1	3.24	0.1
0.5	0.6	0.1	3.05	0
0.6	0.6	0.1	2.94	0.1
0.7	0.6	0.1	2.9	0
0.8	0.6	0.1	2.94	0
0.9	0.6	0.1	3.06	0
1	0.6	0.1	3.13	0

Figure 44: Result of different values of Alpha Parameter



Alpha	Discount	Epsilon	Average wait time per completed episode	Average collisions per episode
0.1	0.1	0.1	3.11	0
0.1	0.2	0.1	3.01	0
0.1	0.3	0.1	2.81	0
0.1	0.4	0.1	3.18	0
0.1	0.5	0.1	3.05	0
0.1	**0.6**	**0.1**	**3.16**	**0**
0.1	0.7	0.1	3.03	0
0.1	0.8	0.1	2.91	0
0.1	0.9	0.1	3.43	0
0.1	1	0.1	2.84	0

Figure 55: Result of different values of Discount Parameter



Alpha	Discount	Epsilon	Average wait time per completed episode	Average collisions per episode
0.1	**0.6**	**0.1**	**3.16**	**0**
0.1	0.6	0.2	3.37	0.1
0.1	0.6	0.3	3.29	0
0.1	0.6	0.4	3.51	0
0.1	0.6	0.5	3.76	0.1
0.1	0.6	0.6	4.12	0
0.1	0.6	0.7	4.09	0.2
0.1	0.6	0.8	4.28	0
0.1	0.6	0.9	4.33	0.1
0.1	0.6	1	4.61	0.1

Figure 66: Result of different values of Epsilon Parameter

B. Discussion

Figure 2 shows the average wait time per completed episode for different parameters which are alpha (learning rate), discount and epsilon. Based on the figure 4, learning rate (alpha) determines the step size in updating the Q-values. In this case, an alpha value of 0.7 seems to strike a balance, leading to a lower average wait time which is 2.90 s. It indicates that a moderate learning rate contributes to more effective learning in the given environment. Moreover, the discount factor influences the agent's consideration of future rewards. A lower discount factor (0.3) in this case seems to result in a lower average wait time (2.81s) which is shown in figure 5, suggesting that giving less weight to future rewards can lead to improved decision-making. According to the figure 6, Epsilon controls the balance between exploration and exploitation. A low epsilon (0.1) suggests a higher focus on exploiting the current knowledge, leading to a lower average wait time (3.16s). However, as epsilon increases, the agent explores more, which can result in longer wait times as the agent takes suboptimal actions to discover new possibilities. In a nutshell, the optimal parameter values depend on the specific characteristics of the environment and the task at hand. Balancing exploration and exploitation is crucial for achieving good performance. Too much exploration (high epsilon) can lead to increased wait times, while too little exploration might cause the agent to miss potentially better strategies (low epsilon). Learning rate (alpha) and discount factor interact in influencing the agent's behavior, and finding the right combination is key to achieving optimal performance.

Furthermore, the learning rate (alpha) affects how much the agent adjusts its estimates based on new information. The slight increase in collisions at alpha=0.4 and alpha=0.6 may indicate that these learning rates result in more dynamic Q-value updates, possibly leading to a few collisions. The discount factor influences the agent's consideration of future rewards. In this case, it seems that varying the discount factor within the given range does not have a significant impact on collision rates, as all values remain at 0. Epsilon controls the balance between exploration and exploitation. The increase in collisions at epsilon=0.7 suggests that, in this scenario, a higher exploration rate results in more collisions. This is expected as the agent explores more and takes riskier actions. In summary, the learning rate (alpha) influences how much an agent adjusts its estimates based on new information. Slight increases in collisions at alpha=0.4 and alpha=0.6 may suggest more dynamic Q-value updates. The discount factor, within the given range, doesn't significantly impact collision rates. Epsilon, controlling exploration-exploitation balance, shows increased collisions at epsilon=0.7, indicating higher exploration leading to riskier actions and more collisions.

Figure 2 shows the average wait time per completed episode for different parameters which are alpha (learning rate), discount and epsilon. Based on the figure 4, learning rate (alpha) determines the step size in updating the Q-values. In this case, an alpha value of 0.7 seems to strike a balance, leading to a lower average wait time which is 2.90 s. It indicates that a moderate learning rate contributes to more effective learning in the given environment. Moreover, the discount factor influences the agent's consideration n of future rewards. A lower discount factor (0.3) in this case seems to result in a lower average wait time (2.81s) which is shown in figure 5, suggesting that giving less weight to future rewards can lead to improved decision-making. According to the figure 6, Epsilon controls the balance between exploration and exploitation. A low epsilon (0.1) suggests a higher focus on exploiting the current knowledge, leading to a lower average wait time (3.16s). However, as epsilon increases, the agent explores more, which can result in longer wait times as the agent takes suboptimal actions to discover new possibilities.

VI. Conclusion

The paper has a conclusion that the Reinforcement Learning Algorithm is suitable to be implemented for traffic light control plan with results performing the best when setting the alpha, discount, epsilon value at 0.1, 0.3 and 0.1 respectively. This will bring the average wait time from 3.46 seconds to 2.81 seconds.

The research can be enhanced by combining other algorithms such as fuzzy logic, particle swarm optimization (PSO), and deterministic algorithms while using real-world scenarios data in virtual environments. This will identify the best combination of algorithms to be used in real-time system.

Acknowledgment

I would like to express my sincere gratitude to Dr. Adeline Sneha John Chrisastum for her invaluable guidance and unwavering support throughout the completion of this research paper. Dr. Chrisastum's expertise, encouragement, and insightful feedback have been instrumental in shaping the quality of my work. Her dedication to fostering a dynamic learning environment and commitment to excellence have significantly enriched my understanding of the research matter. Her mentorship has been a source of inspiration, and I am fortunate to have had the opportunity to learn under her guidance.

References

Abdulhai, B. P. (2003). *Reinforcement learning for true adaptive traffic signal control.* Retrieved from Journal of Transportation Engineering, 129(3), 278–285: https://doi.org/10.1061/(asce)0733-947x(2003)129:3(278

Alam, J. P. (2013). *Intelligent traffic light control system for isolated intersection using fuzzy logic.* Retrieved from Research Gate: https://doi.org/10.13140/RG.2.1.4854.6406

Aradi, S. (2020). *Survey of Deep Reinforcement Learning for Motion Planning of Autonomous Vehicles.* Retrieved from arXiv(Cornell University): http://export.arxiv.org/pdf/2001.11231

Aymeric Vi´e, A. M. (2020, November 6). *Qualities, Challenges and Future of Genetic Algorithms*. Retrieved from SSRN: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3726035

Bonarini, A. (2019). *Genetic Algorithms And Reinforcement Learning : Three Studies On Limits, Improvements And Hybridization*. Retrieved From Politecnico: https://www.politesi.polimi.it/bitstream/10589/152563/5/2019_12_Espositi.pdf

C Vignon, J. R. (2023, January). *Recent advances in applying deep reinforcement learning for flow control: perspectives and future directions*. Retrieved from ResearchGate: https://www.researchgate.net/publication/367476568_Recent_advances_in_applying_deep_reinforcement_learning_for_flow_control_perspectives_and_future_directions

Chian, C. Y., & Kamsin, D. (2023). Implementation of Intelligent Transportation System using Smartphone Sensors to Improve Traffic Conditions. *Journal of Applied Technology and Innovation*, (e -ISSN: 2600-7304) vol. 7, no. 1.

Dezani, H. M. (2014). *Genetic algorithm-based traffic lights timing optimization and routes definition using Petri net model of urban traffic low.* Retrieved from IFAC Proceedings Volumes, 47(3), 11326-11331: https://doi.org/10.3182/20140824-6-za-1003.01321

Hua Wei Guanjie Zheng, Vikash Gayah, Zhenhui Li. (2021). Recent Advances in Reinforcement Learning for Traffic Signal Control: A Survey of Models and Evaluation. *ACM SIGKDD Explorations Newsletter,Volume 22,Issue 2*, 12-18.

Jaun, G., Minhyuck, L., Chulmin, J., Yohee, H., Youngchan, K., & Junwon, K. (2021). Traffic Signal Optimization for Multiple Intersections Based on Reinforcement Learning. *Applied Sciences*, 11(22), 10688. https://doi.org/10.3390/app112210688.

Kolat, M. K. (2023). *Multi-Agent Reinforcement learning for Traffic Signal Control: a cooperative approach.* Retrieved from Sustainability, 15(4), 3479: https://doi.org/10.3390/su15043479

Kumar, R., Nistala Venkata Kameshwer, S., & Vijay K., C. (2023). Adaptive trafc light control using deep reinforcement. *Multimedia Tools and Applications*, https://doi.org/10.1007/s11042-023-16112-3.

Patrick Mannion, Jim Duggan, and Enda Howley. (2016). An Experimental Review of Reinforcement Learning Algorithms for Adaptive Traffic Signal Control. *Autonomic Road Transport Support Systems*.

Reinforcement learning. (2023, April 18). Retrieved from Geeks for Geeks: https://www.geeksforgeeks.org/what-is-reinforcement-learning/

Richard S. Sutton, d. A. (n.d.). *Reinforcement Learning*. Retrieved from https://www.andrew.cmu.edu/course/10-703/textbook/BartoSutton.pdf?fbclid=IwAR0HPCOKOWnK2nN9ijL3Ex23LkEPthhb-xhqWLyv66SbdZJoP7NF_MTF2qg

Shuhrat, B., Ramachandran, C., & Salam, Z. (2021). Recommender systems enhancement using deep reinforcement learning. *Journal of Applied Technology and Innovation*, (e -ISSN: 2600-7304) vol. 5, no. 4.

Silva, G. O. (2022). *On tuning the particle swarm optimization for solving the traffic light problem.* Retrieved from In Lecture Notes in Computer Science (pp. 68–80): https://doi.org/10.1007/978-3-031-10562-3_6

Thomas Rückstieß, F. S. (2010, March). *Exploring Parameter Space in Reinforcement Learning*. Retrieved from ResearchGate: https://www.researchgate.net/publication/227320345_Exploring_Parameter_Space_in_Reinforcement_Learning

Vijay Chahar, S. K. (31 October, 2020). *A review on genetic algorithm: past,present, and future*. Retrieved from ResearchGate: https://www.researchgate.net/publication/344902115_A_Review_on_Genetic_Algorithm_Past_Present_and_Future

Xiaoyuan, L., Xusheng, D., Guiling, W., & Zhu, H. (2019). Deep Reinforcement Learning for Traffic Light. *IEEE Transactions on Vehicular Technology*, 68(2), 1243–1253. https://doi.org/10.1109/tvt.2018.2890726.

Zhide Li， Kaiquan Chen. (2018). Research on Timing Optimization of Reginal Traffic Signals Based on Improved Genetic Algorithm. *IOP Conference Series: Materials Science and Engineering*.

Zibo, M., Tongchao, C., Wenxing, D., Fengyao, J., & Liguo, Z. (2021). Adaptive Optimization of Traffic Signal Timing via Deep Reinforcement Learning. *Journal of Advanced Transportation*, 1–14. https://doi.org/10.1155/2021/6616702.