Finding the Optimal Policy for a Simulated Traffic Intersection with Reinforcement Learning"

By Arlo Rostirolla

Table of Contents

List of Tables

List of Figures

List of Appendices

Introduction

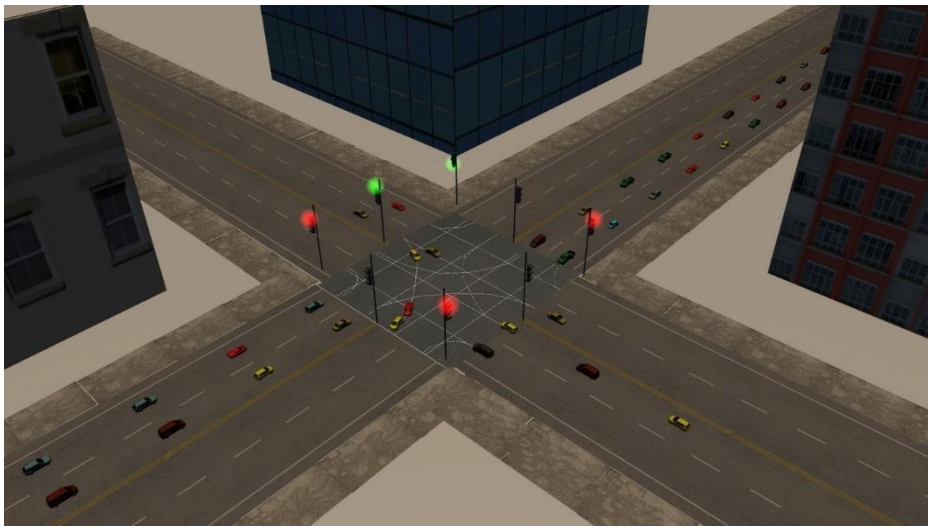
Traffic control is an ideal candidate for optimization via machine learning methods. Inefficient traffic light policies can contribute to pollution, traffic jams, and an increase in commute time for the public. Reinforcement learning is a prime candidate algorithm for this problem, as reward signals are very dense when tied to the throughput of an intersection. With 8 possible actions, and a vector of integers representing the number of cars waiting in each lane, the state space of this problem is not unwieldly. With 8 lanes holding a maximum of 10 cars each, the state space is 10^8, or around 100,000 states. Although not small, it is far less than some other problems such as chess.

Having used RL in the past and seen its effectiveness, we chose this as our project. Previous attempts at implementing a DDQN had been foiled by a very difficult problem space (Berkely Pacman). Other researcher have had more success [3][4], using small action spaces and real intersection video, interpreted with a CNN. It will be argued in this paper that the PPO RL algorithm will be able to optimize the throughput of a real life intersection .

Methods

Simulation

The intersection between Lygon street and Brunswick road in Brunswick was simulated in Unity 2020-30-1. Cars were implemented as Finite State Machines (FSM), which followed preset lines using linear interpolation. Upon spawning, they would follow the first line to the intersection and wait for a green light. Once an action occurred that initiated a green light, the car would then follow the second line passing through the intersection. A ray cast would be sent through the front of the car, and if another car was within 5 metres, the FSM script would pause until that was not the case. **Figure 1.** *Simulation of Lygon street and Brunswick road in Unity*



A collider was placed just inside the intersection, so that once a car entered, a reward of one could be added, the car could be removed from the observation list and then destroyed with a 10 second timer. Throughput was customizable via parameters within the FSM, such as spawning rate and percentage of cars turning versus going straight. A traffic controller script controlled the light animations. It would access the main intersection agent file to check the action/time, and then follow a preset IEnumerator sequence via a switch statement.

Reinforcement Learning

Reinforcement Learning (RL) was implemented using the Unity ML-agents framework [1]. This framework allows easy integration of prewritten RL algorithms with custom unity environments. The observation vector consisted of 8 possible integer values: each representing the number of cars currently waiting in a single unique lane. With this implementation, the state space could be considered number of cars possible per lane to the power of the number of lanes, or 10^8 = 100,000 states. Although a large number, it is quite reasonable in relation to other problems RL could be applied to.

The 8 possible actions are presented in table 1 and generally follow the actions available to most intersections. Either one side of one lane had full access to the intersection (left, straight and right), both lanes running parallel had access to the intersection (straight, left) or both parallel lanes had access to turn right. An extension of this was attempted by implementing a discrete branch with 4 more actions, indicating the duration of the action. Each light action could either run for 10, 15, 20 or 25 seconds, with all running a yellow light for an extra five minutes on top. The reward was set to the number of cars that had passed through the intersection during the episode. Unity ML-Agents 2.01 was used, along with the Python MLAgents 0.29.0 package. Much of the guidance in implementation was gained from [2].

Table 1. *Discrete Action branches available to the agent*

Branch 1	Branch 2 (Extension)
1: Bottom light active (left, straight, right)	1: 15 seconds
2: Bottom and top light active (straight , left)	2: 20 seconds
3: Top light active (left, straight, right)	3: 25 seconds
4: Bottom and top right (right turn only)	4: 30 seconds
5: Left light active (left, straight, right)	
6: Left and right light active (straight , left)	
7: Right light active (left, straight, right)	
8: Left and right (right turn only)	

Results

The initial run was not successful. After increasing the batch, buffer size, and num layers parameters however (See Appendix A1), reward began to improve over time. A graph of this second attempts progress can be seen in figure 2. In this first experiment, all lanes had an equal spawning rate (1 per second) and an equal percentage of cars turning right or left. For evaluation, the real rate of traffic was timed. The throughput of all four directions were logged over a 15 minute timespan. Table 2 shows the calculated traffic rates and Appendix B shows the specific log results.

The real intersection using a static policy let through on average 51 cars per minute. The initial PPO run let through 57 cars per minute at the start, eventually reaching 81 cars per minute at its peak, however, this did not have the throughput parameters of the real intersection.

During evaluation, we wished to test PPO's performance against the baseline static policy, given the differences present between the simulation and the real intersection. It was observed that at this intersection the lights went through turns in clockwise order. The static policy in the simulation was based upon this. Every action was implemented for 15 seconds, before moving to the next. The simulated intersection using a static policy let through 1074 cars over 19.7 minutes, a rate of 54 cars per second.

Figure 2. *Intersection throughput per thousand steps with standard deviation bars (2nd PPO run).*

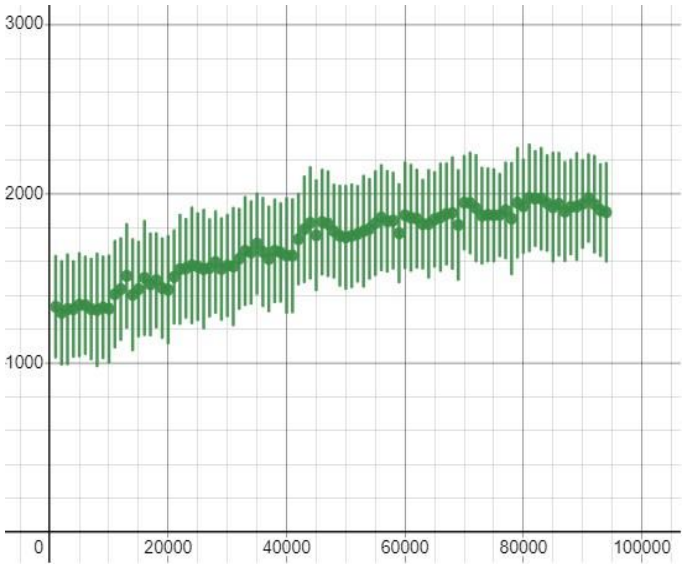


Table 2. *Calculated throughput of Lygon street and Brunswick road*

	% turning left	% turning right	% going straight	Cars per 15 minutes	Cars per minute	Seconds per car
North	11	17	72	174	12	5
West	12	22	66	215	14	4.35
South	21	6	73	171	12	5
East	14	18	68	210	14	4.35

The first attempt at training with the real-life throughput parameters did not result in learning. Attempts were made at changing parameters, such as increasing the time horizon, batch size, buffer size, and adding recurrent memory. The curiosity module was also tried, however the combination of these parameters led to a decrease in reward (See Figure 2). Keeping the hyperparameters but disabling curiosity and memory helped, albeit not enough to increase reward over time (See Figure 3).

The problem was simplified by removing the light timing action branch, setting all timings to 15 seconds, and reverting to the original hyperparameters that worked before. Learning was again observed and can be seen in Figure 5. By step 335,000, the model was letting 68.4 cars per minute through the intersection, 14.4 more a minute than the static policy.

Discussion

PPO RL proved to be a very conducive technique for this problem space. With a relatively small observation space, the algorithm was quickly able to learn an effective policy to optimize traffic throughput. It was quite difficult to synchronize the RL algorithm with the intersection. During early training attempts, it became evident that the lights were going out of sync with the actions, leading to cars running through red lights. It was assumed that this issue would not affect training, given PPO is only concerned with the actions of the cars, and the lights were a cosmetic addition. It was fixed in the final implementation by making the cars receive their signal from the light controller, which received its signal from the PPO agent.

Figure 3. *PPO with curiosity and memory, increased time horizon and batch/buffer size*

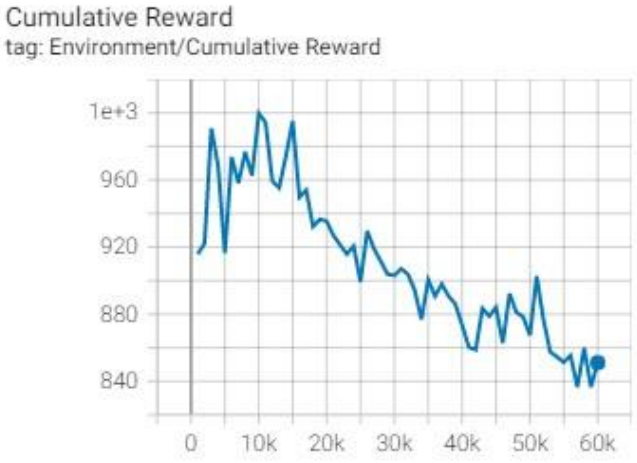
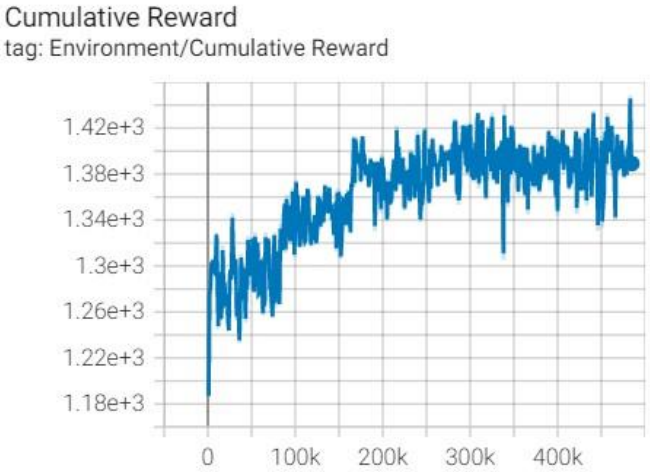


Figure 4. *Increased time horizon, batch and buffer size without curiosity/memory*

Figure 5. *Successful learning in the evaluation trial without timings*



Previously both the car and the light had received their signal from the agent.

The first, successful PPO model (see Figure 2) seemed to have learned to not use actions 4 and 8, where cars in two colinear lanes can only turn right. Some of the car FSMs in the right lane were set to go straight and would hold up the entire lane. The algorithm seemed to prioritize actions 1, 3, 5, and 7, where one lane has access to the entire intersection, and can turn left, right, or go straight. When a lane had built up traffic, this action let through the most cars. The model also prioritized

shorter action times. Once built up traffic had passed the intersection, the throughput is far lower than at the initial green light. It is optimal for the model to change the lights often to maximize actions where there is built up traffic.

Training seemed to break down when the real intersections throughput parameters were inserted. This is likely due to the breaking of symmetry leading to a more difficult problem. Interestingly, removing the extension light timings led to learning. The static policy used at the real life intersection did not have the ability to time lights, and thus it is unsurprising that attempts which used the extension light timings had an overall higher throughput. The static policy simulation only let through three more cars per minute than the real intersection. The policy with light timings let through 36 more than the real intersection at the start of training. It was unfortunate that this was not conducive to our reinforcement learning formulation. Previous research [3][4] had used very small action spaces. The authors of intellilight [3] had 2 actions, either stay on the current green light, or change. It is possible using 32 overall actions, along with real intersection data may have overcomplicated the problem

Collisions between cars were not checked for, given that all cars deterministically followed the road rules. Theoretically, the real intersection should have had an advantage, as more than 10 cars can wait in each lane, and some cars may not choose to follow the speed limit. It was evident though that its potential was held back by the static policy. Allowing higher customizability of the intersection, and using PPO to control it, led to higher rates of throughput.

Conclusion

It was argued in this paper that RL algorithms can lead to higher throughput, and this hypothesis was backed by the data. The initial PPO run with homogenous lane parameters learned a policy that increased the throughput by 24 cars per minute over 2 days of training.

The second attempt, using the real intersections lane parameters, and customizable light timings, started off letting 36 more cars per minute than the real intersection. Without light timings, the PPO model learnt to increase thoroughput by 14.4 cars per minute over 4 days of training. RL would be highly effective at optimizing the throughput of an intersection. One aspect not covered in this project is safety. This simulation assumed perfect drivers following perfect road conditions. Further research would be needed to guarantee that collisions do not occur from normal operation of the lights.

References

[1] "GitHub - Unity-Technologies/ml-agents: The Unity Machine Learning Agents Toolkit", *GitHub*, 2022. [Online]. Available: https://github.com/Unity-Technologies/ml-agents.

[2] *Intel.com*, 2022. [Online]. Available: https://www.intel.com/content/dam/develop/external/us/en/documents/gamedevwithunitymlagentsandintel-optimizedpythonpart2.pdf.

[3] H. Wei, G. Zheng, H. Yao, Z. Li, Intellilight: a reinforcement learning approach for intelligent traffic light control, 2018 ACM SIGKDD Proceedings of the 24th International Conference on Knowledge Discovery & Data Mining, ACM (2018), pp. 2496-2505.

[4] X. Liang, X. Du, G. Wang, and Z. Han, “A deep reinforcement learning network for traffic light cycle control,” IEEE Trans. Veh. Technol., vol. 68, no. 2, pp. 1243–1253, Feb. 2019.

Appendix A

1) Results of the second attempt at training a PPO agent, with equal flow of traffic between lanes

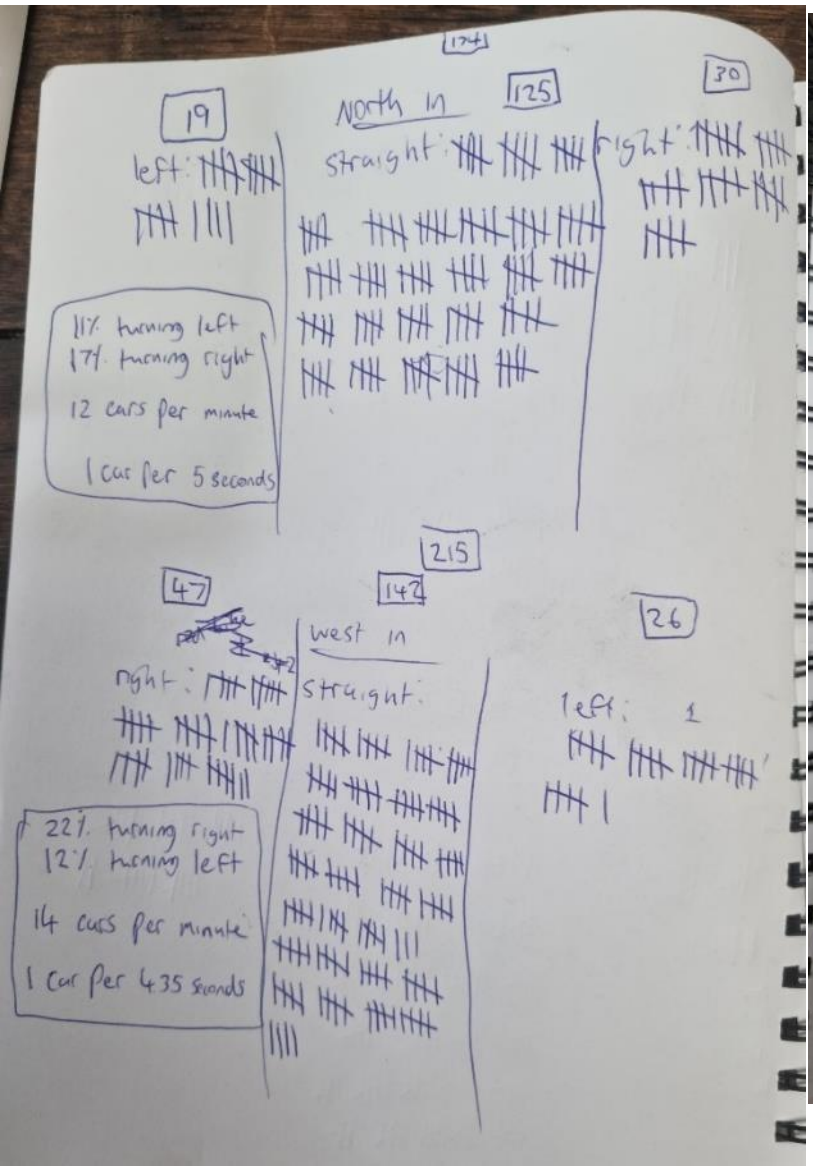
trainer_type: ppo hyperparameters: batch_size: 128 buffer_size: 10240 learning_rate: 0.0003 beta: 0.005 epsilon: 0.2 lambd: 0.95 num_epoch: 3 learning_rate_schedule: linear beta_schedule: linear epsilon_schedule: linear network_settings: normalize: True hidden_units: 128 num_layers: 1 vis_encode_type: simple memory: None goal_conditioning_type: hyper deterministic: False reward_signals: extrinsic: gamma: 0.9 strength: 1.0 init_path: None keep_checkpoints: 5 checkpoint_interval: 500000 max_steps: 999999999999 time_horizon: 5 summary_freq: 1000 threaded: False self_play: None behavioral_cloning: None	[INFO] Intersection. Step: 1000. Time Elapsed: 1403.328 s. Mean Reward: 1333.000. Std of Reward: 296.385. Training. [INFO] Intersection.Step: 10000.Time Elapsed: 14164.037 s. Mean Reward: 1320.741. Std of Reward: 311.420. Training. [INFO] Intersection. Step: 20000. Time Elapsed:27605.172 s. Mean Reward: 1433.640. Std of Reward: 312.986. Training. [INFO] Intersection. Step: 30000.Time Elapsed:40740.845 s. Mean Reward: 1576.376. Std of Reward: 295.890. Training. [INFO] Intersection. Step: 40000.Time Elapsed: 53908.677 s. Mean Reward: 1635.364. Std of Reward: 335.252. Training. INFO] Intersection. Step: 50000. Time Elapsed: 9262.031 s. Mean Reward: 1742.500. Std of Reward: 300.554. Training [INFO] Intersection. Step: 60000. Time Elapsed: 1476.660s. Mean Reward: 1873.039. Std of Reward: 308.957. Training. [INFO] Intersection. Step: 70000.Time Elapsed:14881.142 s. Mean Reward: 1946.932. Std of Reward: 272.474. Training. [INFO] Intersection. Step: 80000.Time Elapsed:28478.814 s. Mean Reward: 1926.356. Std of Reward: 272.099. Training. [INFO] Intersection. Step: 90000.Time Elapsed: 1073.015s. Mean Reward: 1940.066. Std of Reward: 255.914. Training. [INFO] Intersection. Step: 93000. Time Elapsed: 5321.799s. Mean Reward: 1903.343. Std of Reward: 267.680. Training

2). The final attempt at training PPO, using the Lygon x Brunswick intersection rates, and without the light timing action branch

Behaviors: Intersection: trainer_type: ppo hyperparameters: batch_size: 128 buffer_size: 81920 learning_rate: 0.0003 beta: 0.005 epsilon: 0.2 lambd: 0.95 num_epoch: 3 learning_rate_schedule: linear network_settings: normalize: True hidden_units: 128 num_layers: 1 vis_encode_type: simple deterministic: False reward_signals: extrinsic: gamma: 0.9 strength: 1.0 keep_checkpoints: 5 checkpoint_interval: 500000 max_steps: 999999999999 time_horizon: 5 summary_freq: 1000 threaded: False	[INFO] Intersection. Step: 1000. Time Elapsed: 266.751 s. Mean Reward: 1186.894. Std of Reward: 255.890. Training. [INFO] Intersection. Step: 50000. Time Elapsed: 25472.188 s. Mean Reward: 1296.646. Std of Reward: 218.053. Training. [INFO] Intersection. Step: 100000. Time Elapsed: 51954.987 s. Mean Reward: 1346.947. Std of Reward: 217.144. Training. [INFO] Intersection. Step: 150000. Time Elapsed: 77853.348 s. Mean Reward: 1366.261. Std of Reward: 205.665. Training. [INFO] Intersection. Step: 200000. Time Elapsed: 104663.920 s. Mean Reward: 1382.137. Std of Reward: 193.438. Training. [INFO] Intersection. Step: 250000. Time Elapsed: 141895.826 s. Mean Reward: 1387.066. Std of Reward: 175.768. Training. [INFO] Intersection. Step: 300000. Time Elapsed: 163261.488 s. Mean Reward: 1424.723. Std of Reward: 190.038. Training. [INFO] Intersection. Step: 350000. Time Elapsed: 8060.317 s. Mean Reward: 1382.453. Std of Reward: 201.043. Training. [INFO] Intersection. Step: 400000. Time Elapsed: 37272.276 s. Mean Reward: 1343.266. Std of Reward: 198.396. Training. [INFO] Intersection. Step: 450000. Time Elapsed: 63369.169 s. Mean Reward: 1412.989. Std of Reward: 190.866. Training. [INFO] Intersection. Step: 500000. Time Elapsed: 89285.151 s. Mean Reward: 1383.705. Std of Reward: 191.669. Training. [INFO] Intersection. Step: 550000. Time Elapsed: 111919.824 s. Mean Reward: 1374.617. Std of Reward: 199.178. Training. [INFO] Intersection. Step: 600000. Time Elapsed: 134076.785 s. Mean Reward: 1385.330. Std of Reward: 190.993. Training.

Appendix B

1) Fifteen minute observations of the north and west section of Brunswick road x lygon street



2) Fifteen minute observations of the south and east section of Brunswick road x lygon street