*Self-Learning Traffic Mitigation Solution*

Jason Moynihan   
COMP-4200  
University of Massachusetts, LowellLowell, MA  
Jason\_Moynihan@student.uml.edu  
Jordan Urbaez  
COMP-4200  
University of Massachusetts, Lowell

Lowell, MA Jordan\_UrbaezLu@student.uml.edu

*Abstract*—There is a great deal of traffic that is caused by urban sprawl, and subsequently a great number of traffic light led intersections. The mainstream explanation and justification of this method of mitigation argued by civil engineers worldwide has been adopted all over the world. Here, we report the findings of an Artificial Intelligence adaptation of the traditional solution explained above. With the addition of a rational, multi-agent system, the problems caused by urban sprawl can be mitigated much further. This paper will discuss the problem, the approach, the Artificial Intelligence methodologies, and an analysis of results from our agent.

# Introduction

Urban Sprawl and the mainstreaming of the automobile has created an outstanding issue. The current implementation of crossroad traffic handling uses traditional stoplight system with a pre-set cycle period. This cycle period can be changed, but it requires manual adjustment. This current implementation leaves a lot to be desired in terms of optimality. Lots of time is lost in the world waiting in traffic, specifically at stoplights. If the amount of traffic at any light in an intersection could be identified (and compared to the other lights at the intersection), and assuming that there is a pattern of inequal lengths of traffic waiting, the lights could adjust themselves to account for and mitigate that inequality. This begs the question “How can an intersection with multiple lights adjust on their own and adapt to their surroundings?” Researching techniques in reinforcement learning revealed that self-learning agents react to a reward value. An easy, high level example of the logic behind the problem is the following. In a four-light intersection, lights A and C are a pair while lights B and D are also a pair. A and C have the same green time as one another, while B and D have the same green time as one another. If there is too much buildup of traffic at lights A and C, then those two lights need to stay green for longer. Using parameters including difference in traffic waiting at each pair of lights and the difference in green time between the two pairs of lights, we can calculate a reward value. This reward value can be used to add/subtract green time from the pairs of lights. This creates a continuously adaptive system that will constantly attempt to optimize the traffic flow at its intersection. Our hypothesis is that our agent will substantially outperform traditional intersections with partially busy data sets, and only show slight improvement with the edge case and rush hour data sets.

# Literature Review

As humans we use road transportation every day to get around the globe in a fast, efficient way. At the base, of this fundamental way of transportation, is the system we use to control it, intersections. Since these systems aren’t perfect, we are starting to see the harm that this road transportation system can cause. Problems like idling can waste an individual's time on the road. This congestion on streets can cause pollution in the air which can cause lots of harm to those individuals in big cities. These problems will continue to worsen if a solution isn’t solved. For this project, the problem at hand is to create some sort of technology/algorithm to make intersections even more efficient than they are. To solve this problem, many people create algorithms that work together with cameras/sensors that check positioning of cars at each light. These smart traffic lights use artificial intelligence to help improve the flow of traffic and maximize the number of cars that are let through for each light cycle.

A solution to this problem is one that uses reinforcement learning with the agent. Reinforcement learning is a way to put our agent in an unfamiliar environment and have it learn the most efficient way to do a task through actions and rewards. This a very intuitive algorithm to use when trying to solve this problem but many people have tried a plethora of creative ideas.

[1] Although we didn’t try this in our research, some research is showing that deep reinforcement learning might be slightly better in results. A paper by Soumik Sarker showed that deep reinforcement learning obtained quiver design solutions that had the potential to have more complex solutions. This basically means that the light could adjust its times to more variety of cases. They used the function, *rt + γQ π ∗ (st+1,at+1),* to determine the reward given to the agent.

[2] A paper Anurag Kanungo implemented MATLAB’s video and image processing toolbox. The set cameras into an intersection and used MATLAB as their software to identify cars. They convert the images to black and white and determine traffic build up, or traffic density, by how far the line is on that light. They use an algorithm that takes the density they obtained from the camera and determine which light stays what color for how long.

[3] Wei-Hsun Lee takes an interesting route to solving a similar problem. Lee uses a five-state transition roadside unit for a solution to the traffic problem. His states focus on all scenarios like emergency car having to get through, a bus arriving or leaving for travelers, or even just normal everyday traffic for most people.

[4] Some research uses PLC (programmable logic controllers as their system to obtain data. Li Mei uses a PLC with a photoelectric laser sensor that collects traffic signal data. What makes Lin’s paper unique is her use of the input and output assignment table created for her PCL. This gives her a ton of leniency when it comes to the input and output allocation of the traffic light controls.

[5] Surtrac is a smart traffic light control system that already optimizes the flow of traffic through signals and sensors. Their view on the problem is similar to ours, where they want to reduce the time idling on roads with technology in the light’s algorithm. Their algorithm works by taking a decentralized method to traffic control where vehicles phases are independent for each intersection.

# Methodology

The approach to the problem began relatively simple. As outlined in the *Introduction,* the idea was to investigate algorithms that could be used optimize time frames. Again, we were looking for the lights to maintain themselves so that they are always looking to improve based in the situation in front of them. We realized that we were describing reinforcement learning, so that is the path that we took.

The first step for our system was to set up simulations. Regardless of how our algorithm was going to shift due to different amounts of traffic we first needed to simulate the traffic itself. For this, we decided to divide our data sets into three main categories. These included Rush Hour data sets, Partial Busy data sets, and Edge Case data sets. The Rush Hour data sets were meant for rigorous and even unrealistic testing. They flooded traffic ranging from 18-24 cars every 30 seconds to each of the four lights in the intersection. For these test cases, we were not focusing on how many less cars were waiting with our algorithm against the normal non-learning intersection. Instead, we were focused on making sure that the algorithm created a very fair wait time all around. The Partial Busy data sets were meant to simulate reality more accurately. The reason most stoplights struggle with busy traffic is because one pair of lights generally backs up while the other does not. This is the case we were primarily trying to solve, so these data sets were the most important as they represent this type of traffic. The Edge Case data sets for testing validity and integrity of the system. They included cases where there was only traffic going to one pair of lights and very minimal traffic like you may expect to see at an early hour of the morning (2-3 am).

Once we had our system set up for simulation and we had our data sets implemented and formatted properly for use with our agent, we were ready to begin the brains of the project. The algorithm that we chose to use was a modeled reinforcement learning algorithm. Essentially, we would calculate a score value that would represent how well the intersection was doing for that cycle. Each time the intersection would cycle through each light being green, the amount of traffic waiting at each light would be used to calculate the reward value. Markov’s decision process model was the general inspiration for the way we had built our model.

*(S, A, T, R,* ***λ****,)*

This model is the most successful one when it comes to modeling reinforcement learning problems. The traffic lights started at their initial state and adjusted with the reward they were given. For our model, since the algorithm in theory would never reach an optimal limit for every situation, we didn’t create an end state, but a vector of states that would continue to go to one another. The reward function in our model,

*R(s, a, s’),*

would consider a set of two parameters,

*< g, t >.*

Where g is the green time for that specific light and t is the traffic build up for that same light. These parameters were important to the algorithm because this helps stabilize the change that each light would have between cycles.

# Results and Discussions

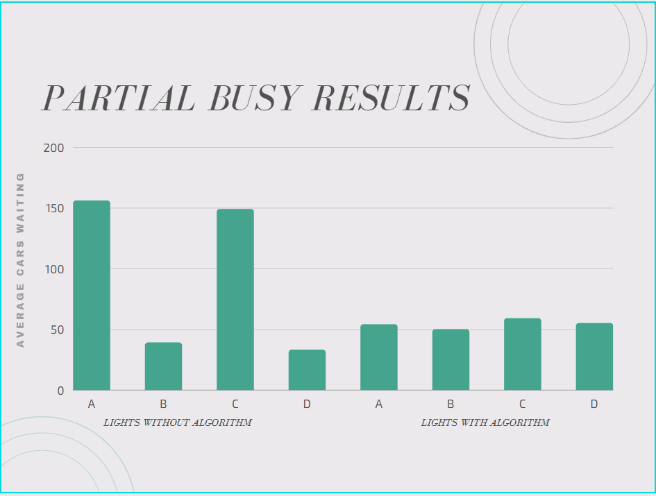
## Rush Hour Data Set Results



*Figure 1 Bar Graph of traffic for Rush Hour tests*

*Figure 1* represents the average number of cars waiting at each light (y axis) against which light they are waiting at (x axis) for all of our 15 rush hour data sets. The first set of data which is on the left-hand side shows the cars waiting at the intersection without learning after 10 minutes. The data on the left side of the graph shows the number of cars waiting after 10 minutes with our algorithm at play. Although the results do not seem to be substantially better, this is because the light was not able to make very many changes. This was since the intersection was being overloaded from every direction, so a normal intersection can handle it almost as well as our algorithm. The results here fit well with our hypothesis. We figured that when a lot of traffic is thrown at our intersection from every direction, there would be slight improvements. The only time the algorithm can really improve the state of the intersection is when there is an unfair wait time, and if traffic hits every side at the same rate, the traffic light can not do much. The overall number of cars waiting after 10 minutes of simulation was 504 in the non-learning intersection while our intersection only had 373 cars waiting. This represents a 26% improvement.

## Partial Busy Data Set Results



*Figure 2 Bar Graph of traffic for Partial Busy tests*

*Figure 2* represents the average number of cars waiting at each light (y axis) against which light they are waiting at (x axis) for all of our 15 partial busy data sets. The Partial Busy data sets are really where our intersection shines. These data sets are made up of pushing a lot of traffic to one pair of lights, simulating a busy road crossed with a non-busy road. This is where the unfair waiting times tend to happen at intersections. These data sets accurately depict the problem that we were trying to solve, and we are happy to show that the problem has even greatly mitigated in these cases. As you can see in *Figure 2*, there is far less traffic waiting overall. Although lights B and D have slightly more traffic waiting at them, the overall fairness is greatly improved. Not only fairness, but the total number of cars waiting at the intersection after 10 minutes of simulation decreased from 377 to 218 which is a 42% decrease. These results supported our hypothesis for sure as these pieces of data were the most representative.

## Edge Case Results



*Figure 3 Bar Graph of traffic for Edge Case tests*

*Figure 3* represents the average number of cars waiting at each light (y axis) against which light they are waiting at (x axis) for all of our 15 edge case data sets. These data sets simulate realistic circumstances, but do not test our algorithm very harshly. This is because a normal traffic light does not build up traffic either with the data sets provided. Regardless, there were instances were there would be a few cars building up at the normal traffic light due another light in a different intersection turning green. Regardless, our algorithm still managed to outperform a traditional stoplight, but these tests do not tell us much. We used these tests as validity to confirm that our light would work in all circumstances. There were 12 cars waiting at the intersection on average after 10 minutes of simulation for a normal light, and there were 0 cars waiting for our learning intersection. These results did support our hypothesis.

# Conclusion

The problems of traffic created by urban sprawl is an ongoing and worsening issue. It has gotten bad enough where the automobile is not a realistic option in busy cities. With the traffic mitigation solution that we have proposed, intersections can become more optimal and present a better source of traffic flow. When these intersections work in conjunction with one another, it creates a ladder effect where one intersection will help the other and the net gain is great. Our work is not complete, however. There is much more that can be done with this project. The real-world applications of this system beg a few questions. How will the intersection work with only 3 lights? What about the turning lanes that make intersections much more complex? These are questions that can be answered with future work. Adding more lanes and/or testing this intersection idea with other machine learning techniques like neural networks could boast even better results.

##### References

1. Kai Liang Tan, Soumik Sarkar, Subhadipto Poddar, and Anuj Sharma, “Deep reinforcement learning for adaptive tarffic signal control,” ASME, 2019, pp. 2-3.
2. Anurag Kanungo, Ayush Sharma, and Chetan Singla, “Smart traffic lights switching and traffic density calculation using video processing,” RAECS UIET, 2014, pp. 2-5.
3. Wei-Hsun Lee and Chi-Yi Chiu, “Design and implementation of a smart traffic signal control system for smart city applications,” Sensors, MDPI, 2020.
4. Lin Mei, Zhang, Lijian, and Wang Lingling, “Intelligent traffic light based on plc control,” IOP Conf. Series: Earth and Environmental Sciences,” 2017, pp. 6-8.

[5] Stephen Smith, Greg Barlow, Xiao-Feng Xie, Olisa Okonkwo, and Lipeng Gong, “Non-market strategy analysis project report,” SURTRAC, 2014,.