Investigating Algorithms to Manage Traffic Signals.

James Frucht and Isaiah Turner
TJHSST
May 2019

Overview

Traffic is becoming an increasingly major problem that results in increased stress on commuters and increased emissions of pollutants. One form of mitigating these effects is using modern developments in computing and artificial intelligence to make traffic signals more efficient to limit the amount of time driver spend waiting at traffic signals. We developed a series of algorithms, which we tested using the SUMO open-source traffic simulation software. We noticed significant improvements compared to current methods of managing traffic signals. We were unable to finish all of the algorithms that we initially intended to create due to time constraints.

Motivation

In the United States, traffic leads to many ecological, sociological and economical problems. According to the Texas A&M Transportation Institute, US commuters spend an average of 42 hours per year delayed by traffic traveling for 22% more time than they would in conditions with no traffic. This wasted time driving leads to 19 gallons of wasted fuel per commuter per year and a cost of $960 per commuter per year. This amounts nationally to 6.9 billion hours spent delayed in traffic per year, 3.1 billion gallons of wasted fuel per year, and a cost of $160 billion per year.

Background

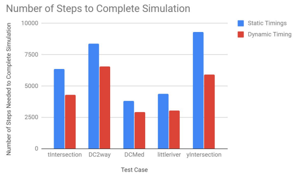
A potential solution form of mitigating this problem is making traffic signals more efficient. By reducing the amount of time drivers spend waiting at traffic signals, commute times can be reduced significantly. We can model making traffic signals more efficient by minimizing the average commute time per driver (considering commutes remain constant).

We used the Simulation of Urban Mobility (SUMO) software. SUMO is an open-source software developed by the German Aerospace Institute that models vehicle and pedestrian traffic. It had the tools necessary to conduct the simulations for our traffic signal: create both artificial test-case road networks and import real-world road map data, manually control traffic signals through their API, and collect data on traffic efficiency..

Methodology

We created various algorithms to test improvements in traffic.

Static Timings: The algorithm calculates the time it would take a single car to travel each controlled stretch of road using the length of the road and the speed limit. These values are then scaled up to a 90-second traffic cycle length, which is common for many traffic lights. As a result, longer stretches of road are allocated more time to account for traffic buildup.



Dynamic Timings: The algorithm was designed to mimic current sensing technologies implemented in traffic signals. Each lane has simulated induction loops placed beneath them to detects the number of cars in the lane at any given time. If a traffic signal detects there are more vehicles in the red light lanes than the green lanes, it switches to the next phase. We also used maximum and minimum times for each phase to limit extreme cases where a phase may never switch.

Dynamic Timings with Communication: Lights switch phases by comparing the number of cars in the red lanes to the cars in the green lanes. However, in networks with multiple lights, the traffic information from each light is sent to neighboring lights. The lights receive the information on potential incoming traffic and weight them when determining whether to switch phases or not. The weighting for the cars at the light was 10, and the weighting for the incoming cars was 1.

We had adjustable parameters for algorithm 3, the maximum and minimum phase length time. In order to find the optimal values for these parameters, we developed a simple hill climber program. The program ran algorithm 3, then adjusted the parameters and repeated the run. If changing the parameters resulted in an improvement, that run was used as a baseline for further iteration.. As a result, the hill climber found optimal parameters for each of our networks.

Results

After completing our testing we compiled our results. The results for our comparison between the static algorithm and the dynamic algorithm with communication are shown to the left. The simple Y-Intersection and T-Intersection had the greatest speedups, of 36.33% and 32.37% respectively. Their simpler nature meant traffic was less bogged down by lane changes and turns into multiple available lanes. The DC 2-way intersection and the Little River Turnpike networks had smaller but still significant speedups. The large DC network had the smallest speedup out of all for the reasons stated above.

Conclusions

Our investigation into algorithms for managing traffic signals showed that efficient traffic signals that use communication between signals can be an effective way to reduce traffic congestion, decreasing the amount of time drivers spend traveling. This has the potential to greatly reduce costs on drivers by limiting gasoline consumption and wasted time. Additionally, it can improve mental and physical health by reducing stress and sedentary behavior.

However, these various benefits would come with many costs that would have to be accounted for when deciding to implement these traffic light algorithms. First, these traffic signals would require significant infrastructure development for creating fast and stable network connections between traffic signals and installing vehicle sensors at all traffic lights. Additionally, these algorithms are far more compute-intensive than current methods of traffic signal management. As a result, traffic signal computers would need to be upgraded and, for centralized algorithms, resources would need to be allocated to running a secure server. The algorithm also requires induction loops or some other method of detecting the number and location of the vehicles at a traffic light. To properly assess the effectiveness of implement more efficient traffic signals, more research must be done to determine the costs required to implement them, and whether they are less than the potential benefits.