Learning Traffic Light Simulation

Lynn Jepsen
Computer Systems Lab, 2006-2007
January 25, 2007

Abstract

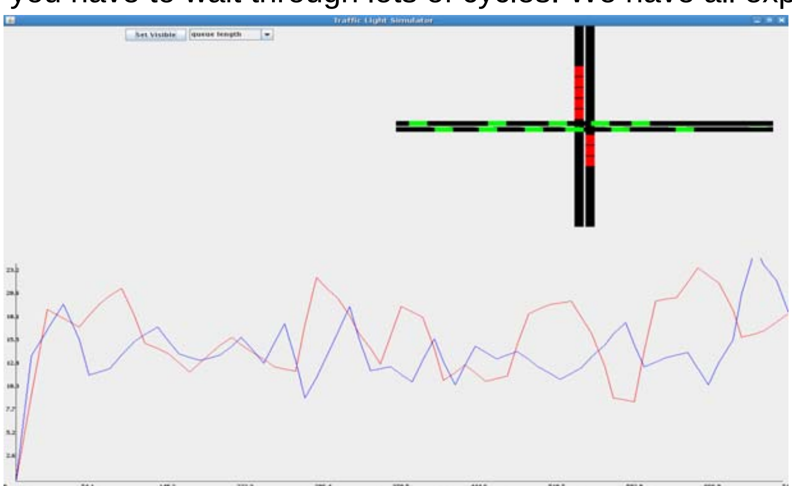
This project is meant to simulate a busy traffic light. The program recognize patterns in the intersection and then uses that information to make the light as efficient as possible. These patterns could be related to the time of day and/or the day of the week. At first I would purposely input recognizable patterns and see if the program would catch it, but eventually the plan would be to possibly use this program at a real intersection. There are many variables' that the program takes into account including traffic density, number of lanes, etc.
Keywords: traffic simulation, efficiency, cars, red light: stop, green light: go, pedal to the metal, vroom vroom, honk

1 Introduction

I want to develop an algorithm that would decrease queue length and wait time while also increasing green light usage by changing the cycle length and the ratio of green light time in each direction. In order to see if this algorithm is working, and gather lots of data very quickly, I built a simulation of a traffic intersection to test the algorithm on. The simulation follows basic rules of the road. Each car takes up so much space on the lane, travels so fast, and has a max acceleration. It is a realistic simulation because it obeys physical laws as well as human laws. People only travel so fast and do not peel out of intersections. The data gathered from the intersection are five separate numbers. First we get the traffic density, the number of cars per minute that pass through one direction. Then we figure the queue length, maximum wait time, and green light usage of the past cycle in the intersection. These are the efficiency variables. We want to optimize them in order to create an efficient intersection. To do this the light algorithm chooses an appropriate cycle length and ratio for green light time in each direction. These factors are chosen by looking at what happened to the efficiency variables when previous cycle length and ratios where used on given traffic densities. Once the cycle is complete we store all five data points so that the light algorithm can use them again.

2 Backgrounds

Traffic lights have always caused problems with traffic flow. If the light has no information about the intersection it is controlling, then you can sit at a red light for what seems hours. Even worse is when the light is much to short and you have to wait through lots of cycles. We have all experienced this. There has



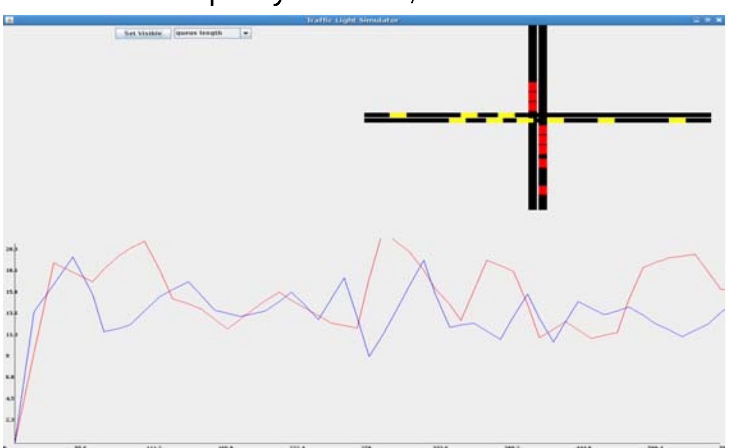
been a lot of research done to try and automate cars, using GPS, so that they all pass through the intersection harmlessly. However, this technology can be expensive, and it would take a long time to install it into all cars in order for the project to work. I think a much cheaper and temporary solution would be to fix the traffic lights we already have with brand new algorithms that would make the light efficient. In order to see if this algorithm is working, and gather lots of data very quickly, we need a simulation of a traffic intersection to test the algorithm on. This way we can see the results pictorially, the aerial view of the intersection, but also graph the efficiency variables so we can see general trends.

3 Developments

The graphical component of this project is how one decides if the light algorithm is successful or not. When queue length and wait time go down, the light is efficient. There are also two separate lines for each intersection. Those lines need to be close to each other; otherwise the intersection favors one direction too much. When various aspects of the light algorithm are changed, you simply have to look at the graphs to decide whether it worked or not. There is also a pictorial component that shows the intersection from an average person's point of view.

…I will put in other light algorithms when I think of different versions…this is a draft, and I have yet to create more light algorithms.

This screen shows the pictorial view of the intersection (yellow cars mean they are looking at a yellow light, etc) as well as the graphical view. This graph shows the queue length over time. It only graphs at the end of each cycle so it doesn't continuously fluctuate up and down. The red line represents the north/south direction while blue is east/west. The graph shows that the intersection is pretty efficient, but still has horrible moments. The graph updates with time.



This is not an exact science because there are too many variables in effect here. You can't possibly minimize wait time to the point where no one waits. Traffic flow is just too erratic to predict well. So by looking at general trends, such as this graph, you can decide if one algorithm is better than the other.

3 Results

I feel like this should be done after I develop more light algorithms and test them against each other.

References

[1] P. Spencer, “Details on the Mathematical Model”, University of Toronto Mathematics, http://www.math.toronto.edu/mathnet/carcompet/model.html, 1997.
I used this to model how each car moves.

[2] B. Sadoun, “An Efficient Simulation Methodology for the Design of Traffic Lights at Intersections in Urban Areas”, SIMULATION Vol. 79, No. 4, pp. 243-251, 2003.
I used this article to give me three ideas for varible to look at in the intersection: waiting time, queue length, and utilization of green light.

[3] K.Patch, “Adaptive Lights Organize Traffic” , Technology Research News, 2005.
Talks about some different ways of maximizing efficiency in isolated traffic lights.The red light keeps track of how many cars are waiting and then