

Traffic Flow Maximization using Evolutionary Algorithm

November 26, 2013

Abstract

Traffic Flow maximization is one of the crucial problems in designing a city. It directly affects the daily life of the people living in that city. It is a complex problem, one that in most cases cannot be deterministically solved. We propose using evolutionary algorithms to solve that problem. We compare existing work and traffic flow with solutions yielded by our evolutionary approach, and our results show that it is beneficial to adopt this strategy when designing traffic light timings.

1 Scope of the Problem

Traffic infrastructure comes in various forms, and optimizing traffic flow in road networks is a task that depends highly on the infrastructure. We commonly consider infrastructure as being in one of two categories: “smart” infrastructure and “legacy” infrastructure. The first makes use of detectors placed in the infrastructure to determine the state of traffic, whereas the second does not.

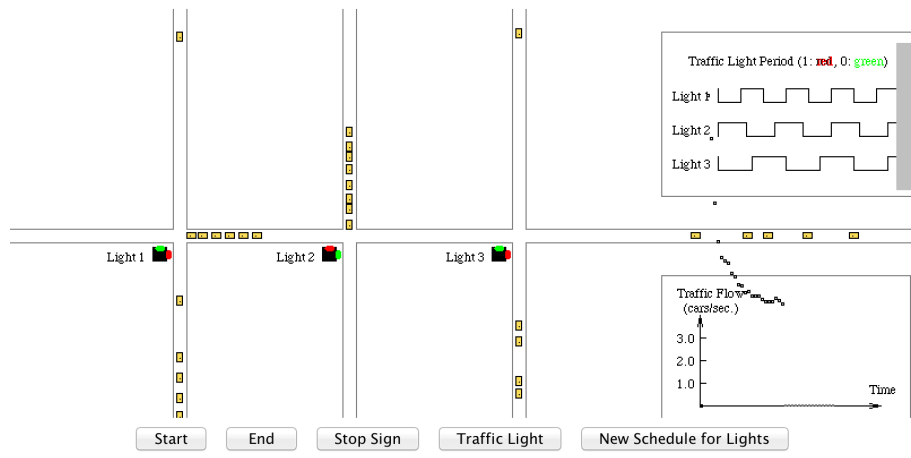
It is reasonable to assume that achieved solutions will perform better as a whole when making use of smart infrastructure. For example, a traffic light that can detect that there is no traffic from East to West, and that vehicles are waiting to go from North to South, can react accordingly and change its state to shorten the wait of these vehicles.

There have been projects in the past where traffic flow was optimized by combining real-time knowledge of traffic and communication between lights. The best-known of these projects is one spearheaded by Carnegie Mellon University in the East Liberty part of Pittsburgh, with excellent results.

However, this previous study’s approach relies heavily on smart traffic lights and detectors, which, although quite practical, are still far and few between throughout the world. In countries such as China or India, where the number of vehicles is growing most rapidly, most roads are equipped with legacy traffic equipment.

An effective approach to solving this problem should be applicable to the maximum amount of scenarios, which is why this paper discusses only optimizing traffic light timings in a legacy environment. However, operating in a legacy

Figure 1: Screenshot of the Simple Java Simulator



environment does not mean that we must forgo all knowledge of the traffic flow. It is reasonable to assume that traffic flow can be measured at specific intersections. The collection of this data can unearth trends in the traffic flow (ie: rush hour traffic). Once this data is collected, any period of time can be split into different sections, where each of these sections has a different, but constant, traffic flow.

In our analysis, we place ourselves in an environment where the traffic flow is a fixed parameter. The above analysis shows that this scenario is relevant.

2 Existing Domain Research

Research on traffic lights optimization began since the emergence of traffic lights. Researchers have come up with several different ways to optimize the traffic lights such as

3 Simulators

When we are developing the algorithms to optimize the traffic lights, one common thing that we will do is to find a simulator. There are tons of simulators on the Internet, we have to choose the one that fits our project the most. The following are the simulators we investigated.

3.1 Simple Java Simulator

This is a simple java simulator. It uses the multi-thread feature of java to speed up and it can automatically generate traffic flow data which are plotted at the right-down corner. Another advantage it has is that you can drag any car from

Figure 2: Screenshot of the Matsim simulator



one position to another. It will automatically adjust the position of the car and continues. Therefore, it enables us to adjust the traffic at any time we want. We can also adjust in real-time the timings of traffic lights. For simple road networks and light traffic flow, it is a great choice. However, in this kind of traffic network, there isn't much to optimize, which is why we didn't choose it as our final simulator.

3.2 MatSim

Matsim is a powerful simulator which you can see from its simulated traffic networks. It is open source, which means that developers can modify it as they wish and adapt it closely to their needs. Also it provides an interactive visualizer which enables users to modify the traffic networks conveniently. At the same time, it provides detailed analysis which can be used by our project.

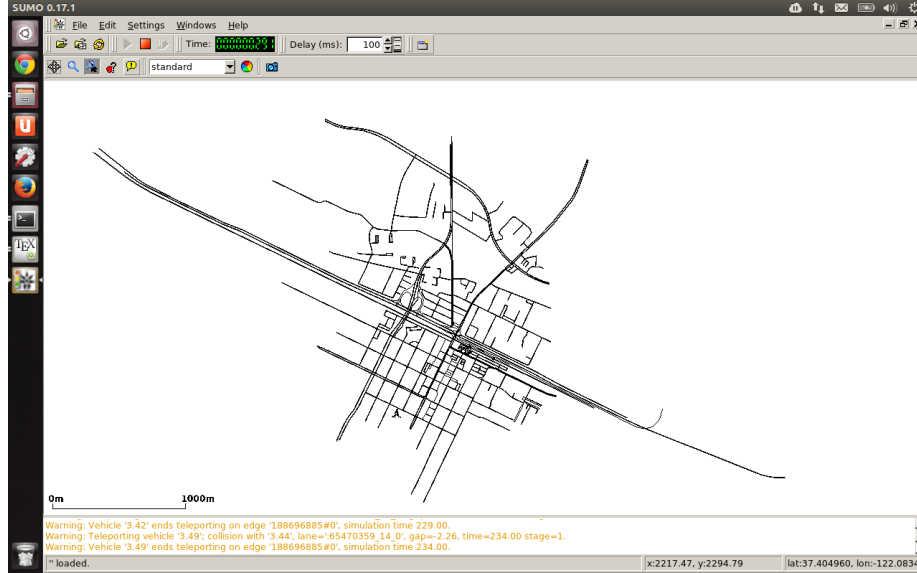
3.3 SUMO

After comparing carefully among several different simulators, we finally chose SUMO which was developed by employees of the Institute of Transportation Systems at the German Aerospace Center.

4 Simulated Traffic networks

In order to test our algorithm, we modeled the traffic network around the Caltrain station located in Mountain View, California. This choice was informed by the fact that it is a well-known intersection among our peers and we have heard

Figure 3: Caltrain Station Simulation



a lot about its long wait times. We investigated the traffic situation around the Caltrain station: during rush hour, we recorded the traffic flow as well as the traffic light timings.

5 The choice of fitness function

In order to best simulate the real world traffic light, we tried several ways to establish the mapping between real world traffic measurement with the fitness function in our evolutionary algorithm. The following are the fitness function we tried in our algorithm.

5.1 Average Speed

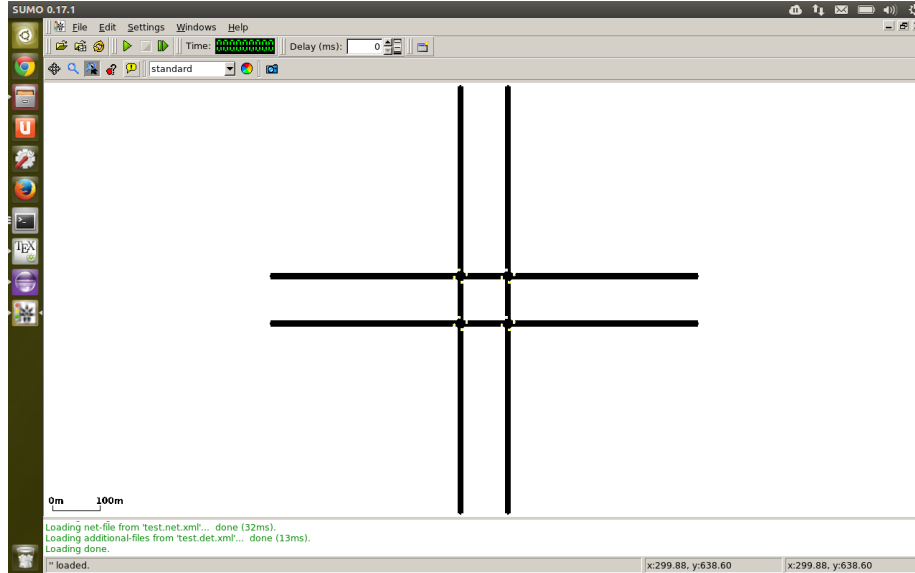
This is the first fitness function we tried. And it is the most intuitive one. An intuitive explanation is that if the average speed of a traffic network is larger than another, then it is more likely to have a greater traffic flow. We tried this fitness in the grid traffic network as shown in figure 4.

5.2 Footnotes

Indicate footnotes with a number¹ in the text. Place the footnotes at the bottom of the page on which they appear. Precede the footnote with a horizontal rule

¹Sample of the first footnote

Figure 4: Basic Simulation



of 2 inches (12 picas).²

Acknowledgments

Use unnumbered third level headings for the acknowledgments. All acknowledgments go at the end of the paper. Do not include acknowledgments in the anonymized submission, only in the final paper.

References

References follow the acknowledgments. Use unnumbered third level heading for the references. Any choice of citation style is acceptable as long as you are consistent. It is permissible to reduce the font size to ‘small’ (9-point) when listing the references. **Remember that this year you can use a ninth page as long as it contains *only* cited references.**

[1] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In G. Tesauro, D. S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp. 609-616. Cambridge, MA: MIT Press.

[2] Bower, J.M. & Beeman, D. (1995) *The Book of GENESIS: Exploring Realistic Neural Models with the GENeral NEural SIMulation System*. New York: TELOS/Springer-Verlag.

²Sample of the second footnote

[3] Hasselmo, M.E., Schnell, E. & Barkai, E. (1995) Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region CA3. *Journal of Neuroscience* **15**(7):5249-5262.