Optimizing average travel time using traffic simulations based on sensor data, with traffic light synchronization

Casey Bowman

University of North Georgia

Introduction

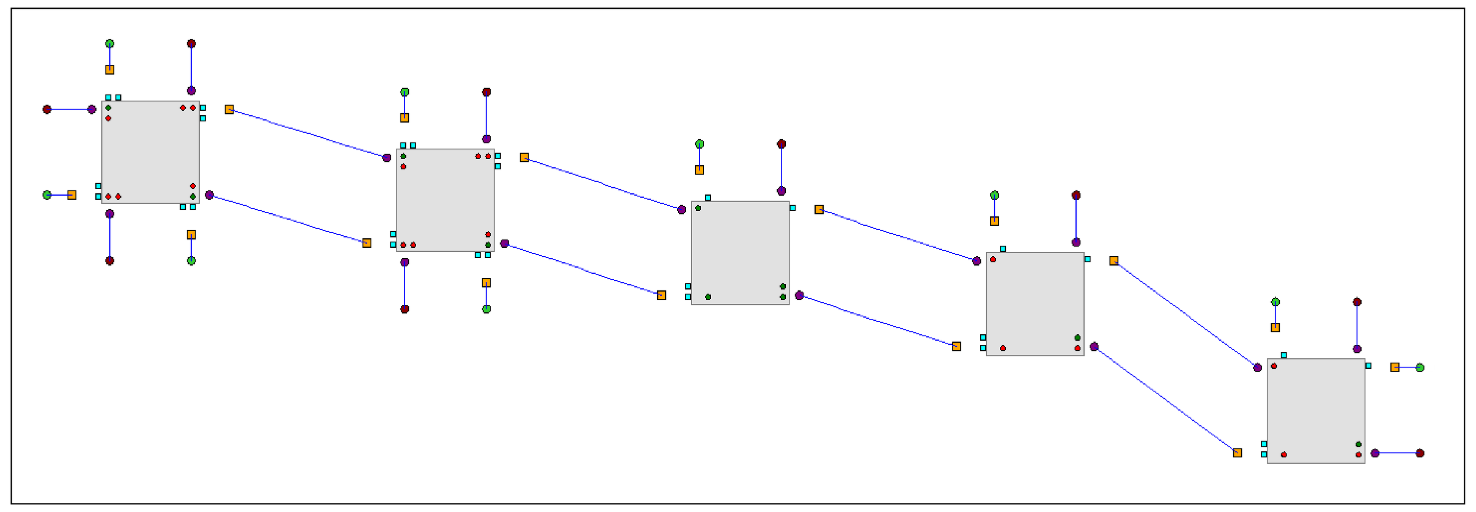
Big data analytics and scalable simulation modeling can be used to improve complex systems and processes. For example, due to their high cost, careful planning of improvements to transportation systems is essential. For this problem, large amounts of sensor data from roads can be used for calibrating traffic models via optimization of traffic light synchronization.

Photo taken by: Fir0002/Flagstaffotos

Simulation Software - ScalaTion

The simulations were created using the software package ScalaTion (Miller, Han, and Hybinette, 2010), which has been programmed using the Scala programming language. The models consist of Sources, Sinks, Transports, Selectors, WaitQueues, Gates, Junctions, and Cars.

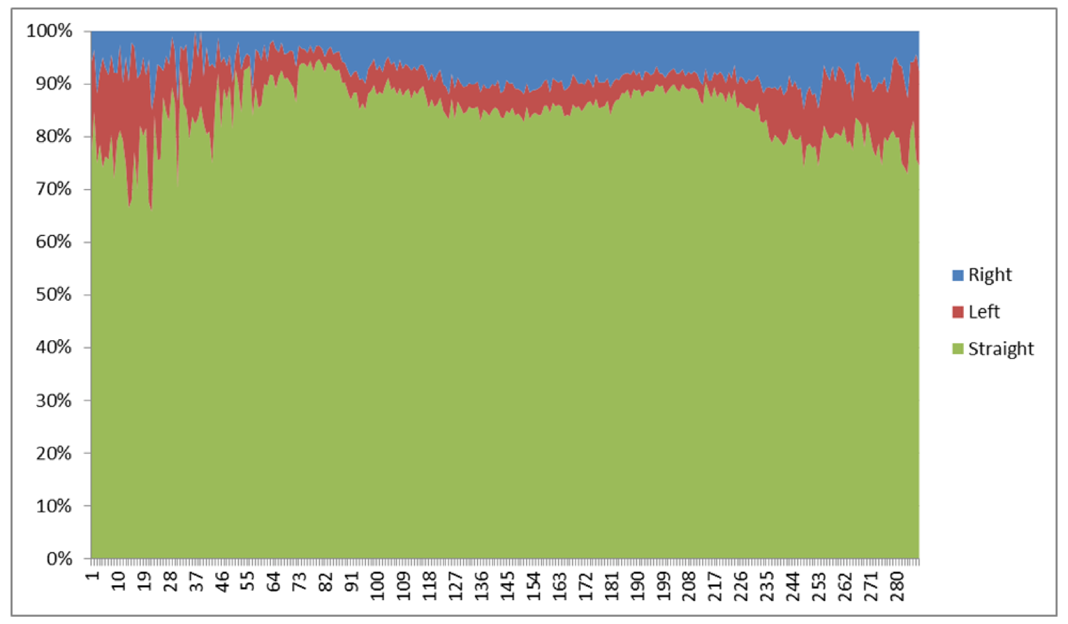
Each source leads to a Selector, which uses the data to determine which direction the vehicle should go. The light blue squares just below the Selectors are WaitQueues, which hold vehicles until the Gates (traffic lights) release them. Vehicles leave the intersections at Junctions and travel to the next intersection, or to a Sink.



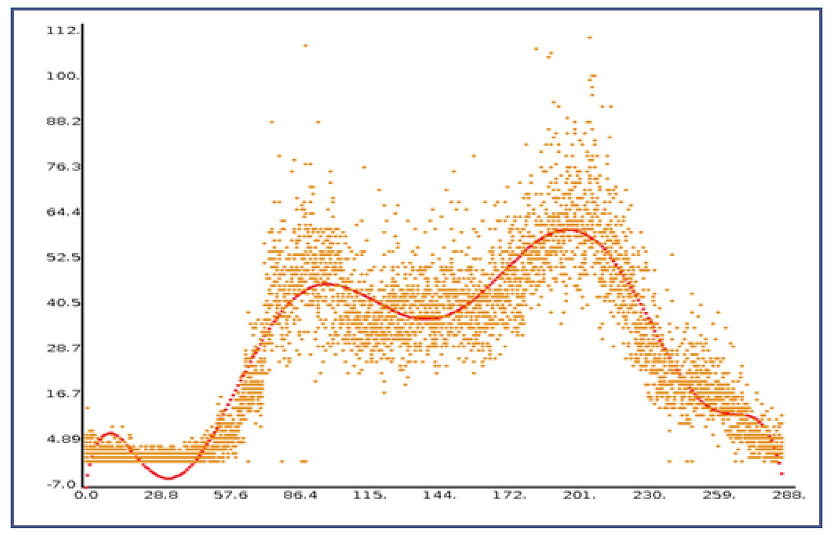
Modeling Vehicle Behavior

As vehicles are generated with the NHPPs, they must move around the simulation model in a realistic way. The choice of direction usually coincides with the lane the vehicle is in. Using the lane-by-lane vehicle counts, the percentage of drivers making each choice can be estimated and used to drive a discrete random variable which chooses directions.

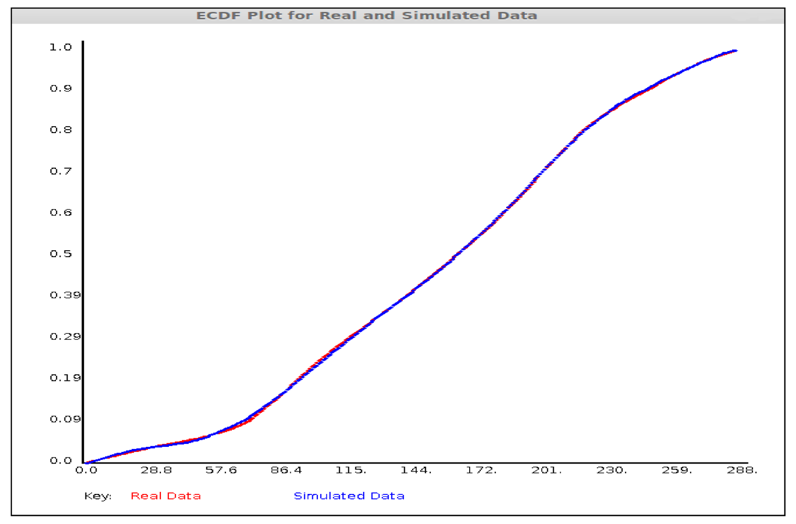
The percentages change throughout the day however. To accommodate this, the percentage data was fit with polynomials, which are used to power the discrete random variable. A U(0,1) uniform random variable is produced, which will fall into one of three percentage groups such as in the figure below.



Analyzing and Modeling Data



Data was collected from the Washington State DOT for a collection of intersections in Kenmore, WA. The data are vehicle counts collected at every 5 minute interval for a 17 week period. We fit this data with polynomials and created non-homogeneous Poisson Processes (NHPPs) (Leemis, 2003) which were validated using a Kolmogorov–Smirnov (KS) test.



Optimization

We minimize the average travel time for all vehicles in the system. The simulation is a function, with this value as output, and traffic light timings as input vector **x**. The input vector is defined as:

x_0 — amount of time lights are green on main road.

x_1 — amount of time lights are green on cross streets.

x_2 — ratio of green time for turn lights.

x_3, x_4, x_5, x_6 — offset values so that intersections enter light cycles at different times.

We assume the total cycle length, and the cycle pattern, is the same for all five intersections.

Optimization is achieved using the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm, which is gradient-based utilizing an approximation to the Hessian matrix. An initial vector of light timings is provided and a gradient is computed. The gradient computes a search direction. A line search routine is then called, which determines the distance to travel along the search direction. The Hessian is then updated.

Results and Future Work

Results are preliminary at this time. Running many simulations consecutively is inefficient, so we have restricted our simulations to 20 minutes of a typical day. The optimization does work for this restricted time period, so we are hopeful making use of Big Data techniques to deal with the large number of simulations will produce good results for longer simulations.

There are several points of emphasis for our future work:

1. Alternative efficiency metrics, such as average throughput.
2. Removing simulation runs using either uniform step-sizes, or a pre-defined sequence of step-sizes. This would eliminate the line search, and the necessary simulation runs that come with it.
3. Implementing the ability to begin the simulation at any time of day, for instance to optimize afternoon rush hour. However, at a busy time of day, there should be many vehicles already in the system. Currently our software only generates vehicles at Sources.
4. Exploiting parallelism in appropriate places, such as running independent simulations simultaneously.

References

Leemis, L. 2003. “Estimating and Simulating Nonhomogeneous Poisson Processes”. In *Proceedings of International Conference in Reliability and Survival Analysis (ICRSA) 2003.*

Miller, J. A., J. Han, and M. Hybinette. 2010. “Using Domain Specific Language for Modeling and Simulation: ScalaTion as a Case Study”. In *Proceedings of the 2010 Winter Simulation Conference*, edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, , and E. Yucesan, 741–752. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Acknowledgements

I would like to thank my advisor, John Miller, for his advice while working on this project. I would also like to thank Mustafa Nural for finding, downloading, and formatting the data used in this project. Their help has been invaluable.