Optimum Number and Position of Signs before a Road Diet

Eric Shao, Ziyang Gao
Carleton College

Abstract

We want to find the optimum number and positions of signs for best traffic efficiency before a road diet. We use Mathematica to build up a simulation program to imitate the situation of a road diet.

Nowadays in big cities, most people have experienced traffic jams and all the traffic jams are caused by road diet, either for construction reason or from an accident. Put signs to warn drivers about the situation should help the driver make better decision and increase the overall road efficiency.

We focus on a two lanes to one lane road diet since two lanes road is the most common road. We find it's best to put 8 signs evenly along the last kilometer. If we can only put one sign, the it's best to put at middle.

Algorithm Logistics

- Set up: we initiated two lanes of car, acceleration lane (lane 1) and merging lane (lane 2). Then we randomly put cars onto the two lanes, with three initial values assigned to each car: speed, position and whether or not want to merge.
- Update road status: we change each car's speed and position based on three constraints: speed limit (SL), position of previous car and length of the merging lane.
- Merge: when a car in lane 2 passing through a merging sign, there is a percentage of chance that he will want to merge into lane1. And he will merge within a certain distance before the end of lane 2.
- Efficiency: we measure the efficiency of the distribution of merging sign by the time it takes for all cars in lane 2 to merge into lane 1.

Picture 1. A Traffic Jam

Picture 2. A Road Closed Sign

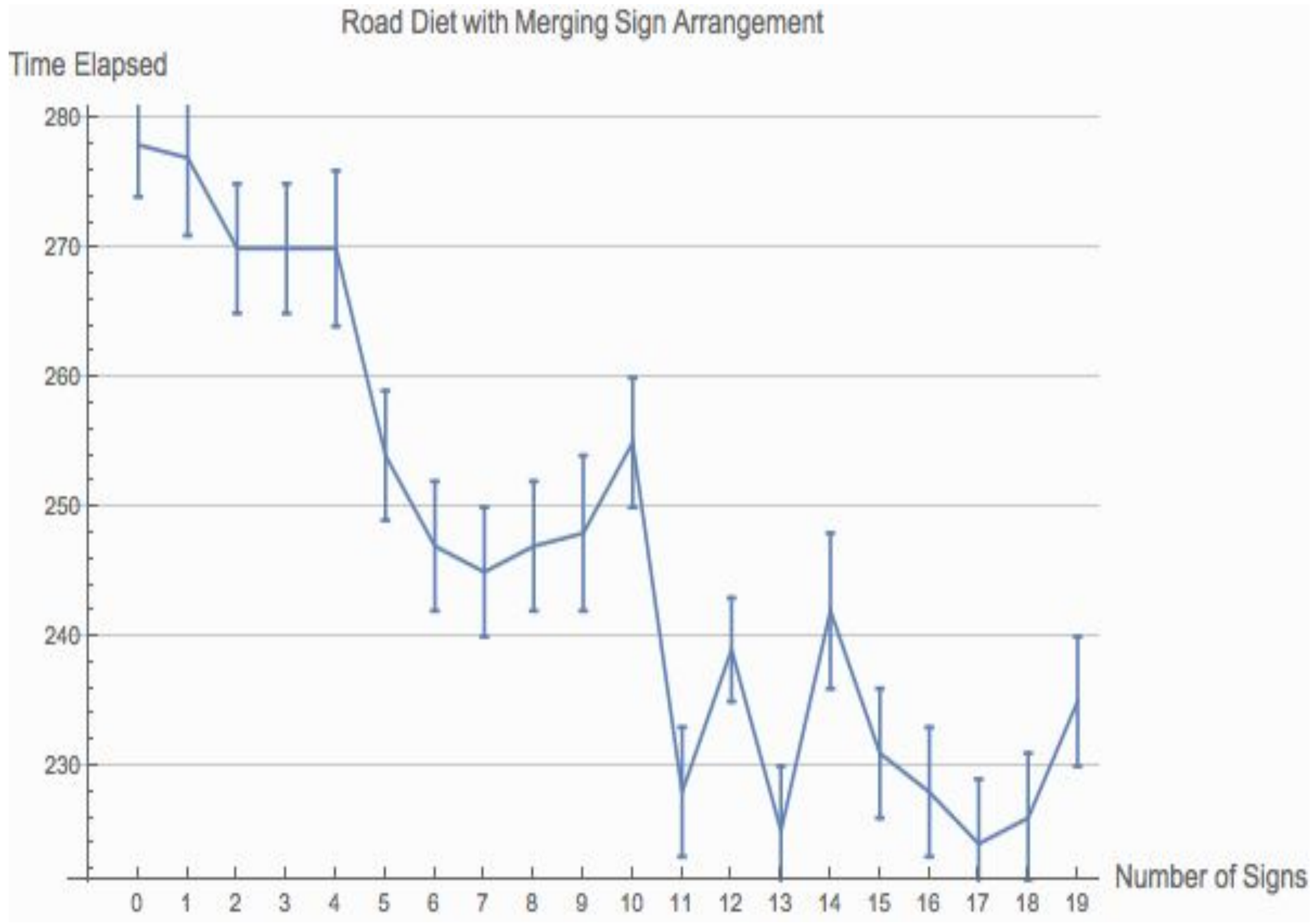

Figure 1: road dieting time with different sign arrangements

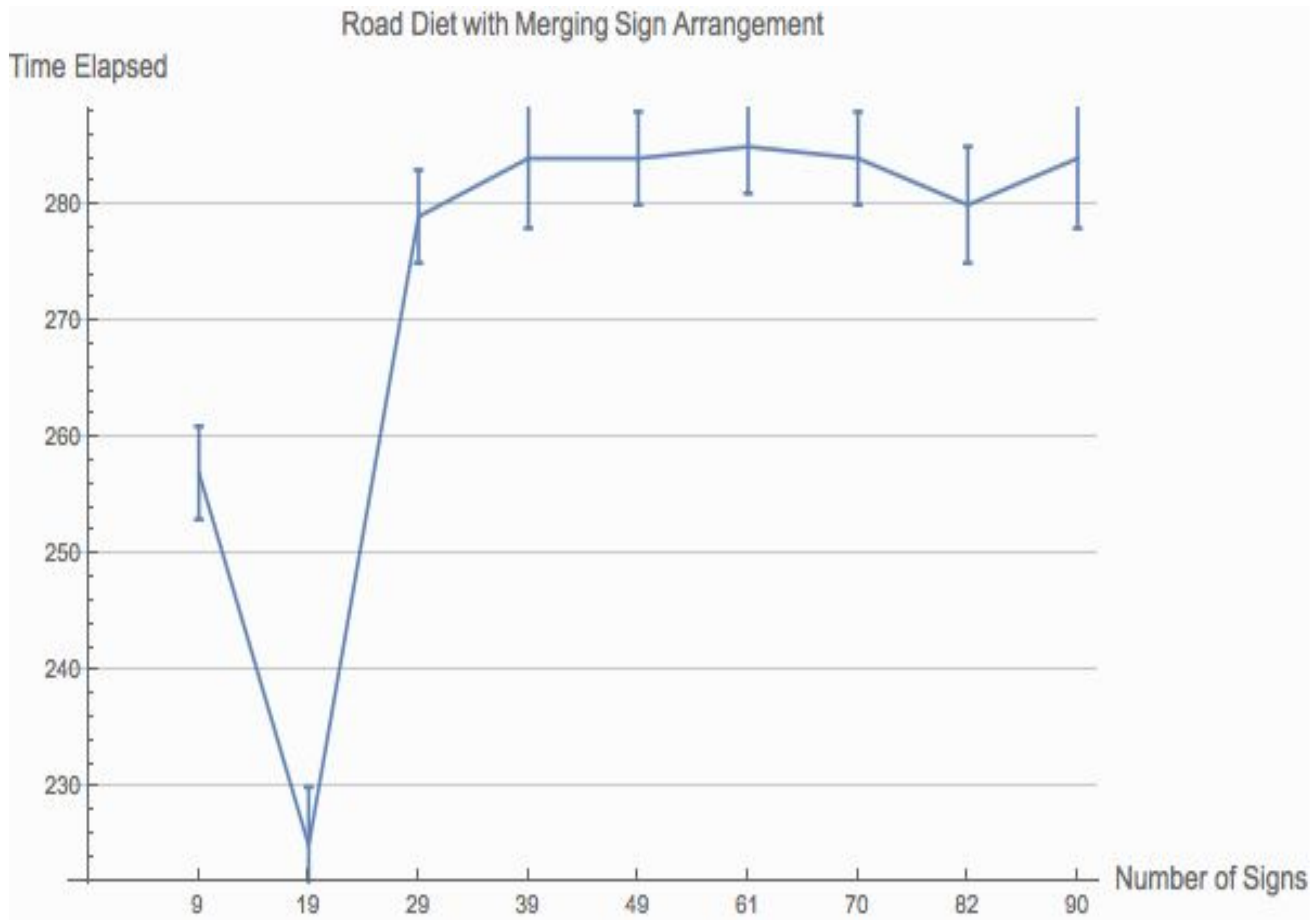

Figure 2: road dieting time with different positions of a single sign

Graph Analysis

We run our algorithm on a lane 2 of 1000m and a 10m/s speed limit.

Figure 1 is the relationship between the total number of merging signs and average time (mean over 20 data samples) needs for all cars in lane 2 to merge into lane 1. We assume that the signs are distributed uniformly over lane 2. We consider number of signs to be integers less than 19, because Mathematica cannot efficiently run on higher boundaries. The downward slope of the graph indicates that the time elapsed for merging generally reduces as number of signs increases to 19.

Figure 2 also shows the correlation between the number of signs and the average time (mean over 20 data samples) for merging. The difference is that the number of signs are from 9 to 90, increasing with a step of 10. The graph shows that the time reaches minimum around 19 signs and goes up before 29. Thus there is no need to put more than 20 signs on the 1000m road.

Conclusions

After testing the model we build and analyzing the data we got from the simulation, we found some ways to potentially reduce the time needs for lane 2 to merge into lane 1.

We reached the least time necessary for merging with 17 signs uniformly distributed along lane 2, with the assumption that our merging lane 2 is 1000 meters long and the speed limit is 10m/s.

Another thing we noticed is that there is not a significant difference of time reduced after the 19th sign, and after sign 29 the merging time needed rises back to the level of 0-2 signs. Therefore, too many signs will hinder the cars on lane 2 from merging into lane 1.

Reference

Beacher, A., Fontaine, M., & Garber, N. (2005). Work Zone Traffic Control: Field Evaluation of Late Merge Traffic Control in Work Zones. *Transportation Research Record: Journal of the Transportation Research Board,1911*, 32-41. doi:10.3141/1911-04

Gaylord, R. J., & Wellin, P. R. (1995). *Computer simulation with Mathematica: Explorations in complex physical and biological systems*. New York, NY: Springer.

Picture 3. A Visualization of a Moment of the Simulation. (Each line is a car)