Matt Miller & Aishanee Wijeratna
MSDS 460: Decision Analytics
4 December 2024

MSDS 460 Final Project: Traffic Flow Simulation

1. Abstract

The problem our group has chosen to tackle is to simulate the traffic flow of an intersection with traffic lights. This will attempt to understand and predict congestion and wait times. With a better understanding of these factors, we could potentially offer the effects of different actions such as adding an additional lane and updating the lights' functionality. Our approach leverages Python to build a discrete foundational simulation model, focusing on the fundamental constraints and variables involved.

Traffic flow simulation has been extensively studied in academic and industrial contexts. There have been many papers and journal entries published on how to effectively solve this problem. An often quoted work is *Traffic Flow Theory: A Monograph*, published in 1975 through the Transportation Research Board. This is considered a seminal work in the field of traffic flow theory although it is based on now arcane computer systems. D.L. Gerlough is credited with other important works in the late 1950s and 1960s on this matter as computers were first being used to simulate real-world conditions like traffic flow (Gerlough & Huber 1975).

More recent literature on the topic of traffic flow is plentiful and can be found through simple searches on the topic through popular scholarly search engines such as arxiv and Google Scholar. The most recent and popular approaches tend to make use of the power of modern technologies like machine learning and neural networks to build upon the work of the pioneers in the 1960s and 1970s.

Additionally, there exists functionality in popular mobile mapping applications, such as Waze, Google Maps and Apple Maps, that can provide predictions on the amount of time it would take to travel a route, based on historical traffic data. Based on this functionality, the assumption is that these applications have a way to understand traffic patterns at various points along routes.

Clearly there has been quite a bit of scholarly and industry work on this problem with clear applications in our interconnected, app-based world. In our project, we did not attempt to solve any new aspects of the problem (as full PhD programs could be committed to this), rather we provide an introductory exploration of traffic modeling,

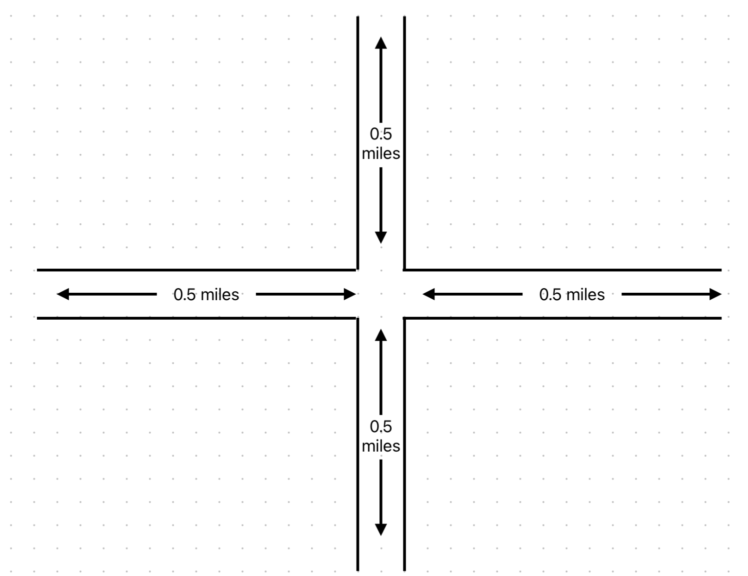
serving as a stepping stone for further study and optimization. The bare fundamentals of traffic flow theory already present many complexities that we felt gave us enough challenge when applying our acquired knowledge from this semester.

2. Data Modeling

We defined the various data points and constraints prior to making a decision on the algorithmic approach that needed to be implemented. The needed data is defined below but it must also be noted that the intersection will be handling traffic in four directions so the algorithm will account for this as well.

2.1 Traffic Volume

Volume is defined here as the average existing amount of cars prior to the traffic light but a previous intersection. In our example, previous intersections were 0.5 miles prior to each side of the intersection/traffic light. For simplicity's sake, we will refer to these 0.5 mile corridors on each side of the traffic light as the 'observation zones'. The length of these will be configurable within the script but we will use 0.5 for demonstrations. As a visual representation:



2.2 Number of Lanes

For the purposes of this project, we have decided that the number of lanes is the same in all four directions. This will be configured within the script but any demonstrations will use two lanes.

2.3 Green Light Frequency

This parameter defines the time and pattern for which a traffic light remains in the green state. For example, a fixed cycle of 30 seconds green, 5 seconds yellow, and 30 seconds red may be used. One way in which traffic flow simulation could be optimized is through dynamic green light frequency based on queue length or time schedule but our script will focus on a fixed pattern for simplicity.

2.4 Average Vehicle Speed

This is the average vehicle speed from one intersection to the rest. This will aid in predicting the average time a vehicle spends in the observation zone as well as potentially queue length at the intersection.

2.5 Size Of Vehicles (Length)

This is the size of the vehicles on the road. In our simulation we chose two sizes, one standard size and one that is 2x size to simulate a semi-truck or larger truck. The proportion of cars that are standard and 2x are set in the script with the 2x size vehicles being generated approximately 5% of the time by default.

2.6 Turn Probability

Although our simulation only handles right turns due to the complexity of left hand turns for a simple simulation, we have added a level of probability that a car will turn right at the intersection. It is set at 30% by default which is approximately based on traffic data we obtained from the intersection in Fort Wayne, Indiana that we are using.

2.7 Time and Date-Based Traffic Patterns

Different times of the day (such as a rush hour in the morning and early evening) as well as the day of the week affect traffic patterns in a regularly traveled route. In more advanced simulations, special days such as holidays would also be accounted for, but for simplicity we have chosen not to account for those.

There are many other variables in a real-world discrete traffic flow simulation. Some of the variables we do not account for but have come across when studying the topic are weather, more varied vehicle size, turning patterns and average following distance. While this is nowhere close to a comprehensive list, we did want to demonstrate our understanding of the complexities of this problem outside of the variables we have defined.

3. Algorithm and Approach

There are a variety of ways this problem can be solved. These range from simple, inexpensive computational methods to incredibly complex and resource intensive neural networks. In more detail:

- Discrete-Event Simulation (DES) could be used to model traffic flow by advancing time to key events, such as vehicle arrivals and light changes, providing an efficient way to capture sequential interactions at intersections.

- Optimization algorithms would focus on improving traffic light timings or lane configurations to minimize congestion and delays.

- Machine learning models, particularly reinforcement learning, could be leveraged to adaptively optimize traffic signal control using historical or real-time data, excelling in dynamic and unpredictable conditions.

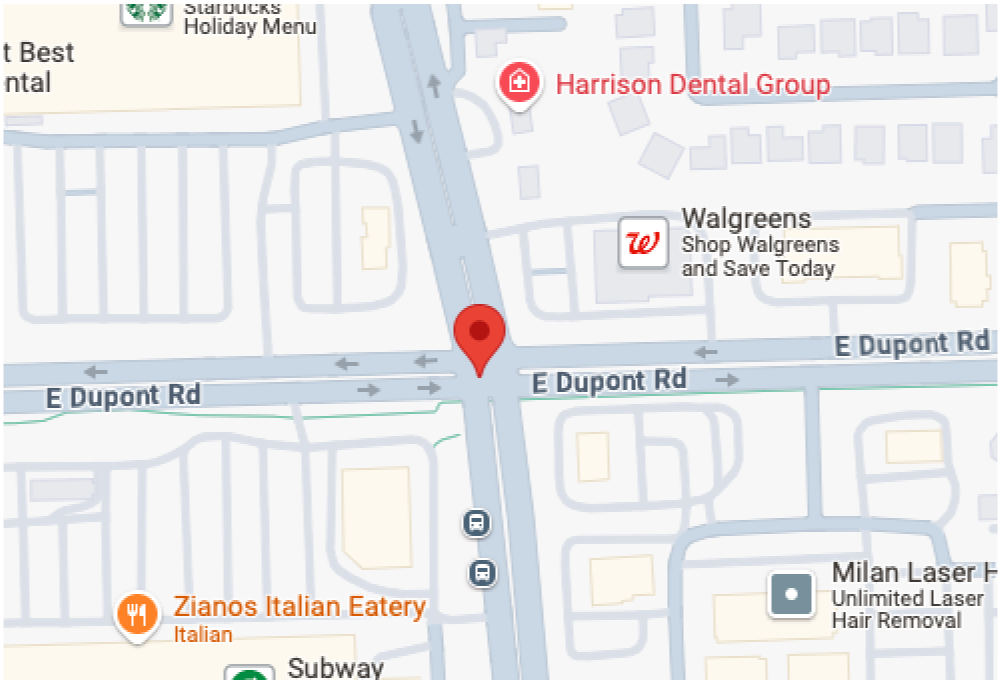
- Finally, network flow algorithms could be utilized to model intersections as networks, analyzing vehicle routing and bottlenecks efficiently but often requiring simplifications of real-world dynamics.

For the purposes of this project, we chose to utilize DES to model traffic flow. We felt that DES was the best choice because it provides a relatively simple and manageable framework for modeling the sequential flow of traffic at intersections. Its event-driven nature allows us to efficiently simulate key traffic interactions, such as vehicle arrivals, departures, and traffic light changes, without needing to track every moment in time. DES also strikes a good balance between realism and computational simplicity, making it ideal for exploring various scenarios, such as adjusting signal timings or adding lanes. Additionally, its modular structure allows us to potentially expand upon this model in the future to include more complex traffic flow behaviors, making it a practical and flexible foundation for this project.

Traffic flow data:
https://www.nircc.com/uploads/1/2/9/8/129837621/coldwaterdupont120.pdf

4. Implementation

We used the SimPy library in Python to simulate traffic flow through a simple four-way intersection controlled by traffic lights. To build an accurate simulation, we used real traffic flow data from a four-way, two-lane intersection in Fort Wayne, Indiana, gathered by the Northeastern Indiana Regional Coordinating Council. We simulated traffic flow at various times of day, and experimented with changing the number of lanes, traffic light timing, and other variables to support our recommendations.



The real-world intersection that our traffic flow simulation is based on.

Time	Vehicles/Minute
6:00AM	12.2
8:00AM	40.5
12:00PM	47.4
5:00PM	66.0

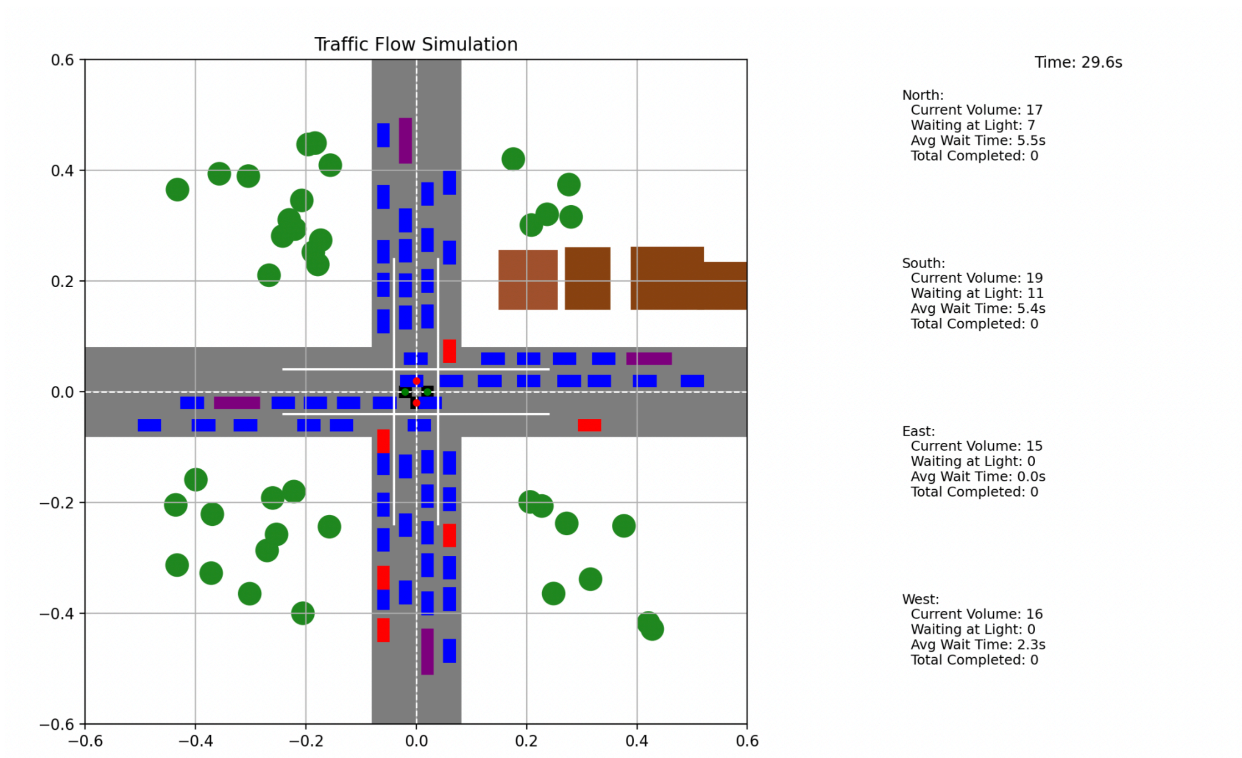
Average traffic flow rates at different times of day for the above intersection.

4.1 Key Simulation Components

We created global variables to allow for the customization of the number of lanes, average traffic flow rates (including volume and speed), traffic light timings, car length and turning probability within the simulation. Although this simulation is based on data from a real-world intersection, we wanted it to be easy to tweak certain variables to see how changing certain attributes of the intersection impacted traffic flow in order to make our recommendations for minimizing congestion.

In the simulation, vehicles are dynamically generated in each lane based on the specified traffic flow rate. We based these traffic flow rates on the real-world traffic flow data shown in the table in section 4.0. Each vehicle is randomly assigned a direction (north, south, east, or west), turning direction (right or straight), and attributes such as position, speed, and wait time, which are updated as the simulation progresses. Traffic lights are set to alternate between green, yellow, and red states for the north-south and east-west directions, with configurable durations for each phase that are again based on the global variables. Vehicles respond to light states, stopping at red lights and crossing the intersection when green. Lanes are modeled as queues, ensuring vehicles maintain appropriate spacing. The simulation tracks key metrics such as the number of vehicles waiting, average wait times, and total vehicles processed for each direction.

Finally, we built out a real-time, animated visualization using the MatPlotLib library in Python to illustrate the intersection layout, traffic light states, and vehicle movements. Statistics for each direction, including currently waiting vehicles, average wait time, amount of vehicles processed and current volume of traffic, are displayed alongside the animation. The average wait time is based on historical data of cars that have previously passed in the specified direction, with the default being set to 50 cars but which is also configurable. This implementation enables easy experimentation with parameters like lane counts, traffic densities, and light timings, allowing us to quickly observe the impact of changes on traffic flow and intersection efficiency.



A screenshot of our simulation in action.

4.2 Assumptions and Limitations

While our simulation is effective at modeling basic traffic flow through an intersection, it has several limitations.

Firstly, the simulation does not allow for left turns within the intersection. Left turns presented a level of complexity that was infeasible for the scope and time constraints of this assignment, as it would require the consideration of protected vs. unprotected left turns. The introduction of left turns would also likely require us to include separate turning lanes, which could affect vehicle positioning, movement, and overall intersection capacity. While the simulation does account for a randomly assigned percentage of vehicles turning right, it does not incorporate traffic rules such as the differing legality for vehicles turning right on red, which can vary by state and significantly impact traffic flow dynamics. Additionally, traffic volume in the simulation is currently measured as the number of vehicles approaching the intersection, but the simulation does not fully capture the impact of traffic light constraints on throughput - it assumes that this singular traffic light is an isolated system, and does not take into account additional congestion that may be present even after vehicles move through the intersection. Finally, the simulation does not consider pedestrian crossings, which can influence traffic light timing and reduce available throughput for vehicles.

We are also aware that in any real-world discrete simulation, there will be anomalous scenarios that, while rare, would affect conditions. For a traffic flow simulation, events such as collisions, road damage, weather conditions and construction could be considered non-standard but could have a potentially significant impact on the output of the system. We also chose not to account for anomalies but are generally aware of their effects.

It is critical to understand that due to these assumptions, our model is a simplification of real-world complexity. Further refinements would be needed to make it more realistic and applicable to specific intersections.

5. Results and Conclusions

While our traffic flow simulation provides a solid foundational model for understanding basic vehicle movement through an intersection, we believe that its significant limitations make it unsuitable for providing actionable recommendations for traffic flow optimization at this time. The lack of left-turn modeling, exclusion of pedestrian interactions, absence of right-turn-on-red logic, and simplification of traffic light constraints highlight some key areas where the model does not fully reflect real-world dynamics.

However, our simulation serves as an excellent starting point for exploring intersection traffic flow. With the modular structure and configurable parameters of DES, it provides a framework that can be expanded upon to include additional complexities, such as dynamic traffic patterns, pedestrian crossings, and left turning lanes. Future enhancements could improve its accuracy and utility, enabling our simulation to become a useful tool for traffic management and optimization.

Works Cited

Chen, Xu, Junshan Wang, and Kunqing Xie. "Trafficstream: A Streaming Traffic Flow Forecasting Framework Based on Graph Neural Networks and Continual Learning." arXiv.org, June 11, 2021. https://arxiv.org/abs/2106.06273.

Gerlough, Daniel L., and Matthew J. Huber. *Traffic flow theory: A monograph*. Washington: Transportation Research Board, National Research Council, 1975.

NIRCC Intersection/Arterial Data, January 14, 2020. https://www.nircc.com/uploads/1/2/9/8/129837621/final_2022-2026_tip_5-3-21.pdf.

Xiong, Liyan, Xinhua Yuan, Zhuyi Hu, Xiaohui Huang, and Peng Huang. "Gated Fusion Adaptive Graph Neural Network for Urban Road Traffic Flow Prediction." *Neural Processing Letters* 56, no. 1 (February 6, 2024). https://doi.org/10.1007/s11063-024-11479-2.