

## Traffic Simulation - Project Plan

*This file was the initial documentation I had planned before any code was written.*

*note: some things are very different than what the final piece turned out to be.*

**The Plan:** Simulate traffic through 4 intersections during theoretical peak hours. Run the data from that simulation through gurobi to see if optimising traffic light sequences can potentially increase traffic flow in these areas.

### **Phase 1 – The Simulation:**

Initial simulation of the traffic along a busy road with 4 intersections. using arrays for roads/lanes to show how busy they are. Each entry in the array is 1 car, so 1=car 0=no car. Can then use that to generate a heat map for visuals based on how “full” the arrays are compared to their set maximum (road length).

#### **Needed:**

- Data for the numbers of vehicles along each street/road.
- Statistical distributions of the vehicles along the roads, Probabilities of where each will go throughout the peak times.

#### **Future:**

- Simulation visuals, mapping agents on top of a simple map.
- Switch to a 6 lane main road model to allow for individual turning lanes – increase complexity but captures larger roads.
- Merge a 3 lane road into 2 (? For fun mostly – creates another bottleneck, push Gurobi even further)
- Allow for right turns – add extra lanes at lights to allow that.

### **Phase 2 – The Optimisation:**

Run the given output from phase 1 through Gurobi to see if we can optimise the traffic light sequences in the area to increase traffic flow.

#### **Needed:**

- Sets:
  - Lanes (Left lane, straight lane (Lane 1, Lane 2 – Lane 3 later if possible)).
  - Intersections (Traffic lights – Forward, left turn, right turn (right turn only on some lanes))
- Data – Road congestion for each road/street (from phase 1).
- Variables:
  - Traffic light timers (1 if green, 0 if red) for all sets of traffic lights.
  - How LONG cars have been waiting on each road.
- Objective – Maximise traffic flow across all roads (minimize total wait times)
- Constraints:
  - Max time a light can be red – this should link light timers and how long cars have been waiting variables.
  - Minimum time a light can be green.
  - Lights at same intersection must obey traffic laws, i.e., if north/south lights are green, east/west lights must be red.

- Flow constraint.
- Cars in lanes must not exceed max road length

### **Phase 3 – Validation:**

After finding a solution from gurobi (if one was found), run that data back into the initial simulation for comparison.

### **Python Files:**

**Traffic\_Data** – Intersection data, probabilities for vehicles entering which intersection.

**Traffic\_Simulation** – Takes Traffic\_Data and runs the simulation using a standard timing sequence, e.g., 30s/30s.

**Traffic\_Optimisation** – Takes Traffic\_Data and Traffic\_simulation and inputs it into gurobi to find the optimal timers for the traffic lights at each intersection.

**Traffic\_Comparison** – Just a comparison graph to compare the 2 methods.