CS166 - LBA project

Minerva University

CS166 - Modeling and Analysis of Complex Systems

Stênio Alves de Assis

Prof. Tambasco

April 7th, 2024

CS166 LBA Project

Simulation

The code simulates urban traffic by analyzing the congestion of streets using a graph network through the motion of vehicles. Cars are distributed in a network with random nodes being assigned as their initial and final destinations. In each step, they move from their current node to the next node in the optimal path (the shortest path) to their final destination. Each node is a point of intersection between roads, while each edge is a road where cars move from node to node. At the end of the iteration, we have a map of the network showing the streets with a more relative number of cars represented by the intensity of red and the width of the edge. The model has the following parameters, variables, assumptions, and rules:

Parameters

- Graph: Any city/neighbor network using the OSMnx package osmnx.graph_from_address()
- Number of cars: the number of cars distributed in the network with initial and final destinations.
- Number of steps: the number of iterations in which we will run the simulation
- Jam threshold: We decided that the maximum number of cars in an edge to create a jam is the time to travel the road multiplied by 100.

    - Estimating a critical density (d_c) of cars per unit of length that causes a jam, we say it is equal to the inverse of the average spacing between cars (s):

      $$d_c = \frac{1}{s}$$

      Assuming the space between cars in a congested condition is s = 10 m = 0.01 km, the critical density d_c = 1/0.01 = 100 cars/km.

      The total number of cars to start a jam (N_c) for the road length (L) is:

      $$N_c = d_c \times L$$
      $$N_c = 100 \times L$$

      Since length ∝ time, we will use time instead of length.

Variables

- Number of cars in which node
- Number of cars in which edge while moving between nodes
- Car shortest distance between initial and final destination

Assumptions

- Directed graph, since roads can be one or two ways.
- Edges are not roads, but segments of roads between two intersections.
- It is a closed network; cars do not enter or leave the network during the simulation.
- The simulation stops when the number of steps is over or when all cars reach their final destination.
- A random node is chosen to be the departure and destination, which equals the chance of each node being selected.
- We disregard many factors influencing traffic, such as walkers, varying car speeds, traffic lights, vehicles, and driver behaviors.

Rules

- Distributed cars in the network with random initial and final destination
- Calculate their shortest path.
- In each iteration, move cars along the edges.
- If there are more cars than the threshold in an edge, create a jam and let the car in the same edge for one more iteration.
- Finish the simulation after all the steps.
- Plot the final network with edges with different widths and red intensities according to the number of cars.

Advantages and Limitations

The simulation models the cars (individual agents) and predicts jams on roads. These results can be used to optimize urban plans to modify road aspects such as maximum allowed speed, the traffic light time, and the need for unnecessary stops to make those roads less likely to be congested.

As mentioned earlier, this network does not account for factors that significantly affect traffic jams, such as the presence of traffic lights and the presence of public buildings, such as schools, hospitals, gyms, and governmental facilities, that will affect the traffic of people and cars on those roads.

One way to have a more accurate model is to have a jam threshold for each edge. We can define this value to be chosen as a parameter instead of being defined inside the code. However, this would still mean that this value is arbitrary and equal to all roads, independent of their lengths, maximum allowed velocity, and time travel. However, for a better simulation, it would be better to have a value proportional to the travel time of each road, which is calculated from the maximum speed and the length of the road for each road. Then, in the code, for each edge, we compute the minimal number of cars to start a jam, proportional to the road travel time. This value is multiplied by a factor of 100 to allow cars, derived as the critical density (see Simulation: Parameters).

Visualization

Beforehand, no visualization was being shown. After fixing the code, the same network was formed after the steps with the nodes omitted and the edges with width and color intensity proportional to the number of cars. For example, if the maximum number of cars on the roads is 5, roads with 2 and 3 cars will have around 50% of the red intensity, and their width will be changed proportionally, with a maximum value of 10. In Figure 1, we can see the network before the simulation is run. Figure 2 shows what the same network looks like after 100 steps with 1000 cars. In Figure 2, we can see the roads with higher and lower congestion.

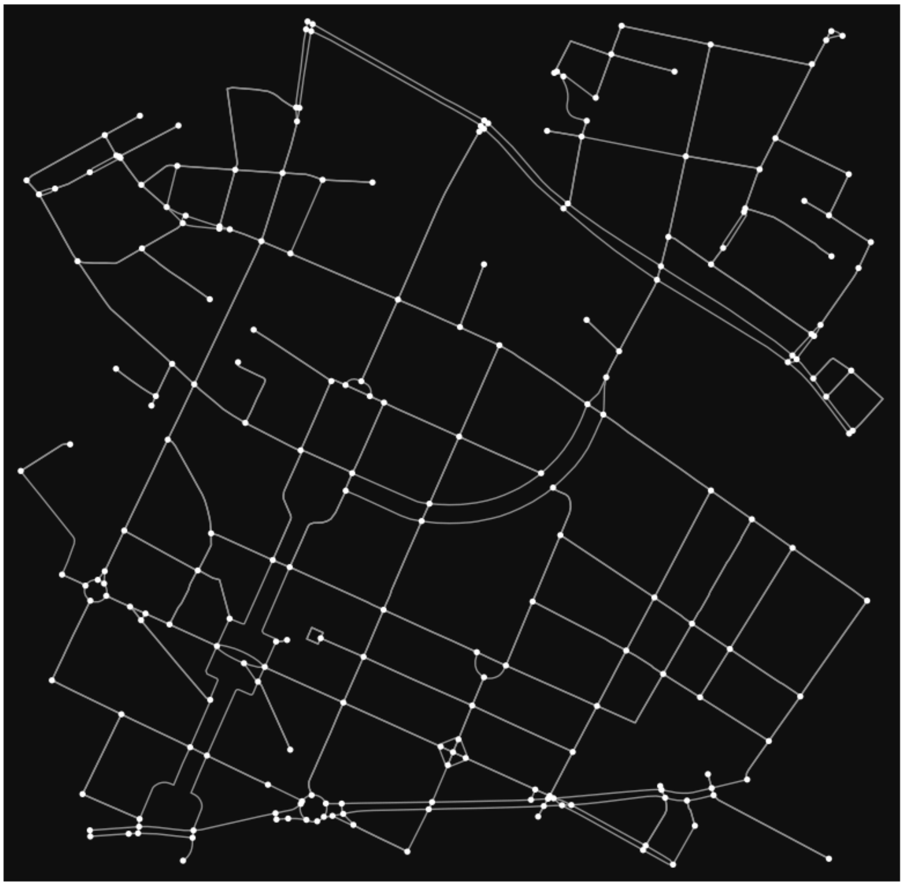

Figure 1: Driving network with edges as roads and nodes as intersections of Adalbertstraße 58, Berlin, Germany

Figure 2: Network of roads of Adalbertstraße 58, Berlin, Germany after Traffic Simulation. Higher intensity of red and wider roads means roads that are more congested after the simulation.

Theoretical and Empirical Analysis

In order to decide which metric best represents the empirical results, a correlation coefficient was computed for each of the following metrics and the number of cars in the edges and the nodes after 100 simulations with 1000 cars and 100 steps. The metrics used were the following:

For edge:

- Edge betweenness centrality

For nodes:

- Betweenness centrality
- Degree centrality
- Closeness centrality
- Node degrees
- Pagerank
- Load centrality

The correlation coefficient is seen in Figure 3. When computing the correlation coefficient, we used the number of cars in the edges for the edge betweenness centrality and the number of cars in the nodes for the other metrics. As observed, edge betweenness centrality is the metric with the highest correlation (~0.70). More central edges have a higher betweenness centrality. Since it is expected that those edges are more likely to be present in the shortest path between two nodes, it is more likely that more cars will pass between them. Those roads are expected to be more congested. Then, we will use the edge betweenness centrality as the primary metric for the congestion simulation.

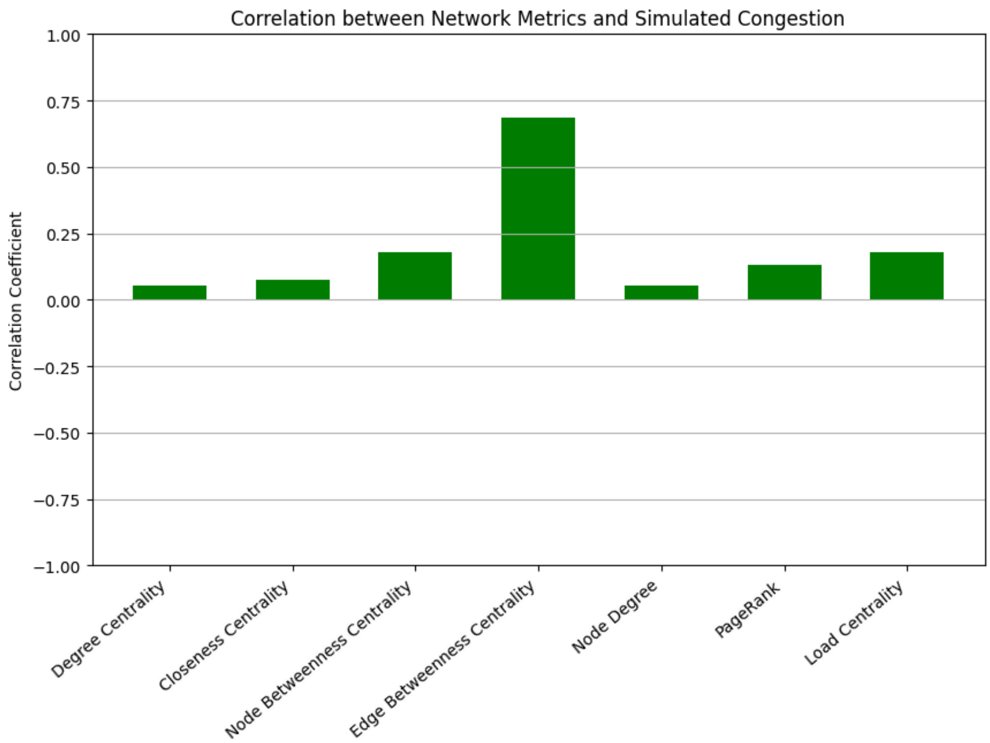

Figure 3: Correlation between Network Metrics from networkx and the simulated congestion. We observe the high correlation between the edge between centrality and the number of cars on edges.

Edge Betweenness Centrality

We can calculate the edge betweenness centrality (EBC) for an edge *i* as:

$$EBC(i) = \frac{1}{(n-1)(n-2)} \sum_{j \neq i, k \neq i, j \neq k} \frac{N_{sp}(j \xrightarrow{i} k)}{N_{sp}(j \rightarrow k)}$$

Where *n* is the number of edges, and $N_{sp}(j \xrightarrow{i} k)$ is the shortest path between edges *j* and *k* that passes through edge *i* while $N_{sp}(j \rightarrow k)$ is the shortest between *j* and *k*.

Comparison of EBC and Number of Cars

Figure 4a) shows the average percentage of cars for each edge, while Figure 4b) shows the EBC for each edge. Although the plots have some noise,, we still see similarities in the trend between those plots.

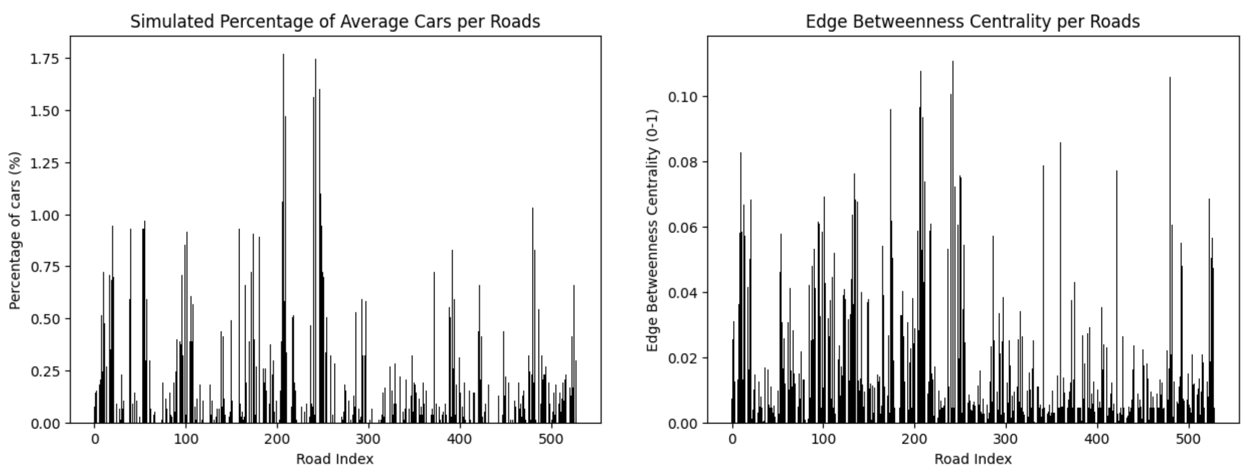

Figure 4a) Simulated Percentage of Average Cars per road.
Figure 4b) Edge Betweenness Centrality per road.

A better way to visualize which roads are more congested is by plotting the top 3% of the plot. Figure 5 shows the top 3% of the congested roads from the simulation and the top 3% of roads with higher EBC measurement with 53.33% of accuracy, measured as the percentage of shared edges. We can see that many nodes are present in both plots, indicating the EBC is a good predictor for the congestion of roads.

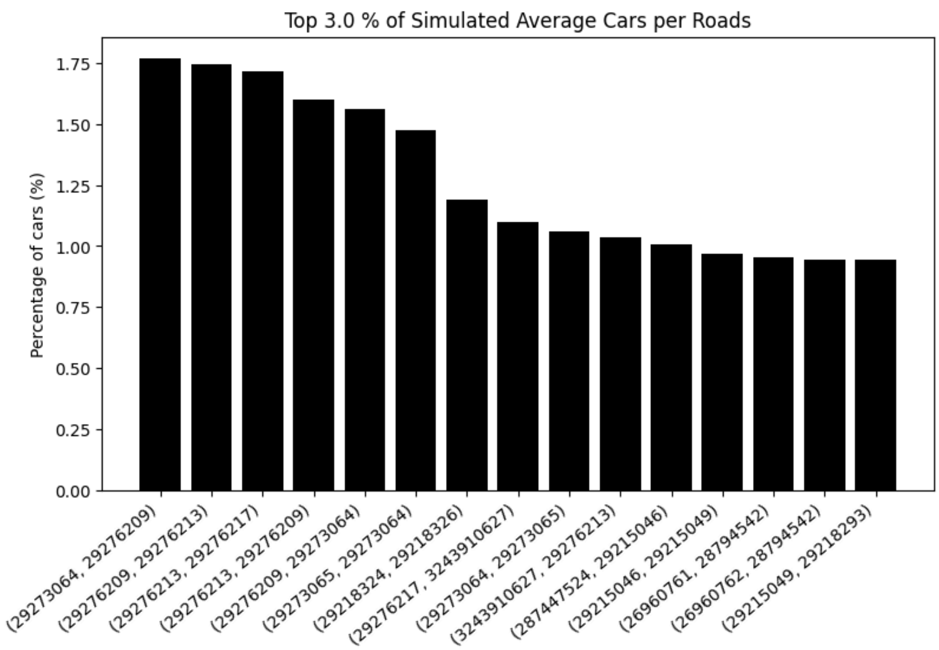

Figure 5a) Top 3% simulated average cars per roads

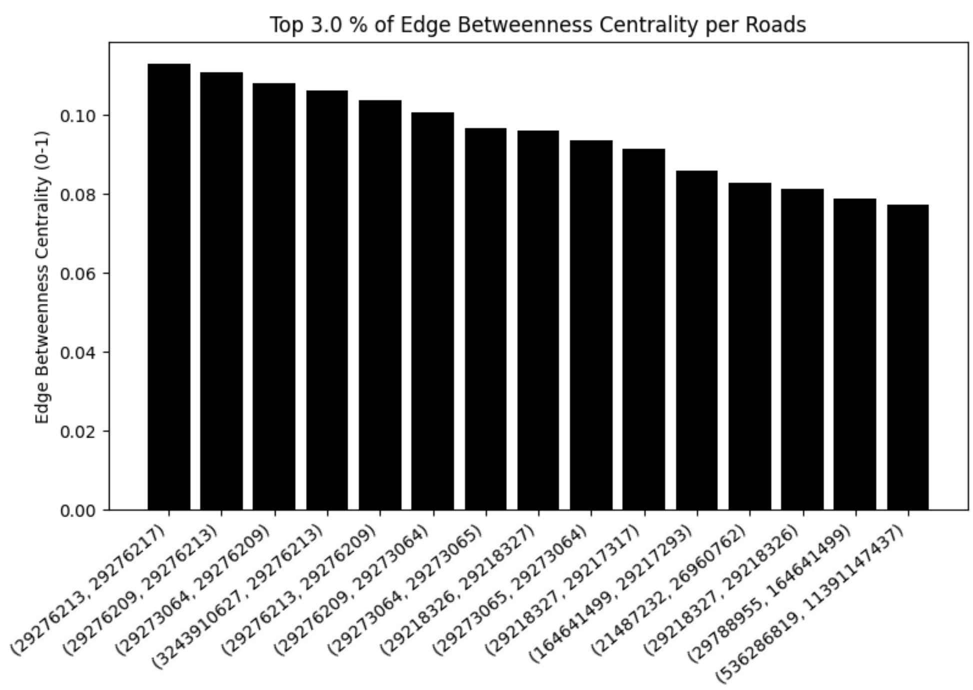

Figure 5b) Top 3% of edge betweenness centrality per roads

The edge betweenness centrality (EBC) is an effective way to predict which roads will be congested. The edge with higher EBC (29276213 – 29276217) is the third most congested edge, while the most congested node (29273064 – 29276209) is the third node with higher EBC.

London Traffic Simulation

Now, the model should be applied to a famous street in London called City Road, one of the most busy roads in Islington. The road is near the Minerva residential hall. The network is shown in Figure 6.

Figure 6: Driving network with edges as roads and nodes as intersections of City Road, London, England

In Figure 7, we can predict with 60% accuracy the five most congested roads in Islington borough in London. More simulations with a higher number of iterations and a better estimation of the average number of cars in the area as parameters can produce an estimation with a higher accuracy.

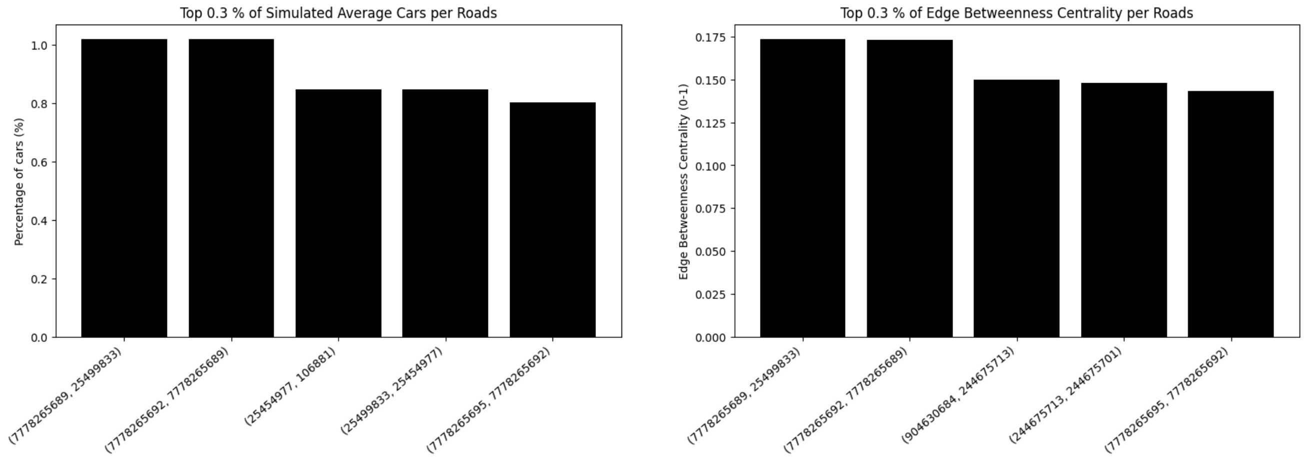

Figure 7: Top 0.3% of simulated average cars and edge betweenness centrality per roads.

The most congested edge is (7778265689, 25499833), which also has the highest betweenness centrality. In Figure 8, we see in red that the most congested edge is a small segment of the City Road nearby the Old Street Station.

Figure 8: The most congested edge in Islington is a segment of the City Road (arrow in red)

Appendix

The code used for this assignment can be accessed here:
CS166 LBA.ipynb

AI statement

Grammarly was used for grammar proofreading and Chat-GPT was used for helping debugging.