Reinforcing the Bangladesh roadwork

Network model generation

EPA133A: Advanced Simulation
Assignment 3

Daijie Huang
Milan Moleman
Nelene Augustinus
Roelof Kooijman
Simone Hoogendijk

Delft University of Technology

Reinforcing the Bangladesh roadwork

Network model generation

by

Daijie Huang

Milan Moleman

Nelene Augustinus

Roelof Kooijman
Simone Hoogendijk

Student Name	Student Number
Daijie Huang	5155118
Milan Moleman	5415764
Nelene Augustinus	5404916
Roelof Kooijman	5389437
Simone Hoogedijk	5405084

Instructor: A. Verbraeck
Instructor: Y. Huang
Project Date: 2, 2024 - 4, 2024
Faculty: TPM faculty

Style: TU Delft Report Style, with modifications by Daan Zwaneveld

Contents

1 Introduction

Background

Bangladesh faces natural disasters such as earthquakes, river flooding and cyclones more frequently [1]. As Bangladesh is a delta, the country is divided by a complex network of rivers and canals. This makes Bangladesh highly dependable on infrastructure such as bridges and roads that can keep regions connected. Therefore, the Bangladesh Infrastructure Upgrades project, initiated by the World Bank, aims to enhance the nation's transportation infrastructure. By analysing its vulnerability and robustness to natural disasters, possible policy interventions on specific bridges and roads can be assessed.

Aim of study

This analysis focuses on the flow of traffic and the collapsing of bridges on Bangladesh's main roads - N1 and N2 - as well as their side N-roads longer than 25 km.

To better understand how, on a system level, the collapse and maintenance of bridges may lead to traffic congestion, a simulation approach is used. The effect of bridge maintenance on traffic congestion behaviour is complex. Many small interactions together result in emerging traffic congestion.

Applying a simulation approach is beneficial in several ways. First, by simulating the underlying mechanisms and the resulting emerging behaviour, the system itself can be better understood [3]. This is helpful when designing policy interventions and assessing their impact on the situation. Moreover, using simulation allows for exploring 'what if' scenarios without actually having to test these in real life.

In this research, agent-based modelling (ABM) is used. On the one hand, ABM enables the modelling of the individual agents (vehicles in this case) and their interactions within the traffic system. On the other hand, since it allows for interactions between agents, it facilitates the simulation of emergent behaviour that arises from their interactions.

To better formulate the model and its corresponding experiment design, the research question is: How do bridges with different conditions collapse and hence impact traffic on Bangladesh's main roads (N1 and N2) and their side roads longer than 25 km?

Structure of report

The report presents the following structure: Chapter 2 will explain the conceptual model and its components. Whereas Chapter 3 describes the formalisation of the model and its implementation, Chapter 4 discusses the experimental setup used for the analysis. Chapter 5 gives the simulation results under different scenarios. Lastly, Chapter 6 discusses the limitations and possible improvements of the model.

2

Model conceptualisation

As stated in the introduction, a simulation model will be used to understand the emerging behaviour of traffic flows as a result of bridge breakdowns. This chapter will shortly describe the model on a conceptual basis.

General conceptualisation

The model is conceptualised following four main components that are derived from real life. Trucks can drive over roads (links). These roads can be connected through intersections. Roads can drive in both directions, meaning that from both the beginning as well as the end of a road trucks function as a source and a sink. In each timestep, vehicles (trucks) are created at the SourceSinks. Consequently, the vehicles are assigned a path through the road and bridge network. The paths can be assigned in three different ways, namely a straight routing type, a random routing type, and a shortest path routing type. For this project, the N1 and N2 road in Bangladesh, together with its intersecting side N-roads that are longer than 25 km will be used. Figure 2.1 shows an overview of the relation between bridges, links, intersections and trucks.

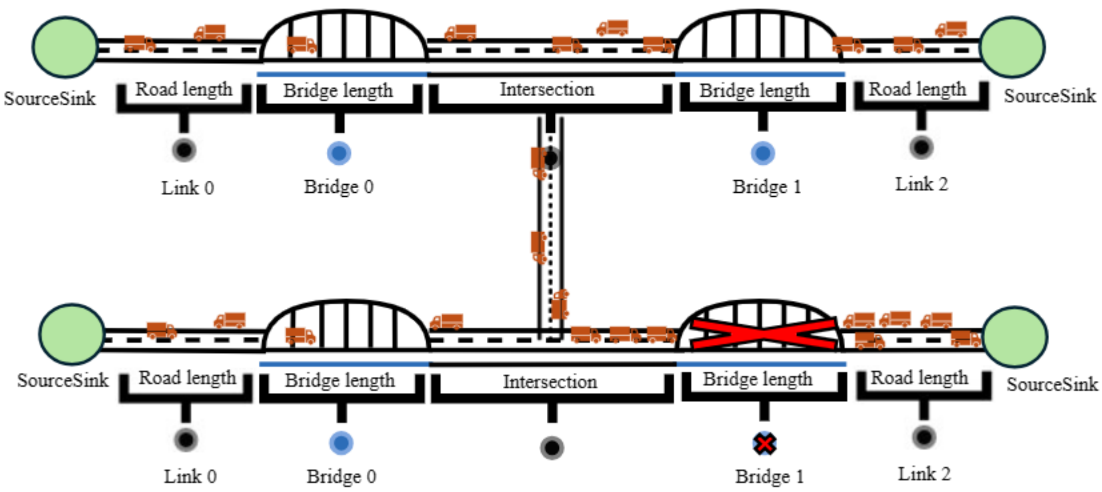

Figure 2.1: Conceptual overview of SourceSinks, Bridges, Links and Intersections

The bridges within the network can have different conditions, following the Bridge Condition Survey (BCS) of the Bangladesh government [6]. Table 2.1 below shows an overview of these bridge conditions and what they mean. A bridge might collapse based on its condition. Due to a bridge collapsing, trucks could get delayed as collapsed bridges have to be repaired.The delay time will vary based on the length of the bridge. For the time that the trucks have a delay time they will wait before a bridge. This is shown at the bottom of Figure 2.1. Bridge collapses could therefore have a serious impact on the continuity of the traffic flows in the Bangladesh road network.

Table 2.1: Possible bridge conditions.

Bridge condition	Explanation
A	No damage
B	Minor damage
C	Major elemental damage
D	Major structural damage

Table 2.2 below shows an overview of the different components used.

Table 2.2: Descriptions for each component.

Model component	Description
Source	Generate vehicles, entrance of the road
Sink	Remove vehicles, exit of the road
SourceSink	Generate and remove vehicles, ”2-way traffic”, represented as the starting and ending of the road
Bridge	Bridge along the road, with different conditions, possible to break down
Link	Road segments in between the bridges and
Intersection	Connect one road to the other
Vehicle	Truck

Key performance indicators

Measuring the model’s performance is crucial. It evaluates the results and guides decision-making processes accordingly. Table 2.3 shows the key performance indicators used to assess model behaviour.

Table 2.3: Key Performance Indicators.

KPI	Explanation
Average driving speed [km/h]	The actual average speed in km/hour of a vehicle
Total bridge collapse	The cumulative total amount of bridges collapsed
Total bridge collapse per category	The cumulative total amount of bridges collapsed, per category bridge
Average driving time [min]	Average driving time to reach destination

Considering the ethical dimensions of transport network criticality metrics given in the study of [4] it can be said that the chosen metrics in this study are utilitarian in nature. All four of the metrics reflect the overall performance and efficiency of the traffic. Furthermore, the used road networks represent only the main roads. So the metrics together with the scope of the road network focus on performance for all users and do not consider any disadvantage user groups or areas and there for this project focuses on utilitarianism and not egalitarianism. A reason why egalitarianism is not considered is that including egalitarian criticality metrics was to hard considering the time scope and already obtained skills during the course to implement these metrics.

3 Formalisation and Implementation

This chapter will elaborate on the formalisation and its implementation of the conceptual model discussed in the previous chapter. First, a general overview on how data is used and structured will be provided. Consequently, the data preparation process will be discussed in more detail, as well as the formalisation of creating intersections, generating the road network and assigning routing methods.

General overview

Figure 3.1 provides a summary of the entire process.

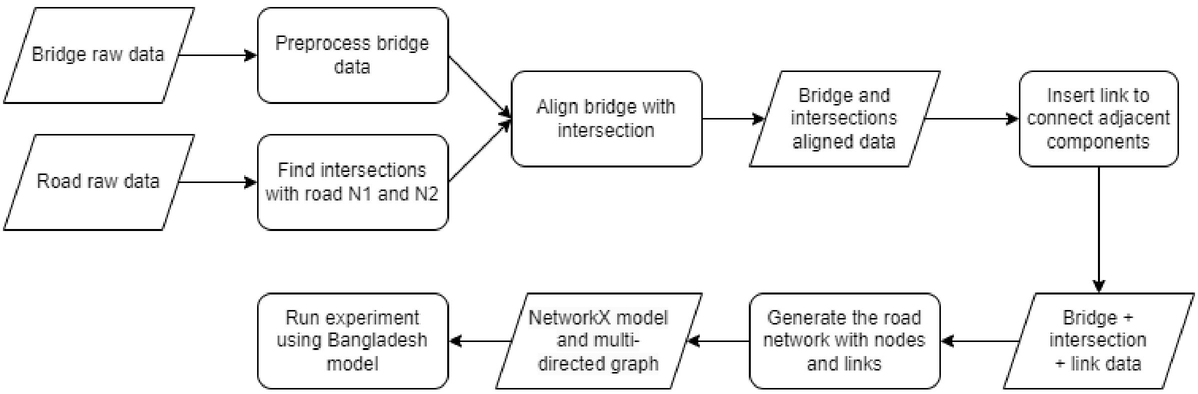

Figure 3.1: Process summary.

Data preprocessing

To create the model, data on the bridges and roads are required. The data for the bridges is given in an Excel file from Verbraeck [8]. The current state of the bridge data files is not usable for the model. Hence, it first needs to be cleaned. Below, the data cleaning process is discussed.

Bridge

For this assignment, the whole BMMS overview dataset was handled. This was done to make sure bridge data of all roads could potentially be used when investigating intersections.

Figure 3.2 shows the bridge data preprocessing process. Generally, the preprocessing process consists of dealing with missing values and duplicates of bridges without having separation in left or right directions.

The bridge data is cleaned in two steps. First, missing values are handled. Afterwards, duplicates are processed, studied, and handled too.

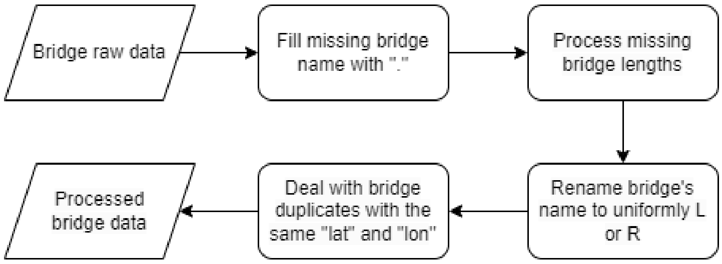

Figure 3.2: Bridge data preprocessing.

Handling missing values.

Having missing values may introduce inconsistency in the dataset, hence the missing values are checked. In total, 315 missing values are found in the "name" column of the dataset while 9 missing values are found in the "length" column. The missing "name" values were filled in with a ".", as the proper name is difficult to retrieve when missing. The missing "length" values are imputed using two different methods. The first option is imputing the missing values by using the value of a bridge that has the same chainage. If there is no bridge with a similar chainage, the average length of all bridges on the road will be used. By applying these two methods, all missing values are resolved.

Process the duplicates. Besides missing values, a lot of duplicates with the same latitude and longitude are retrieved from the duplicate analysis. The N1, N2, N5, and N8 roads have more than 50 duplicates, yet these are also large roads. Having multiple bridges at the same location might occur due to two different reasons.

The first reason is that some bridges are two-directional and both have a left and a right lane. The left and the right side of the bridge are included in the dataset as two separate entities. These entities are indicated with either 'L' or 'R'. In that case, having duplicate bridges is not an issue, as the truck will either drive on the left or right side of the bridge, based on the direction it's driving.

The second reason is that some rows are real duplicates of a bridge. This is the case when the longitude and latitude are the same and the bridge is meant to facilitate the same direction of traffic, for example, if two bridges both have an "R" value. In this case, a decision must be made to decide which of the duplicates must be kept. This is done based on the bridge condition. This project aims to study the driving, delay, and overall travel time for vehicles under different scenarios and identify which bridges are critical and vulnerable. For more reliable results, the worst conditions for the same bridge are used. Hence, bridge duplicates with the best conditions are dropped. This will assume the worst possible case for the scenario analysis. If the conditions are the same for the same bridge, the duplicates are randomly selected.

Road

The model will simulate the N1, N2 and their intersecting N-sideroads that are longer than 25 km. Provided for this project is the road data in the form of location reference points (LRPs). It is decided to refrain from using this road data to generate the network. The bridges themselves can already depict the road. Furthermore, the bridges come with a chainage, hence a location relative to the road. Although adding the LRPs will provide extra detail in the road shape, it is expected this does not add extra value in fulfilling the purpose of the simulation model.

Creating links. To connect bridges, a link is added between every two unique bridges. Bridges with an identical latitude and longitude are not connected through a link since they are identified as the same. Adding these links results in a continuous road on which bridges are interspersed with links. The length of a link is calculated by subtracting the chainage of the previous bridge of the chainage of the next bridge. To get the actual length of the link, half of the length of the previous bridge and half of the length of the next bridge are subtracted. The chainage of a link is calculated by taking the average of the chainage of both the previous and the next bridge. At last, the latitude and longitude of a link is determined by interpolating between the latitude and longitude coordinates of both the previous and the next bridge.

Creating intersections

To create a graph for the N1 and the N2 and their side roads it is required to find the intersections between the N1 and N2 and their side roads. For the model, only N roads that are longer than 25km are considered. Figure 3.3 gives an overview of the determination of the intersections. Starting with the road data from the "roads.csv" file, linestrings are made from the LRPs corresponding to the same road. For these lines, it is checked if they are N roads and if they are longer than 25km. All the lines that meet these conditions are buffered. Buffering the linestring makes them bigger, hence the linestrings will cover a bigger area. This makes it easier to find intersections because the road LRPs do not always properly connect. Figure 3.4 *a* shows two linestrings not connecting. When buffered, they do connect as shown in Figure 3.4 *b*. These linestrings were saved in a dataset. For simplicity reasons no difference was made between the buffered linestrings and normal linestrings. The dataset of the linestrings of the N roads only contains the name of the road and its linestring. A similar dataset was made for the N1 and N2 roads only.

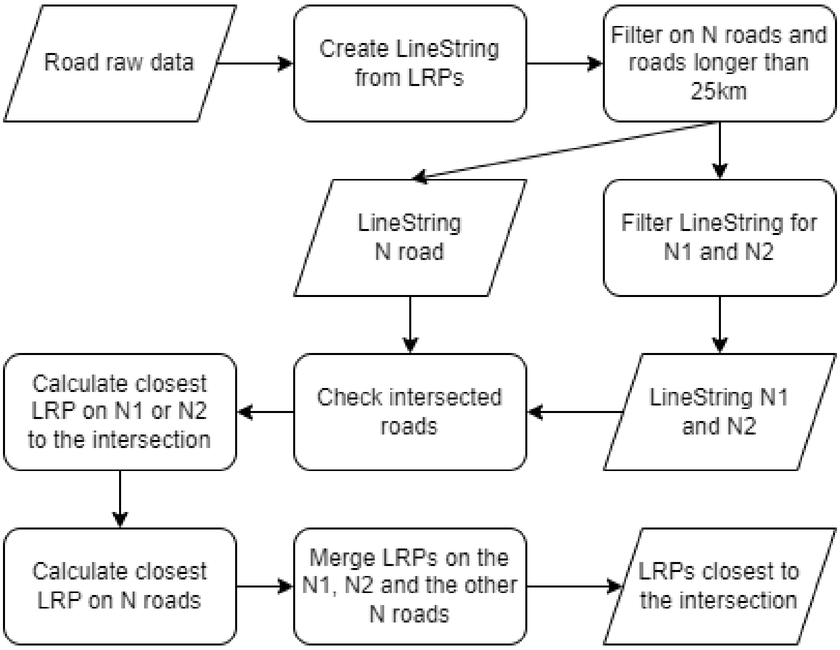

Figure 3.3: Find intersections with road N1 and N2.

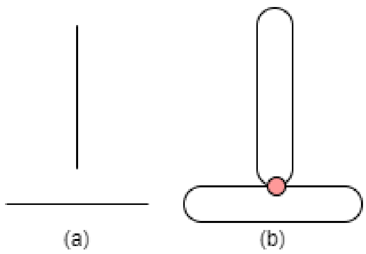

Figure 3.4: Example for buffer.

After the dataset of linestrings and corresponding road names for the N roads was made, it was investigated whether there were any intersections of N roads with the N1 or N2 road. If they intersected, the closest LRP's from both the N1/N2 side of the intersection as well as the other side of the intersection were calculated. The result of all found intersections was merged with the LRP data of the cleaned roads dataset into one dataset. The result is a dataset that contains the LRP's of the two sides of the intersections, with an extra column that links these LRP's. Figure A.1 shows the intersections as points and the road they intersect with.

Generating the road network

After cleaning the data, retrieving all intersections with roads N1 and N2, and adding links between bridges, the road network was generated. For this, the NetworkX library was utilised. A directed graph was initialised since the network includes roads in both ways. For every road, each bridge, link, intersection, or sourcesink was added to the network as a node. For each consecutive pair of nodes, an edge is included. In addition, the edge pairs were reversed to assess the effect of two-way roads.

For each intersection, the corresponding nodes on the intersected roads were retrieved from the intersections dataset created earlier. For these nodes, an edge was created. The length of the edge was set to zero since the latitude and longitude is the same for both nodes. In principle, these two nodes represent the same points since they share the same coordinates and have a distance fixed to zero. Yet, for simplicity, two nodes are created instead of one. However, computing the shortest path and its distance is not affected due to the fact that the length of the edge is zero.

Whereas the intersected edges have a fixed distance to zero, the distance of normal edges was computed using the chainage of the nodes. The absolute value of the difference in chainage is utilised as distance between the nodes.

The corresponding road network that was generated is visualised in Figure 3.5

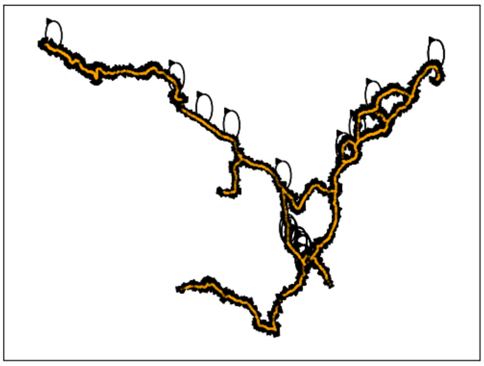

Figure 3.5: Road network using NetworkX

Assigning routing methods

In this section, the method for choosing the route of the truck will be discussed. When initializing the model, a model parameter 'routing type' can be set to either 'straight', 'random' or 'shortest'.

Straight and random routing

The straight routing type results in a straight path from the given 'sourcesink' to the last 'sourcesink' of the same road. The random routing type results in a path from the given 'sourcesink' to a random 'sourcesink' in the network. These methods were not altered from the original model.

Shortest path routing

The shortest routing type results in the shortest path from the given 'sourcesink' to a randomly chosen 'sourcesink' in the network. Using the shortest path method adds value to the model as more realistic traffic behaviour can be simulated. This is because it is likely that, if rational behaviour is assumed, people choose the fastest way to their destination. Additionally, by assigning shortest paths, the travel behaviour is randomized which also corresponds more to real-life travel behaviours.

Using NetworkX to compute the shortest path The shortest path can be computed using a heuristic that is derived from graph theory. Using the model graph attribute G, The NetworkX library was used to calculate the shortest distance from a sourcesink to another randomly chosen sourcesink. The *shortest path* NetworkX method can use either the Dijkstra or the Bellman-Ford algorithm. For this project, the choice was made to use the Dijkstra algorithm. The Dijkstra algorithm is more efficient compared to Bellman-Ford in terms of time complexity [5]. Additionally, the need for using Bellman-Ford disappears as no negative edges were used in simulating. It is important to be aware that the different distances between nodes are taken into account by setting the shortest path's method parameter *weight* to the 'distance' attribute of the edges between nodes. One of the disadvantages of Dijkstra's Algorithm is that it conducts a blind scan, resulting in an increased processing time [2]. To decrease this processing time, a dictionary of all shortest paths for origin, destination pairs is created that keeps track of newly computed paths. When a shortest path between to nodes has to be found, this dictionary is first checked. If the path already exists, then there is no need to compute the path again, reducing processing time of the simulation.

In the model class, the *get route* method assigns the correct routing method based on the set model parameter. Figure 3.6 provides an overview of this assignment of a routing method and the corresponding actions related to the selected routing method.

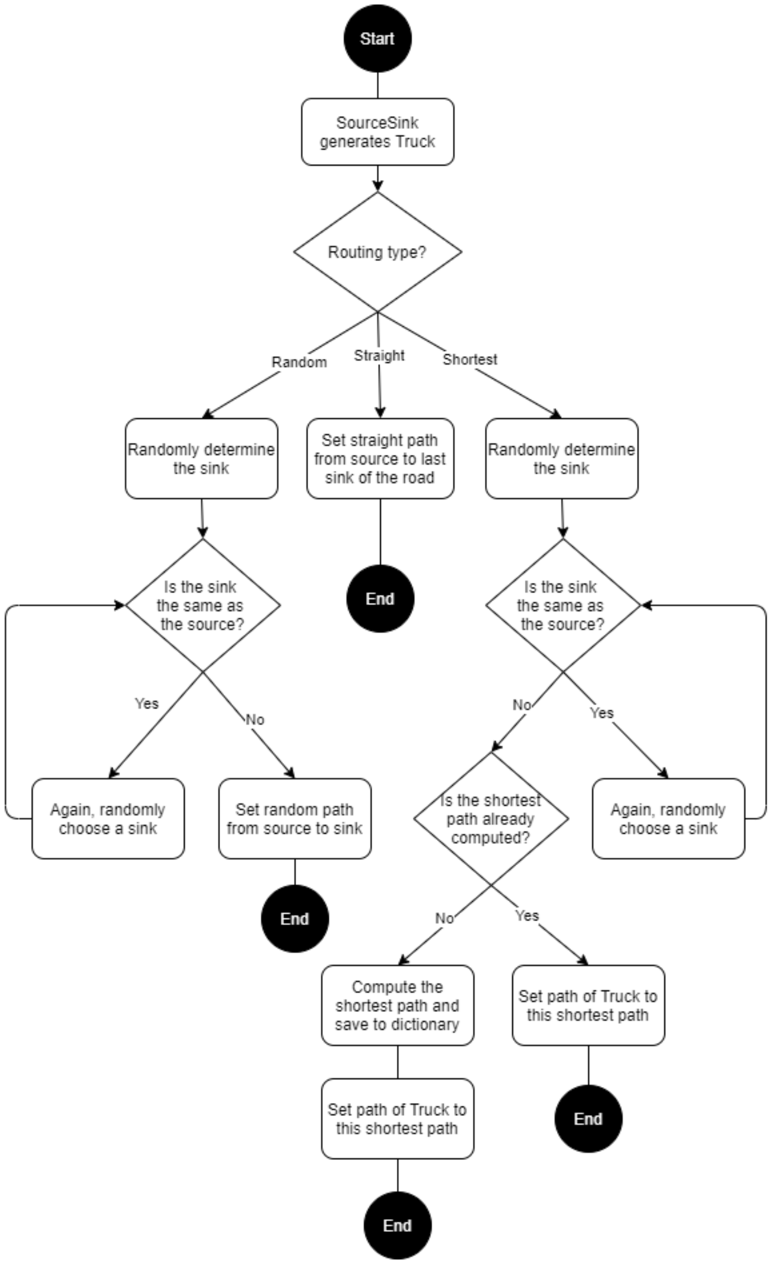

Figure 3.6: Flow chart of the routing method

4

Experimental design

This section describes the choices for the setup of the experiments. At first, the experimental setup will be explained. Afterwards, a short description of the scenarios will be given.

Reasons for experimental design

Before looking into the experiment itself, the reasons for doing experiments need to be discussed. According to Sanchez et al.[7], these can be the following reasons. First, recall that the project aims to study the impact of bridges with different conditions on Bangladesh‘s traffic, design experiments can help develop a basic understanding of the model’s behaviour. Furthermore, experimental designs can quantify research goals, i.e., quantify the impact of bridges as driving time. Only with quantifiable parameters it is possible to compare the merits of various decisions or policies.

Scenarios

For this study, a total of five experiments will be done, one for each scenario. Each experiment involves 10 iterations, resulting in 50 runs in total.

The scenarios differ in the probability of a bridge breaking down, which is different for each condition. An overview of the probability per bridge condition per scenario is given in Table 4.1. The probability that the bridges with different conditions break down increases per scenario, where scenario one has the lowest probabilities and four the highest.

Table 4.1: Scenarios with different probabilities.

Scenario	Cat A %	Cat B %	Cat C %	Cat D %
0	0	0	0	0
1	0	0	0	5
2	0	0	5	10
3	0	5	10	20
4	5	10	20	40

Though having only probability differences in each scenario is considered elegant in design. Yet, this subtle design is a trade-off between capturing meaningful KPIs and minimising the complexity of the experiment [7]. Furthermore, the chosen scenarios are designed with significant differences in the probabilities of bridge breakdowns across conditions, which can already give useful results for a preliminary analysis.

Model run

Every model run will consist of 7200 steps. Every step represents a minute and 7200 minutes implies five days in total. Per five minutes a truck is created at a given source. This means that the model will generate 1440 trucks in total.

Replications

Due to stochastic effects, a single model run is not reliable. Therefore, each experiment will involve 10 iterations.

Any increase in replications compared to having no replications will be beneficial for the reliability of experiment results. However, there is a certain limitation to only using 10 replications for the experiments. Normally, more replications would be better as with only 10 replications there is still a significant chance that not all stochastic effects are included. However, due to time constraints, the choice was made to only do 10 replications for each experiment.

5

Results

In this section, the results of the experiments that are conducted will be discussed. To analyse the results of the model, several Key Performance Indicators are used. These involve the average driving time, the average delay time, the average speed of the trucks and the number of collapsed bridges.

Average driving time and delay time

Figure 5.1 shows the average driving time it takes for a truck to travel from a source to sink. The average driving time does not control for the travel distance, which differs per vehicle. For this the average speed is used. However, from Figure 5.1 it can be concluded that the average driving time increases in correspondence with the order of the scenarios. Hence, an increase in the chance of bridges to collapse also increases the average driving time of vehicles. Average driving time grows exponentially in the first three scenarios, whereas the last two scenarios result in a linear increase of driving time.

Figure 5.2 shows the average delay time. Whereas the average driving time focuses on the average total travel time of all vehicles, the average delay time corresponds to the average delay time that is assigned to a collapsed bridge. Even though this metric might not be the most insightful when assessing network criticality, it is useful to validate model behaviour in the different scenarios. Just as with the average driving time, the average delay time per bridge increases along with the scenarios. This corresponds to expected model behaviour, as with increased chances of bridge collapse, more bridges are likely to collapse, resulting in more bridges having an increased delay time (as opposed to bridges that are not collapsed having no delay time). As with each scenario the probabilities of bridge collapse are doubled, the delay time follows exponential behaviour.

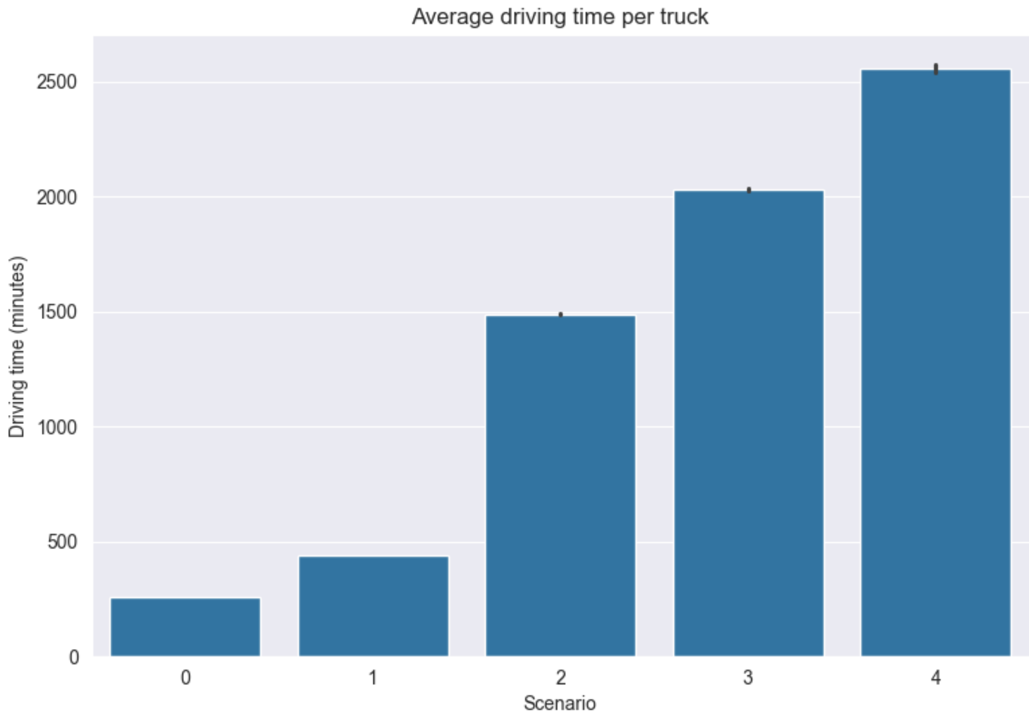

Figure 5.1: Average driving time per vehicle

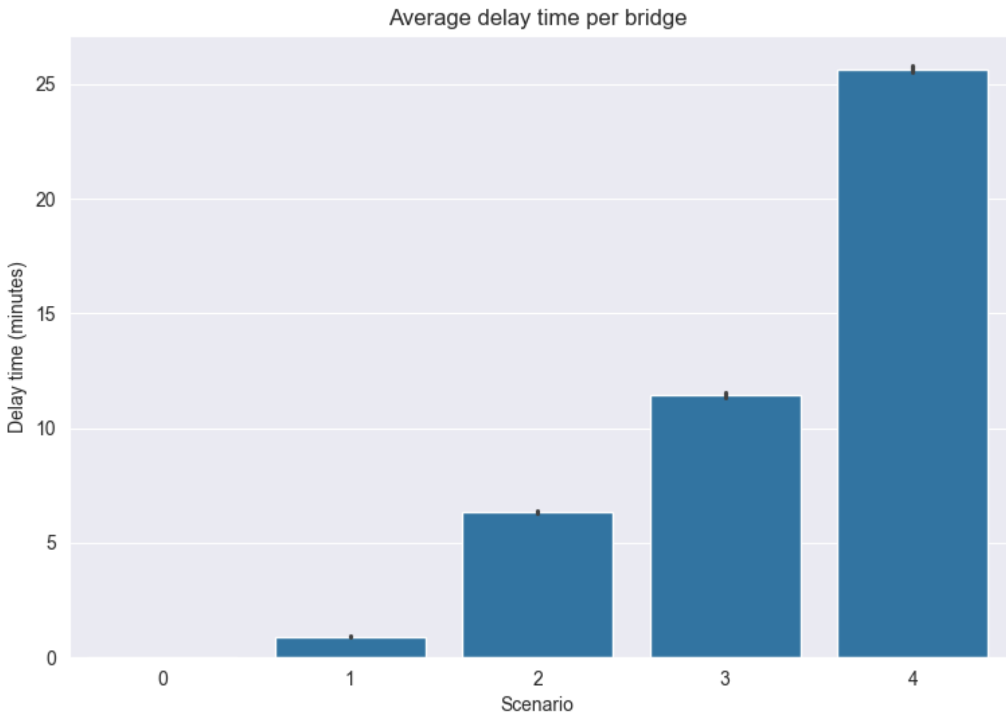

Figure 5.2: Average delay time per bridge

Average speed

Both the average driving and delay time do not account for the distances travelled by vehicles, yet heterogeneity exists in the travel distances for each vehicle. Since the shortest path between a source and sink is computed for a vehicle, each vehicle encountered different travel distances. The length of the shortest path is used to retrieve the travel distance per vehicle. The average speed per vehicle is shown in Figure 5.3.

Figure 5.3 shows that average speed is almost 48 km/h for the base case, in which no bridges collapse. However, when a probability for collapsing is included, the average speed decreases. Whereas the average speed is reduced in the first scenario, the last three scenarios show a more significant decrease. The average speed equals approximately 33 km/h in the first scenario, 9 km/h in the second scenario, 6 km/h in the third scenario, and 2 km/h in the last scenario. An interesting finding is that both the second and third scenarios denote an almost similar average speed per truck. Yet, the probabilities for bridge collapse significantly differ between these scenarios. It is unclear why this behaviour occurs. This could be grounds for further research.

In addition, the results show that a brief increase in average speed is denoted in the first model steps for all scenarios. While the average speed in the base case grows afterwards, the average speed reduces for all scenarios with collapse probabilities. These finding align with the observation that after one model step some bridges already collapse. After collapsing, the average speed decreases heavily. The magnitude of this number of bridges and that collapse determines the magnitude of the decrease in average speed of trucks. Hence, a clear link between the probability of bridges collapsing and average speed can be established, just like the driving and delay time of trucks.

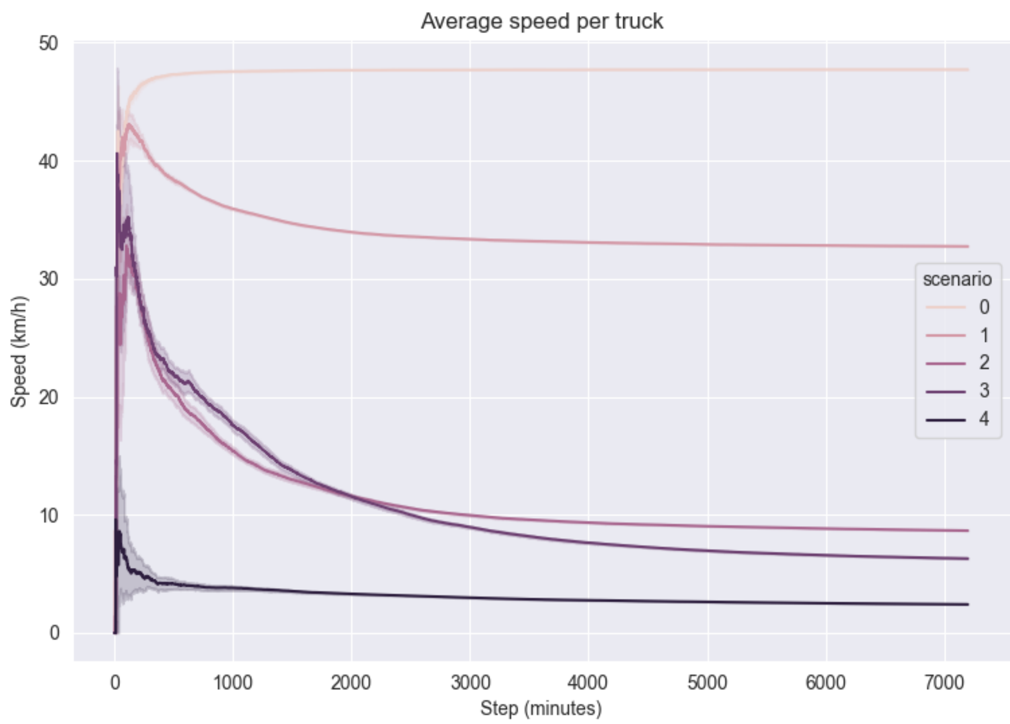

Figure 5.3: Average speed per vehicle

6 Reflection

The purpose of the model is to represent a certain aspect of a real-life system. In this study, trucks drive over the N1, N2 and their side roads longer than 25 km. Constructing a model that mirrors the real system is impossible and unwanted. The main limitations of the model are given below. Also, possible improvements and extensions will be discussed.

(1) The current model only focuses on the N1, N2 and their side roads, which may result in an incomplete depiction of the transportation network. The model could overlook alternative routes and pathways utilized by trucks, which may introduce biases and hinder the model's ability to provide a comprehensive understanding of traffic flow dynamics and network resilience.

(2) For the intersections, N204 and N207 have two intersections with N2, but only one for each road is taken into account. This means tha a lot of trucks will have to drive to the used intersection to the N2, while, in reality, the truck could be faster when it drives over the other intersection. This will lead to higher driving times then expected. To include the second intersections, the used methods of calculating shortest path should expanded.

(3) This model does not take into account more complex road behaviour. For example, when a truck encounters a collapsed bridge, it might choose to drive to the next nearest bridge if it is close enough. This would then result in other bridges being used more often when a nearby bridge is collapsed. Consequently, this could result in faster deterioration of this bridge condition. Another example is road congestion behaviour. The more cars are waiting at a bridge the more congestion.

(4) Aside from complex road behaviour, there are also limitations concerning bridge collapse behaviour. The model does not account for repair time and gradual deterioration of bridges over time due to weather exposure and traffic density. These simplifications can lead to less realistic model results.

Future research could enhance the outcomes by several extensions:

- **Additional roads**: Include additional roads to provide a more comprehensive view of the transportation network. This would help mitigate bias and offer a more egalitarian perspective by considering alternative routes and their impacts on traffic flow. Research could further explore data-driven approaches to identify and incorporate other relevant roads into the model.
- **Explore additional network criticality measures**: This could involve integrating metrics on transportation network analysis to capture additional aspects of network vulnerability and resilience [4].
- **Account for bridge repair and deterioration**: Extend the model to include factors such as bridge repair time and deterioration due to traffic density. Incorporating these factors would provide a more realistic representation of bridge collapse behaviour and its long-term implications for transportation network resilience.

- **Simulate realistic travel flows**: This could involve incorporating spatial and temporal variations in traffic patterns based on factors such as population density, economic activity, and time of day.

References

[1] Asian Disaster Reduction Center (ADRC). *Information on Disaster Risk Reduction of the Member Countries*. 2019.

[2] Admin. *Dijkstra Algorithm - GATE CSE Notes*. Mar. 2023. URL: https://byjus.com/gate/dijkstra-algorithm-notes/.

[3] Lawrence Blume. *Agent-Based Models for Policy Analysis*. July 2015. URL: https://www.ncbi.nlm.nih.gov/books/NBK305903/.

[4] Jan Kwakkel Bramka Arga Jafino and Alexander Verbraeck. "Transport network criticality metrics: a comparative analysis and a guideline for selection". In: *Transport Reviews* 40.2 (2020), pp. 241–264. DOI: 10.1080/01441647.2019.1703843. URL: https://doi.org/10.1080/01441647.2019.1703843.

[5] Brian Patrao. "Dijkstra's vs Bellman-Ford Algorithm - Brian Patrao - Medium". In: (Apr. 2023). URL: https://medium.com/@brianpatrao1996/dijkstras-vs-bellman-ford-algorithm-383e4771c2cb.

[6] Roads and Highways Department. *BRIDGE CONDITION SURVEY MANUAL*. 2005. URL: https://www.rhd.gov.bd/Documents/HDM/BridgeConditionSurveyManual/Index.pdf.

[7] Susan M. Sanchez, Paul J. Sanchez, and Hong Wan. "Work smarter, not harder: A tutorial on designing and conducting simulation experiments". In: *2021 Winter Simulation Conference (WSC)* (2021).

[8] A. Verbraeck. *bridges*.

A

Additional Figures

Figure A.1: Intersection points and there roads

B
Bonus Assignment

The current intersections are estimated using the LRPs in the road database. However, these estimations will not always be accurate. To control if the estimations are accurate, they will be compared to the intersections that are calculated with a shapefile containing the roads of Bangladesh. The calculation of the intersections with the shapefile is the same as explained in the ”Creating Intersections” section in Chapter 3, only the preprocessing is different.

The shapefile does not properly indicate which road parts belong to which N roads. So instead of filtering on N roads, the shapefile filters on which road parts are referenced to as primary or trunk. For these primary/trunk roads, the intersections with the N1 and N2 are calculated. Identifying the N1 and N2 from the shapefile was possible because, fortunately, the N1 and N2 are identifiable using the ”ref” column of the shapefile data. The shapefile returns more intersections than the road data. So, it is then checked which intersection of the shapefile is closest to the road data intersection. For these two points, a line is created. To calculate the distance in meters, the coordinate reference system (CRS) must be changed to one that is based on meters and not latitude and longitude. The used CRS is EPSG:3857. With the new CRS, the length of the line is given in meters and indicates how far the estimated points from the road data are from the estimates from the shapefile.

Figure B.1 shows both points, blue for the intersection of the shapefile and the red points for the road data (called CSV data because the data is in a .csv file). The line is also shown in white. From Figure B.1, it is visible that the blue points accurately indicate the intersections, while the blue ones are slightly off. Figure B.2 depicts the gap between the point from shapefile data and the road data in kilometers. It is notable that the intersection to the N1 and the N2 are both around 7 km. Looking closely at the plot for the N1 and N2 in Figure B.1 shows that both intersections are the same, but one from N1 to N2 and the other from N2 to N1. There is a difference in these two because the intersection for one is based on the LRPs on the N1 and the other the LRPs of the N2.

Based on the results, it can be said that the inaccuracy of the estimates used in the model is minimal. Assuming that the shapefile intersections are correct, and the estimated intersections in the model are completely in the wrong direction, then the maximum extra driving would be two times the distance between the estimate and the shapefile point. This effect would be worst in the case of the N1 to the N2 or from N2 to the N1. Worst case, it would be 14 km that the driver must drive extra to compensate for the wrong intersection. This would be 18 minutes of extra driving time, assuming a driving time of 46 km/h. Unfortunately, the connection between the N1 and N2 will be a crucial one because, as shown in Figure A.1, the intersection is in the middle of the road network connecting the upper part with the bottom part of the road network. The other intersections have a smaller difference. They are all under 1.8 km, which is less than 3 minutes of extra driving time for a truck. So it can be concluded that possible wrongly estimated intersections will only have a significant negative influence for the trucks crossing from the N1 to the N2 or from the N2 to the N1.

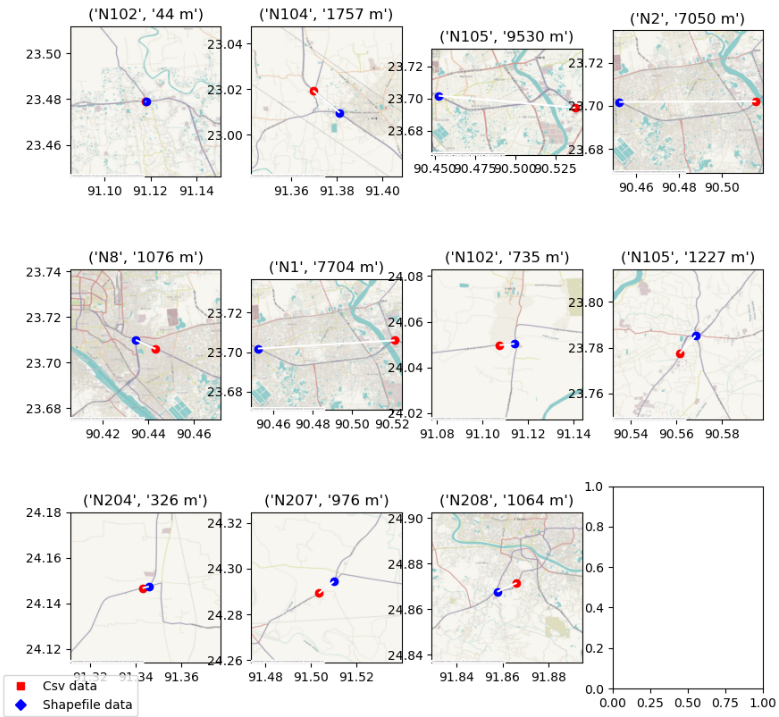

Figure B.1: Comparison estimated intersection points for the road data and shapefile data

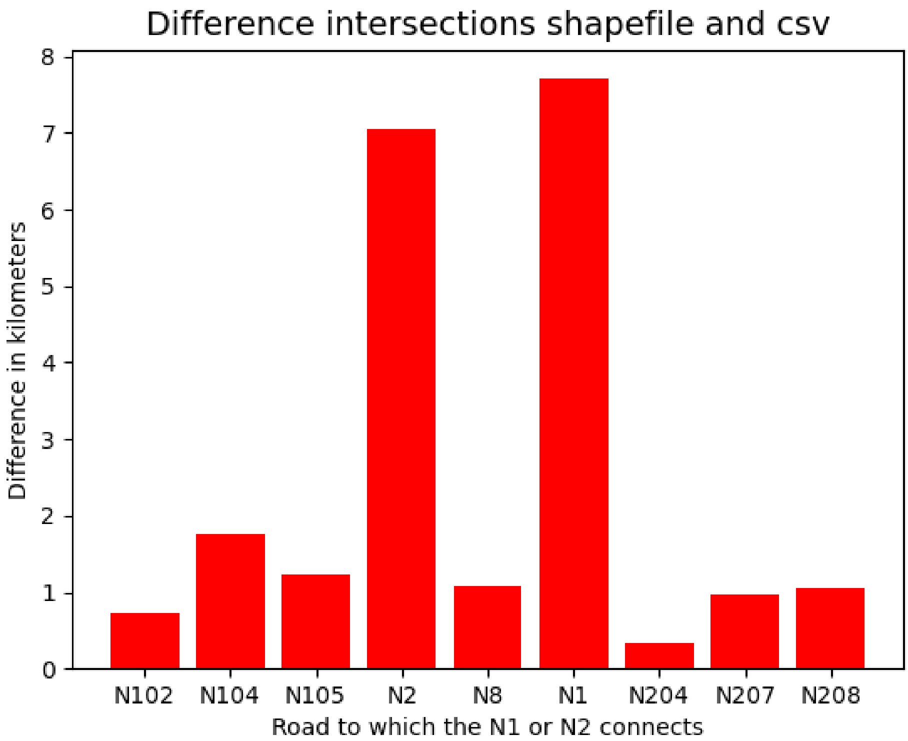

Figure B.2: Distance between points of intersections in km