UNIVERSITY OF THE
WITWATERSRAND,
JOHANNESBURG

School of Computer Science and Applied Mathematics

INVESTIGATING SHORTEST DISTANCE AND LOWEST CONGESTION ROUTING IN AN URBAN ROAD TRAFFIC NETWORK USING A MICROSCOPIC AND A MACROSCOPIC MODEL

Esmi Eunicé Conradie

Supervisor: Prof. Byron Jacobs

A dissertation submitted to the Faculty of Science, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Master of Science.

2020, Johannesburg

Declaration

I declare that this dissertation is my own, unaided work. It is being submitted for the Degree of Master of Science at the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination at any other University.

____

(Signature of candidate)

_____________ day of _____________ 20 _____________ at _____________

Glory to God. To my family who inspire me.

Acknowledgements

I would like to thank my supervisor Prof. Byron Jacobs for his invaluable guidance. It has been a sincere pleasure working with him. I would also like to extend my gratitude to the National Research Foundation for their financial support.

Abstract

Increasing road traffic [1] is costing time, money and increasing emissions [12-14]. Reducing congestion will reduce CO_2 emissions [5]. We explore the use of dynamic navigation with real-time traffic information and its impact on congestion. A microscopic and a macroscopic model of an urban road traffic network is used to investigate the impact of static shortest distance routing and dynamic routing with local and global knowledge of congestion. An agent-based simulation model and a fluid dynamic model is built for different road networks. The macroscopic model approach, combining methodologies from Angstmann et al. [6] with the viscous Burgers' equation and the numerical solution method from Wani and Thakar [7] simplifies having a partial differential equation model on a network structure based on a road system. As expected a decrease in travel time with dynamic routing where congestion could be avoided was observed on a large fabricated road network for the microscopic model. This suggests mobile navigation with real-time traffic information should improve congestion and travel times. Even though some of the same road networks were modelled in the microscopic and macroscopic models, the results are difficult to compare.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Mathematical modelling aims to describe the physical world in terms of mathematical equations and constructs, to help us gain a better understanding of the world around us and to, from these insights, hopefully improve the systems we have studied. In modelling vehicular traffic we try to address what is for many of us a familiar set of problems. How do I get to my destination? How do I get there on time? How can congestion be minimised? How can individuals impact traffic? These are some of the questions that have been studied since the 1950s [8-14]. Increasingly the focus is not only on minimising traffic and congestion to benefit individuals with less time spent on the road, but also on the positive environmental effect this is expected to have [5]. Traffic flow modelling is more relevant today than it has ever been.

Grote et al. [1] state that increased urbanisation is leading to increased road traffic volumes and therefore increased emissions of greenhouse gases, such as CO_2, and other pollutants. Grote et al. [1] explain that this is compounded by congestion and the resultant stop-and-go driving which further increases emissions. CO_2 emissions can be lowered by improving traffic operations, specifically through the reduction of congestion according to Barth and Boriboonsomsin [5]. Congestion also negatively impacts a number of other areas. As seen in Weisbrod et al. [2] it not only has a detrimental effect on air quality, but also causes a loss of personal time for individuals and higher costs and reduced service areas for businesses for workforce, suppliers and customers. They show a number of areas in which businesses are affected: higher travel costs due to longer travel times, higher vehicle operating costs, congestion related accident costs and increases in costs such as inventory cost due to delayed freight or service delivery. Not only businesses are experiencing the economic impact of congestion, the costs to individuals are also increased. The *2019 Urban Mobility Report*, a study conducted in the United States of America including 494 urban areas by Schrank et al. [3], estimates that the travel time of Americans in 2017 was raised by 8.8 billion hours due to congestion, with an additional 3.3 billion gallons of fuel purchased at a cost of 166 billion USD. These represent significant costs and Levy et al. [4] show a smaller, but still substantial monetary value linked to the public health impacts of congestion. Overall, the need to reduce congestion is a pressing problem.

What is congestion? Congestion as defined by Goodwin [15]: "Congestion is defined as the impedance vehicles impose on each other, due to the speed-flow relationship, in conditions where the use of a transport system approaches its capacity." Studying routing of vehicles can bring insights into the causes of congestion and the measures that can be taken to alleviate it. Combining and comparing the results from different modelling methodologies for the same problem should help us gain a deeper understanding of the system studied and its mechanics. The first approach to modelling traffic was a macroscopic one and one that is still used today: looking at the flow of traffic as if it were a fluid [8]. With advances in our computing ability many more possibilities have arisen, especially for microscopic traffic models. agent-based modelling is a natural fit for traffic modelling because vehicles, with their drivers, are intelligent entities within this paradigm.

The way we live today is heavily influenced by the information we seemingly always have readily at hand. And so our driving is also influenced by the availability of real-time traffic information and optimised routes. How does this inform the driving decisions we make? And what is the cumulative effect of that on the system?

1.2 Traffic flow modelling

Traffic flow modelling was introduced in 1955 by Lighthill and Whitham [8] who studied traffic flow using ideas from fluid dynamics. In their model the traffic flux is expressed as the product of the traffic density and velocity. Their model is a first order model, that is, it consists of one equation. This model assumes conservation of vehicles and that velocity is a function of density only [16]. The same model was also derived independently by Richards [17] in 1956. This is known as the Lighthill-Whitham-Richards model and it is the simplest partial differential equation model of traffic flow. Some shortcomings of this model are instantaneous changes in vehicle speed, the failure to describe platoon diffusion and the inability to cope with the instabilities of stop-and-go traffic. Also, it functions poorly for light traffic by not recognizing a distribution of desired velocities [10, 18]. Further, in the first order Lighthill-Whitham-Richards model traffic is restricted to equilibrium states as stated by Zhang and Jin [19].

To address these limitations of the first-order traffic model, a second order model was developed by Payne [9]. In Payne's model, the first equation is the same as the Lighthill-Whitham-Richards model. A second equation is added which defines acceleration [19]. This second equation includes relaxation, increasing or decreasing the velocity to reach a desired speed, convection, dependence of velocity on moving vehicles entering and exiting a given section of highway and anticipation - changing speed according to traffic conditions ahead [18]. Payne's model is a nonequilibrium model and according to Zhang and Jin [19] it is more realistic in this respect than the first order model of Lighthill-Whitham-Richards that preceded it.

Payne's second order model was heavily criticized by Daganzo [10]. Daganzo showed that this type of model could result in cars having negative speed, which causes wrong-way travel under certain conditions. According to Daganzo this is a failure of any continuum model of traffic which smooths out all discontinuities in density. Furthermore, the Payne model cannot describe queue end behaviour as well as the Lighthill-Whitham-Richards model [10]. Daganzo emphasises the difference between vehicles and fluids to show the shortcomings of higher order fluid dynamic models applied to traffic flow, highlighting first that vehicles have personalities rather than being uniform particles. Second, the width of a shock in traffic encompasses a few vehicles. Last, a fluid particle responds to stimuli in front of it and behind it and vehicles mostly to frontal stimuli. For this reason vehicles are anisotropic particles. Another feature of the Payne model is that the characteristic speed of a wave can become larger than the mean flow rate of the vehicles. If we interpret this characteristic speed as the distribution of information that means that vehicles receive and react to information from behind, violating the anisotropic quality ascribed to traffic flow [10, 18]. According to Zhang [20] this violation of the anisotropic property is the price most higher order models have to pay to be able to handle non-equilibrium phase transition. The first order model does not include non-equilibrium states.

These criticisms led to revisions of the second order model. Aw and Rascle [11] believe that they address the concerns of Daganzo [10] by replacing the space derivative in the Payne model with a convective derivative. They agree with Daganzo on the shortcomings of Payne's second order model, such as the violation of the anisotropic quality. In contrast to Payne's model, their second order model has anisotropic vehicles. Zhang [20] independently changed the momentum equation similarly to Aw and Rascle. In this way, through the choice of momentum equation a higher order model can preserve anisotropy [16]. Notably, Zhang's anisotropic second order model is derived from a car following model.

The above interpretation of the characteristic speed is not universally agreed upon. Helbing and Johansson [21] argue that the interpretation of the characteristic speed of the wave as the speed of information can sometimes be misleading and that having characteristic speeds faster than the average speed of the vehicles in the model does not mean there is a theoretical inconsistency of the model as stated by Daganzo [10]. However, Zhang [22] strongly disagrees with this stance, siding with the findings of Daganzo on the characteristic speeds.

While Daganzo [10] criticised the modelling of traffic as a fluid as seen above, Zhang [16] calls it natural to model dense traffic on a large time-space scale in this way as it displays features similar to those of fluids.

Traffic models may be categorised according to their level of detail [12] which leads to three categories of traffic flow models. The highest level of detail is found in microscopic models where vehicles are modelled individually. Also included in this category are submicroscopic models where parts of the vehicles are specified. Mesoscopic models aggregate the entities modelled into larger units while macroscopic models treat the system as one flow without distinction of entities. We will now briefly discuss each of these categories.

Microscopic models include car following models, sub- and microscopic simulation such as agent-based modelling, particle models and cellular automation. Particularly, agent-based modelling is becoming prevalent for the modelling of traffic and transport systems. In this study an agent-based model of traffic is developed. Chen and Chang [13] provide an overview of the applications of agent-based modelling in traffic. A reason for the rise of agent-based modelling for traffic applications is given by Mastio et al. [23], who state that as more road users have access to real-time traffic information the behaviour of the networks becomes harder to analyse and predict. They argue that agent-based modelling is therefore the most relevant way to accurately depict this system. For instance, Raney and Nagel [14] simulate traffic in Switzerland using an agent-based model where agents' route choices are influenced by their memory of route performance. Weyns et al. [24] model anticipatory vehicle routing where drivers collaboratively interpret the local traffic situation. Rothkranz [25] implements a system where agents are personal intelligent travelling assistant systems.

For many of the agent-based models the agents move across a road network. Routing of agents on a network can be accomplished using a path-finding algorithm. Several such path-finding algorithms exist and Dijkstra's shortest path algorithm is well-known and commonly used as mentioned by Tucnik et al. [26]. Dijkstra [27] proposed an algorithm to find the minimal distance between nodes in a network. In this way an agent can find the shortest route from its source node to its destination node. An overview of the applications of path-finding algorithms in general and Dijkstra's algorithm in particular is given by Tucnik et al. [26]. They compare Dijkstra's algorithm to others and analyse the performance of path-finding algorithms in a large-scale multi-agent environment. Dijkstra's algorithm is widely used in agent-based modelling of traffic systems. Some examples are given in the references [29-37].

The next category of traffic models is the mesoscopic models. Mesoscopic models include headway distribution models, cluster models, gas-kinetic continuum/multi-class/multi-lane models, generic gas-kinetic and platoon based multi-lane multi-class models. An overview of mesoscopic models is provided by Hoogendoorn and Bovy [12]. Mesoscopic models do not distinguish individual vehicles. In these models, the behaviour of the vehicles is described in aggregate probabilistic terms as small groups of vehicles represent traffic [12].

One type of mesoscopic models, headway distribution models, assume the time between the passage of successive vehicles, that is the headways, are identically distributed independent variables [12] and according to Li and Chen [38] they show the uncertainty inherent in drivers' car following behaviour while describing the stochasticity of traffic concisely.

Another type of mesoscopic models, cluster models study homogeneous clusters of vehicles defined by their velocity and the size of the cluster, which is usually dynamic [12].

Mesoscopic models also include gas-kinetic continuum models which were developed by considering the movement of vehicles as the motion of gas particles [39], in contrast to the classical approach of Lighthill-Whitham-Richards type models which consider this as a fluid. According to Hoogendoorn and Bovy [12] these models describe the reduced phase density which reflects the velocity distribution function of an individual vehicle and can be seen as a mesoscopic representation of the macroscopic traffic density. The first gas-kinetic traffic model was developed by Prigogine and Herman [40, 41]. Van Wageningen-Kessels et al. [39] tell us that this model is improved by Paveri-Fontana [42] who allows correlation between the behaviour of nearby vehicles. Further developments in gas-kinetic models include the multi-class model by Hoogendoorn and Bovy [43], a multi-lane Paveri-Fontana model by Helbing [21] and a platoon based model by Hoogendoorn [45].

The last category of traffic flow models is macroscopic models. Macroscopic models include systems of partial differential equations [12]. Models of the Lighthill-Whitham-Richards and Payne types discussed earlier, as well as their extensions and their semi-discrete and discrete forms form part of this class of models [12]. These categories, microscopic, mesoscopic and macroscopic, do not necessarily stand alone: models can be derived from models in other categories or used together or in hybrid models to study different aspects of the same problem [39]. Here, we will use a microscopic and a macroscopic model to study traffic on a network. Previously we mentioned that Zhang [20] derived their second order macroscopic model from a microscopic car following model. Some examples of hybrid models are mentioned next. In Aw et al. [46] and Moutari and Rascle [47] the authors combine the Aw-Rascle second order fluid dynamic traffic model with a microscopic version. Bourrel and Lesort [48] create a hybrid model from a vehicular model and a Lighthill-Whitham-Richards type model. Leclercq [49] presents a hybrid Lighthill-Whitham-Richards model encompassing the macroscopic model and a microscopic resolution of the same model. Hybrid models can be useful in situations were detailed results are needed for some parts of the system and less detailed results are sufficient for the rest [39]. Van Wageningen-Kessels et al. [39] mention the development of hybrid models as an opportunity for future work.

When comparing model categories we see there are advantages to each type. Microscopic models are useful for capturing individual behaviours, but are computationally intensive and the results may be difficult to interpret. In macroscopic models detail is lost, but according to Hoogendoorn and Bovy [12] the parameters are easier to validate and calibrate as there aren't that many parameters and they are easier to observe and measure than those used in other models. Comparatively, macroscopic models are efficient for large networks. De Marchi and Gerdts aim to combine agent-based microscopic modelling with macroscopic modelling on more complicated road networks such as a grid including a traffic circle [50]. Traditionally, traffic flow modelling considered a single road. These single road models can be applied to the edges of a network where these represent roads when one wants to model the traffic on a road network. However, according to Helbing et al. this is complicated to implement at the nodes of the network [21] which correspond to the intersections. Another approach to including a network in a macroscopic model is followed by Angstmann et al. [6]. When deriving equations governing the movement of particles from a continuous time random walk on a network, the network is incorporated by means of an adjacency matrix. This is the approach followed in my research to include the road network in the macroscopic model. Helbing et al. [44] model traffic flow on a network and use a permeability factor at nodes to represent traffic lights at intersections. This idea of a permeability factor is extended by Rarità et al. [51], where values of 0 or 1 represent the different phases of traffic lights and values in between these enable them to model other types of intersections. According to Bretti et al. [52] traffic flow at junctions is underdetermined even after the prescription of conservation laws for cars and they introduce a right of way parameter to uniquely solve Riemann problems at the junctions. They also stress the importance of traffic flow simulation as static models do not realistically model heavily congested urban road networks [52].

Another important consideration is raised by Kaddoura and Nagel [53], who show the importance of including traffic incidents in traffic flow modelling due to the impact thereof on the system and on travel times. They model incidents as a reduction in road capacity.

Many of the models discussed so far consider traffic as a fluid and, fluid properties can be interpreted for traffic flow. Diffusion in a traffic model can represent drivers looking at traffic ahead. Li [54] uses a non-linear convection dominant diffusion equation, constructed by adding a diffusion term to the Lighthill-Williams-Richards model, to model traffic on a uniform highway without junctions which is then solved numerically. Similarly, viscosity can be linked to driver memory, which is that the current speed depends on the previous speed because of the inertia of the vehicle and the reaction time of the driver. This was shown by Zhang [56] who derived a macroscopic fluid dynamic traffic model from a microscopic car following model which included driver memory. Their resultant macroscopic model included viscous effects.

This relationship between a microscopic and a macroscopic traffic model of the same system is explored by Sun et al. [55]. They derive a macroscopic continuum traffic flow model from a microscopic car-following model and show this macroscopic model is more realistic and meaningful than one not derived from a microscopic model.

An equation that was proposed by Bateman [57] in relation to the motion of a viscous fluid and later used by Burgers [58] when he was studying turbulence is now commonly referred to as the viscous Burgers' equation and is given by

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = v\frac{\partial^2 u}{\partial x^2}, \tag{1.1}$$

where t is the temporal and x the spatial variable and v the viscosity coefficient. The applicability of the viscous Burgers' equation to traffic flow problems is shown by Binatari [59]. Velasco and Saavedra [60] derive the

Burgers' equation for traffic flow from the kinetic Reduced Paveri-Fontana equation. Binatari [59] asserts that solving the viscous Burgers' equation for traffic flow problems exactly is complicated and merits the use of an alternative solution method. One such alternative solution method is the Crank-Nicolson type finite difference scheme proposed by Wani and Thakar [7] to solve the viscous Burgers' equation. In our study the viscous Burger's equation is used to represent traffic and it is solved using the aforementioned scheme.

1.3 Aim

The aim of this study is to compare the representation of an urban road network and its associated traffic flow and analyse the effect of prescribing different routing objectives using both a microscopic agent-based model and a macroscopic model based on fluid dynamics.

1.4 Research questions

We aim to answer the following questions:

- How can we capture drivers following the least congested route to their destination based on their knowledge of upstream congestion?
- How does the difference in routing influence individual travelling times and the overall congestion of the system?

- How can we represent routing of vehicles on a network in a macroscopic fluid dynamic traffic network model?
- How do the results of the macroscopic and microscopic models of the urban traffic network compare?

Chapter 2

Microscopic model for traffic flow

2.1 Introduction and background

In this chapter we discuss the assumptions made when we built the model and the methodology we used to implement it. Next, we describe the model and outline the various routing options used. We give inputs, data used, and outputs from the model. The results from these outputs and their interpretation is discussed in Chapter 4.

2.2 Assumptions

"A model which took account of all the variation of reality would be of no more use than a map at the scale of one to one." - Joan Robinson, Essays in the Theory of Economic Growth, 1962

Assumptions are necessary to reduce the model complexity and define working parameters. The assumptions made for the agent-based model are detailed in this section.

2.2.1 Road network

The assumptions made regarding the road network are the following:

- All roads in the network are considered to be single lane roads and no turning lanes are included.
- Conservation of vehicles is assumed on each road.
- The capacity of a road is calculated to be the length of the road divided by the length of a vehicle and rounded down to the nearest integer.
- The road network is modelled as a data structure and consists of a list of nodes and a list of edges. The relationship between the nodes and edge is stored within the edges.
- Intersections are modelled as different instances of the same type with their own attributes, such as intersection type, and behaviours, such as

the changing of traffic lights.

- Two types of intersections are present in the model, namely non-phase traffic lights and all-way stops. Small roundabouts are represented by all-way stops as this is what they are reduced to in heavier traffic. The data used does not consistently distinguish intersections where only drivers approaching from one direction have to stop, therefore the decision was made to treat all intersections other than traffic lights as all-way stops.

2.2.2 Vehicles

Vehicles were modelled with the following assumptions:

- Vehicles will be modelled as agents.
- A finite number of vehicles are considered.
- It is assumed that vehicles all have the same length which includes the minimum following distance for modelling purposes. The exact length used and other specific values are given in Section 2.6. Thus a uniform fleet is modelled and no allowance is made for different vehicle types such as motorcycles.

2.2.3 Driving behaviour

The following assumptions characterise driving behaviour in the model:

- No overtaking is modelled.
- No public transport modes are included in this model.
- If possible vehicles will move at the maximum speed prescribed for the road.
- A set minimum following distance is prescribed and captured in the length of the vehicles.
- Vehicles are assigned their destination at the start of their trip and this destination remains unchanged for the duration of the trip. This is the same irrespective of the routing methodology employed. The routing can be static or dynamic, but the destination remains unchanged.

2.2.4 Simulation

There are only a few assumptions pertaining to the simulation:

- The simulation uses a time step length of one second.
- The model runs have a simulated time duration in the order of hours.

2.2.5 General

Only one general assumption was made:

- The International System of Units (SI) is used for time and length quantities.

2.3 Modelling methodology

In this section we outline the approach taken in modelling the road traffic system under consideration.

The urban road network is represented in the model as a network where the nodes of the network are the intersections and the edges make up the roads. Each edge of the network is considered to be a stretch of road where the number of vehicles is conserved; we explicitly impose the restriction that no vehicle may be removed from the stretch of road at any point except at the nodes. There are only single lane roads in this network and no overtaking is considered. Also, every edge has a maximum speed and capacity.

A finite number of vehicles is simulated travelling from a certain node to another to mimic people on their commute home from work. Nodes are considered to be in business or residential areas. Homes and businesses are along roads, but vehicles can only leave roads at intersections. This is a simplification made for modelling purposes. These source and destination nodes may all be the same node or clusters of neighbouring nodes.

Vehicles are modelled as agents in the agent-based model and each vehicle is assigned its own route. Each agent has its own departure and destination nodes. This route is the shortest route in terms of distance between each pair of departure and destination nodes and is calculated beforehand using Dijkstra's algorithm. Intersections also have associated agents to model traffic lights schedules and priorities at intersections which are specified beforehand.

In this chapter we consider two different scenarios of routing dynamics. The first, mentioned in the preceding paragraph, considers static shortest distance routing where the second introduces a recalculation of the route at each intersection including the congestion as well as the distance. We expect that the difference in dynamics will result in faster travel times for the second scenario. This will likely not be the globally optimal travel time since the distance travelled still plays a big role, and the distance travelled could increase since avoiding congestion may result in longer routes. However, there should be an improvement in how long it takes a vehicle to reach a certain destination from a certain source when congestion is factored into the routing rather than just taking the shortest distance route while disregarding traffic conditions. Future work could test more variations on the routing and perhaps find a way to ensure optimal travel time.

In the second scenario the routing of the agents becomes more dynamic with the agents evaluating the densities on the road edges ahead at each node while still taking into account the distance to their own final destinations. The dynamic routing is done in two ways, one where the agent has a global view of congestion, that is, the agent knows how congested every road in the network is and the other where the agent has only a local view of congestion. This local view is where the agent only knows the congestion of the roads connected to the intersection it is currently at. The first view corresponds to someone using a navigation system where congestion is taken into account in the routing and the second to someone whose navigation doesn't take congestion into account, but can see how busy the road is ahead and lets that influence their route. We expect routing with a global view to outperform routing with a local view as decisions made on congestion in the immediate vicinity of the vehicle can lead the vehicle into areas where there is even more congestion. Of course, this is also dependent on the size of the network. When there are only a few roads on the route, we expect that the benefits of knowing the global congestion rather than the local congestion will become negligible. We also expect the dynamic routing options to give better results than the static routing in high congestion scenarios and similar results when there is little congestion in the network.

Object orientated programming languages are particularly suitable for agent-based models, as such Java is used to develop this model.

The road network and the agents representing the vehicles with their routes and routing choices are simulated over a period of time calculating the necessary values at every time step.

2.4 Base model

The agent-based model was developed in the Java programming language using the BlueJ development environment. It comprises six classes: *Edge*, *Vertex*, *Graph*, *DijkstraAlgorithm*, *ABSimulation* and *Vehicle*. Information pertaining to the first five classes is contained in the appendix and the *Vehicle* class will be discussed in this section. Figure 2.1 illustrates the relationships between the various classes.

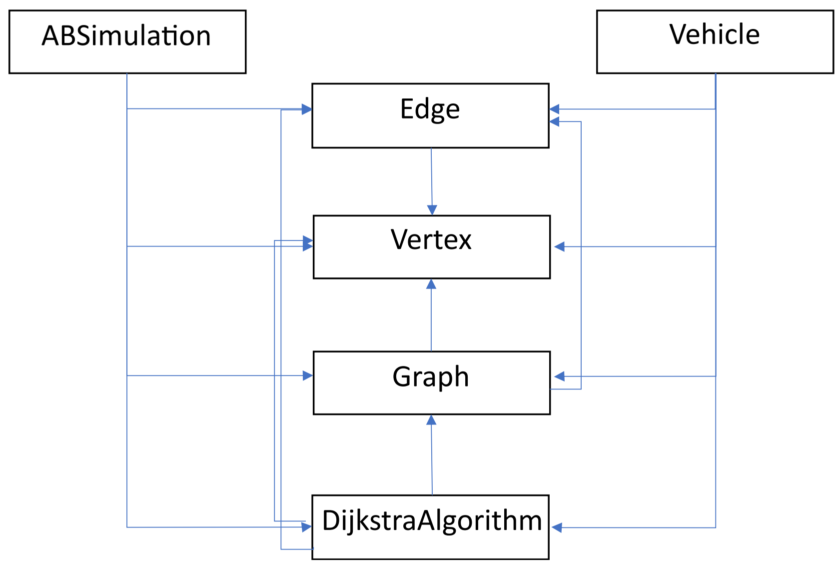

Figure 2.1: A representation of the classes in the development environment.

2.4.1 *Vehicle* class

The *Vehicle* class has variables to keep track of each vehicle's properties. These include information about its route, current position, identity, speed, the type of road it is currently on, its time in the system and information about the graph and queues. The class also includes a method to update its location, time and speed variables at every time step. The logic employed in this method is detailed in the next few paragraphs. In Figure 2.2 the queue operations are shown.

If the vehicle has not finished its route yet, and the current time is greater than its arrival time, it will be updated. Firstly, its travel time is increased by the value of the time step. The next step is determined by the vehicle's location relative to intersections. If the vehicle is moving away from an intersection it should be removed from the queue at its departure node.

If it is at an intersection with its *location* set to zero, that is, it has arrived at the intersection, but not moved into the next road, it must be ascertained whether it has finished its route now. If it has, the boolean variable *Finished* is set to *true* and it is removed from all queues. If its current intersection is a multi-way stop, or a single-way stop where the vehicle is at a stop, it must determine whether it is the first vehicle in the queue at that intersection. If the current intersection is a traffic light and the vehicle is in the stop direction, it is set to not being first in the queue for that intersection and removed from the queue associated with the non-stop direction if necessary. In the non-stop direction of a traffic light controlled intersection, the vehicle is set to be first in the queue and removed from the queue associated with the stop direction if necessary.

If the vehicle is at an intersection with its *location* equal to the length of its current road, first check whether its current destination node is the same as its final destination node. If it is, it is set to *Finished* and removed from all queues. If they are not the same, get its next road and its nodes. If its next road isn't full, set the *CurrentDestinationNode* as the *CurrentDepartureNode* and the *location* to zero. Set the new *CurrentDestinationNode* and new

CurrentRoad and move to the correct road queue. If the vehicle is at a stop at an intersection, that is, at a multi-way stop intersection or at a single-way stop intersection in the stop direction, it is added to the end of the intersection's stop queue. Then, if it is first in this queue, *FirstInQueue* is set to true, else it is set to false. Having arrived at a single-way stop in the non-stop direction, *FirstInQueue* will be set to true and it will be removed from the non-stop queue. If the intersection the vehicle has arrived at is a traffic light and it is in the non-stop direction, the same steps are taken as with the non-stop direction at the single-way stop. In the stop direction of the traffic light *FirstInQueue* is set false and the vehicle is removed from the non-stop queue. The arrival time at the intersection is set to the current time.

If the vehicle has not finished its route and its *location* is not zero and if it is moving in a non-stop direction and it is within ten units from the end of its road, it is added to the end of the non-stop queue.

After all this has been set, the next step is to determine whether or not the vehicle will move forward. First it is assumed that the vehicle will move and *VehicleWillMove* is set to *true*. If there is no space in front of the vehicle for it to move, this will be set false. This includes cases where the *location* is zero, but *FirstInQueue* is *false*. The vehicle will not move if it has arrived at an intersection in the stop direction within the last five time steps. Next, the speed of the vehicle is assigned. If the vehicle will not move its *CurrentSpeed* is set to zero. If there is another vehicle on the same road in the same direction within ten units ahead of the vehicle, travelling at a lower speed, the speed is determined according to the following equation:

$$CurrentSpeed = CurrentSpeed - \frac{CurrentSpeed - \text{Speed of other vehicle}}{\text{Location of other vehicle} - Location}. \tag{2.1}$$

If neither of these adjustments to the speed need to be made, it defaults to the speed limit. When a vehicle approaches and intersection where it will have to stop, such as driving towards and all-way stop or a red traffic light, it will slow down. When it is within ten units of the intersection, but not at the intersection yet, its current speed will be adapted in the following manner:

$$CurrentSpeed = \frac{CurrentSpeed \times (RoadLength - Location)}{10}. \tag{2.2}$$

The vehicle's location is updated to the location plus the distance it will move in the next time step except if this value is greater than the length of the road it is currently on. In this case the location is set to the road's length.

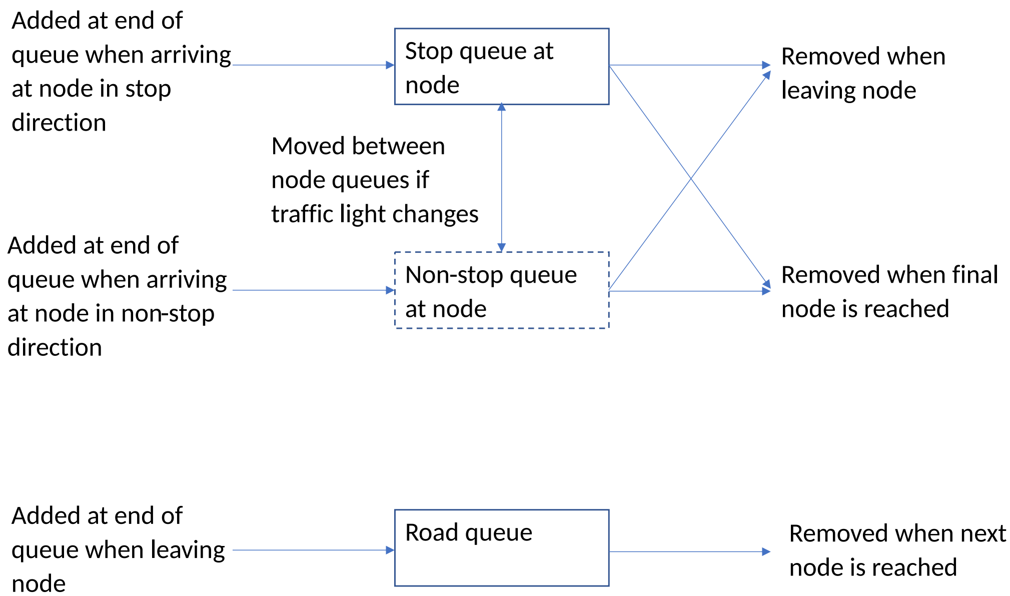

Figure 2.2: A diagram of the queues in this class and their operation.

2.5 Alternate routing

Different routing options were tested using the model. Changes made to the model to implement this are discussed in this section. In the base case of the model, routes are calculated before agents are created and remain static throughout the model run. These routes are purely based on the shortest distance from the source node to the destination node. In the other scenarios discussed in this section, the assigned route is examined and possibly changed at each intersection. The routing algorithm now not only takes into account the shortest distance to the destination, but also the congestion on the network.

2.5.1 Dynamic routing based on global knowledge of network congestion

Initially a shortest distance based route is still calculated from the initial departure node to the final destination node. At the start of the *updateVehicleLocationTimeAndSpeed* method of the *Vehicle* class an updated list of *edges* called *adaptededges* is generated. The edges in this list have a weight calculated as follows:

$$\text{Weight of } \textit{adaptededge} = (\text{Weight of } \textit{edge}) \times (1 + \text{Congestion of } \textit{edge}). \quad (2.3)$$

It is then ascertained whether for this vehicle, if it has not finished its route, it has arrived at an intersection during the previous time step. If this is the case, its route is recalculated. Again, the *DijkstraAlgorithm* class is utilised, but instead of edge weights that are merely the length of the edges, it is now presented instead with an adapted edge weight based on the length and penalised according to the congestion of the edge. The algorithm in this section summarises this logic.

for *number of edges* **do**
  weight of *adaptededge* = (weight of edge)× (1 + congestion of edge);
end
if *not finished with route* **then**
  if *arrived at node in previous time step* **then**
    recalculate route using *DijkstraAlgorithm* and *adaptededges*;
  end
end

Algorithm 1: Logic added for dynamic routing.

2.5.2 Dynamic routing based on local knowledge of network congestion

This scenario is handled very similarly to the one outlined above where an agent has global knowledge of congestion. Here it is however assumed that the agent only knows about the congestion on certain roads in its vicinity. This vicinity has been restricted to roads originating from the intersection it has just arrived at due to the scale of the maps implemented.

2.6 Data and inputs

The inputs into the model will be discussed in this section, particularly the original inputs. The various inputs used to study different scenarios and the difference in results this produces will be detailed in the chapter discussing the model results. Inputs concerning the vehicles include the number of agents to be simulated and their interarrival times, as well as the average vehicle length plus the following distance. Other simulation specific input is the duration of a simulation run. A time interval for the traffic light changes as well as their initial states have to be set.

Other inputs deal with the map on which the simulation is run. The number of nodes and number of edges have to be specified. Each node has an identifying number and a type: multi-way stop, single-way stop or traffic light. Every edge has an identifying number, a source node, a destination node, a weight or length and a type: stop direction or non-stop direction.

The map data used in the different simulation runs is fabricated in some instances and real in others.

2.6.1 Small fabricated map

Initially a small test map was devised consisting of six nodes and seven bidirectional roads, therefore 14 edges. A car length of nine metres was used. We simulated 2500 agents over a period of 3000 seconds with random interarrival times - the maximum time between two consecutive arrivals was three seconds. A list of arrival times was constructed using the interarrival times, arrival times were then assigned to agents. Agents can only be active in the model when the current model time is greater than their arrival times. A sufficiently large number of agents were needed to cause some congestion in the network to test the impact of routing while taking congestion into account and through experimentation it was found that this can be achieved with 2500 agents. Too high a number of agents coupled with too short interarrival times lead to agents not being able to enter the network timeously therefore increasing the number of agents only increases network congestion up to a point whereafter the agent simply pile up waiting to enter the system. With too low a number of agents, there was no congestion. To see the effect of the number of agents on the system, the number of agents was varied and then the number of cars per road over time was compared to the capacity of each road. At the chosen number the number of cars on a road would increase, but also decrease. The simulation duration is dependent on the number of agents and chosen in a way that all agents can reach their destinations. Sources and destinations of agents were also selected randomly while ensuring that the source and destination nodes are not the same. The selection was done randomly as the map is too small to accommodate clustering of source and destination nodes and to ensure traffic is distributed across the network. All the stops were initially treated as multi-way stops. The map used is shown in Figure 2.3:

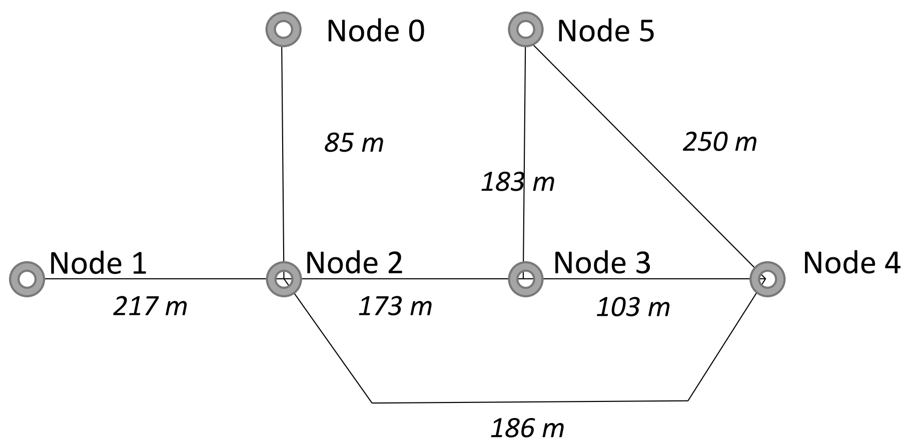

Figure 2.3: A small test map.

The following adjacency matrix (2.4) shows the distances between the nodes in this map. Blank elements in the matix indicate no connection between the nodes. All lengths of edges are in metres. In this map all roads are bi-directional.

$$A = \begin{bmatrix} 0 & & 85 & & & & \\ & 0 & 217 & & & & \\ 85 & 217 & 0 & 173 & 186 & & \\ & & 173 & 0 & 103 & 183 \\ & & 186 & 103 & 0 & 250 \\ & & & 183 & 250 & 0 \end{bmatrix}. \tag{2.4}$$

2.6.2 Large fabricated map

A large network was constructed in the form of a $n \times n$ grid. This grid consists of $(n+1)^2$ nodes and $2n(n+1)$ bidirectional edges. n was chosen as $n = 24$, resulting in 625 nodes and 2400 unidirectional edges. This grid is illustrated in Figure 2.4. Each edge has a length of 200 metres. We chose a uniform length for the edges as this means that routes can be changed more easily when there is congestion without influencing the distance of the route by much or at all. Therefore this provides an environment where congestion can dissipate across the network when it is included in routing and the effect of this can then be seen in the resultant travel times.

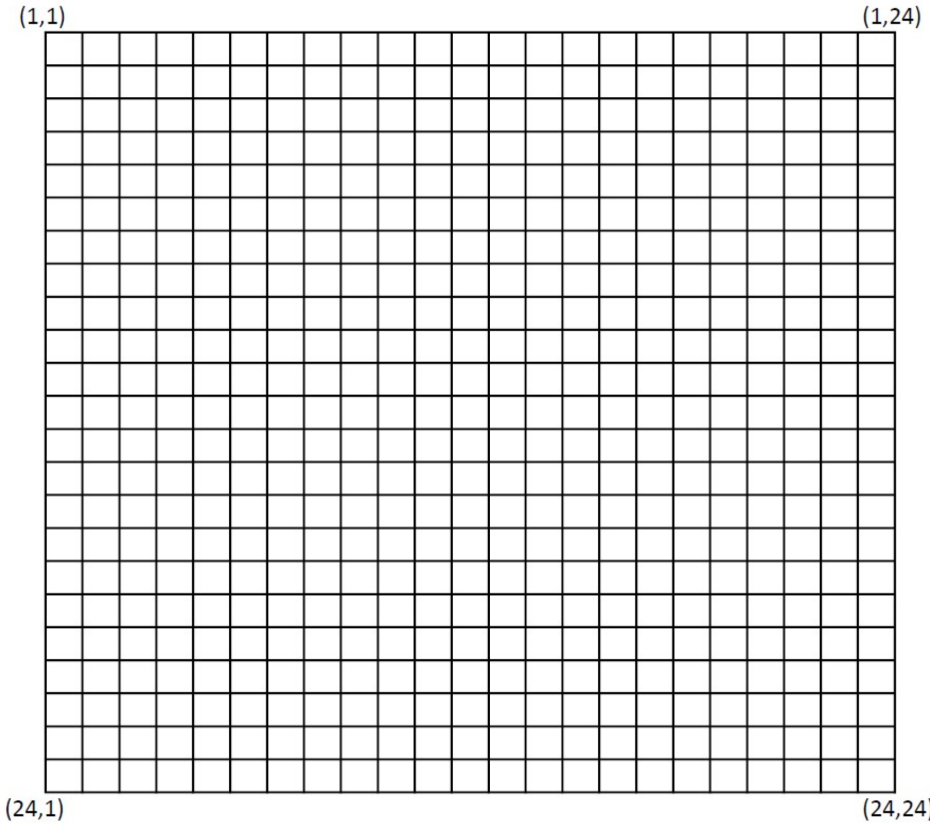

Figure 2.4: A large grid map.

At first the sources and destinations were chosen at random, while making sure that the selected source and destination are not the same node. There were 25000 agents which had random interarrival times, with at most two seconds between arrivals. The simulation ran for 3000 seconds. As discussed in 2.6.1 the number of agents and the interarrival time were chosen to cause congestion on the network and the simulation duration is dependent on the number of agents. The length of an average vehicle remained nine metres.

All stops were set to multi-way stops. In later runs the source and destination nodes were clustered to reflect commuters more accurately. Figure 2.5 shows this clustering.

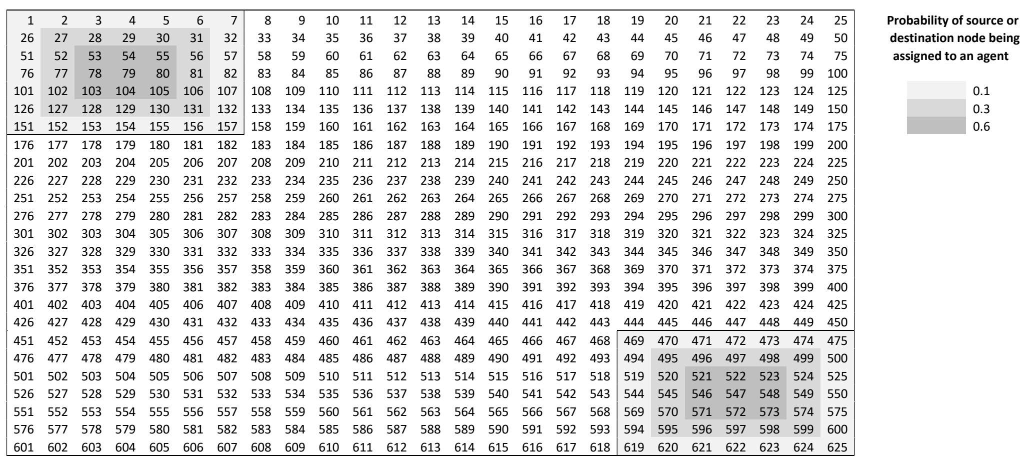

Figure 2.5: Clustering source and destination nodes in a large grid map.

As shown in Figure 2.5, the clustered nodes were divided into three categories with different probabilities of being assigned as source or destination. Nodes coloured in the lightest grey have a probability of 0.1 of being chosen, the nodes in the medium grey have a probability of 0.3 of being chosen and the nodes coloured in the dark grey have a probability of 0.6 of being chosen. This is implemented as a cumulative distribution in the model. A random number between zero and one is generated using Java's built-in random number generator. Using this number one of the sets of nodes is selected. The node that is finally assigned is chosen randomly from this set with an equal probability. Lower numbered nodes were used to choose source nodes from and higher numbered nodes to chose destination nodes from.

2.6.3 Small real map

Next a small real map was built into the model to test the model using real-world information. The real-world map data was obtained from Open Street Map (openstreetmap.org). Open Street Map is an open data initiative [61] where non-expert volunteers freely generate map data through uploaded GPS traces. The data is licensed under the Open Database License. Credit for the data goes to the Open Street Map Foundation and its contributors. It was decided to use a familiar area so that the accuracy of the data could be verified. The area that was chosen is shown in Figure 2.6.

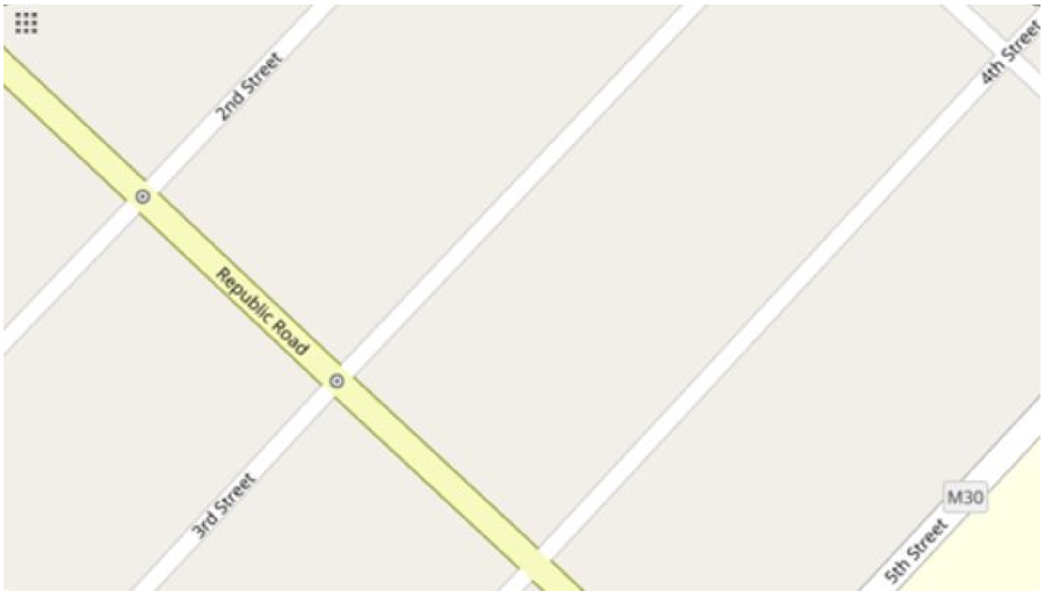

Figure 2.6: A small real map.

This map shows four parallel streets, 2nd Street, 3rd Street, 4th Street and 5th Street. Republic Road is shown as crossing 2nd Street, 3rd Street and 4th Street. A small section of 2nd Avenue is shown crossing 4th Street. The intersections of Republic Road with 2nd Street and 3rd Street are traffic circles. At the intersection of 4th Street and Republic Road there are stop signs in 4th Street and not in Republic Road. The intersection of 4th Street and 2nd Avenue also has stop signs in 4th Street and not in 2nd Avenue. This area was manually selected and exported on Open Street Map. This resulted in data for a slightly different area than the one shown in Figure 2.6 above - the information contained in the exported file was of not exactly the same area than the selected one. Plotting the points and roads contained in the file gives the following network:

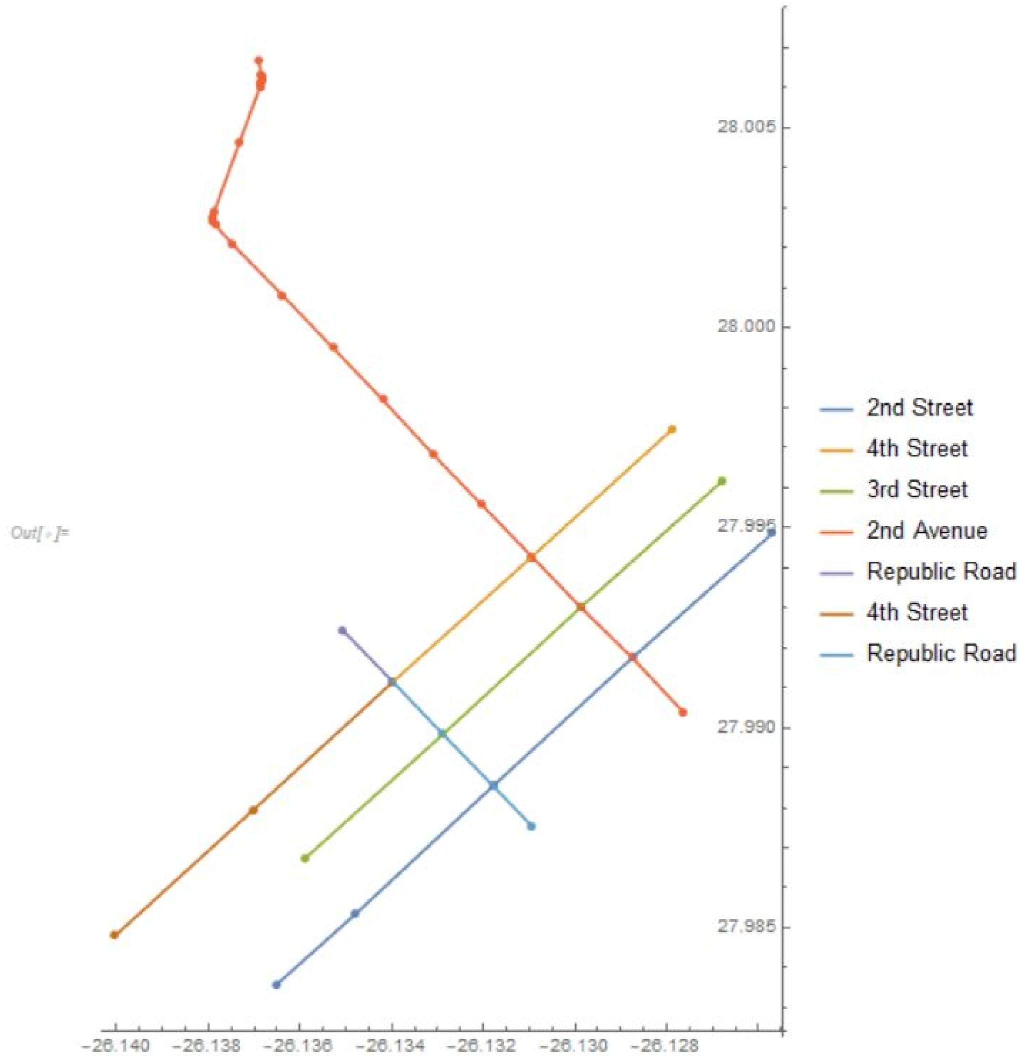

Figure 2.7: A plot of the points and roads in the data obtained for the small real map.

Data exported from Open Street Map comes in the form of an XML (eXtensible Markup Language) formatted file with a .osm file extension. This file contains data pertaining to points known as nodes, ways and relations.

Any point with coordinates is a node. A way is a collection of nodes. Ways can describe roads or pieces thereof, footpaths and the outlines of buildings among other things. Relations show relationships between nodes and ways to capture things like routes and turn restrictions [62]. Only nodes and ways were used to construct the map in the model. In the file a node has an identification number, coordinates, information about when and by whom it was uploaded and updated. Extra information is given through tags. These tags are entered by the various individuals who contribute to Open Street Map and this leads to the integrity of the data being inconsistent. There is also a lack of standardisation of data which makes processing the data more difficult. A way has an identification number, a list of ordered nodes and information about its creation and updated state, as well as tags. Elements of interest for this study are tagged "highway" which means they describe roads used by motor vehicles. Other important tags were for nodes that were tagged "traffic_signals", "mini_roundabout" and "stop". What is needed for the agent-based model from the .osm file are the nodes corresponding to network nodes in the model, that is, intersections and terminating nodes with the type of intersection if applicable, the node's coordinates and which other nodes it is connected to. From this information the necessary inputs to the model can be inferred. To gain this information the .osm file was processed using Wolfram Mathematica.

For this particular area it was noted that the intersections with traffic lights and traffic circles were marked as such in the Open Street Map data, but for the intersections where there are only stop signs in one of the crossing streets, there are no tags indicating the type of intersections. This led to the decision to treat all intersections without traffic lights as all-way stops. The file was imported into Mathematica. First the nodes and ways were found, with their identification numbers and types and the node references for each way. Intersection nodes were identified as belonging to more than one way. Terminating nodes were found at the ends of ways. Sets of connected intersection nodes were extracted. For sets of more than two nodes the nodes of the set were ordered according to their way's ordering. These sets were then broken up into pairs. Pairs of terminating nodes with their neighbouring intersection nodes were also constructed. These pairs were then translated into coordinate pairs. From the coordinate pairs the length of the edges could be determined using Mathematica's GeoDistance function. This then gives all the map inputs needed for the model. This process is shown in Figure 2.8. The model was run with 2500 agents with random interarrival times over a period of 3000 seconds. There was a maximum of three seconds between arrivals. This number of agents yielded the necessary amount of congestion determined as explained in Section 2.6.1.

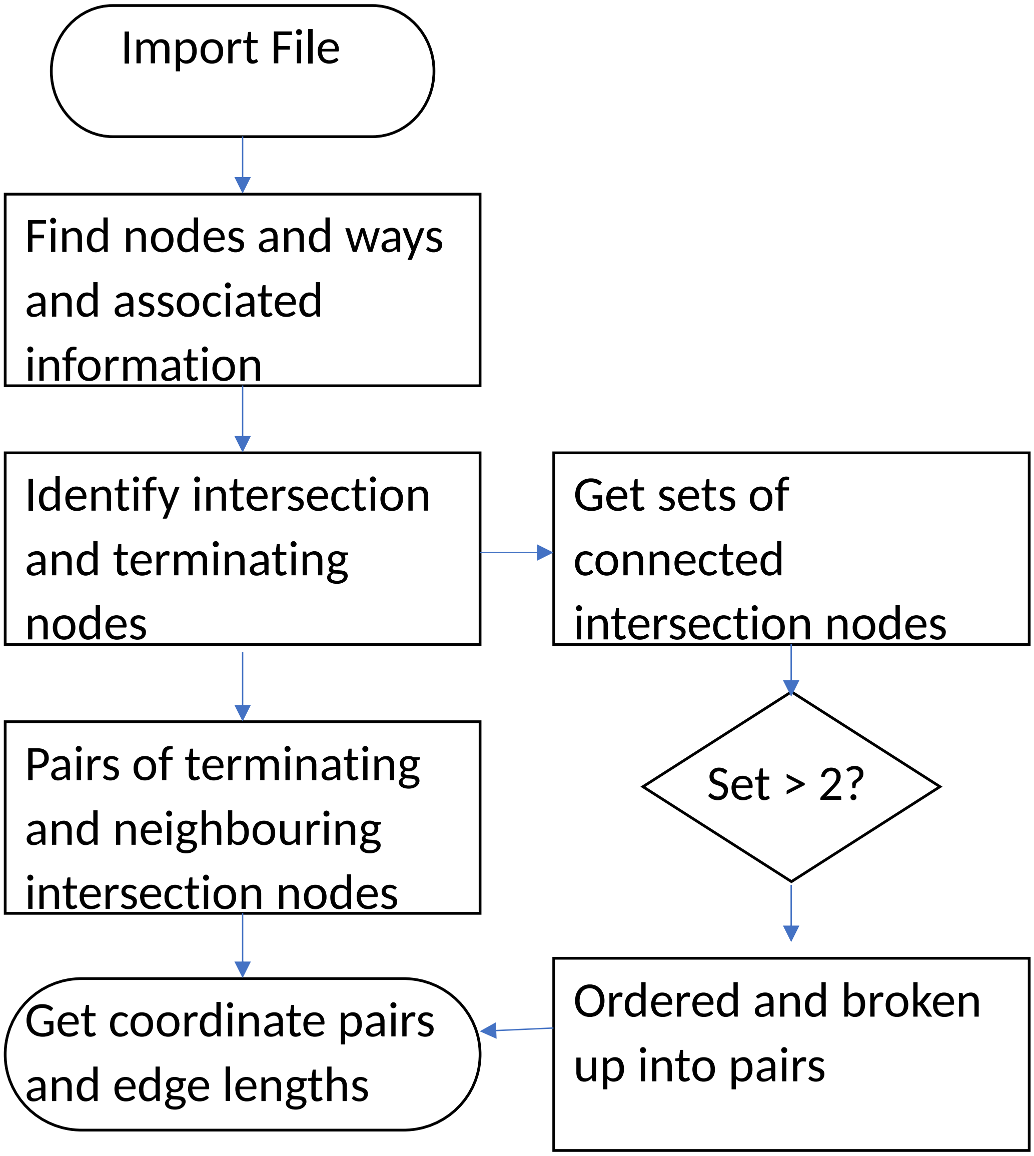

Figure 2.8: A flow chart showing the data processing.

2.7 Output

Output from the model is written out to four separate text files. The file "Vehicles.txt" contains information about the different vehicles in the model. Information relating to the different roads in the model is written to "Roads.txt". "TrafficLights.txt" contains the data pertaining to traffic lights. These files are updated at every time step. The information in the text file "Routes.txt" is written out at the beginning of the run, before any agents are created. It details the initial shortest distance route calculated for each agent, based on its source and destination nodes. The data contained in these text files are analysed in R and the results of this analysis presented in Chapter 4. The exact data written out is presented in Table 2.1 below.

2.8 Conclusion

This chapter discussed the way in which the agent-based model was envisaged and realized. Its results will be discussed together with the results from the partial differential equation model described in Chapter 3. In future work the agent-based model could be expanded to include a large network based on a real map. Data for this map could be obtained from Open Street Maps and would have to undergo thorough cleaning and validation due to the inconsistencies in the data discussed in Section 2.6.3.

Table 2.1: Content of Output Text Files.

Vehicles.txt	**Roads.txt**	**TrafficLights.txt**	**Routes.txt**
Current model time	Current model time	Current model time	Route Number
Vehicle identification number	Road identification number	Intersection type	Ordered list of nodes in route
Whether route is finished	Road length	End node of current road	
Initial departure node	Road capacity	Edge type	
Final destination node	Road queue size		
Current departure node	Identification numbers of vehicles on road		
Current destination node			
Current road			
Location on road			
Travel time			
Current speed			
Activation time			

Chapter 3

Macroscopic model

3.1 Introduction and background

This chapter considers a macroscopic model for the description of traffic flow. This is the classical approach and has been taken by many different researchers [12] since the the first models by Lighthill, Whitham and Richards [8, 17]. Macroscopic models are still prevalent today due to their efficiency, especially for larger networks [12]. Here we use Burgers' equation similarly to Binatari [59], who shows how Burgers' equation can be derived for traffic flow. Also, Velasco and Saavedra [60] explain the relationship between Burgers' equation and the Lighthill-Whitham-Richards model and they derive Burgers' equation from the Reduced Paveri-Fontana equation found in Paveri-Fontana [42]. This indicated that it is a viable approach due to its strong links with well-known and commonly used traffic models. After stating the assumptions made and the methodology used to create the macroscopic traffic model, based on a partial differential equation, this chapter will also relate how the road networks were approximated for use in the model and detail the derivation of the partial differential equations. Next, the scheme for the solution of the equations is given and the inputs and outputs are discussed. We use the results as a point of comparison with the agent-based model and expect they will support the findings from the microscopic agent-based model. All these results are given in the following chapter.

3.2 Assumptions

This section contains the assumptions for the macroscopic model. These include assumptions regarding the road network and the vehicles modelled, as well as initial conditions.

3.2.1 Road network

Assumptions regarding the road network follow:

- Conservation of vehicles is assumed on each road.
- The capacity of a road is not taken into account, as such we do not limit the number of vehicles on a specific road at a given time.
- The road network is represented by an adjacency matrix.

- No differentiation is made between intersections in the sense that all intersections are treated the same regardless of their type, that is, whether they are traffic lights, all-way stops, roundabouts or single direction stops.

3.2.2 Vehicles

Vehicles were modelled with the following assumptions:

- Vehicles will be modelled as particles.
- Vehicles are not considered individually, and is viewed as the density of vehicles at a particular point.
- A uniform fleet is modelled and no allowance is made for different vehicle types such as motorcycles as no distinction between vehicles is made.

3.2.3 Driving behaviour

Assumptions with regards to the driving behaviour in this model:

- No public transport modes are included.
- The speed of the vehicles will be influenced by the density.

3.2.4 Initial conditions

Some arbitrary values for the density are assigned to the nodes at time 0.

3.3 Modelling methodology

An urban road network is represented as a network where the intersections correspond to nodes and the road sections between intersections to edges of the network. Conservation of vehicles on the edges of the network is assumed. An adjacency matrix of the network is used to include the chosen road network in the model. The vehicles are considered to be particles moving across the network. In the microscopic model each vehicle was an agent and here each vehicle is a particle. The macroscopic model views the traffic as the aggregate movement of the particles and does not distinguish individual particles. The viscous Burgers' equation is chosen as the partial differential equation governing the movement of the particles. When the constitutive equation for flux in the Lighthill-Whitham-Richards model is modified by introducing a diffusion coefficient, this leads to the viscous Burgers' equation [59, 60]. Velasco and Saavedra [60] also derive the viscous Burgers' equation from the Reduced Paveri-Fontana kinetic equation [42] which is a well-known model for traffic flow. The viscous Burgers' equation is a non-linear equation and has both diffusion and advection. The equation is given as

$$u_t + uu_x = Du_{xx}, \tag{3.1}$$

where u in this case represents the density of the particles, or traffic, and x is the spatial and t the temporal variable. D is the diffusion coefficient. The Laplacian u_{xx} will be described through the use of an adjacency matrix based on the network as was done in Angstmann et al. [6]. In this way the network need not be uniform as the discretisation lattice is not uniform as per usual, and nodes can have multiple connections to other nodes and this number of connections can vary from node to node. Diffusion mimics the dynamic lowest congestion routing in the microscopic model where vehicles have a local knowledge of traffic as the particles move to areas of lower concentration. Diffusion is the spread of particles from a higher to a lower concentration. When diffusion governs the movements of the vehicles, they will move to an area with a lower concentration of vehicles. Therefore, if an edge the vehicle can move to has the same or a greater concentration of vehicles, that is, is similarly or more congested, the vehicle will favour another path. In the agent-based model with dynamic routing based on local knowledge of congestion, a vehicle will also choose the least congested of the edges to which it can proceed. In order to render the model in equation (3.1) well posed, we are required to impose initial and boundary conditions. Although the spatial domain is represented by a network, with non uniform meshing, we impose zero-flux boundary conditions on the domain. This is a modelling device used to enforce the conservation of vehicles in the domain, that is no vehicles may leave the network. An appropriate distribution of vehicles across the spatial domain is selected as the initial condition. As the viscous Burgers' equation is parabolic, a Crank-Nicolson type finite difference scheme is used to solve the system. This scheme was proposed by Wani and Thakar [7]. The system is then solved iteratively for the density in MATLAB as it has strong support for matrix operations.

3.4 Modelling the network

An adjacency matrix is generated to describe the network on which the propagation takes place. This adjacency matrix plays a part in the formulation of the network Laplacian as will be seen in Equation (3.9).

3.4.1 Small fabricated map

The small fabricated map discussed in Section 2.6.3 can be viewed as is shown in Figure 3.1:

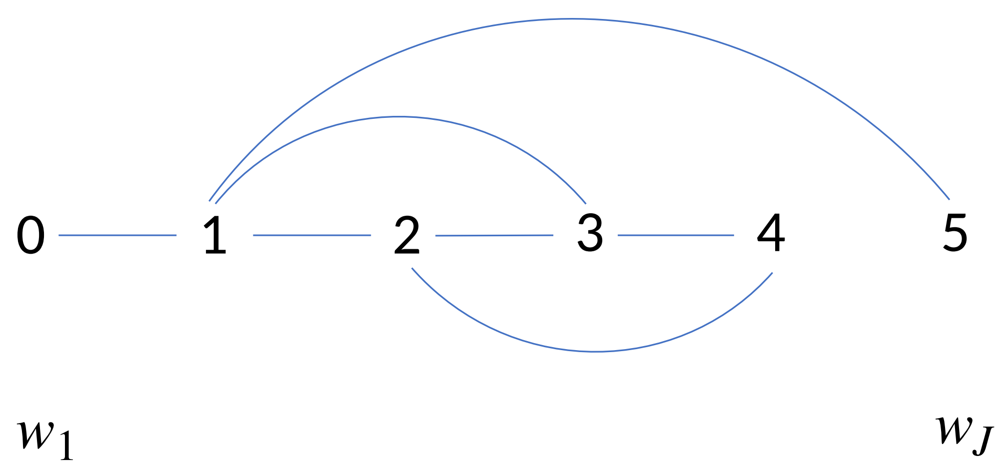

Figure 3.1: A representation of the small fabricated map for the purposes of the adjacency matrix.

Nodes in this network correspond to points in space where the equations will be evaluated, $w_1, w_2, ..., w_J$, where J is the number of nodes. The adjacency matrix indicates the connections between nodes with a value of 1 if there is a connection and a value of 0 if there is edge between the nodes. For example, in this network $A_{1,1} = 0$ as node 0 as there is no edge connecting it to itself and $A_{1,3} = 1$ as node 0 is connected to node 2. The adjacency matrix for the small fabricated road network is given by

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.2}$$

3.4.2 Small real map

The following gives the adjacency matrix for the small real map as described in Section 2.6.3:

$$A = \begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0
\end{bmatrix}. \tag{3.3}$$

3.4.3 Sixteen node grid

A uniform grid network consisting of 16 nodes was also tested to illustrate the influence of the network architecture of the small real map on the solution. The grid is shown in Figure 3.2:

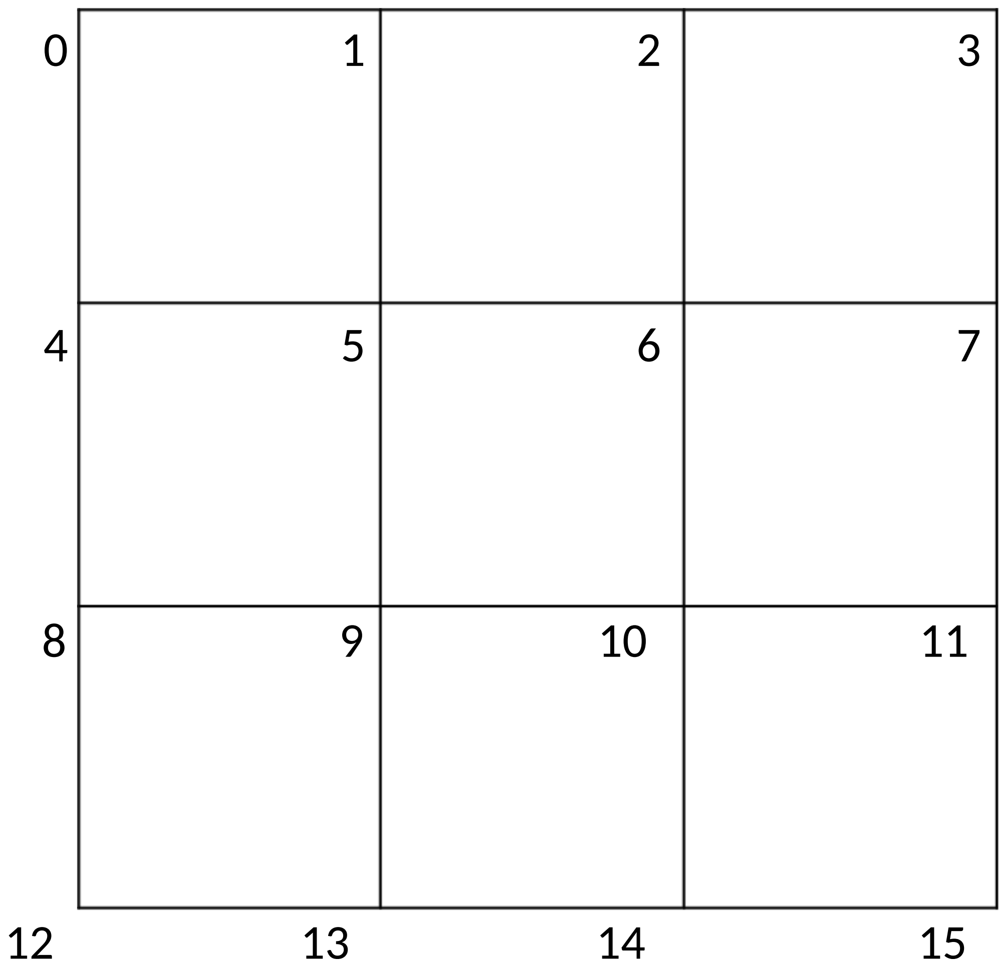

Figure 3.2: A uniform grid consisting of 16 nodes.

The adjacency matrix associated with this grid is:

$$A = \begin{bmatrix}
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0
\end{bmatrix}. \tag{3.4}$$

3.5 Derivation of equations

The partial differential equation chosen to represent the system, is the viscous Burgers' equation (3.1) which as mentioned in Section 3.3 is closely linked to well-known traffic models. Setting up the Laplacian u_{xx} is done following the method in Angstmann et al. [6]. The Laplacian is approximated by some matrix, in this case using the adjacency matrix A to implement the specific network in the system. We consider a network whose vertices form the set

$$\mathbf{W} = \{w_1, w_2, ..., w_J\}, \tag{3.5}$$

where the number of vertices is given by J. As seen in equations (3.2) and (3.3) the adjacency matrix A is given by

$$A_{i,j} = \begin{cases} 1 & \text{if vertices } w_i \text{ and } w_j \text{ are connected} \\ 0 & \text{otherwise} \end{cases}. \tag{3.6}$$

As in Case A of Angstmann et al. [6] we assume that the waiting time on every vertex is the same and that the jumping probability across edges is equal for a certain vertex. Let

$$\alpha(w_j) = \alpha_A \qquad\qquad \forall\ w_j \in \mathbf{W}, \tag{3.7}$$

and let

$$\lambda(w_j, w_i) = \frac{1}{k_i}, \tag{3.8}$$

then the network Laplacian can be expressed as

$$L_{i,j} = \alpha_A \left(\frac{1}{k_i} A_{i,j} - \delta_{i,j} \right). \tag{3.9}$$

Each vertex w_j has a degree k_j that is defined as the total number of vertices that it is linked to. This is also equal to the sum of each row of the adjacency matrix. The coefficient

$$\alpha_A = \frac{\text{waiting time density}}{\text{probability that a particle does not jump}}, \tag{3.10}$$

is assumed to be constant and for this model is set to $\alpha_A = 1$. A higher value of α_A indicates a higher degree of diffusion. The probability density of jumping to a vertex w_j at a given time t after arriving at a vertex w_i at an earlier time can be expressed as the product of the jump density and the waiting time density. We have assumed that the waiting time is the same on all the vertices and that the probability of jumping is equal for a given vertex. The jump density is symbolised by λ and $\sum_{j=1}^{J} \lambda(w_j, w_i) = 1$ for fixed w_i. Further, $\delta_{i,j}$ is an identity matrix where the entry at (i, j) is given by the Kronecker delta at (i, j). The Laplacian given in Angstmann et al. [6] leads to the following expression where $\mathbf{u}_x$ is the partial derivative of $\mathbf{u}$ with respect to x, that is the spatial partial derivative of the traffic density:

$$\mathbf{u}_t = \mathbf{L}\mathbf{u} + \mathbf{u}\mathbf{u}_x, \tag{3.11}$$

with $\mathbf{L}\mathbf{u} = D\mathbf{u}_{xx}$. We know from Angstmann et al. [6] that

$$L(u) = u_{i+1} - 2u_i + u_{i-1}, \tag{3.12}$$

and $\mathbf{u} = (u(w_1, t), ..., u(w_J, t))$. Here $L(u)$ denotes the Laplacian at u and u_i is the value of u at the node i. The initial condition stated earlier can now be formalised. We can give the initial condition as

$$u(w_j, 0) \in (1, 0, 0.5, 0, 0.25, 0) \qquad j \in [1, 6], \tag{3.13}$$

for the small fabricated map and

$$u(w_j, 0) \in (0, 0.1, 0, 0.1, 0, 0.1, 0, 0.1, 0, 0.1, 0, 0.1, 0, 0.1, 0, 0.1) \qquad j \in [1, 16], \tag{3.14}$$

for the small real map.

3.6 Finite Difference Method

The methodology from Wani and Thakar [7] is followed to construct a finite difference approximation for the system. We discretize Du_{xx} as is done in a Crank-Nicolson scheme, u_t by forward difference and uu_x by central difference at $t = t_n$ and $t = t_{n+1}$. Here $t_n = t_0 + n\Delta t$, with Δt the step size in t and $w_j = w_1 + j\Delta w$ with $1 < j < J$. At time $t = t_n$ the numerical solution at $w = w_j$ is given by u_j^n. So the solution at $t = t_n$ is $\mathbf{u}^n = (u_1^n, u_2^n, ..., u_J^n)^T$.

The following approximation is obtained

$$\begin{aligned}&\frac{u_j^{n+1}-u_j^n}{\Delta t}+\frac{1}{4\Delta x}[u_j^n(u_{j+1}^{n+1}-u_{j-1}^{n+1})+u_j^{n+1}(u_{j+1}^n-u_{j-1}^n)]\\&=\frac{D}{2(\Delta x)^2}(u_{j+1}^{n+1}-2u_j^{n+1}+u_{j-1}^{n+1}+u_{j+1}^n-2u_j^n+u_{j-1}^n)\end{aligned}. \tag{3.15}$$

The discretisation steps in equation (3.15), specifically for the advective term, are chosen so that the discretisation is linear. This enables the Von Neumann stability analysis, while preserving accuracy and stability typically associated with Crank-Nicolson schemes [7]. The finite difference stencil for this method is shown in Figure 3.3.

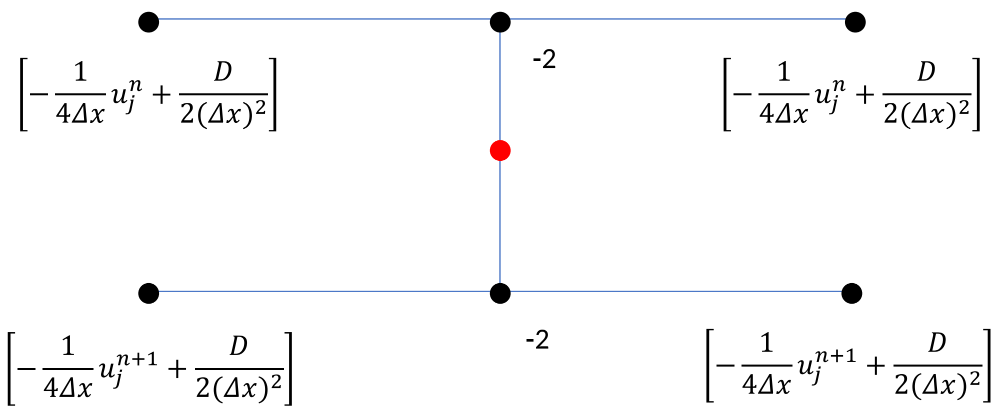

Figure 3.3: A finite difference stencil of the method.

This scheme is consistent according to Wani and Thakar [7]. The finite difference scheme is a numerical approximation of the partial differential equation and the solution to the finite difference scheme is an approximation of the exact solution of the partial differential equation. If the approximation error grows exponentially the numerical scheme is deemed unstable. It is possible to define conditions, usually limits on the size of the steps in the temporal and spatial parameters, for which the numerical scheme is stable. Often, this places a limit on the time step for a given spatial step. During Von Neumann stability analysis the solutions of the partial differential equation in one spatial dimension can be expressed using the Fourier inversion formula with wave number θ. We can consider the stability for an individual wave number since for a linear equation a linear combination of solutions is also a solution. The solution at a time step depends on the solution at the previous time step in the following way: $u^{n+1} = \xi u^n$, where ξ is the amplification factor. If $|\xi| < 1$, then the solution is stable and damping, if $|\xi| = 1$, the solution is neutrally stable and if $|\xi| > 1$ the solution is unstable and growing. We substitute u_j^n by $\xi^n e^{Ij\Delta x\theta}$, where I is the imaginary unit. Let $m = \Delta x\theta$, $p = \frac{\Delta t}{4\Delta x}$ and $r = \frac{D\Delta t}{2(\Delta x)^2}$. Then

$$\begin{gathered} \xi = 1 - p[(\xi e^{Im} - \xi e^{-Im}) + \xi(e^{Im} - e^{-Im})] \\ +r(\xi e^{Im} - 2\xi + \xi e^{-Im}) \end{gathered}. \tag{3.16}$$

Given that $e^{Im} - e^{-Im} = 2I\sin(m)$ and $e^{Im} + e^{-Im} = 2\cos(m)$ we have

$$\begin{gathered} \xi = 1 - \xi p(2I\sin(m)) - p(2I\sin(m)) \\ +\xi r(2\cos(m)) - 2\xi r + r(2\cos(m)) - 2r \end{gathered}, \tag{3.17}$$

leading to

$$\xi = \frac{1 - 2pI\sin(m) + 2r\cos(m) - 2r}{1 + 2pI\sin(m) - 2r\cos(m) + 2r}. \tag{3.18}$$

For this problem we can substitute $p = 0.01$ and $r = 0.08$ as $p = \frac{\Delta t}{4\Delta x}$ and $r = \frac{D\Delta t}{2(\Delta x)^2}$ with $\Delta t = 0.01$, $\Delta x = 0.25$ and $D = 1$. Thus we have

$$|\xi| = \left|\frac{1 - 2pI\sin(m) + 2r\cos(m) - 2r}{1 + 2pI\sin(m) - 2r\cos(m) + 2r}\right| \leq 1, \tag{3.19}$$

and therefore the scheme is stable for the chosen values of D, Δt and Δx.

3.7 Solution of equations

To solve the system we take equation (3.15), substitute in equation (3.12) and approximate $(u_{j+1}^n - u_{j-1}^n)$ with the matrix B^n given by

$$B^n = \begin{bmatrix} 0 & u_2^n & 0 & \dots & & 0 & -u_J^n \\ -u_1^n & 0 & u_3^n & 0 & \dots & & 0 \\ 0 & -u_2^n & 0 & u_4^n & 0 & \dots & 0 \\ & & & \ddots & & & \\ 0 & \dots & 0 & -u_{J-3}^n & 0 & u_{J-1}^n & 0 \\ 0 & \dots & & 0 & -u_{J-2}^n & 0 & u_J^n \\ u_1^n & 0 & \dots & & 0 & -u_{J-1}^n & 0 \end{bmatrix}. \tag{3.20}$$

Then we can rewrite equation (3.15) as

$$\begin{aligned} u^{n+1} - \frac{\Delta t}{2(\Delta x)^2} L u^{n+1} + \frac{\Delta t}{4\Delta x} B^n u^{n+1} \\ = u^n + \frac{\Delta t}{2(\Delta x)^2} L u^n - \frac{\Delta t}{4\Delta x} B^n u^n \end{aligned}. \tag{3.21}$$

The condition number of a system $\mathbf{Ax} = \mathbf{b}$ is given by

$$K(A) = \|A\| \left\|A^{-1}\right\|, \tag{3.22}$$

where in this case

$$A = \left[\delta_{i,j} - \frac{\Delta t}{2(\Delta x)^2} L + \frac{\Delta t}{4\Delta x} B^n\right], \tag{3.23}$$

and $\mathbf{x} = u^{n+1}$ with

$$\mathbf{b} = u^n + \frac{\Delta t}{2(\Delta x)^2} L u^n - \frac{\Delta t}{4\Delta x} B^n u^n. \tag{3.24}$$

If the solution of the system $\mathbf{Ax} = \mathbf{b}$ is sensitive to small changes in the elements of A or $\mathbf{b}$, then A will have a large condition number and is ill-conditioned. If the system is well-conditioned the condition number is close to one. The well-conditioning determines how accurately a system of equations can be solved. We know from Belsley et al. [63] that if data is known to d significant digits, then a condition number of the order of magnitude of 10^r can lead to inaccuracies in the $(d-r)$th significant digit of the solution. For the small fabricated map the condition number $K(A) = 1.1699$ and for the small real map $K(A) = 1.2418$. Therefore both are well-conditioned and can be solved with high accuracy.

3.8 Data and inputs

Contained in this section is the specific data and inputs to the macroscopic model. Two road networks were modelled given by the maps in Figure 2.3 and Figure 2.6. Section 2.6.3 elucidates how the data from the small real map were processed to provide model inputs. The number of nodes in each map, six for the small fabricated map and sixteen for the small real map was needed as input to this model, as well as the degree for each node given by k_j. For the small fabricated map $k_j \in (1, 4, 3, 3, 2, 1)$ and for the small real map $k_j \in (1, 1, 1, 4, 4, 1, 1, 4, 4, 1, 1, 4, 4, 1, 1, 1)$. The time step size $\Delta t = 0.01$ and the step size in space $\Delta x = 0.25$. For the small fabricated map the number of iterations for the solution was set to fifty and for the small real map it was set to 500. The Laplacian matrix L for the system as given by equation (3.9) is also an input to the model. For the small fabricated map L is given by

$$L = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ 0.25 & -1 & 0.25 & 0.25 & 0 & 0.25 \\ 0 & 0.\dot{3} & -1 & 0.\dot{3} & 0.\dot{3} & 0 \\ 0 & 0.\dot{3} & 0.\dot{3} & -1 & 0.\dot{3} & 0 \\ 0 & 0 & 0.25 & 0.25 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix}, \tag{3.25}$$

and for the small real map by

$$L = \begin{bmatrix}
-1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.25 & 0 & 0.25 & -1 & 0.25 & 0 & 0 & 0.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.25 & 0 & 0.25 & -1 & 0.25 & 0 & 0 & 0.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.25 & 0 & 0 & 0.25 & -1 & 0.25 & 0 & 0 & 0.25 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.25 & 0 & 0 & 0.25 & -1 & 0.25 & 0 & 0 & 0.25 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & - & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.25 & 0 & 0 & 0.25 & -1 & 0.25 & 0 & 0.25 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.25 & 0 & 0 & 0.25 & -1 & 0.25 & 0 & 0.25 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1
\end{bmatrix}. \tag{3.26}$$

L for the 16 node grid, where $k_i = 1 \; \forall i, i \in [1, 16], i \in \mathbb{Z}$, is given by

$$L = \begin{bmatrix}
-1 & 0.5 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/3 & -1 & 1/3 & 0 & 0 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1/3 & -1 & 1/3 & 0 & 0 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.5 & -1 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/3 & 0 & 0 & 0 & -1 & 1/3 & 0 & 0 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.25 & 0 & 0 & 0.25 & -1 & 0.25 & 0 & 0 & 0.25 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.25 & 0 & 0 & 0.25 & -1 & 0.25 & 0 & 0 & 0.25 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1/3 & 0 & 0 & 1/3 & -1 & 0 & 0 & 0 & 1/3 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1/3 & 0 & 0 & 0 & -1 & 1/3 & 0 & 0 & 1/3 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.25 & 0 & 0 & 0.25 & -1 & 0.25 & 0 & 0 & 0.25 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.25 & 0 & 0 & 0.25 & -1 & 0.25 & 0 & 0 & 0.25 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 0 & 0 & 1/3 & -1 & 0 & 1/3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & -1 & 0.5 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 0 & 0 & 1/3 & -1 & 1/3 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 0 & 0 & 1/3 & -1 & 1/3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0.5 & -1
\end{bmatrix}. \tag{3.27}$$

3.9 Output

Output is given in the form of graphs generated in R from the solutions obtained in MATLAB. These graphs plot the density at a node over time and therefore show how the density in the system evolves.

3.10 Conclusion

The model described in this chapter is very different from the agent-based model in the previous chapter, even though they were built to represent the same systems. The next chapter will now look at the differences and similarities in the results obtained from the two models.

Chapter 4

Results and discussion

Our expectation of the results in this chapter is that the output from the two different modelling approaches will show some agreement as they are based on some of the same systems. Since the two approaches and their inputs are very different as seen in the assumptions for the models, we may however see results that are seemingly unrelated. For the agent-based model we hope to see a positive impact in congestion for the dynamic routing options which take congestion into account when compared to static shortest distance routing. We anticipate that the diffusion in the macroscopic model will have an effect on traffic and that it will lead to less congested networks.

4.1 Microscopic model

This section explores the results from the runs of the microscopic agent-based model. First the travel times along routes are given, then the results pertaining to the assignation of routes to vehicles and last, results showing the utilisation of networks' roads. We want to see whether dynamic routing with traffic information rather than static shortest distance routing will have an effect on the routes vehicles take and on their travel times through the system. This will help to assess the possible impact of these routing choices on congestion in the system. We also consider the flow of vehicles through the network and how this can be interpreted.

4.1.1 Vehicle routing

When we look at vehicle routing in this section, we consider different aspects. First, we investigate the spread of vehicles across the different possible routes with static routing and then look at the extent of the impact on route assignment when the routing is dynamic instead of static. The spread of the vehicles across routes gives an indication of the spread of vehicles across the network and therefore of how evenly spread we expect traffic to be. There are a finite number of source and destination node pairs and only a few routes between each of those pairs. In the case where we apply static shortest distance routing this number of routes between pairs of source and destination nodes is reduced as only the shortest route or possibly a few routes of equal length can be selected. We can see this in that, for the small fabricated map, there are 30 unique pairs of source and destination nodes and 30 unique routes, thus one route per pair of source and destination nodes. The same is true for the small real map, with 240 unique pairs of source and destination nodes and the same number of unique routes. For the large fabricated map however, there are 50 unique pairs of source and destination nodes and 1866 unique routes. This is due to the fact that the selection of source and destination nodes was severely restricted and that the large grid affords many routes of the same length between nodes that are far apart, as these are. The clustering of source and destination nodes is thought to be more representative of rush hour traffic where vehicles travel between residential areas and business districts. Future work could combine this with a more realistic road network.

We expect that, if we send a large number of vehicles through a network, each of these routes will be utilised numerous times. For the smaller maps, where source and destination nodes were assigned at random, we expect a fairly even spread of vehicles across routes. For the large map, where the source and destination nodes were clustered as described in section 2.6.2, we expect that some routes will see heavier traffic than others. Since we are still looking at outputs from static shortest distance routing, we expect clear clustering on some routes since vehicles will only take the shortest routes to their destinations and there is no dispersion due to traffic avoidance included yet. The data presented in Figures 4.1, 4.2 and 4.3 were sorted ascendingly to better discern the spread of cars across the routes in each map. In Figures 4.1 and 4.2, we see a random, but fairly even spread of vehicles across networks. In Figure 4.1 we that all routes were used by a number of cars. Some routes were used more than others, but the increase across routes is consistent. In Figure 4.2 there are many more routes and although all routes are utilised, some are used much more than others. The spread is still fairly even as can be seen from the gradual increase. Looking at Figure 4.3 we can see the effect of the restrictions placed on the source and destination nodes in the much higher usage of some routes in comparison to others. Towards the left of the plot many routes have a low utilisation. On the other end of the graph some routes have a very high utilisation compared to the rest. There is also clear evidence of the clustering of the routes in the large jumps in number of cars. Here, we don't see the gradual climb of Figures 4.1 and 4.2. Clearly, the larger map has areas where congestion is concentrated. When we compare this with Figures 4.4 and 4.5 we see a similar pattern. These figures show the utilisation of the routes at the end of the simulation run and so represent the final routes taken by the vehicles. Looking at the dynamic routing, both local and global, the cars are more evenly spread than in the case of the static routing. However, the strong clustering of cars on certain routes is still present. This is not unexpected as the dynamic routing takes into account both congestion and distance and would therefore still have a low utilisation of some longer routes.

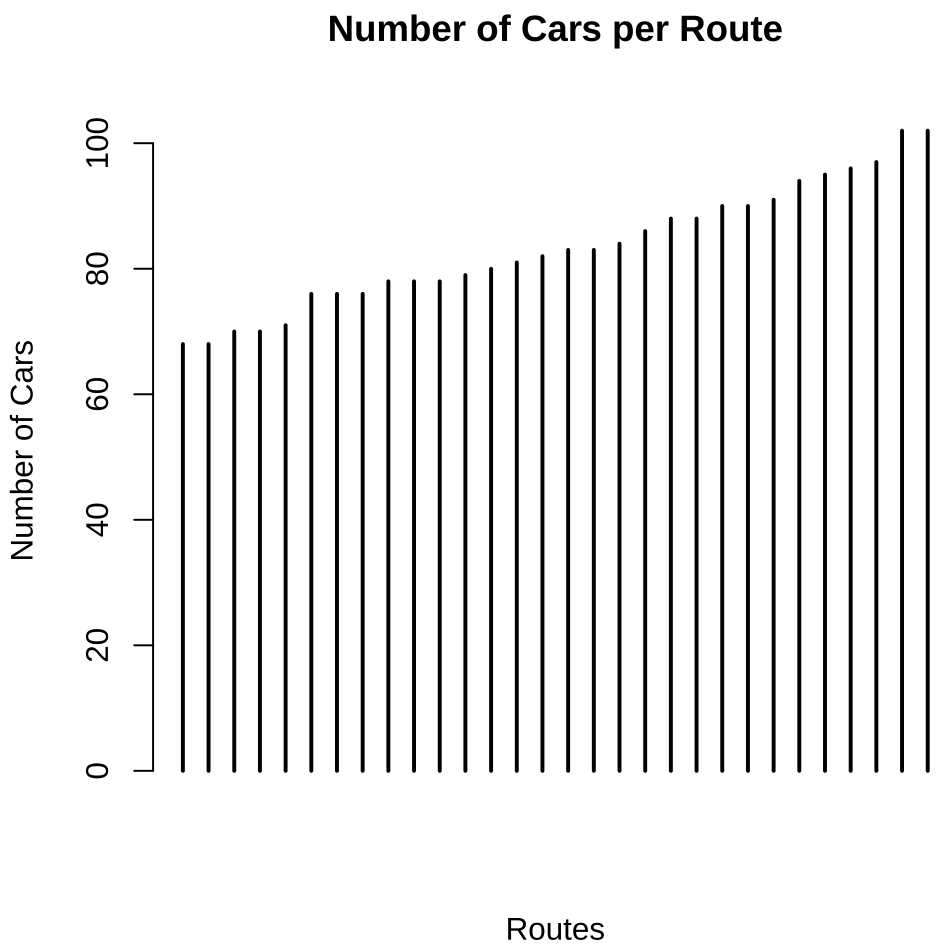

Figure 4.1: A graph showing the number of cars per route for the small fabricated map with static shortest distance routing.

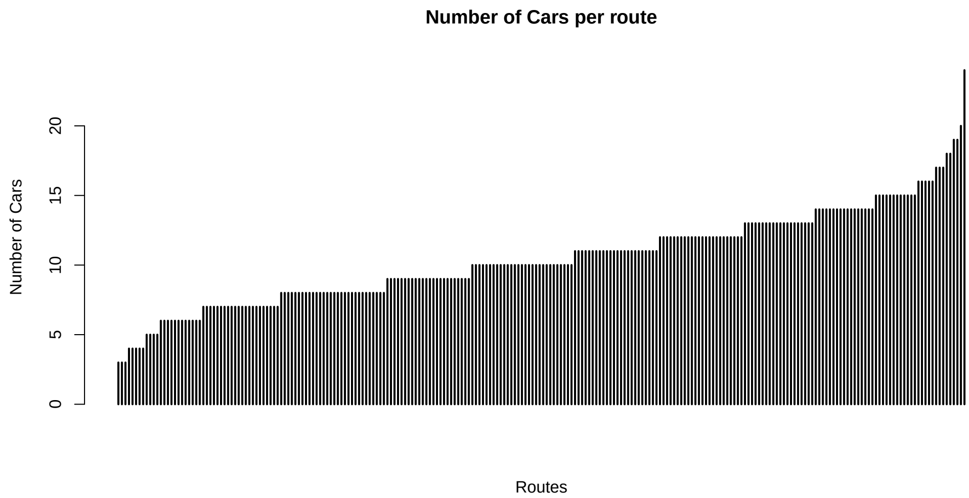

Figure 4.2: A graph showing the number of cars per route for the small real map with static shortest distance routing.

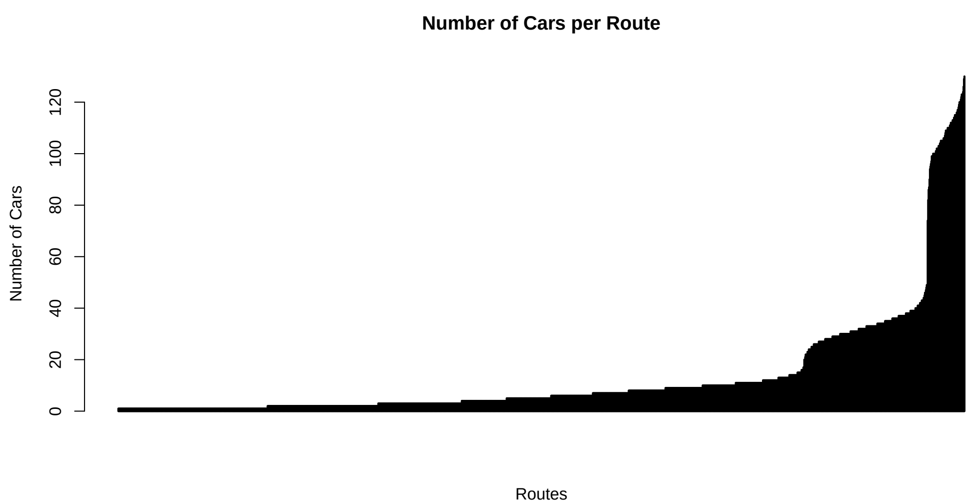

Figure 4.3: A graph showing the number of cars per route for the large fabricated map with static shortest distance routing.

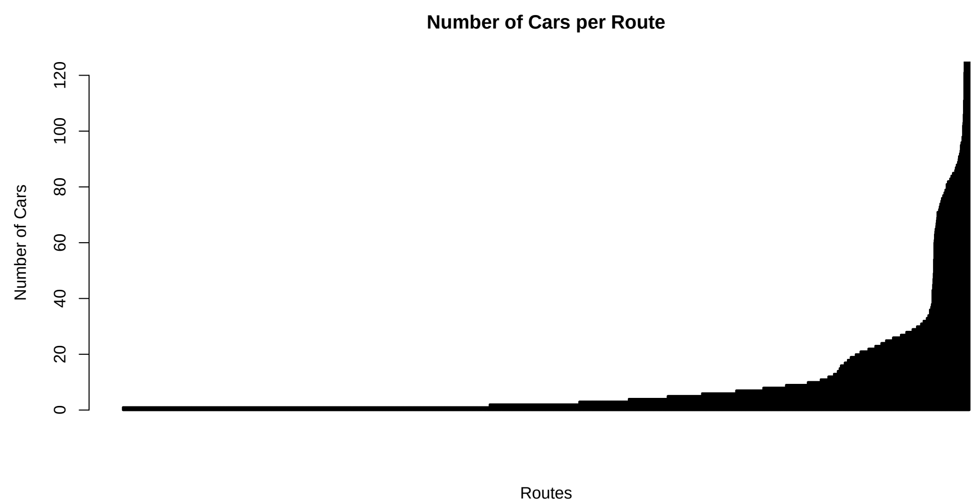

Figure 4.4: A graph showing the number of cars per route for the large fabricated map with dynamic global routing at the end of the simulation.

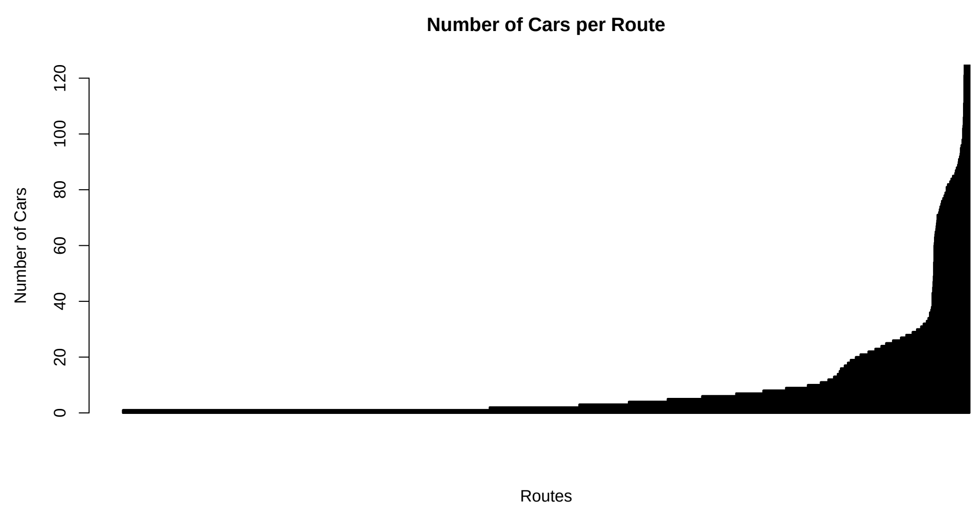

Figure 4.5: A graph showing the number of cars per route for the large fabricated map with dynamic local routing at the end of the simulation.

We can compare the number of distinct routes assigned during a simulation run with the number of vehicles in that run to get a sense of the impact of dynamic routing. In the case of shortest distance routing every vehicle gets assigned one route for the duration of the simulation, therefore the number of routes assigned is equal to the number of vehicles. So, for the 2500 vehicles, 2500 routes are assigned. We expect that this number will be higher with dynamic routing: if a vehicle changes its route due to congestion, this is a new assigned route. And this is what we observe here, where for dynamic routing with a global perspective the small fabricated map with 2500 vehicles has 4201 distinct routes and the small real map with 2500 vehicles has 7142 distinct routes. For the small fabricated map the number of assigned routes are nearly twice as many as the number of vehicles. From this we can deduce that a large portion of the vehicles' routes were influenced by traffic while using dynamic routing. The small real map has an even higher ratio of assigned routes to vehicles, there are almost three times as many. This indicates that some vehicles may have changed routes on more than one occasion. The small real map is larger than the small fabricated map and therefore affords more routing options between nodes, this can explain why more changes take place in the small real map.

Evidently, the dynamic routing was utilised by vehicles when it was available. To see if it had a positive impact on congestion, as we expect it will, we must look at other outputs, such as the travel times discussed in the next section.

4.1.2 Travel times along routes

As we saw in the previous section, making the vehicles' routing dynamic instead of static and then also taking into account the nearby or overall congestion of the network impacts the vehicles' routes. But what kind of effect does this have on the system and its congestion? We expect the impact to be a positive one - that the system will be less congested and that individual vehicles will be able to navigate the congestion better. To assess this impact we consider the travel times of vehicles per route. Naturally, having more and less traffic on a route will introduce variation in the travel times of vehicles on that route. We are interested in the spread of travel time as well as the upper and lower limits of this travel time. To readily visualize this we have represented the travel times as boxplots.

Boxplots give an indication of the limits and spread of the data. The thick line inside the box represents the median of the values while the sides of the box are the first and third quartiles of the data. Whiskers extend outside the box to show the range of values considered as normal, every point outside this range is seen as an outlier and shown on the plot as a circle.

In the model, vehicles' speeds default to the speed limit on the road. As stated in the assumptions, this speed limit cannot be exceeded. Travel time variance is therefore caused by the following factors: slowing down to approach an intersection where it will stop; being stationary at an intersection to stop or wait for a green traffic light or space to move into the next road; slowing down when approaching a slower vehicle. We assume that the effect of stopping and waiting for traffic lights will be similar for all of the scenarios, so most variation in the travel times on routes will be due to varying degrees of congestion. Less congestion means shorter travel times and more congestion means longer travel times. The speed limit, together with the length of each road, places a physical lower limit on the travel time for each route.

Looking at Figures 4.6, 4.7 and 4.8 we can see the variation in travel time on each possible route on the small fabricated map. Each figure represents a routing choice, in Figure 4.6 static shortest distance routing option was applied, in Figure 4.7 dynamic global routing and in Figure 4.7 dynamic local routing. We see a slight decrease in the travel time variation with dynamic routing. We expect a positive influence and here it can be seen, however the effect is quite small. The size of the map plays an important role here: routes are short so there is not a lot of opportunity for them to be changed for better or for worse by dynamic routing and on a short route this will result in a small change. Also, there is little difference between local and global knowledge of the network for the dynamic routing as local traffic knowledge already encompasses a large part of the network. Thinking about this in terms of real experience, the question is how much travel time can be improved through routing if you are only travelling a couple of blocks and the answer is that it cannot be improved much due to the short distance travelled and the physical limitations such as the speed limit and reasonable acceleration. Figure 4.6 shows the spread of travel times per route with static shortest distance routing on the small fabricated map with Figure 4.7 showing dynamic routing with global information and Figure 4.8 dynamic routing with local information on the same map. These boxplots show that some routes had barely any variation in travel times while others had much more.

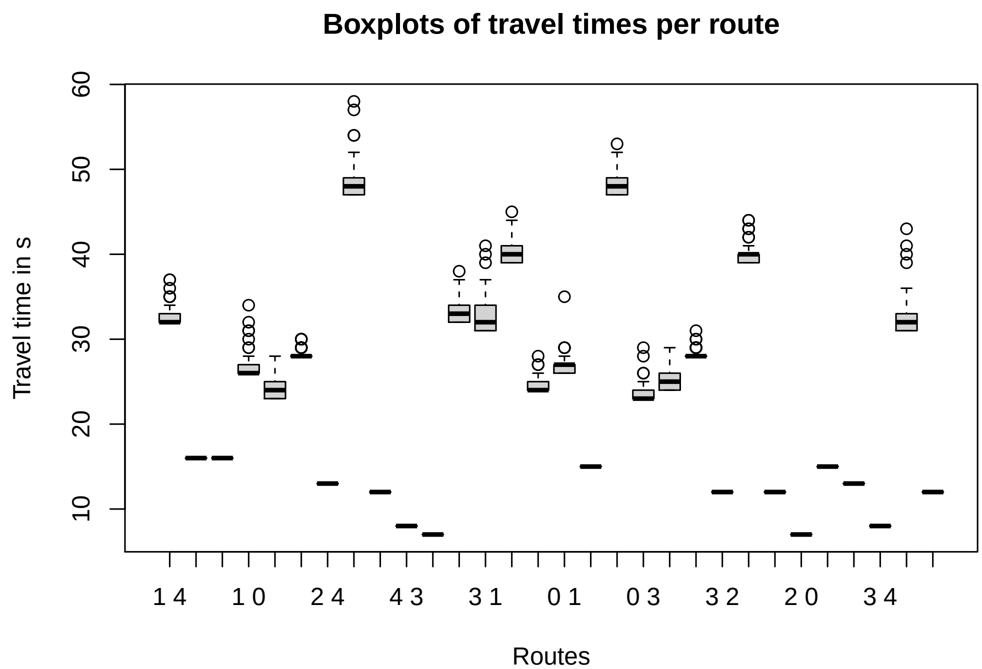

Figure 4.6: Boxplots showing the spread of travel times per route for the small fabricated map with static shortest distance routing.

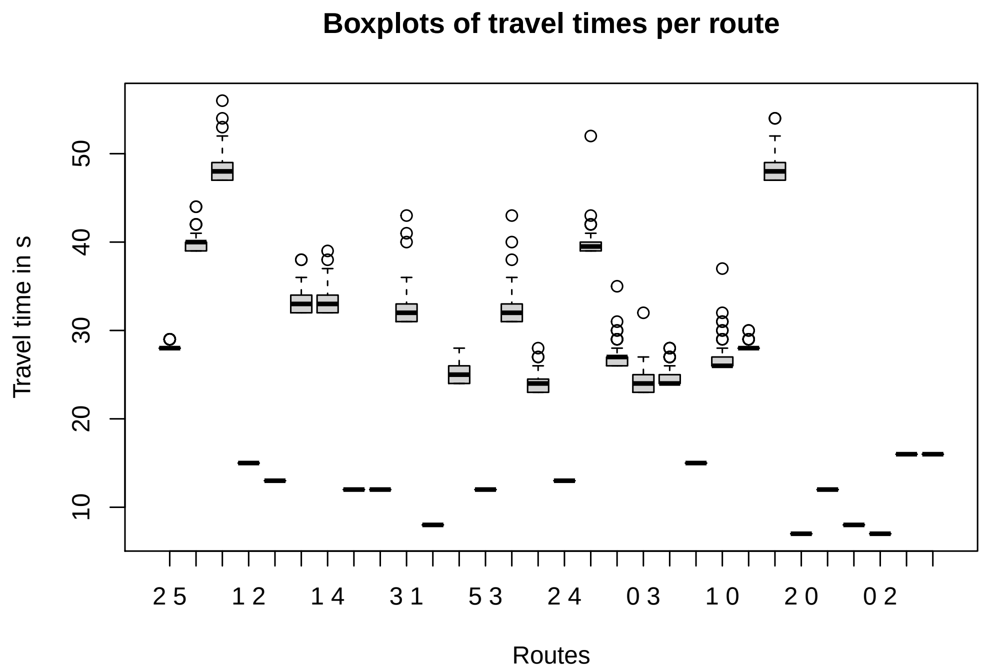

Figure 4.7: Boxplots showing the spread of travel times per route for the small fabricated map with dynamic routing.

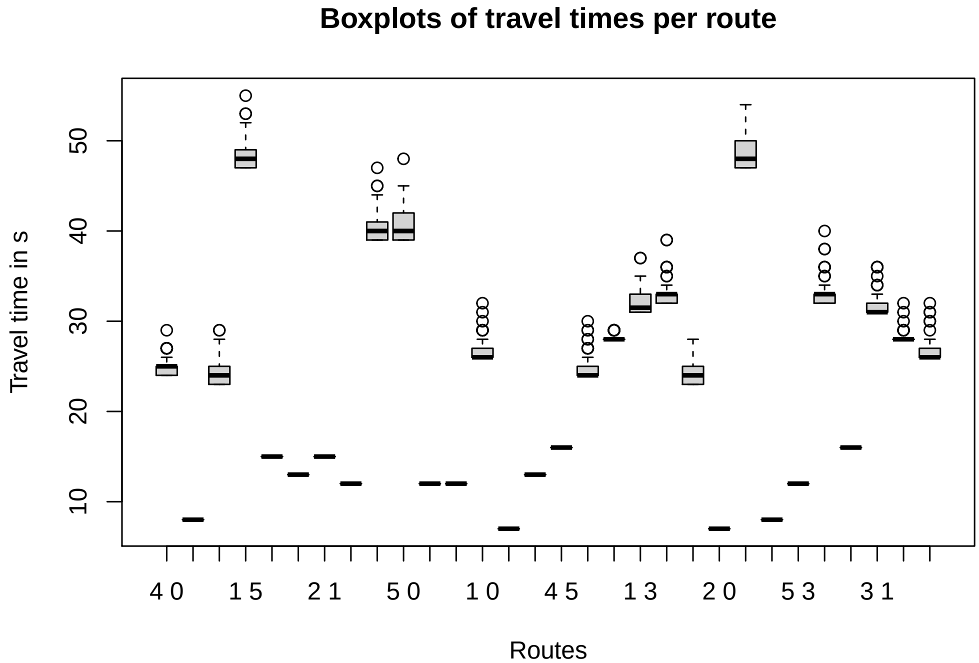

Figure 4.8: Boxplots showing the spread of travel times per route for the small fabricated map with dynamic local routing.

Moving on to the slightly larger small real map we expect similar results, but perhaps a larger positive impact form dynamic routing. We look at Figures 4.10, 4.9 and 4.11 for static shortest distance, global dynamic and local dynamic routing on the small real map. Low congestion in the system could explain the overall smaller variation in travel time. Looking at the upper limits of the travel times we can see that travel times were much shorter in the global dynamic routing scenario. In this sense the global dynamic routing outperforms both the static shortest distance routing and the local dynamic routing options. Interestingly, there is the most variation per route for the bulk of the routes with dynamic global routing. Thus global dynamic routing also results in less predictable travel times. However, this is still a small variation in travel time and it could possibly be ascribed to the fact that when everyone in the system is utilising global dynamic routing the areas of congestion are bound to change making the system itself and therefore the routing less predictable. Figure 4.9 shows the spread of travel times per route with static shortest distance routing on the small real map with Figure 4.10 showing dynamic routing with global information and Figure 4.11 dynamic routing with local information on the same map. These boxplots show that most routes had barely any variation in travel times.

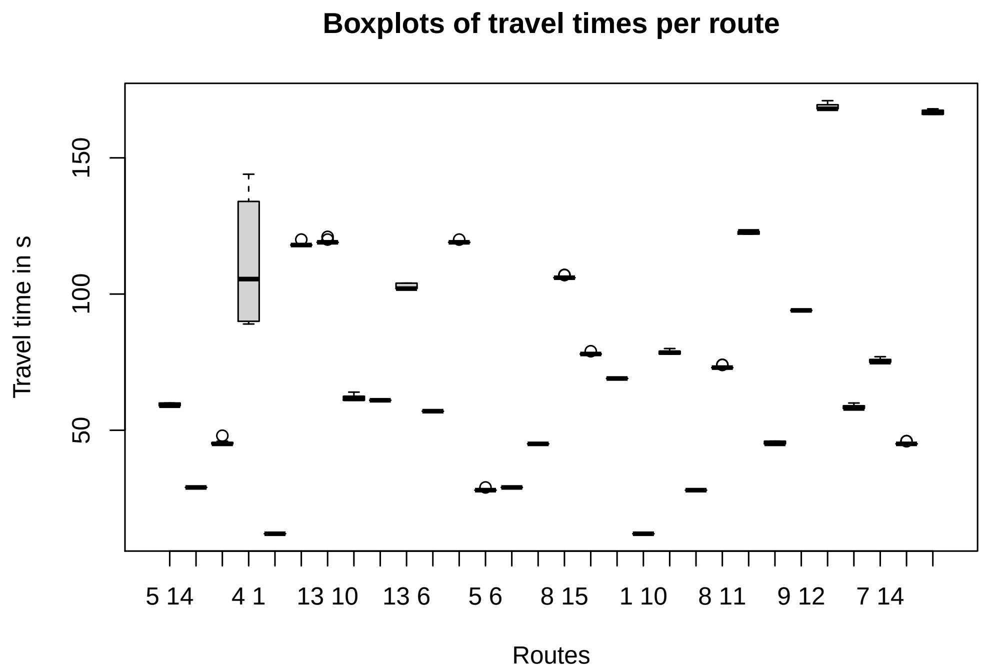

Figure 4.9: Boxplots showing the spread of travel times per route for the small real map with static shortest distance routing.

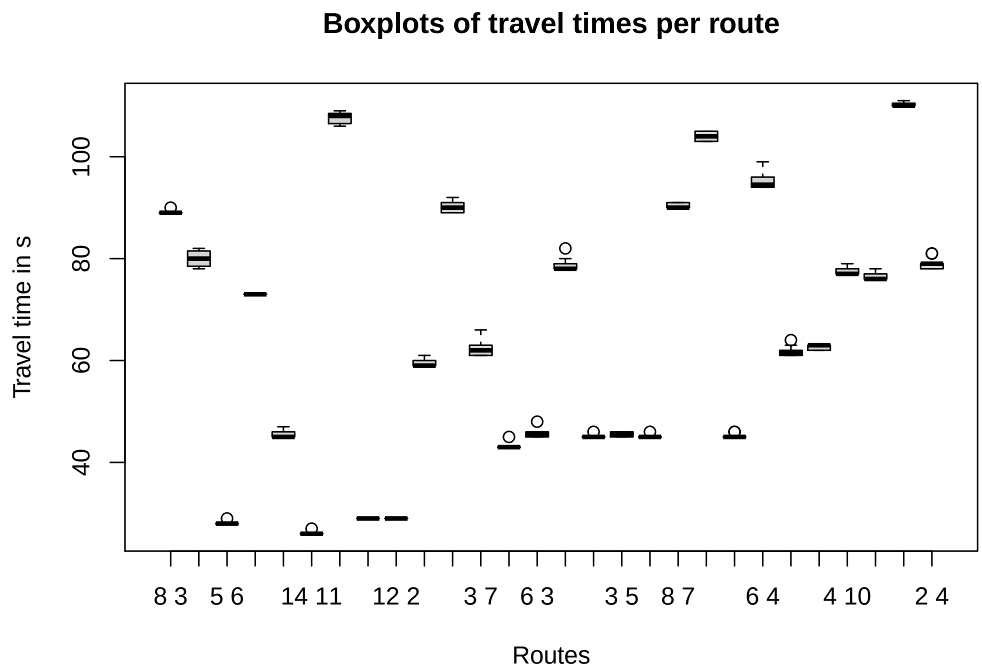

Figure 4.10: Boxplots showing the spread of travel times per route for the small real map with dynamic routing.

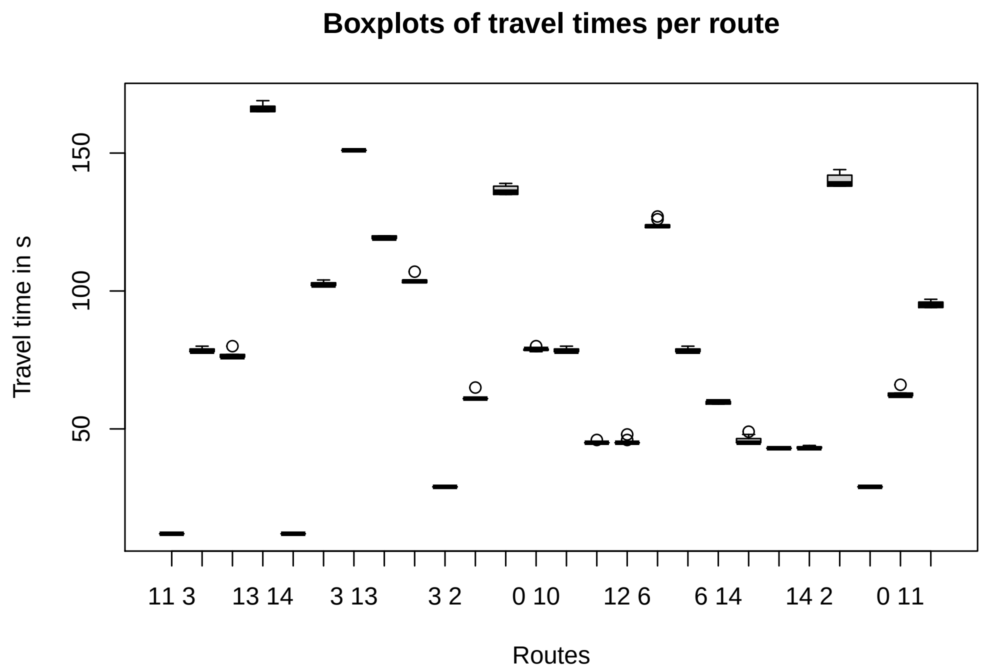

Figure 4.11: Boxplots showing the spread of travel times per route for the small real map with dynamic local routing.

We expect the effects of the different routing options to be even more pronounced for the large fabricated map. Figures 4.12, 4.13 and 4.14 show travel times on the routes of the large fabricated map for static shortest distance routing, global dynamic routing and local dynamic routing, respectively. Comparing the static to the dynamic routing we can see a definite reduction in both the variability and in the travel times when global dynamic routing is used. The large and fairly open grid provides space for the dispersion of traffic. In future it could be of interest to implement a large real map. We expect to see the positive influence of dynamic routing on this system.

The effects of bottlenecks in a large network could be studied. Figure 4.12 shows the spread of travel times per route with static shortest distance routing on the small fabricated map with Figure 4.13 showing dynamic routing with global information and Figure 4.14 dynamic routing with local information on the same map. These boxplots show a lot of variation with static shortest distance routing, likely caused by the resulting congestion and barely any variation in travel times for the dynamic routing options.

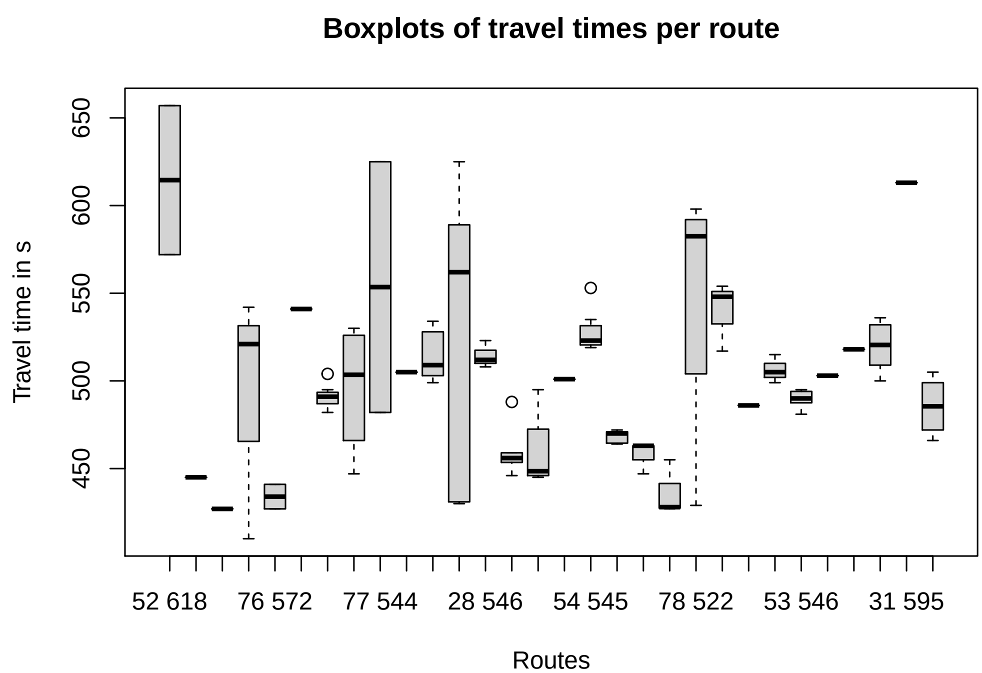

Figure 4.12: Boxplots showing the spread of travel times per route for the large fabricated map with static shortest distance routing.

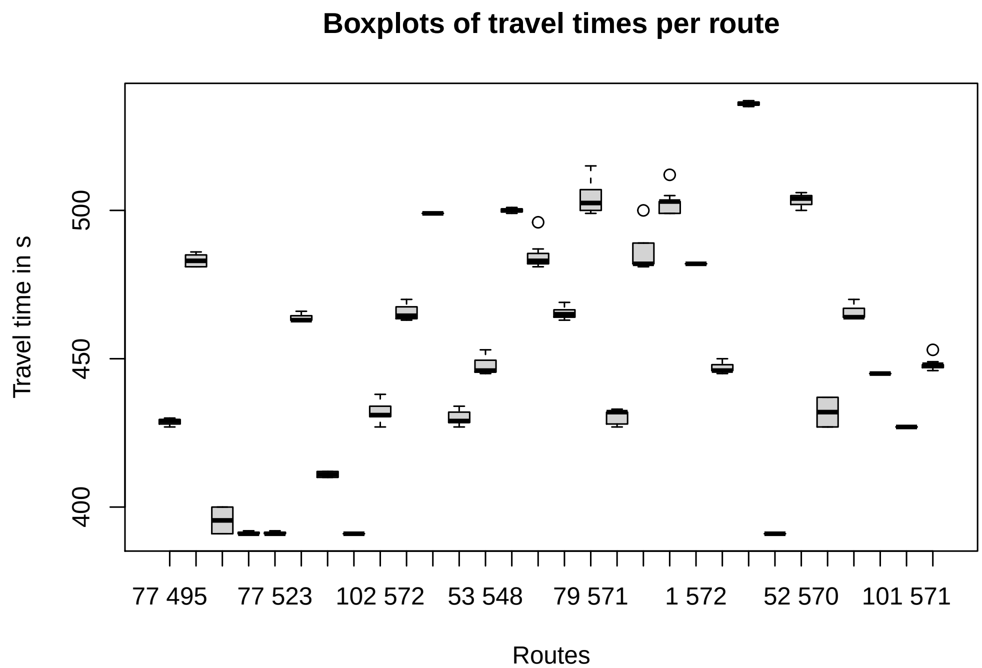

Figure 4.13: Boxplots showing the spread of travel times per route for the large fabricated map with dynamic routing.

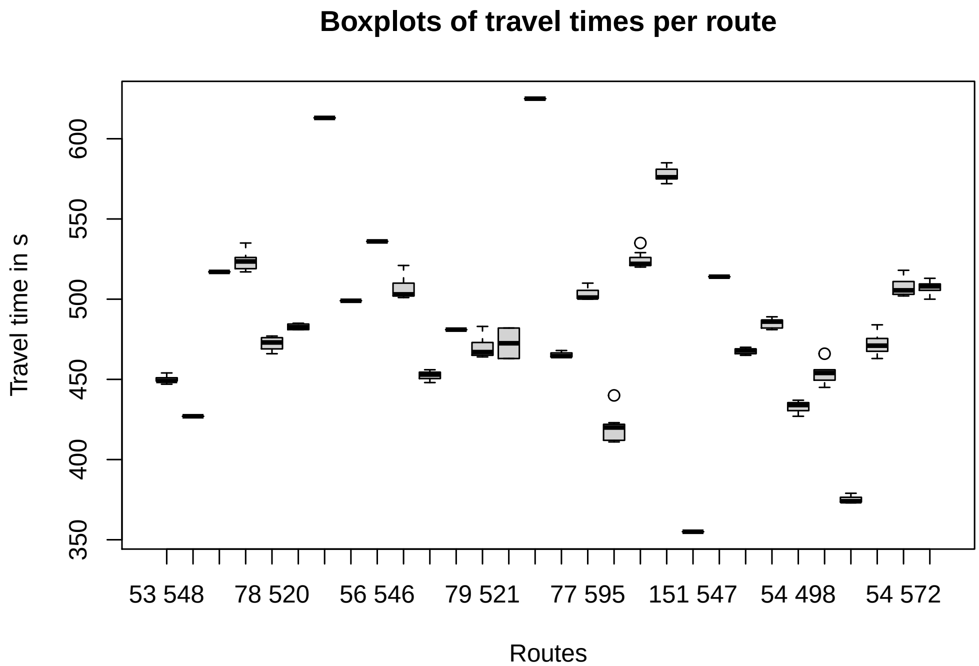

Figure 4.14: Boxplots showing the spread of travel times per route for the large fabricated map with dynamic local routing.

Table 4.1 shows the mean travel time per vehicle for each scenario. For the small fabricated map we see almost no difference in the values, regardless of the routing option. For the small real map there is a small decrease with global dynamic routing and an increase with local dynamic routing. Looking at the large fabricated map, we see a large decrease with global dynamic routing and a large increase with local dynamic routing.

A decrease in travel time with dynamic routing was expected. This was seen most clearly on the large fabricated road network with dynamic global routing and should be explored in large models with different network struc-

Table 4.1: Mean travel time per vehicle.

Static shortest distance	**Small fabricated map**	**Small real map**	**Large fabricated map**
Static shortest distance	22.53	66.88	505.32
Dynamic global	22.58	65.67	452.22
Dynamic local	22.52	70.54	585.82

tures. All of this suggests mobile navigation with real-time traffic information should improve congestion and travel times.

4.1.3 Flow of vehicles on roads

This section deals with the flow of traffic across the roads in the network. The number of cars on a road over time was plotted for each road of each map for each routing method. These graphs provide a view of the movement of vehicles through the network and are important in validating that the model works as it should. In a way this is like looking at an animation of traffic through the system. It helps us to see whether vehicles move along roads in unexpected ways such as getting stuck. We gain a sense of the utilisation of the roads in the network. We do not expect to see large pile-ups of traffic that remain throughout the simulation and we anticipate that some roads will be used more frequently than others. There are seven roads in the small fabricated map, twenty five in the small real map and over two hundred utilised roads in the large fabricated map as shown in Table 4.2. When factoring in the different routing options with this, it results in a volume of graphs too big for inclusion in this document. This section includes a couple of representative graphs as examples.

Table 4.2: Number of roads utilised per map.

Map	**Number of roads utilised**
Small fabricated map	7
Small real map	25
Large fabricated map	200

Figure 4.15 shows one of the roads from the small fabricated map with static shortest distance routing. The number of vehicles using the road varies with time and the road is frequently empty, but not for long. After a certain point in the simulation the road is not used any more. This occurrence is dependent on the relation of the time length chosen for the simulation to the number of vehicles and their interarrival and travel times. All the roads in this map are empty at the end of the simulation and this represents all the vehicles having arrived at their destination node. Herewith we are reassured that no vehicles are stranded somewhere in the network, instead of moving along as they should. The number of vehicles on the road increases and decreases as vehicles arrive on the road, progress along it and leave it for another road. This is exactly the behaviour we expect and require.

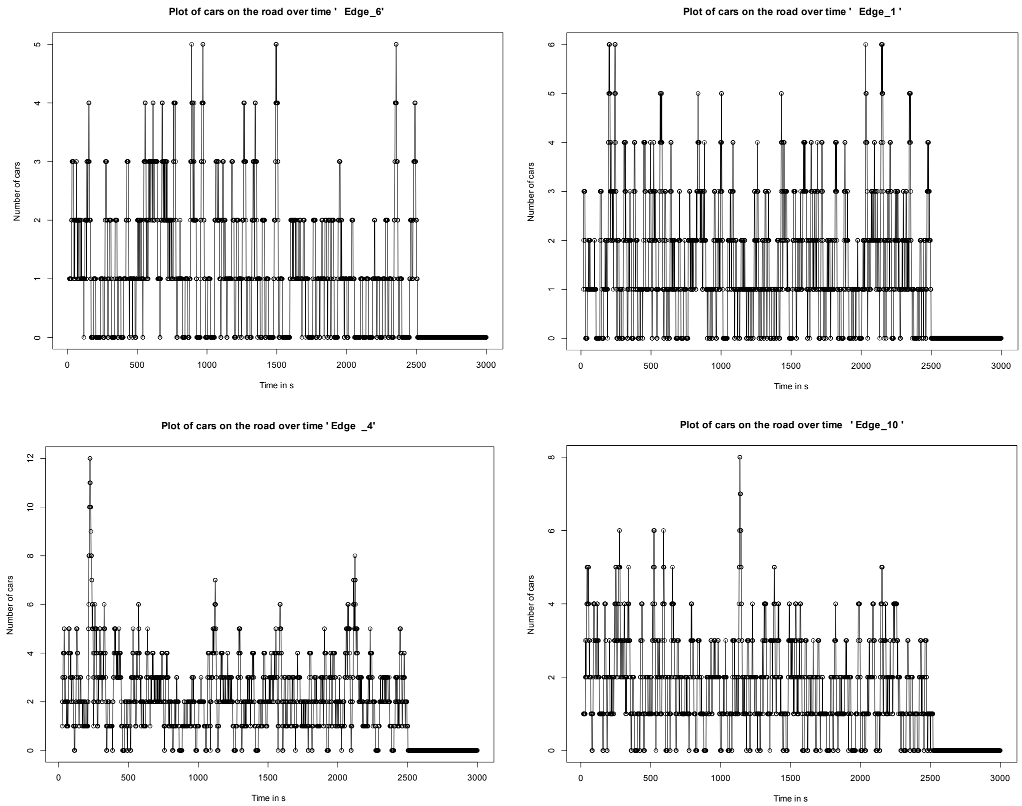

Figure 4.15: Four graphs showing the number of cars on a road over time for the small fabricated map with static shortest distance routing.

The plot in Figure 4.16 shows a road in the small real map with static shortest distance routing. This road is only empty near the end of the simulation run. A similar utilisation is seen for a number of other roads in the small real map for any type of routing, while the others have graphs similar to Figure 4.15. Again, all the vehicles have reach their destinations within the model run time. Figure 4.16 illustrates that some roads in the small real map have a much higher utilisation than others. This is likely due to the source and destination nodes assigned. Even though congestion builds on these roads, it is diminished again as there is movement along the roads.

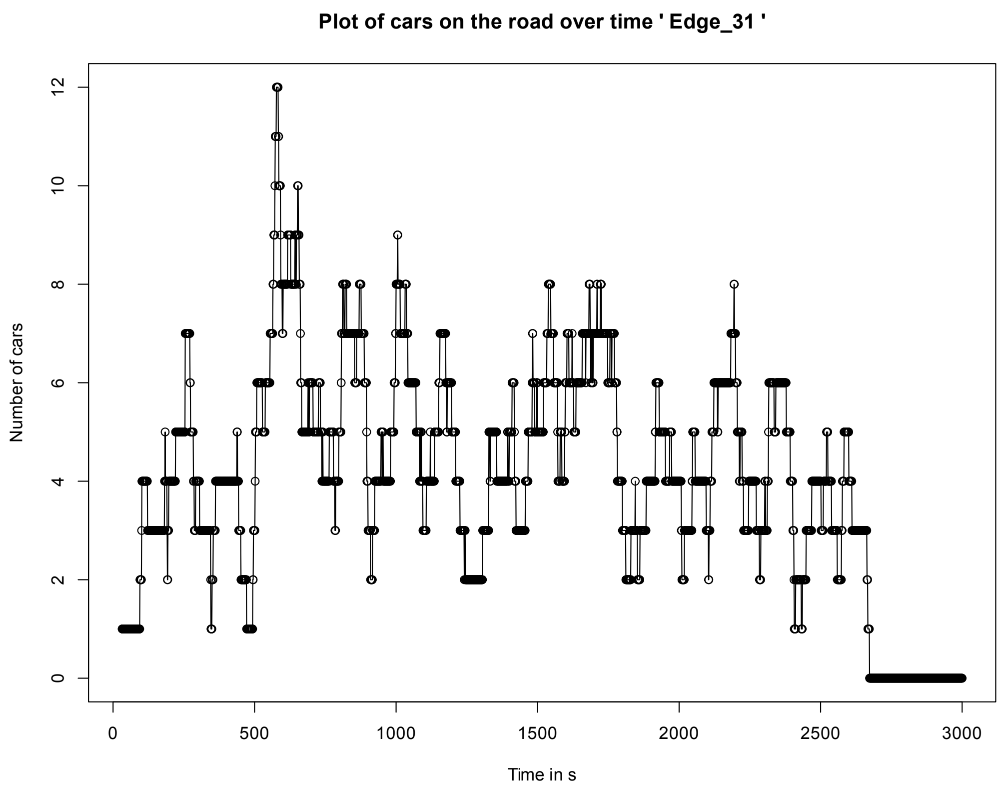

Figure 4.16: A graph showing the number of cars on a road over time for the small real map with static shortest distance routing.

The following three figures, Figure 4.17, Figure 4.18 and Figure 4.19 show roads from the large real map with static shortest distance routing. These three plots are representative of all the road plots for the large real map, regardless of routing. The road in Figure 4.17 is used very frequently, the road in Figure 4.18 less frequently and the road in Figure 4.19 almost never.

This ties in with the discussion in section 4.1.1 on how the utilisation of the available roads is affected by the restriction on source and destination nodes. We would expect to see this type of usage pattern. The large fabricated map is included here again in Figure 4.20 to help explain this type of use. Clearly in this simulation the model run time was insufficient for all the vehicles to finish their routes, however, from the increasing and decreasing numbers of vehicles on the roads, we still see that vehicles flow through the system unimpeded.

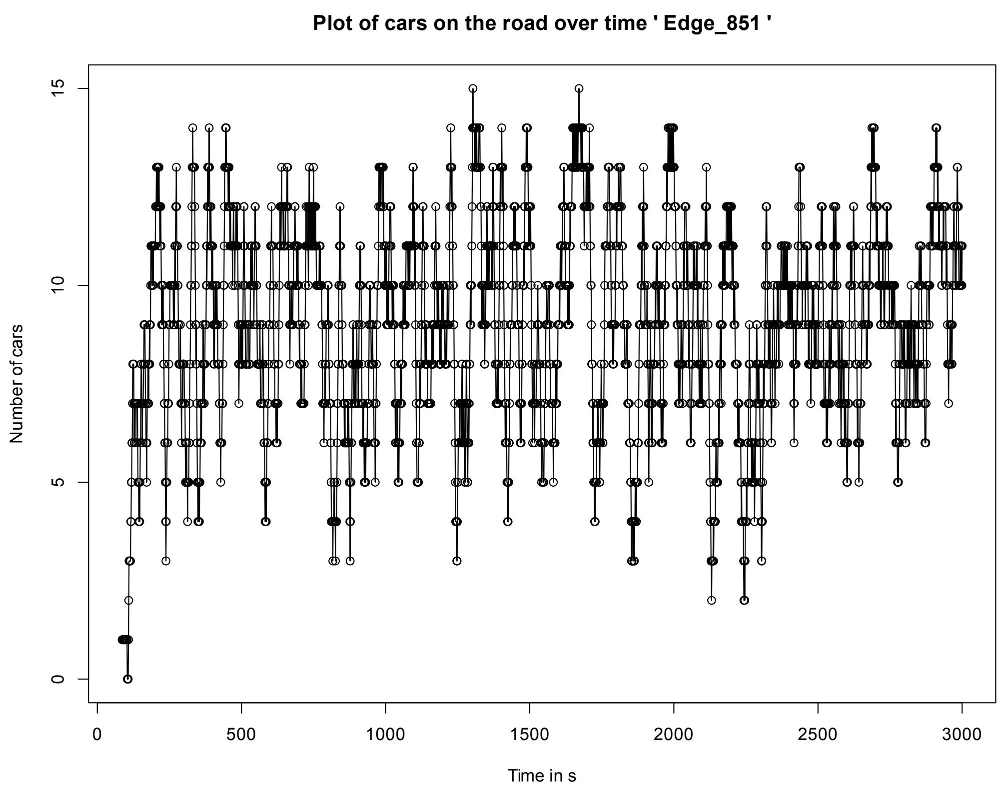

Figure 4.17: A graph showing the number of cars on a road over time for the large fabricated map with static shortest distance routing.

Figure 4.18: A graph showing the number of cars on a road over time for the large fabricated map with static shortest distance routing.

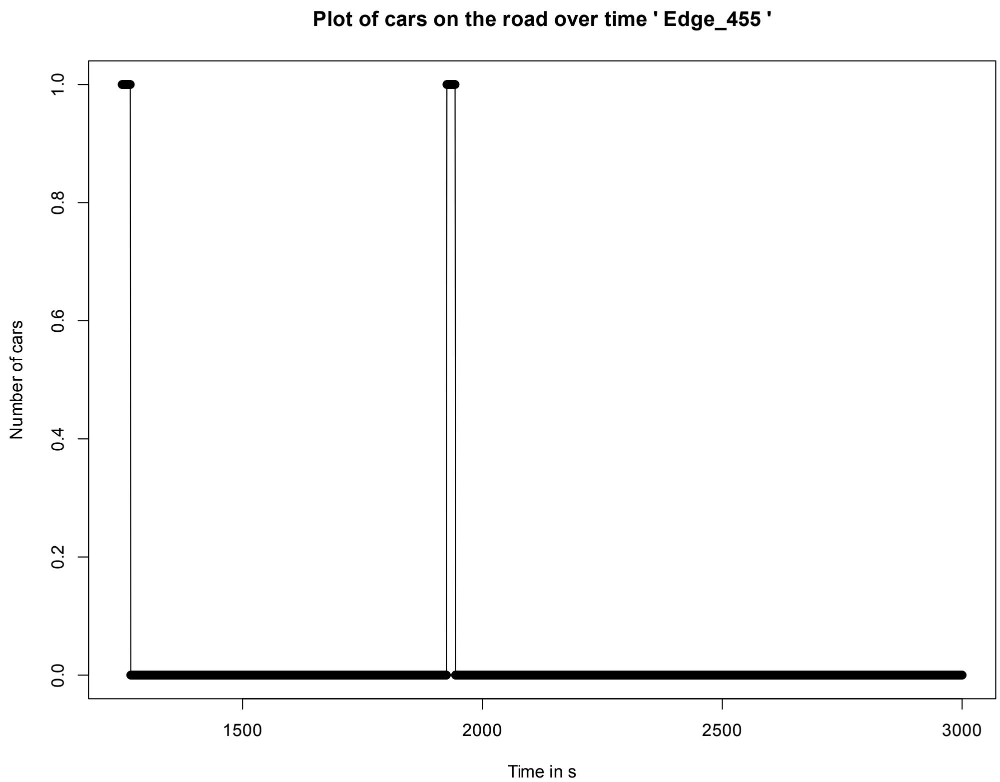

Figure 4.19: A graph showing the number of cars on a road over time for the large fabricated map with static shortest distance routing.

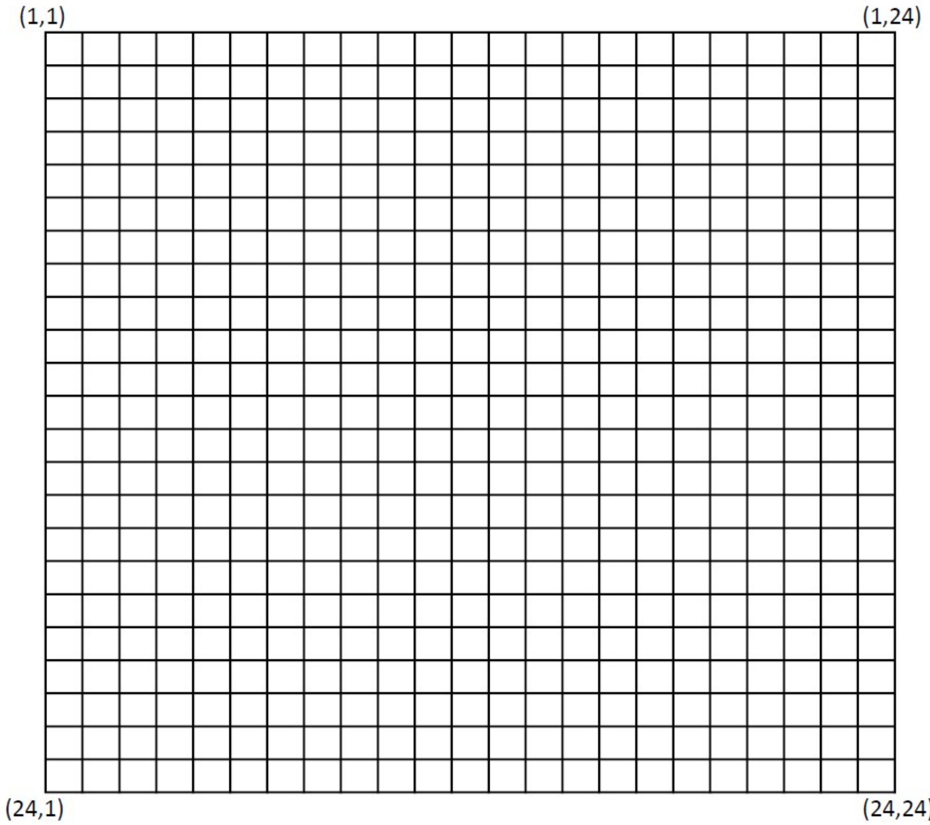

Figure 4.20: A large grid map.

4.2 Macroscopic model

The output from the macroscopic model is the value of the density, u at each node at each time step, that is, how much traffic there is at a certain point in the network at a given time. Plotting this density against time shows us how traffic is moving through the system with time. Diffusion in this model could be interpreted as mimicking the dynamic lowest congestion routing in microscopic model, when vehicles change their route based on local knowledge of congestion they are essentially moving toward areas of lower concentration. This, coupled with the fact that the same systems were implemented in both models leads us to expect that the results in this section will support the results from the agent-based model which we discussed in the previous section.

Plotting the density against time results in a similar way of looking at the system as in section 4.1.3, where the flow of vehicles in the agent-based model was discussed. We can see the movement of traffic through the network and we notice in the figures below that the number of vehicles at a given point in the network varies with time, just as illustrated in Figure 4.15. The same restrictions related to the smaller map size discussed previously, apply here. There is not a lot of space for the dispersion of traffic to take place. Here we have no discernable routes like we had for the agent-based model.

The iterative solution of the initial value problem on the small fabricated map is shown in Figure 4.21 below. The x-axis is time t, the y-axis is density u and each line on the graph represents one of the nodes of the network. The graph shows the convergence of this system to a steady state where the density is uniformly distributed across the nodes in the network. Dense traffic at a node in the initial state is dissipated across the network in time. This could be attributed to diffusion.

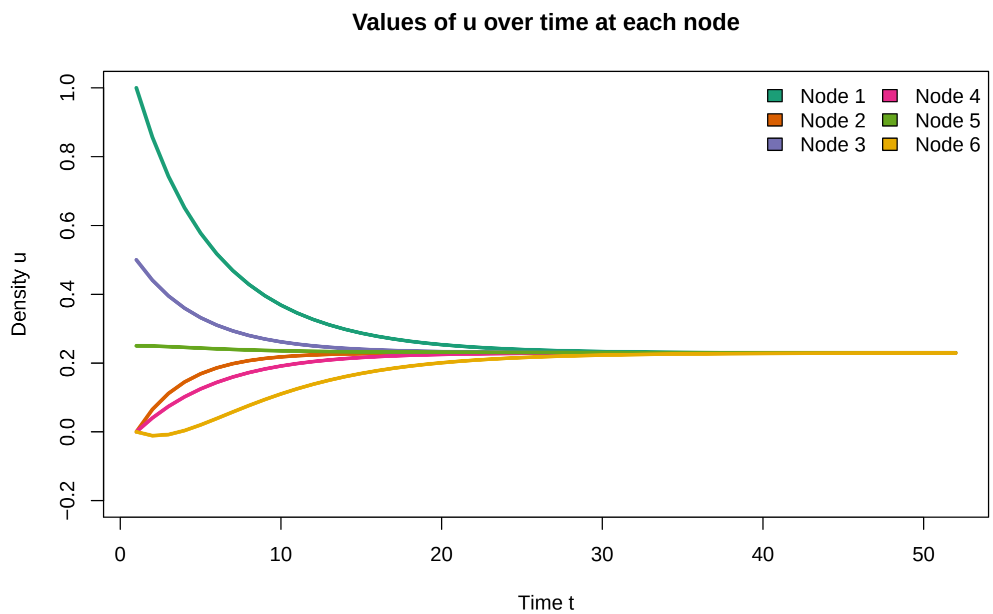

Figure 4.21: A graph of the changes in the value of u over time for each node for the small fabricated road network.

The iterative solution of the initial value problem on the small real map is shown in Figure 4.22 below. The x-axis is time t, the y-axis is density u and each line on the graph represents one of the nodes of the network. The graph shows that at each node, the solution takes the form of a periodic oscillation with an increasing amplitude. We see traffic increasing and decreasing at each node as we saw for the microscopic model in section 4.1.3. Also in that section, we mentioned that some roads in the small real map have a higher utilisation than others, see Figure 4.16. Here we see that some nodes experience greater fluctuations in traffic than others, which have relatively stable traffic.

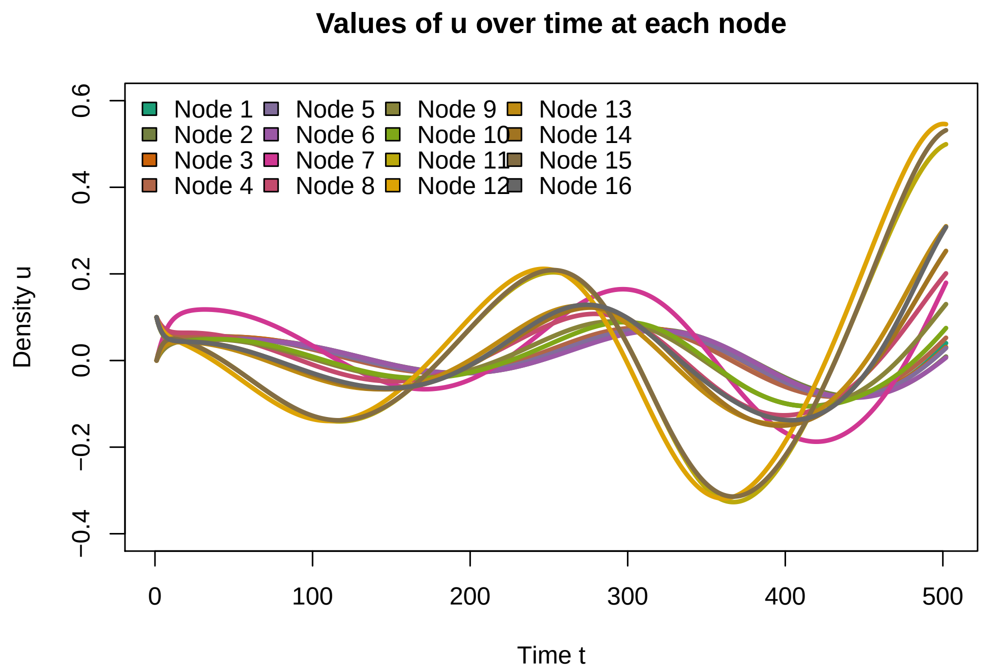

Figure 4.22: A graph of the changes in the value of u over time for each node for the small real road network.

Figure 4.23 plots density u on the z-axis and time t and node x on the x- and y-axis, respectively. The nodes on the y-axis are the 16 nodes in the network. The scale for time t on the x-axis is the same as the time scale in Figure 4.22 above, from 0 to 500. The same scale as in Figure 4.22 is also applicable to the values of u, on the y-axis in Figure 4.22 and on the z-axis in Figure 4.23. This graph is similar to the one above when viewed into the yz-plane. From the perspective into the xz-plane it shows the evolution of the solution u over time. We can see periodic increases and decreases in the density of the network as a whole. Traffic density present in the initial state of the network dissipates and then the traffic builds again and dissipates again. A uniform distribution of traffic density across the network is not achieved as with the small fabricated map and the uniform grid. This is likely due to the differences in network architecture between the maps as discussed at the end of this section.

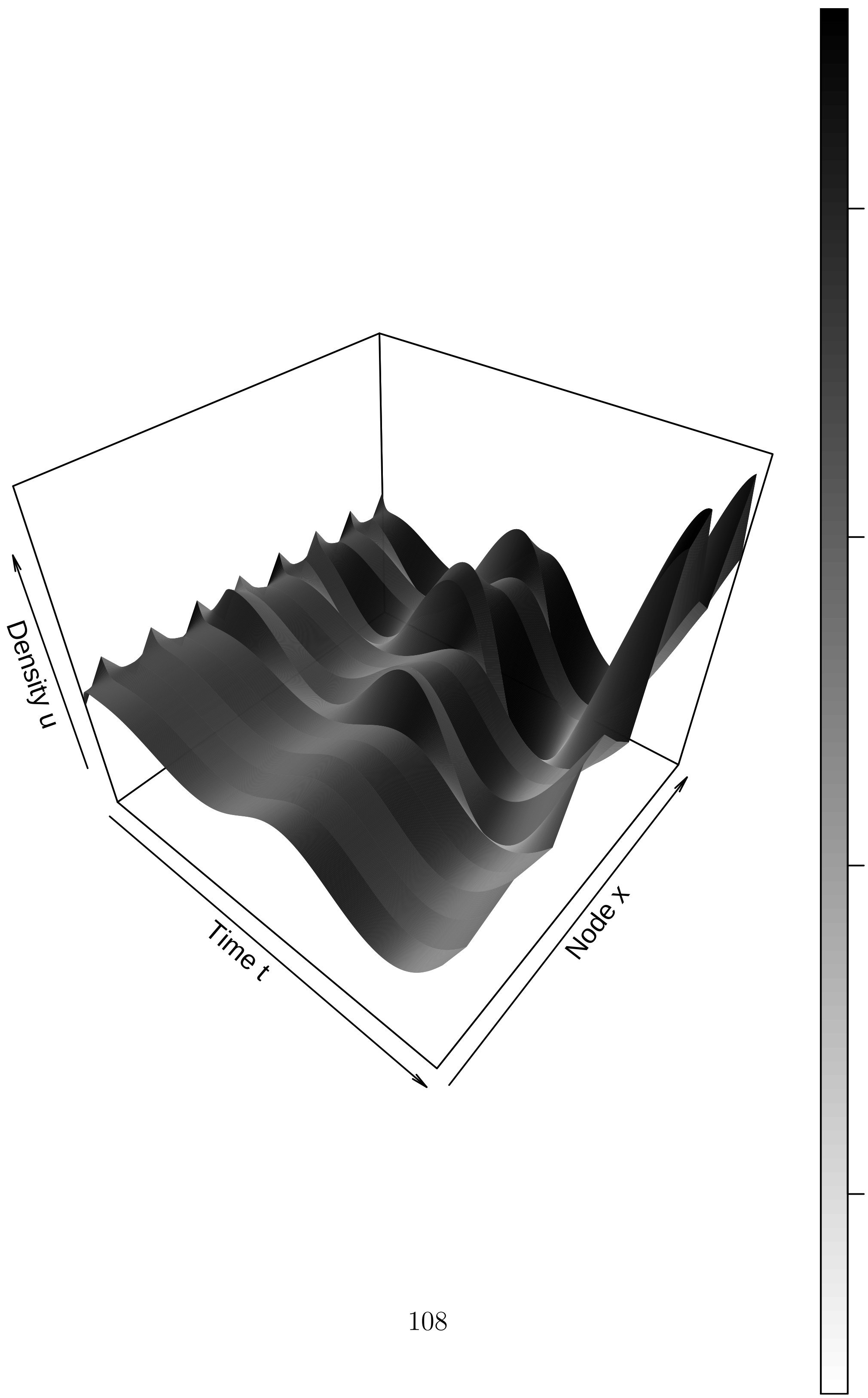

Figure 4.23: A graph of the changes in the value of u over time and space for the small real road network.

Below, in Figure 4.24 the same data as in Figure 4.23 is plotted in a similar way. The axes and scales are the same, but the values at each node is shown distinctly.

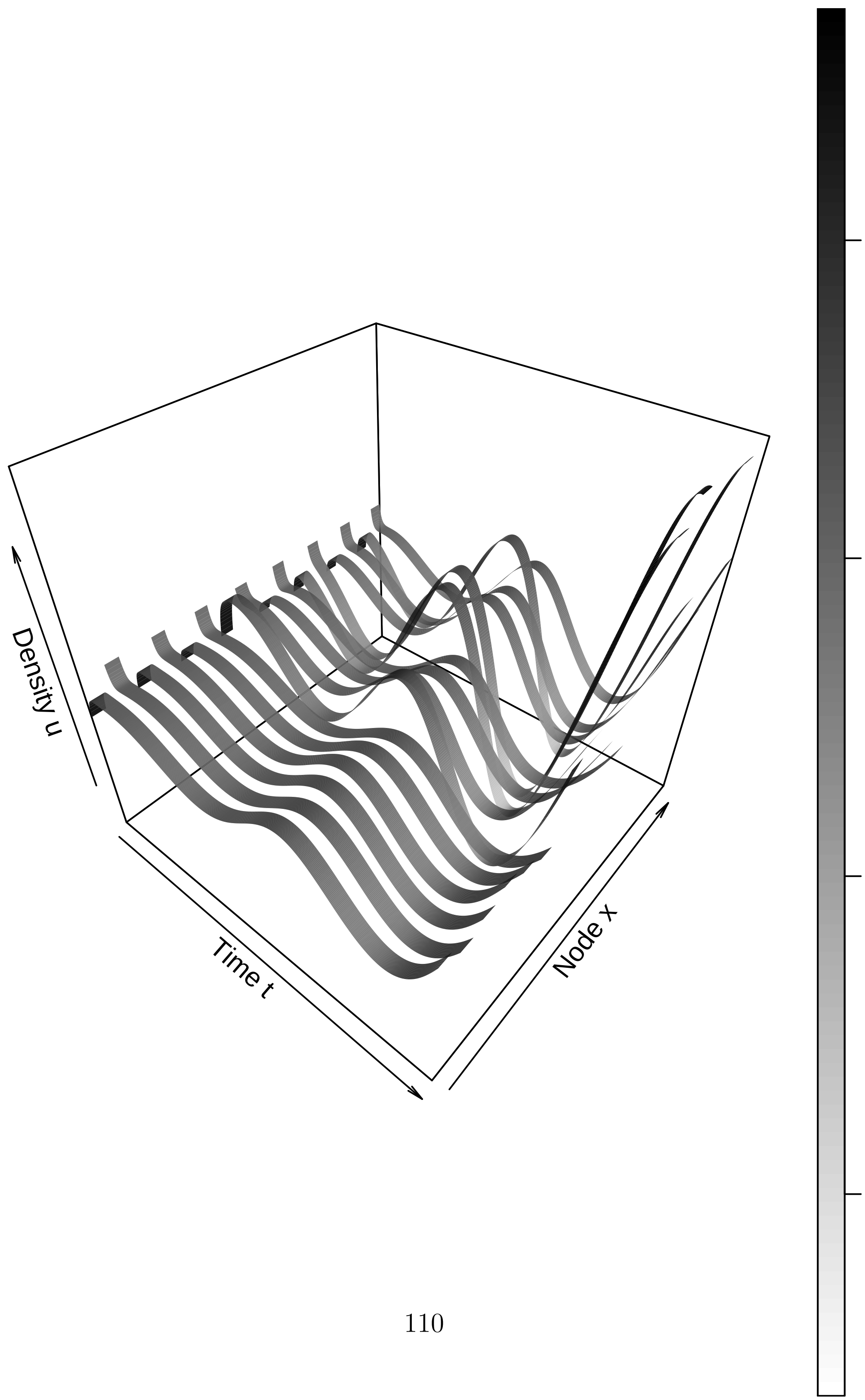

Figure 4.24: A graph of the changes in the value of u over time for each node for the small real road network.

Next the plots for the 16 node uniform grid is shown. In Figure 4.25 the x-axis is time t, the y-axis is density u and each line on the graph represents one of the nodes of the network. The graph shows that the traffic density across the network increases and then dissipates converging to a uniform distribution of density across the nodes of the network.

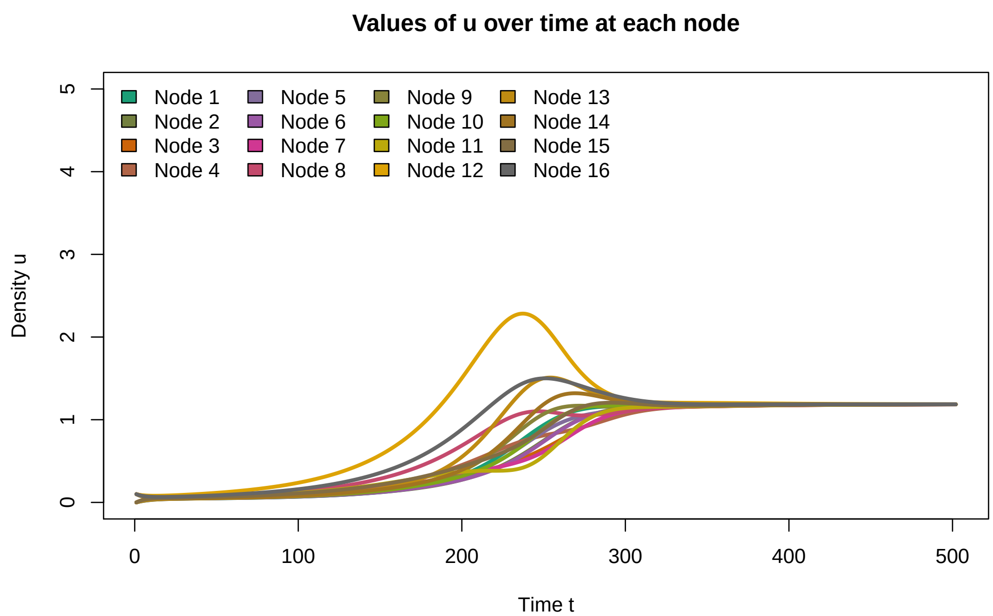

Figure 4.25: A graph of the changes in the value of u over time at each node for the 16 node uniform grid.

In Figure 4.26 density u is shown on the z-axis and time t and node x on the x- and y-axis, respectively. The nodes on the y-axis are the 16 nodes in the network. The scale for time t on the x-axis is the same as the time scale in Figure 4.25 above, from 0 to 500. The same scale as in Figure 4.25 is also applicable to the values of u, on the y-axis in Figure 4.25 and on the z-axis in Figure 4.26. This graph is similar to the one above when viewed into the yz-plane. From the perspective into the xz-plane it shows the evolution of the solution u over time. Traffic density present in the initial state of the network dissipates and then the traffic builds again and dissipates again. A uniform distribution of traffic density across the network is achieved.

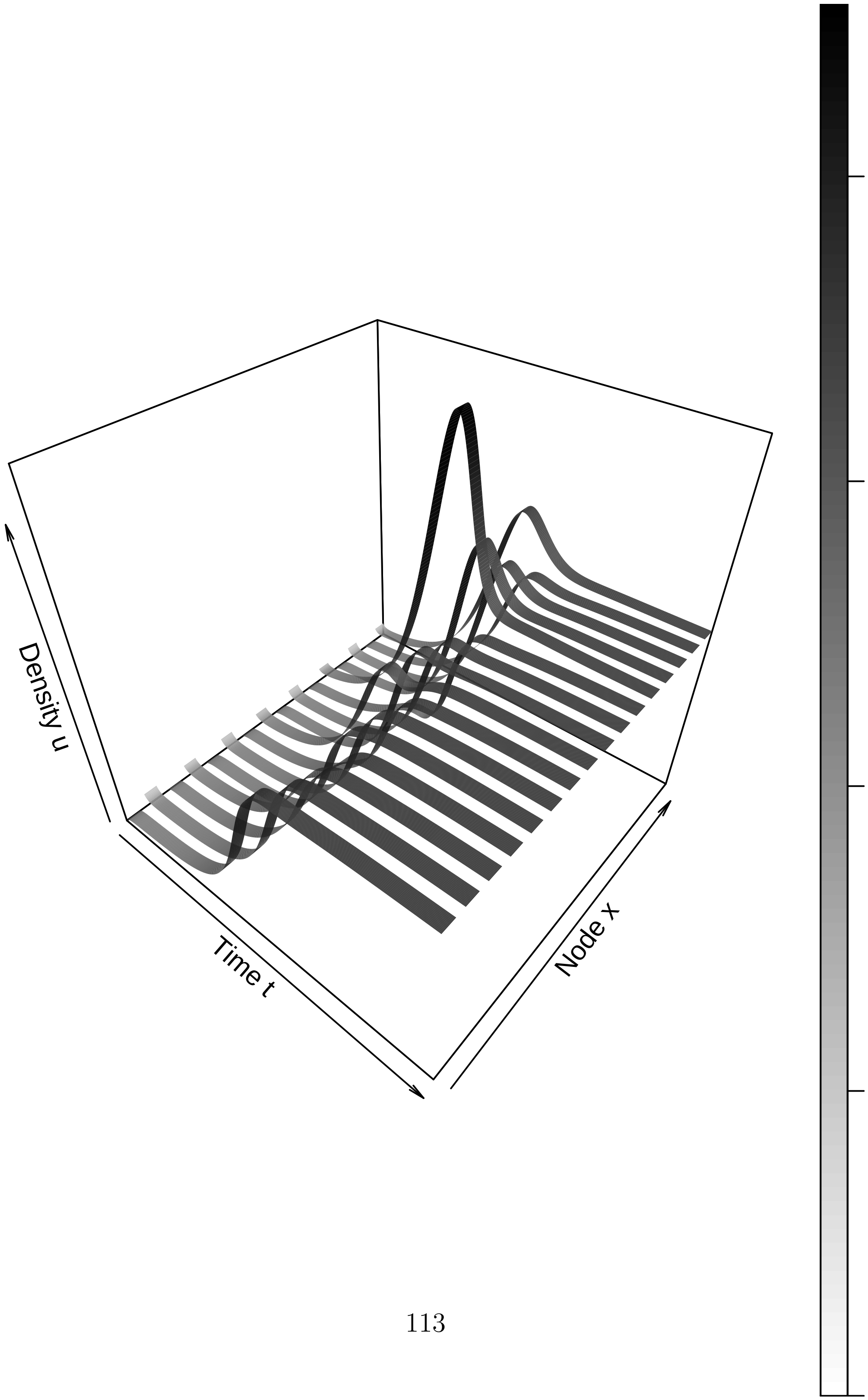

Figure 4.26: A graph of the changes in the value of u over time for each node for the 16 node uniform grid.

When we look at these figures we see that for the small fabricated map and the uniform grid, the traffic is spread evenly across the nodes after a while, remaining unchanged thereafter, but for the small real network, the traffic density continues to rise and fall and it does not get evenly distributed across the nodes. Why do we not see the same kind of spread here? If we go back to Figures 2.6, 3.1 and 3.2, we see that for the small fabricated map and uniform grid most, if not all, of the nodes are connected to multiple other nodes. When we look at the small real map, we see that the majority of the nodes are endpoint nodes connected to only one other node. Possibly, the way in which the map limits were chosen causes less freedom of movement resulting in areas of congestion. When too many vehicles accumulate at an endpoint, they only have one other node to go to. There are ten endpoint nodes in the small real map and six middle nodes. In Figure 4.24 you can see six nodes where there is a smaller variation in density over time and ten with greater variation. The variation at the endpoint nodes leads to variation at the middle nodes, but here there are more choices of nodes to move to and variation is diminished. This suggests that the network architecture plays an important role in the behaviour of the solution and that it should be chosen with care.

Chapter 5

Conclusions

Where does this research fit into the framework of traffic modelling? We know from Hoogendoorn and Bovy [12] and Van Wageningen-Kessels et al. [39] that traffic models are usually categorised according to their level of detail as microscopic, mesoscopic or macroscopic. In microscopic models individual vehicles are traced, in mesoscopic models vehicles are seen as aggregates of particles and in macroscopic models the flow of traffic is modelled as a fluid flow. Hoogendoorn and Bovy [12] also mention that microscopic models often describe microscopic attributes of traffic, while macroscopic models better describe the macroscopic attributes of traffic flow. The agent-based model that was developed falls into the category of microscopic models as it distinguishes vehicles and the behaviour of an individual can be singled out. The partial differential equation model is a macroscopic model, looking at traffic as a flow. In this chapter we will contemplate the relation of these models to existing traffic models.

5.1 Microscopic model

The agent-based modelling paradigm is well suited to traffic operations. According to Chen and Chang [13], these models are powerful applied to large scale dynamic environments such as traffic systems. Also, Manley et al. [28] note that in urban road traffic the macroscopic flow of traffic is a direct result of the microscopic behaviours and interactions of many individuals and therefore agent-based modelling can represent urban traffic systems realistically. Mastio et al. [23] argue that the real time information available to drivers increases the complexity of traffic, making it difficult to predict and limiting the applicability of more traditional approaches. The availability of this information and the drivers' reaction to it can easily be incorporated in an agent-based model as was done in this study.

In our study the vehicles were represented as agents. This is a common approach [13, 28, 32] and as Mastio et al. [23] state, it is relevant for agent-based simulations of traffic to be centered on individuals as their reactions and behaviour is what drives the traffic patterns observed on a higher level. Manley et al. [28] also model the road segments as agents. In Weyns et al. [24] vehicles have delegate agents that investigate traffic conditions and communicate with each other so that vehicle routing becomes a colloborative process. Rothkranz [25] gives each driver a personal intelligent travelling assistant system and these systems are the agents in their model. Many other studies use agents to manage operations in their models or as traffic control instruments [13].

Route or path planning is an integral part of most agent-based simulation models of traffic. Tucnik et al. [26] provide an overview of algorithms used to this end. They state that Dijkstra's algorithm is seen as one of the best for finding the shortest route, although it can be slow for large networks. This is the algorithm we employed in our model to calculate the static shortest distance route and which was adapted by means of weighting according to length and congestion for the dynamic routing scenarios. An adapted form of Dijkstra's algorithm was also used by Raney and Nagel [14], Raney et al. [32], and Zhao et al. [30].

Manley et al. [28] argue that people are mostly unable to identify the fastest route, as they are bounded by their own knowledge and memory. They therefore propose that a model where vehicles take the fastest routes to their destinations is unrealistic. In our static routing scenario, agents take the shortest distance route, not the fastest one, to their destinations. In this simplified way we aimed to capture the fact that people often take a suboptimal route to their destination due to their inability to know which route would be fastest, but they would still take a route closer to the optimal route, based on their existing knowledge and memory. In Raney and Nagel [14] and Raney et al [32], the authors incorporate driver's memory of route performance to select better future routes. As Mastio et al. [23] note, there is a lot of real time traffic information available to drivers currently. Weyns et al. [24] envision a system where vehicles, traffic lights and traffic control systems all communicate to reach mutual beneficial routing for all. Such a system would require intelligent traffic systems and vehicles. In our dynamic routing scenario we look at an alternative with more accessible technology - what if everyone used their mobile navigation systems to find the fastest route? It is likely that this approach will not result in the best possible scenario for everyone, as could be reached through collaboration, but that there will be a benefit in reducing the congestion of the system using existing and readily accessible technology. We see an improvement in our model from dynamic routing. Rothkranz [25] report a definite improvement in their model when using dynamic rather than static routing.

5.2 Macroscopic model

The macroscopic model we studied is based on partial differential equations and follows the idea that Lighthill, Whitham and Richards [8, 17] had when they originally started modelling traffic - that the flow of traffic can be described in terms of fluid dynamics. Like the seminal Lighthill-Whitham-Richards model, our model is also a first order model in the sense that it comprises one equation. All the first models for traffic, regardless of their order, were defined on a single stretch of roadway and many insights concerning traffic flow, especially on highways were gained from these. However, traffic flow and particularly congestion is not restricted to long straight stretches of highway, but frequently occur in the dense networks of roads in urban centres. Naturally, traffic models should also be applicable here. A common approach is to apply the fluid model to each individual road in the network and then using some extra information to solve the problem at the intersections. Helbing et al. [44] call setting this kind of model for networks that are irregular or complex challenging, notably concerning boundary conditions at the road junctions. By including interactions at the intersections in detail in their model, they managed to see behaviour as would be expected at traffic lights. Studying irregular networks, Rarità et al. [51] take a two-phased approach, first applying the Lighthill-Whitham-Richards model on the roads and then solving at the intersections through the introduction of permeability parameters. Bretti et al. [52] have a similar approach, but include a right of way parameter to solve Riemann problems at junctions uniquely. In our model we included the network in a simpler way, following the approach in Angstmann et al. [6]. In this manner even complex, irregular networks where nodes can have multiple connections and need not all have the same number of connections, can be easily incorporated into the model. This approach gives less information about interactions at intersections than the aforementioned approaches, but is much simpler and faster to implement. This can be beneficial if a higher level model is required in a short amount of time.

The equation used to describe the traffic density across the network in our model, is the viscous Burgers' equation. The Burgers' equation as a model for traffic flow was derived from the well-known mesoscopic Reduced Paveri-Fontana equation, a gas-kinetic model of traffic flow by Velasco and Saavedra [60]. Binatari [59] applied the equation to traffic and found that obtaining the exact solution was more complicated than solving the problem numerically. A numerical method for solving the Burgers' equation based on the Crank-Nicolson method was proposed by Wani and Thakar [7] and this was the numerical method we employed for solving our model. According to Zhang [56] viscosity can be linked to the effect of driver memory in traffic models. Li [54] added a diffusion term to the Lighthill-Whitham-Richards model reflecting drivers' ability to look ahead and respond to nonlocal changes in traffic conditions. This corresponds to our expectation that the diffusion in our traffic flow model has a similar effect to lower congestion routing. It also affirms our interpretation of the results where we see an eventual even spread of traffic that we see as attributable to this diffusion.

5.3 A microscopic and a macroscopic approach

According to Hoogendoorn and Bovy [12] the correct level-of-detail of a traffic model depends on its application. Macroscopic models are better at describing macroscopic features of traffic flow and microscopic models better at describing microscopic details. There are other benefits and disadvantages to each of these modelling approaches as we have also found in our study. Though the idea of the agent-based simulation model is simple, it was complicated and time-consuming to program and a lot of processing was needed to process its inputs and outputs. A lot of information could be gained from this model and it is much closer to a real traffic system. Specifying the various routing approaches was relatively easy in this model. Conversely, the partial differential equation model of the system and the finite difference method used to approximate its solution was simpler to derive and program, although the concept of the model is more sophisticated. Existing methodologies could be harnessed for the development of this model. The macroscopic model is further removed from reality and it is difficult to specify details of the system. The results are also more complex to interpret, especially in terms of the realistic road network.

Even though some of the same road networks were modelled in the microscopic and macroscopic models, the results do not seem cohesive. Future work could include ensuring stronger alignment of the models. Sometimes, microscopic and macroscopic models can be derived from each other, such as was done by Sun et al. [55]. They derived a continuum traffic model from a microscopic car following model and state that this model is more realistic and meaningful than a continuum model not arrived at in this way. Aw et al. [46] and Moutari and Rascle [47] take this a step further: after deriving a continuum traffic flow model from a microscopic follow-the-leader model, they combine the two models into one hybrid model. Van Wageningen-Kessels et al. [39] deem this area of traffic models, namely hybrid models, an important focus area for the development of future traffic models.

Appendix A

agent-based model documentation

A.1 *Edge* class

A.1.1 Description

Objects of this class represent the edges of the network, which in turn represent roads.

A.1.2 Properties

Properties of the *Edge* class.

```
String id
int weight
```

```
String EdgeType
double capacity
double congestion
Vertex source
Vertex destination
```

A.1.3 Property descriptions

String *id*

```
setId
getId
```

Identifies the edge.

int *weight*

```
setWeight
getWeight
```

Gives the length of the edge.

String *EdgeType*

```
setEdgeType
getEdgeType
```

Either *StopDirection* or *NonStopDirection*. Used to determine stopping behaviour of a vehicle on this road when it reaches the next intersection.

double *capacity*

```
setCapacity
getCapacity
```

Calculated by dividing *weight* by the vehicle length used in the model, where the vehicle length encapsulates the actual length of the vehicle as well as the set minimum following distance. The number of vehicles that can be accommodated on this road if all vehicles are at the minimum following distance from each other.

double *congestion*

```
setCongestion
getCongestion
```

Calculated by dividing the number of vehicles on the road by its *capacity*.

Vertex *source*

```
getSource
setSource
```

Identifies the node at the start of the edge.

Vertex *destination*

```
setDestination
getDestination
```

Identifies the node at the end of the edge.

A.2 *Vertex* class

A.2.1 Description

Objects of this class represent the edges of the network, which in turn represent roads.

A.2.2 Properties

```
String id
String name
String IntersectionType
```

A.2.3 Property descriptions

String *id*

```
setId
getId
```

Identifies the node and contains only numerical characters.

String *name*

setName

getName

Identifies the node.

String *IntersectionType*

```
setIntersectionType
getIntersectionType
```

Gives the type of intersection: *TrafficLight*, *AllWayStop* or *SingleWayStop*.

A.3 *Graph* class

A.3.1 Description

Associates a given list of nodes with a given list of edges.

```
List<Vertex> vertexes
List<Edge> edges
```

A.3.2 Properties

A.3.3 Property descriptions

List⟨Vertex⟩ *vertexes*

A list of all the nodes or vertexes.

List⟨Edge⟩ *edges*

A list of all the edges.

A.4 *DijkstraAlgorithm* class

A.4.1 Description

Implements the Dijkstra algorithm for a given map consisting of nodes and edges and specific source and destination nodes to return a path. This path is an ordered list of nodes and represents the shortest path between the source and destination nodes as measured by the weights assigned to the edges.

A.4.2 Properties

```
List<Vertex> nodes
List<Edge> edges
Set<Vertex> settledNodes
Set<Vertex> unSettledNodes
Map<Vertex, Vertex> predecessors
Map<Vertex, Integer> distance
```

A.4.3 Methods

```
DijkstraAlgorithm
void execute
void fineMinimalDistances
int getDistance
List<Vertex> getNeighbors
```

```
Vertex getMinimum

boolean isSettled

int getShortestDistance

LinkedList<Vertex> getPath
```

A.4.4 Property descriptions

List⟨Vertex⟩ *nodes*

All the nodes in the network.

List⟨Edge⟩ *edges*

All the edges in the network.

Set⟨Vertex⟩ *settledNodes*

Nodes already settled into place.

Set⟨Vertex⟩ *unSettledNodes*

Nodes not settled yet.

Map⟨Vertex, Vertex⟩ *predecessors*

Defines relationships between nodes.

Map⟨Vertex, Integer⟩ *distance*

Associates lengths.

A.4.5 Method descriptions

DijkstraAlgorithm(Graph *graph*)

Creates a copy of the array of nodes and edges to work on.

void execute(Vertex *source*)

Moves nodes from *unSettledNodes* to *settledNodes* in order and calls *findMinimalDistance*.

void findMinimalDistance(Vertex *node*)

Finds minimal distances between nodes. Calls *getDistance*.

int getDistance(Vertex *node*, Vertex *target*)

Gets distance between nodes from edge weights.

List⟨Vertex⟩ getNeighbors(Vertex *node*)

Gets a node's neighbours.

Vertex getMinimum(Set⟨Vertex⟩ *vertexes*)

Get the minimum node using *getShortestDistance*.

boolean isSettled(Vertex *vertex*)

Returns whether the node is in the set of settled nodes.

int getShortestDistance(Vertex *destination*)

Gets the shortest distance associated with the destination node.

LinkedList⟨Vertex⟩ getPath(Vertex *target*)

Checks whether a path exists and orders it.

A.5 *ABSimulation* class

A.5.1 Description

Contains the main method. Creates objects and performs most initialisation. Runs the simulation and writes the output.

A.5.2 Properties

```
int NumberOfAgents

int[] InterarrivalTimes

Random randomNumbers

int IterationDuration

int NumberOfNodes

int NumberOfEdges

ArrayList<LinkedList> ListOfAllRoutes

LinkedList<Integer>[] AllStopQueues

LinkedList<Integer>[] AllNonStopQueues

LinkedList<Integer>[] RoadQueues

ArrayList<Vehicle> allVehicles

List<Vertex> nodes

List<Edge> edges

Graph graph

int carLength

Formatter vehicleOutput

Formatter roadOutput

Formatter trafficLightOutput

Formatter routeOutput
```

A.5.3 Methods

```
void main
void openFiles
void InitialiseNetworkArchitecture
void CreateAgents
void addLane
```

A.5.4 Property descriptions

int *NumberOfAgents*

The number of agents in the model.

int[ ] *InterarrivalTimes*

The agents' interarrival times.

Random *randomNumbers*

Random number generator.

int *IterationDuration*

The duration of the simulation.

int *NumberOfNodes*

The number of nodes in the network.

int *NumberOfEdges*

The number of edges in the network.

ArrayList ⟨LinkedList⟩ *ListOfAllRoutes*

A list of all the routes created for agents.

LinkedList⟨Integer⟩[ ] *AllStopQueues*

All the queues associated with edges of which the *EdgeType* is *stopDirection.*

LinkedList⟨Integer⟩[ ] *AllNonStopQueues*

All the queues associated with edges of which the *EdgeType* is *nonStopDirection.*

LinkedList⟨Integer⟩[ ] *RoadQueues*

All the queues associated with edges.

ArrayList⟨Vehicle⟩ *allVehicles*

All the vehicles in the model.

List⟨Vertex⟩ *nodes*

All the nodes in the model.

List⟨Edge⟩ *edges*

All the edges in the model.

Graph *graph*

Combines the nodes and edges. See *graph* class.

int *carLength*

The length of a vehicle in the model. This includes minimum following distance.

Formatter *vehicleOutput*

Writes to the text file *Vehicles.txt.*

Formatter *roadOutput*

Writes to the text file *Roads.txt.*

Formatter *trafficLightOutput*

Writes to the text file *TrafficLights.txt.*

Formatter *routeOutput*

Writes to the text file *Routes.txt*.

A.5.5 Method descriptions

void main(Graph *String args[ ]*)

Calculates interarrival times. Calls *openFiles*, *InitialiseNetworkArchitecture* and *CreateAgents*. Generates source and destination nodes and calls *DijkstraAlgorithm* to calculate routes. Writes to text files. Assigns vehicles to queues. Performs the simulation - at every time step, updates traffic lights if applicable, calls *updateVehicleLocationTimeAndSpeed* for each vehicle and calculates congestion.

void openFiles()

Opens text files for output.

void InitialiseNetworkArchitecture(int *NumberOfNodes*)

Creates nodes and edges and sets their initial properties.

void CreateAgents(int *NumberOfAgents*, ArrayList ⟨LinkedList⟩ *ListOfAllRoutes*, Graph *graph*)

Creates agents and sets their initial properties.

void addLane(String *laneId*, int *sourceLocNo*, int *destLocNo*, int *duration*, String *laneType*)

Adds new edges.

Bibliography

[1] Grote, M., Williams, I., Preston, J. and Kemp, S., *Including congestion effects in urban road traffic CO2 emissions modelling: Do Local Government Authorities have the right options?*. Transportation Research Part D. 2016; **43**:95-106.

[2] Weisbrod, G., Vary, D. and Treyz, G., *Measuring Economic Costs of Urban Traffic Congestion to Business*. Transportation Research Record: Journal of the Transportation Research Board. 2003; **1839**(1):98-106.

[3] Schrank, D., Eisele, B and Lomax, T., *2019 Urban Mobility Report*. The Texas A&M Transportation Institute. 2019.

[4] Levy, J. I., Buonocore, J. J. and Von Stackelberg, K., *Evaluation of the public health impacts of traffic congestion: a health risk assessment*. Environmental Health. 2010; **9**(1):65.

[5] Barth, M. and Boriboonsomsin, K., *Real-World Carbon Dioxide Impacts of Traffic Congestion*. Transportation Research Record: Journal of the Transportation Research Board. 2008; **2058**(1):163-171.

[6] Angstmann, C. N., Donnelly, I. C. and Henry, B. I., *Pattern formation on networks with reactions: A continuous-time random-walk approach.* Physical Review E. 2013; **87** 032804.

[7] Wani, S. S. and Thakar, S. H., *Crank-Nicolson Type Method for Burgers Equation.* International Journal of Applied Physics and Mathematics. 2013; **3**(5):324-328.

[8] Lighthill, M. J., and Whitham, G. B., *On Kinematic Waves. II. A Theory of Traffic Flow on Long Crowded Roads.*. Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences. 1955; **229**(1178):317-345.

[9] Payne, H. J., *Models of freeway traffic and control*, Mathematical Models of Public Systems. 1971; **1**:51–61.

[10] Daganzo, C., *Requiem for second-order fluid approximation to traffic flow*, Transportation Research. 1975; **29**(B):277-286.

[11] Aw, A. and Rascle, M. *Resurrection of second order models of traffic flow?*, SIAM J. Appl. Math. 2000; **60**:916-938.

[12] Hoogendoorn, S. P. and Bovy, P. H., *State-of-the-art of vehicular traffic flow modelling.* Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering. 2001; **215**(4):283-303.

[13] Chen, B. and Chang, H. H., *A Review of the Applications of Agent Technology in Traffic and Transportation Systems*. IEEE Transactions on Intelligent Transportation Systems. 2010; **11**(2):485-497.

[14] Raney, B. and Nagel, K., *Truly agent-based strategy selection for transportation simulations*. In 82nd Annual Meeting of the Transportation Research Board, Washington, DC. 2003.

[15] Goodwin, P., *The Economic Costs of Road Traffic Congestion*. ESRC Transport Studies Unit University College London. 2004.

[16] Zhang, H. M., *New perspectives on continuum traffic flow models.*. Networks and Spatial Economics. 2001; **1**(1-2):9-33.

[17] Richards, P. I., *Shock Waves on the Highway*. Operations Research. 1956; **4**(1):42-51.

[18] Stoilova, V., Nikolov, E. and Nikolova, N., *Analytical Deriving of Second Order Model of Payne from First Order Lighthil-Whitham-Richards Model*. Cybernetics and Information Technologies. 2013; **13**(4):54-62.

[19] Zhang, H. M. and Jin, W. L., *Kinematic wave traffic flow model for mixed traffic*. Transportation Research Record. 2002; **1802**(1):197-204.

[20] Zhang, H. M., *A non-equilibrium traffic model devoid of gas-like behavior*. Transportation Research Part B: Methodological. 2002; **36**(3):275-290.

[21] Helbing, D. and Johansson, A. F., *On the controversy around Daganzo's requiem for and Aw-Rascle's resurrection of second-order traffic flow models*. The European Physical Journal B. 2009; **69**(4):549-562.

[22] Zhang, H. M., *Comment on "On the controversy around Daganzo's requiem for and Aw-Rascle's resurrection of second-order traffic flow models" by D. Helbing and AF Johansson*. The European Physical Journal B. 2009; **69**(4):563-568.

[23] Mastio, M., Zargayouna, M., Scemama, G. and Ranab, O., *Two distribution methods for multiagent traffic simulations*. Simulation Modelling Practice and Theory. 2018; **89**:35–47.

[24] Weyns, D., Holvoet, T. and Helleboogh, A., *Anticipatory vehicle routing using delegate multi-agent systems.* In 2007 IEEE Intelligent Transportation Systems Conference, pp. 87-93. IEEE, 2007.

[25] Rothkrantz, L.J.M., *Dynamic routing using the network of car drivers.* In Proceedings of the 2009 Euro American Conference on Telematics and Information Systems: New Opportunities to increase Digital Citizenship, pp. 1-8. 2009.

[26] Tucnik, P., Nachazel, T., Cech, P. and Bures, V., *Comparative analysis of selected path-planning approaches in large-scale multi-agent-based environments*. Expert Systems with Applications. 2018; **113**:415-427.

[27] Dijkstra, E. W., *A Note on Two Problems in Connexion with Graphs.* Numerische Mathematik. 1959; **1**:269-271.

[28] Manley, E., Cheng, T., Penn, A. and Emmonds, A., *A framework for simulating large-scale complex urban traffic dynamics through hybrid agent-based modelling.* Computers, Environment and Urban Systems. 2014; **44**:27-36.

[29] Grau, J. M. Salanova, and Romeu, M. A. E., *Agent based modelling for simulating taxi services.* Procedia Computer Science. 2015; **52**:902-907.

[30] Zhao, B., Kumar, K., Casey, G. and Soga, K., *Agent-Based Model (ABM) for City-Scale Traffic Simulation: A Case Study on San Francisco.* International Conference on Smart Infrastructure and Construction 2019 (ICSIC) Driving data-informed decision-making. ICE Publishing, 2019.

[31] El Fazziki, A., Benslimane, D., Sadiq, A., Ouarzazi, J. and Sadgal, M., *An agent based traffic regulation system for the roadside air quality control.* IEEE Access. 2017; **5**:13192-13201.

[32] Raney, B., Cetin, N., Völlmy, A., Vrtic, M., Axhausen, K. and Nagel, K., *An agent-based microsimulation model of Swiss travel: First results.* Networks and Spatial Economics. 2003; **3**(1):23-41.

[33] Scheltes, A. and De Almeida Correia, G.H., *Exploring the use of automated vehicles as last mile connection of train trips through an agent-*

based simulation model: An application to Delft, Netherlands. International Journal of Transportation Science and Technology. 2017; **6**(1):28-41.

[34] Ciechanowicz, D., Pelzer, D. and Knoll, A., *Simulation-based approach for investigating the impact of electric vehicles on power grids*. 2015 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC). IEEE, 2015.

[35] Sadiq, A., El Fazziki, A., Ouarzazi, J. and Sadgal, M., *Towards an agent based traffic regulation and recommendation system for the on-road air quality control*. SpringerPlus. 2016; **5**(1):1604.

[36] Van Duin, J. R., Van Kolck, A., Anand, N. and Taniguchi, E., *Towards an agent-based modelling approach for the evaluation of dynamic usage of urban distribution centres*. Procedia-Social and Behavioral Sciences. 2012; **39**:333-348.

[37] Balmer, M., Cetin, N., Nagel, K. and Raney, B., *Towards truly agent-based traffic and mobility simulations*. Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004. IEEE, 2004.

[38] Li, L. and Chen X. M., *Vehicle headway modeling and its inferences in macroscopic/microscopic traffic flow theory: A survey*. Transportation Research Part C: Emerging Technologies. 2017; **76**:170-188.

[39] Van Wageningen-Kessels, F., Van Lint, H., Vuik, K., and Hoogendoorn, S., *Genealogy of traffic flow models*. EURO Journal on Transportation and Logistics. 2015; **4**(4):445-473.

[40] Prigogine, I. and Herman, R., *Kinetic theory of vehicular traffic*. Transport Research International Documentation. 1971; **100**.

[41] Prigogine, I., *A Boltzmann-like approach to the statistical theory of traffic flow*. Theory of traffic flow. 1961.

[42] Paveri-Fontana, S. L., *On Boltzmann-like treatments for traffic flow: a critical review of the basic model and an alternative proposal for dilute traffic analysis*. Transportation research. 1975; **9**(4):225-235.

[43] Hoogendoorn, S. P. and Bovy, P. H. L., *Modelling multiple user-class traffic flow*. Transportation Research B. 2000; **34**(2):123-146.

[44] Helbing, D., Siegmeier, J. and Lämmer, S., *Self-organized network flows*. arXiv preprint physics/0702173. 2007.

[45] Hoogendoorn, S. P., *Multiclass continuum modelling of multiclass traffic flow*, PhD Thesis, Delft University of Technology, 1999.

[46] Aw, A., Klar, A., Rascle, M. and Materne, T., *Derivation of continuum traffic flow models from microscopic follow-the-leader models*. SIAM Journal on applied mathematics. 2002; **63**(1):259-278.

[47] Moutari, S. and Rascle, M., *A hybrid Lagrangian model based on the Aw–Rascle traffic flow model*. SIAM Journal on Applied Mathematics. 2007; **68**(2):413-436.

[48] Bourrel, E. and Lesort, J. B., *Mixing microscopic and macroscopic representations of traffic flow: Hybrid model based on Lighthill–Whitham–Richards theory*. Transportation Research Record. 2003; **1852**(1):193-200.

[49] Leclercq, L., *Hybrid approaches to the solutions of the "Lighthill–Whitham–Richards" model*. Transportation Research Part B: Methodological. 2007; **41**(7):701-709.

[50] De Marchi, A. and Gerdts, M., *Traffic Flow on Single-Lane Road Networks: Multiscale Modelling and Simulation*. IFAC-PapersOnLine. 2018; **51**(2):162-167.

[51] Rarità, L., D'Apice, C., Piccoli, B., Helbing, D., *Sensitivity analysis of permeability parameters for flows on Barcelona networks*. Journ. Diff. Equat. 2010; **249**(12):3110-3131.

[52] Bretti, G., Natalini, R. and Piccoli, B. *Numerical approximations of a traffic flow model on networks*. NHM. 2006; **1**(1):57-84.

[53] Kaddoura, I. and Nagel, K., *Using real-world traffic incident data in transport modeling*. Procedia Computer Science. 2018; **130**:880–885.

[54] Li, L. Y., *A Continuum Model of Traffic Flow on Road Networks*. The 3rd International Conference on Transportation Information and Safety, Wuhan, 2015.

[55] Sun, L., Jafaripournimchahi, A., Kornhauser, A. and Hu, W.,*A new higher-order viscous continuum traffic flow model considering driver memory in the era of autonomous and connected vehicles*. Physica A. 2020; **547**:123829.

[56] Zhang, H. M., *Driver memory, traffic viscosity and a viscous vehicular traffic flow model*. Transportation Research Part B: Methodological. 2003; **37**(1):27-41.

[57] Bateman, H., *Some recent researches on the motion of fluids*. Monthly Weather Review. 1915; **43**(4):163-170.

[58] Burgers, J. M., *A mathematical model illustrating the theory of turbulence*. Advances in applied mechanics. 1948; **1**:171-199.

[59] Binatari, N., *A comparison between Cole-Hopf Transformation and Homotopy Perturbation Method for Viscous Burger Equation in Traffic Flow*. J. Phys.: Conf. Ser. 2019; **1320** 012030.

[60] Velasco, R. M. and Saavedra, P., *A first order model in traffic flow*. Physica D: Nonlinear Phenomena. 2007; **228**(2):153-158.

[61] OpenStreetMap Foundation, *about*. https://www.openstreetmap.org/about [accessed 28 November 2020]

[62] OpenStreetMap Wiki contributors, *Main Page*, OpenStreetMap Wiki, 19 July 2020, https://wiki.openstreetmap.org [accessed 28 November 2020]

[63] Belsley, D.A., Kuh, E. and Welsch, R.E., *Regression diagnostics: Identifying influential data and sources of collinearity*. Vol. 571. John Wiley & Sons. 2005.