A Discrete Simulation Model for Heterogeneous Traffic Including Bicycles on Urban Road Networks

Jelena Vasić

MEng

A thesis submitted in partial fulfilment of

the requirements for the degree of

Doctor of Philosophy

to the

School of Computing

Faculty of Engineering and Computing

Dublin City University

Supervised by Prof. Heather Ruskin

January 2014

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of PhD is entirely my own work, and that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: ______________________________ (Candidate)

ID No.: ___________

Date: ___________

Dedicated

to my parents, Ann and Rastko,

my children, Ada and Luka, and

most of all, even though it goes without saying,

to Anthony

Acknowledgements

First of all, I would like to thank Prof. Heather Ruskin for supervising my efforts and teaching me most of what I know about scientific research, but also for always being available to read, even not-so-great, unfinished material, and provide invaluable input. I would like to thank all my colleagues in LG01, past and present, who made my time in DCU an interesting and enjoyable period and, in particular, Dr Marija Bezbradica who has given me excellent advice and much practical help.

I would like to acknowledge the Irish Research Council, who funded the first three years of my studies, through an Embark Initiative scholarship.

Finally, thanks to my family: my sister, Barbara, who has provided me with a calm working environment and dinners; her cats, Misquit and Mališa, who have been the right kind of company through the long working nights; Anthony, who has had to do just about everything lately and who has unquestioningly supported me in this endeavour from beginning to end; and to my children, Ada and Luka, who have been patient but will, I hope, soon be doing all those things left for "after Mammy finishes her PhD".

Contents

Abstract

A Discrete Simulation Model for Heterogeneous Traffic
Including Bicycles on Urban Road Networks

Jelena Vasić

Environment and health-related concerns mean that pedal-bicycles as an alternative mode of urban transport are gaining ground, with study of motorised/non-motorised traffic mix a topic of practical interest in transportation science and traffic modelling.

This thesis reports on a simulation model, developed for heterogeneous traffic on city networks with *ad hoc* lane-sharing, characteristic of Dublin streets. While based on simple cellular automaton rules, the vehicle movement model also accounts for vehicle type heterogeneity and network-specific factors, including the resolution of conflicts and effects of driver decisions on movement dynamics. The model has been implemented as an agent-based simulation framework. Its spatial component is based on a modular design that facilitates straightforward scenario configuration and scalability. In order to perform large network simulations, the framework has been adapted for parallel processing. Issues of both static and dynamic load balancing are considered.

While detailed field data are not available for heterogeneous traffic on urban networks, which precludes precise quantitative validation, sensitivity analysis of the model was performed with a wide range of parameters and values. Macroscopic whole-network measures are defined and used to study a number of scenarios, the most manifest property of which is the contrast between slow and fast, vulnerable and less vulnerable agents in the traffic mix.

List of Figures

List of Tables

Chapter 1

Introduction

As 'greening' of all aspects of human activity becomes mainstream, transportation science is also increasingly focused around sustainability. Modal co-existence between motorised and non-motorised urban road traffic is, in this context, of particular interest in traffic flow modelling. The work described in this thesis centres on mixed car and bicycle traffic on urban infrastructure suitable primarily for homogeneous motorised flows, typical of Dublin and old urban centres in Europe. The flows in question are characterised by low bicycle volumes and the urban network by narrow roads and a scarcity of bicycle lanes, which are only partly connected.

1.1 Motivation

The study of the bicycle as a means of transport and its promotion and integration into 'mainstream' traffic is the topic of much recent research (Cervero et al., 2009; National Sustainable Transport Office, 2009; Pucher et al., 2010). There are also various models of bicycle traffic that focus on level of service, other quantifiers of bicycle facilities, routes and required facilities, as well as bicycle-suitability of existing infrastructure for motorised traffic (Taylor and Damen, 2001). Finally, literature reports on a number of general, space-continuous simulation models that include bicycle traffic (Khan and Maini, 1999; Gould and Karner, 2009) and commercial simulators that also facilitate the inclusion of bicycle-related elements in their configurations (Barceló et al., 1999;

Fellendorf and Vortisch, 2010). More recent work on traffic in developing countries (e.g. Jia et al., 2007; Gundaliya et al., 2008; Xie et al., 2009) specifically studies interactions between bicycles and motorised traffic, both on roads and at intersections. However, as with any other kind of transport mode, bicycle flow share in overall traffic and the style of integration vary from country to country and, among other things, depend on the limitations of infrastructure and societal attitudes. Hence, while the physical dynamics of moving bicycles, in or outside of the context of other traffic, oftern can be generalised, the specific conditions found in Dublin warrant attention, especially since there have not been studies that take a comprehensive look at interaction between motorised and non-motorised modes in conditions of narrow space sharing, which is the dominant means of bicycle inclusion in Dublin traffic.

The questions that can be asked are applicable, with changes in specifics, to any kind of network and traffic mix: How does the network, as it is, perform with only one, only the other or with various combinations of the two types of traffic (motorised and non-motorised)? Can the whole-network performance be improved by changes to the allocation of street space, movement direction, signalling or traffic rules? Can safety be improved in any of these ways? What is the best measure of network performance and does this differ between modes? Or - *more specifically* - what negative impact and/or benefits would be seen in different traffic flows if a number of connecting roads were to be made 'cycle-only' in the city? Or, would such a change encourage people to cycle more? And so on. These questions can be answered either by conducting real life experiments or by modelling and the former is impractical in most cases, calling for a suitable model - one with knowledge of the specific properties of the system in question.

There are sophisticated and widely used simulation products, such as VISSIM (Fellendorf and Vortisch, 2010) or AUNSIM (Barceló et al., 2005) that allow for bicycles in their models and which would typically be used by organisations dealing with traffic to ask the broad questions listed above. However, apart from being expensive, these base their vehicle movements on particular proprietary models and would also need to be configured for the types of interaction between vehicles of particular interest here.

1.2 Research objectives

The high-level aim of the project, of *constructing a model with the power to summarise a bicycle-inclusive traffic system in terms of useful qualitative and quantitative information*, was preconditioned by in-depth understanding of:

- the inputs and their relationships, most importantly the different involved vehicle types and the network, including its topology and other properties
- what kind of outputs would be useful and how these could be extracted from the model; also how such data could be processed and used
- the most suitable modelling techniques with respect to the requirements for facility of network and parameter configuration and fast repeated derivation, which precluded theoretical approaches in favour of computing

The main objective of the research was to *design and implement a bicycle-inclusive traffic model* that would incorporate all the elements of the problem space, namely:

- vehicle movement: intrinsic properties and interaction with environment including other vehicles, infrastructural features and network topology
- the network space: defined to (i) allow occupancy by different types of vehicles (heterogeneity) and (ii) navigation using vehicle movement rules

A question that arose from the beginning and repeatedly re-emerged during the research was whether the challenge of combining all these diverse elements in a model could be met with a relatively simple and elegant solution i.e. a clear but *comprehensive formulation*, similar to that of models for traffic on a road stretch (e.g. Nagel and Schreckenberg, 1992), which would be attractive for its ‘portability’ of definition, allowing relatively facile description and re-creation of complex scenarios. On the other hand, it was essential to incorporate all known features of the modelled domain, including different types of known bicycle-motorised vehicle interaction (e.g. Chen et al., 2009), conflicts at intersections (Allen et al., 1998a), routing decisions and, indeed, simple movement ahead on the road.

Tightly associated with the process of defining a model is its *validation*, through comparison with real data or by other means, if these are capable of provding evidence of validity. Our objective was to validate the defined heterogeneous network traffic model by collecting field data and, if needed, carrying out a feedback procedure to calibrate the model until it was satisfactory with respect to the data. The scope of the validation was to remain limited to only the most basic model elements and behaviours so as not to become unmanageably lengthy. However, confidence in the building blocks would provide the first step towards confidence in the validity of the entire structure of the model.

The computational dimension of the project posed an additional requirement of ensuring reasonable performance and investigating *parallel computing possibilities*. An interesting question not tackled in literature is the direct comparison of parallel processing when using shared memory on the one hand and the messaging paradigm on the other for traffic flow applications. It was an aim of the research to include parallel implementations and carry out some exploration of how they performed.

Finally, identifying suitable measures and data presentation formats for *traffic system evaluation* was seen to be an important part of the project since these would give meaning and utility to the model. Providing illustrative examples of model application was the final objective set for the project timeframe. This would complement the model-building work by providing an early taste of its usefulness.

1.3 Thesis overview

The organisation of the thesis is as follows. Chapter 2 constitutes the literature review, setting the scene for the work reported on in the other parts of the thesis. This is followed by Chapter 3, describing the entire model defined for a generalised case of traffic including bicycles and cars on networks. This includes the description of the spatial and participant behaviour aspects of traffic. This chapter also includes the analysis and high-level design required for a parallel implementation of the model. Chapter 4 deals with high-level implementation aspects of the simulation system, i.e., those that are tightly related to the implemented model (rather than to, e.g., programming techniques). Substantial sections of the chapter are dedicated to the implementation of routing zones and to parallel implementation performance. The penultimate chapter, Chapter 5 reports on the output of the performed simulations and provides validation and analysis of those data. Chapter 6 concludes the thesis, summarising the work and results and giving an overview of possible directions for future work that builds on the research carried out so far.

Chapter 2

Literature Review

Transportation and, specifically, traffic flow modelling has been an area of growing interest ever since the motorised vehicle volumes on roads became inconvenient and dangerous enough to warrant the introduction of formal traffic control. For such control to be applied properly, research is required on traffic properties and infrastructure demands. Its ultimate aim is the understanding of pertinent dynamics, needed in the development of plans and measures for minimising congestion[1] and maximising safety and, through these, meeting the economic, environmental and political demands related to the movement of people in general. In recent years, and with populations and economies growing in most parts of the world, the constraints within which traffic science operates also increasingly involve sustainability, society and health related elements (Brown et al., 2009) and are inevitably becoming self-referential, in that traffic itself is a large part of the the problem that needs to be solved. These trends have brought about the revival of alternative transport modes, including the pedal bicycle, in developed countries and increased interest in preserving those modes in the developing world.

In contrast to this scene, set through a multitude of heterogeneous factors at play, lie the elements of concrete systems: transportation network structures, vehicle capabilities and the most common goal of traffic participants, which is to get somewhere in the

[1] In Gazis (2002) the story is recounted of how traffic theory was directly applied in a successful experiment to reduce congestion in the Lincoln Tunnel in New York.

shortest possible time and avoid accidents on the way[2]. These parameters, while not exact, can be defined well enough to appeal to intuition by suggesting that it should be possible to 'capture' traffic flow in a model as a first step towards shaping it to the complex needs of its users.

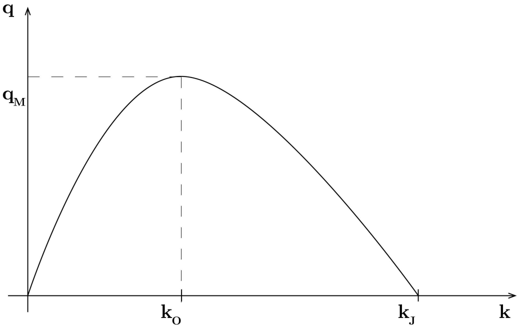


Figure 2.1: Illustration of fundamental diagram shape, where q_{M} is the maximal flow, k_0 is the traffic density at which the maximal flow is achieved and k_{J} is the jam traffic density.

It was in 1912 that, for the first time, the traffic counts in New York showed more motorised vehicles than horse drawn ones (Brown et al., 2009). Twenty years or so later, traffic measurements and attempts to define the attributes of roads with regard to traffic were well under way. Greenshields (1935) presented relations between traffic density and average speed in experimental data and reported the earliest traffic modelling attempts, which still stand as a rough guide, in mathematical terms, on road traffic (Schoenhof and Helbing, 2009). He derived the fundamental diagram (Figure 2.1) from the measured density/speed relation and the flow equation:

$$q = u \times k \tag{2.1}$$

where q is the traffic flow (in number of vehicles that pass the point of observation per time unit), u is the space-average velocity at the point of observation and k is the density at the point of observation (in number of vehicles per unit of length). All

[2] An interesting statistic, quantifying the success with which drivers in Germany achieve this last goal, is given in Nagel et al. (2003), where the probability of accident per interaction is reported to be 10^{-6} or 10^{-7} (an interaction is any adjustment made by a driver in reaction to another driver's behaviour or to the environment and happens every few seconds).

fundamental diagrams found in the literature, including network-related macroscopic fundamental diagrams, whether theoretically or experimentally derived, correspond to this fundamental diagram in the following ways: a) they contain the point $(0, 0)$, b) as the density increases, the flow first increases, then decreases and c) the flow becomes 0 again at some density value, usually referred to as jam traffic density. The shape of the curve, however, varies considerably: as new sets of data were collected and interpreted, ever more sophisticated models (or model modifications) were developed to explain these.

With time, the need arose for traffic to be described and quantified at different levels and from different points of view, hence the range of research areas and related model types encompassed today by transportation science (Hall, 2003). These include environmental (Massink, 2009), health-related (Woodcock et al., 2009), economic (Cavill et al., 2008) and safety (Dabbour and Easa, 2008) considerations, urban planning (Tiwari et al., 2007; Blanco et al., 2009), infrastructure design (Aultman-Hall, 2004), long-term participant behaviour (Heinen et al., 2010), traffic forecasting models employed in long- and short-term traffic planning (Boyce, 2007), flow models, which capture the physical properties of traffic (Nagel et al., 2003), models of capacity, accepted time gaps and others that focus on particular traffic parameters for use by engineers in the design of traffic systems or other (Brilon et al., 1999) and traffic control models (Papageorgiou et al., 2003), but still the list is not exhaustive. Further, the results of transportation research are applied by engineers and planners, directly or indirectly, through tools such as simulators (Fellendorf and Vortisch, 2010) and Adaptive Traffic Control Systems (Stevanovic, 2009). However, the main focus of this review are models involved with *real-time traffic dynamics*.

Modelling of real-time traffic dynamics has two immediately identifiable facets (or components): (i) a spatial one, which deals with the representation of infrastructure and its occupancy by vehicles and (ii) a behavioural one, which describes traffic participant positioning and movement within the spatial context. In the case of a network, apart from the immediate space of roads and intersections, which is negotiated physically by vehicles, there is also *topology*, which offers an abstract view, for the study of

(a) control strategies and rules (these, together with the spatial aspect, complete the infrastructure) and (b) route choice by participants (which adds another layer to traffic behaviour). We use these relations to define a reference model, shown in Table 2.1, which offers a classification of the different areas covered by real-time traffic models and, through their relationships, shows the interdependence between the different disciplines that study them. The different aspects of real-time traffic are usually in narrow focus and often exclusively the domain of a particular community such as physicists, computer scientists, planners or engineers, which means that a synthetic view of 'how it all fits together' is not normally seen in literature. In order to provide a sketch of the full landscape of real-time traffic related research, this literature review gives some examples of non-flow work in the following section, defining concepts that feature in the thesis, even if they are not its focus. The remainder of the chapter is divided into three sections on traffic flow literature, in the order of increasing relevance to the research reported in the thesis.

	Infrastructure	**Participant behaviour**
Occupiable space	Spatial model	Physical movement rules
Network topology	Control and rules of the road	Route choice

Table 2.1: A network traffic modelling reference model: traffic models can be grouped by domain (infrastructure vs. behaviour) and by spatial level of abstraction (immediate occupiable space vs. network topology)

2.1 Non-flow network-related research

The main non-flow aspects of real-time traffic research, control and routing models, have been driven mostly by requirements of engineering, with research directed towards avoiding congenstion and optimising flows in existing networks, as well as planning changes in infrastructure, such as the building of new roads or changes to intersection types.

Control

Traffic control and, more recently, *intelligent transportation systems* (ITS) represent the most visible end-results of traffic-related models in action. The functioning of these systems, however, is underpinned by a large body of research that includes flow modelling with control strategies taken into consideration. Papageorgiou et al. (2003) provided a comprehensive overview of traffic control research, covering control at isolated intersections, fixed-time coordinated control and traffic-responsive coordinated control, both in urban and motorway networks, and concluded that there is a large gap between theory and practice in this area. A more recent review, focused on urban traffic coordinated controls, can be found in Li (2012). The ultimate purpose of the systems described in these reviews is practical application, of which the most widely used examples are *adaptive traffic control systems* SCOOT and SCATS (Stevanovic, 2009).

Traffic control research has also been approached from the traffic flow modelling perspective. For example, a control scheme with real-time local adaptivity was proposed by Laemmer and Helbing (2008) for self-controlled traffic lights with phases that roughly correspond to flows to which they are applied. Also, a simple congestion relief strategy for cities was designed by Gonzales et al. (2008), which refers to a city model based on traffic 'reservoirs' and involves directing traffic from the more densely occupied to the emptier of these.

Routing

In general, *transportation forecasting* models are used to estimate the likely impact of investment and policy decisions relating to transport (National Research Counci, U.S., U.S.). The entire forecasting process consists of a number of activities, including data collection and analysis, both in the social and technical domains, and the application of these data in models designed to answer questions pertinent to planning. The development of such models is where technical research is required. The generally adopted transportation modelling approach is the four-step model (FSM), first developed as part of the Chicago Area Trasportation Study in the 1950s (McNally, 2007), which, even if not actually followed in this explicit form (Transportation Research Board, 2010), is useful for reference. The steps are: (i) trip generation, (ii) trip distribution, (iii) mode choice and (iv) *traffic assignment* (Hensher and Button, 2000). The first three steps model demand: general, specific to start and end points and specific to modes, respectively. The demand and the properties of the network are all taken into account in the fourth step, the only one that deals with real-time traffic dynamics, namely, determination of link flows in a transportation network. The most significant early work on this topic was Beckmann et al. (1955), where assignment methods were developed according to Wardrop's (1952) first principle. This principle states that a *user equilibrium* is reached when no unilateral change that a participant in network traffic can make to their route would improve their travel time (the author also formulated a second principle, which relates to *system equilibrium* and states that this is reached when the average travel time in the network is minimal). More recently, Boyce et al. (2005) reviewed the impact and influence of that publication in the 50 years since it came out. Equilibrium assignment models involve two types of network equilibrium: variable demand (of which Beckmann et al., 1955, is one) and those combining trip distribution and assignment (the second and fourth step of the FSM).

A direct connection was made between traffic flow modelling and route assignment in Daganzo (1995b,a), where underlying traffic performance was investigated, for a single link and a network, respectively.

A multi-modal assignment model due to Modesti and Sciomachen (1998) considered the problem of the shortest path through the network, with weight on links dependent on cost, time and modal inclinations of a user. In Garcia and Marin (2005) route assignment was done with mode combinations taken into account. Wardman et al. (2007) presented a mode-choice model with special focus on cycling and of particular interest in the UK, while Tiwari et al. (2008) modified the standard method for estimating passenger car units (PCUs) to apply to truly heterogeneous traffic (such as that common in India). Simulation-based assignment models also exist, for example the agent based one described in Balmer et al. (2004) looked at individual origin-destination points and individual learning instead of regions defined by an *origin-destination table*, which is the usual way of specifying demand in a transportation forecasting model.

Gap acceptance

Another area of research closely related to traffic flow is the estimation of crucial parameters of traffic, required in network flow models and control strategy design. One such parameter is the *critical gap* and its estimation was the subject of a comprehensive review by Brilon et al. (1999). The methods summarised therein defined the status of so-called *gap acceptance theory*, which is closely related to estimating the capacity values of unsignalised conflicting flows. For example, Brilon and Wu (1999) used accepted gaps to estimate the capacity of a two-stage priority intersection. A review of capacity estimation was provided by Akçelik (2007). The reported work was all based on accepted gap estimation models, headway distribution models (which concern themselves with the disribution of time-gaps between subsequent vehicles in a flow) or some combination of the two. An alternative to time-gap acceptance was suggested by Wang and Ruskin (2003), where the method of Minimum Acceptable sPace (MAP) is defined.

2.2 Empirical data analysis

In order for any useful model to be developed, the properties of what is modelled must be understood. Hence, a very important research effort is focused on reporting empirical data and interpreting these to derive properties of traffic and enable model templates to be formulated. Much research has been conducted around explaining and organising information on the behaviour of traffic in the congested part of the fundamental diagram. A single-line diagram, like that shown in Figure 2.1 corresponds to equilibria at different densities. On the other hand, an empirical fundamental diagram looks like the one shown in Figure 2.2. Here, the *free-flow* section, which is the linear section at low densities, is fairly well defined, while the *congested* section, which follows the point of maximal flow, displays wide scatter, but also hysteresis (first spotted by Edie, 1963).

One of the most interesting and, in traffic flow science, controversial, pieces of work

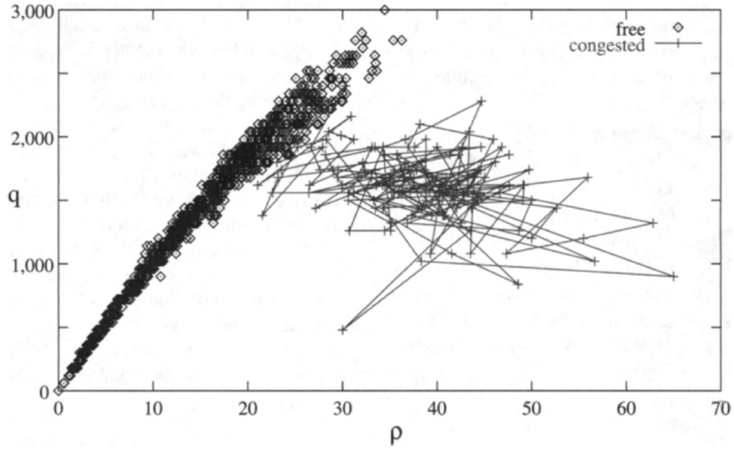


Figure 2.2: Empirical fundamental diagram, from Nagel et al. (2003).

was due to Kerner and Rehborn (1996a,b, 1997), on the phases of traffic flow. They described traffic as analogous to a thermodynamic system and, using data measured on the A5 motorway in Germany, identified three phases of flow: *free flow*, *congested traffic* and what the authors term *synchronised flow*, of which there are three subtypes (constant-velocity, constant flow and density and patternless). A detailed one-to-one analogy with thermodynamic systems was suggested in Nagel et al. (2003). The crux of this *three-phase traffic theory* was that it postulated the existence of three states between which transitions occur spontaneously. These authors showed, based on traffic data and independently of phase-based interpretation, that wide traffic jams can move upstream in a stable manner if met by a steady influx of vehicles and that the outflow from the front of these jams is about 33% lower than the maximally achievable flow value on the same road. In Daganzo et al. (1999), data from a Canadian road was used to repeat Kerner's findings on the behaviour of jams, but with explanations other than spontaneous formation, such as, for example, one particularly slow driver or a slight narrowing of the road, for their appearance. In Neubert et al. (1999) an objective method, using correlations and autocorrelations of macroscopic attributes, for identifying the different traffic phases, was introduced. Three-phase theory remains controversial. Results of empirical data analysis, which do not support intrinsic traffic flow phases can be found in Windover and Cassidy (2001). Their findings suggested a smooth fundamental diagram - one that is not in accordance with this theory. In agreement with the three-phase argument of Neubert et al. (1999), however, are results reported by Knospe et al. (2002), but for a road with no speed limit. In further criticism of the three phase model Schoenhof and Helbing (2009) and Treiber et al. (2010) state that all traffic patterns can be represented with simpler two-phase models. The argument advanced is that the validity of a model lies in whether it is capable of reproducing reality, including the different traffic regimes and patterns, and that the number of phases identified or named is irrelevant.

Network traffic data from Yokohama, Japan, was analysed by Geroliminis and Daganzo (2008), to reveal the existence of a *macroscopic fundamental diagram* (MFD), which is the density-flow diagram arrived at by plotting flow in the network, averaged over all links, against global density. Buisson and Ladier (2009) analysed a large amount of data from Toulouse, France, and were able to identify differences between the MFDs resulting from slightly different conditions, including the distance of the loop detectors from traffic lights, the presence/absence of traffic lights and type of road (city or highway). Analysis of network data from Twin Cities Metropolitan Area Freeway network in Minnesota, USA, by Geroliminis and Sun (2011), confirmed the viability of the MFD. However, the authors qualified the result with a proviso that a network, in order to display a well-defined MFD, must have the property that it produces the same distribution of densities at any two moments with equal traffic loads, regardless of the origin-destination table[3].

Specific data can be used for the analysis of particular features of a traffic system. Chen et al. (2005) analysed time-series data from freeways in California's San Francisco Bay Area, to conclude that car-pool lanes were causing congestion, rather than relieving it. Cassidy et al. (2008) analysed the same data further but included spatial considerations and came to the opposite conclusion, that car-pool lanes 'do their job' of reducing congestion.

Wang et al. (2008) used data from Shenyang, Tianjin and Shijiazhuang in China to calculate *bicycle conversion factors*[4] for various manoeuvres, e.g., straight passage

[3] This specifies regions in a network that act as origins and destinations of traffic and expected/measured volumes between each origin-destination pair.

[4] The ratio of car capacity flow to the measured bicycle flow.

through and conflict at an intersection. Tiwari et al. (2008) confirmed that mixed traffic data collected in two sites in India comply with Equation 2.1.

2.3 Traffic flow models

In contrast to the non-flow aspects of real-time traffic science, which relate to engineering, flow modelling is closer in topic to statistical physics. Theoretical models have traditionally been constructed by physicists through analogies drawn with physical systems (Helbing, 2009b). More recently, i.e. in the last 20 years, it has become techologically possible to exploit computational models of traffic flow, which require a lot of processing power to yield usable results (Schadschneider, 2008).

Traffic flow models can be classified broadly by the level at which traffic is represented, i.e., as macroscopic, microscopic and so-called mesoscopic models. In macroscopic models vehicles are viewed *en masse* and the focus is on aggregate properties of traffic, such as average velocity, density and flow. These describe the dynamics of traffic without reference to individual vehicles. Microscopic models are based on measurements or assumptions about the attributes and behaviour of individual vehicles, which provide a starting point for the description of a traffic system, but allow the derivation of macroscopic properties, either theoretically or by simulation. Mesoscopic, a term sometimes used for gas-kinetic models (Hoogendoorn and Bovy, 2001), do not strictly fit either of the two other categories and are macroscopic in terms of chosen variables but microscopic in terms of spatial dimensions. Models can also be classified into theoretical and simulation models, depending on how information is derived from initial descriptions. The latter intuitively applies only to microscopic models: as an ‘imitation of reality’, simulation should include some representation of individual vehicles; but, the term is also used where more abstract concepts, such as traffic density distribution, are computed (e.g. Helbing and Treiber, 1999; de Palma and Marchal, 2002; Sewall et al., 2010). Finally, there is a difference between models that focus on motorway traffic, i.e. on traffic flowing mainly undisturbed, in one direction, over a long stretch of road, and models of urban traffic, which is affected by the topology of the network. These classifications are somewhat crude and do not always do justice to the complex work that has been done in traffic flow modelling. However, they provide a basis for navigation through the abundant material that has been produced on the subject.

2.3.1 Motorway models of traffic flow

Since the spatial aspect of a stretch of road is uncomplicated and generally constant, motorway models of traffic flow focus only on the dynamics of driver/vehicle units themselves and on interactions between them. These can be affected by vehicle capabilities, drivers' psycho-physical properties and their proclivity towards compliance with road traffic rules, apart from external conditions such as the width of the road and density of traffic.

Macroscopic models

The first physical world-inspired model of traffic flow, of macroscopic type, was the *fluid dynamic (or kinematic wave) model* of Lighthill, Whitham and Richards (Lighthill and Whitham, 1955; Richards, 1956). Starting with the assumption of a smooth fundamental diagram (such as the one in Figure 2.1) and applying the law of conservation as expressed for fluids, the authors described how disturbances spread through traffic in the form of kinematic waves and developed a theory for quantifying the speed at which such disturbances move. Time-space diagrams of traffic flow could then be generated under different conditions, such as a bottlenecks and at traffic lights. Most macroscopic models, like Lighthill, Whitham and Richards' (LWR), are based on the continuity equation:

$$\frac{\partial \rho}{\partial t} + \frac{\partial q}{\partial x} = 0 \quad \text{with} \quad q = \rho V \tag{2.2}$$

where ρ is the traffic density, q is flow, V is the average vehicle velocity and x and t are the spatial and time variables. Expanding this equation with additional terms and inclusion of additional equations for models of higher order were developments that improved the representation of real traffic features over that achieved by the LWR model. These higher order models aimed at accounting for non-equilibrium dynamics with an additional equation for velocity, including acceleration, convection (transport to other vehicles), anticipation (aimed-for velocity) and relaxation (approach to aimed-for velocity) (Payne, 1971) and at smoothing out discontinuities (shock-waves) of the LWR model by the inclusion of a viscosity term in the equation (Kühne, 1984; Kerner and Konhäuser, 1993). While some researchers considered these higher-order models unnecessary for most purposes (e.g. Daganzo, 1995c), new ones continued to be developed. A further improved representation, which included a third equation, for velocity variance, in the set and which corrected many of the problems with earlier models (e.g. densities above the physically possible and negative velocities) was due to Helbing (1995, 1996). Aw and Rascle (2000) provided a correction for characteristic velocities higher than the maximal vehicle velocities and Jiang et al. (2002) reproduced stop-and-go traffic.

Another macroscopic approach is that taken by queueing models, such as Vandaele et al. (2000) and Helbing (2003). These quantify traffic dynamics using *travel time* as the main variable, in contrast to fluid-dynamic models, which focus on velocities.

Gas-kinetic (mesoscopic) models

The first gas-kinetic model was proposed by Prigogine and Andrews (1960). A Boltzmann-like equation was used to describe the distributions, in a phase space (time, space, speed), of a single particle (i.e. vehicle):

$$\frac{\partial f}{\partial t} + v\left(\frac{\partial f}{\partial x}\right) = \left(\frac{\partial f}{\partial t}\right)_{\text{rel.}} + \left(\frac{\partial f}{\partial t}\right)_{\text{coll.}} \tag{2.3}$$

where f is the phase space distribution function $f(x, v, t)$ and x, v and t represent the spatial, velocity and time dimensions. The overall time derivative of the function then consists of a *relaxation* and a *collision* term, where the former represents a vehicle's aim towards a desired velocity and the latter stands for vehicle interactions, rather than literal collisions. Gas-kinetic models are often used for the analytical derivation of macroscopic equations, for example, in Helbing (1996), which improved on the original gas-kinetic model by adding terms for driver behaviour and taking into account the finite size of vehicles. More recently, gas-kinetic models for multi-lane traffic (Helbing, 1997), applicable to congested traffic Helbing (1998) and for multiclass traffic (Hoogendoorn and Bovy, 2000) have been developed.

Car-following models

Car following models are classified as microscopic, in that their initial focus is on the individual vehicle. They are based on formulae for the time- or distance-spacing between cars, under assumptions such as 'drivers will keep a safe distance from the vehicle ahead' but 'will not lag unnecessarily behind'. The car-following rules, once defined, are used to form a bigger picture of traffic flow, through derivation of macroscopic attributes and relationships and simulation of their operation. The earliest work in this area includes that by Herrey and Herrey (1945), who defined the *influence length* of a vehicle as including reaction, braking and vehicle length components, and space-time diagrams of *influence stripes*, from which the maximum capacity and optimal velocity were derived. Another early attempt (Pipes, 1953) investigated car-following as a traffic modelling method by mathematical expression of the rules set out in the California Vehicle Code, for different behaviours of the leading car. The work done in the Basic Science Lab of General Motors in the late 50s and early 60s (Brackstone and McDonald, 1999), resulted in the well-known Gazis-Herman-Rothery (GHR) formula of car-following, the choice of parameters for which motivated much car-following research that followed. Publications in this context included Chandler et al. (1958), describing initial experiments and theories of car-following, Herman et al. (1959), in which the propagation of disturbances and stability were examined, Gazis et al. (1959), who derived macroscopic relationships from car-following rules and Gazis et al. (1961), in which the *GHR formula* was proposed, and given as:

$$a_n(t) = cv_n^m(t)\frac{\Delta v(t-T)}{\Delta x^l(t-T)} \tag{2.4}$$

where t is the current time, x the vehicle position, v the velocity, n the ordinal number of the vehicle to which the formula is being applied, T the driver reaction time and m, l and c are constants used to calibrate the model.

Another group consists of models that take the *optimal velocity* (OV) approach, first proposed in Newell (1961) and, more recently, revived in Bando et al. (1995). With the OV approach vehicles are assumed to adjust their speed towards an *optimum speed* that depends on the distance from the preceding vehicle, rather than simply towards the current speed of the preceding vehicle (the approach taken in the GHR model). Tanaka et al. (2008) applied the OV model to a mix of slow and fast vehicles on a multi-lane highway. Ge et al. (2004) have used following rules based on more than one vehicle ahead, while Tang et al. (2009) added driver memory, both resulting in OV models that exhibit greater stability.

Among other car-following contributions, that described in Helbing and Tilch (1998), is based on a model originally defined for pedestrians (Helbing and Molnar, 1995). In it, cars are governed by three generalised forces (not of action-reaction type), specifically, (i) the acceleration force, (ii) the force based on 'desire to keep a safe distance' and (iii) a fluctuating force. Treiber et al. (2000) used a car following model, with empirical boundary conditions, to reproduce the traffic states near inhomogeneities observed in data from several German freeways. Newell (2002) further proposed a very simple theory where each vehicle follows 'in the footsteps' of the leader, by realising the same time-space trajectory (except for translation in space and in time). Kerner and Klenov (2002) used a space-continuous microscopic model with car-following, including stochastic elements derived from parameters of a realistic fundamental diagram, to reproduce system behaviour reflecting the three-phase model of traffic (e.g Kerner and Rehborn, 1997). Empirical *departure headways*[5] at signalised intersections were reproduced and explained by Jin et al. (2009), in terms of interactions between queueing vehicles, with the help of a fairly complex set of car following rules.

[5] A departure headway is the time difference between the moments when two vehicles following each other cross the stop-line, as they start moving after being stationary (in the queue) for some time.

Cellular automata

Cellular automata (CA), as a mathematical concept, first appeared in the 1940s, suggested for the study of self-reproducing organisms by von Neumann (1944). The approach was popularised in the 1970s through Conway's Game of Life (Sarkar, 2000) and studied in detail by Wolfram (1986). An initial traffic model, based on CA and proposed by Cremer and Ludwig (1986) went largely unnoticed, until the seminal work of Nagel and Schreckenberg (1992) was published.

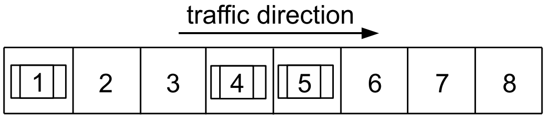


Figure 2.3: Illustration of a one-dimensional CA lattice with vehicles.

In the simplest form of a traffic cellular automaton model, the road is represented by a one-dimensional lattice of cells, such as the one shown in Figure 2.3. Each vehicle takes up one cell on the lattice and the lattice is populated with vehicles according to given initial conditions. In each discrete time period vehicles are moved on the lattice according to specified update rules. For example, the rule might be that a vehicle move forward until it reaches the farthest unoccupied cell, in which case the vehicle in cell numbered 1 in Figure 2.3 would move to the cell numbered 3. These models, in essence, incorporate both car following and Boltzmann-like concepts, adapted for computer simulation. The space and time dimensions of the gas-kinetic models translate into cells of a lattice and discrete time periods; the velocity dimension of the gas-kinetic models and car following rules translate into CA update rules.

The paper that stimulated an avalanche of effort (reviewed, e.g., in Maerivoet and De Moor, 2005) on cellular automaton (CA) models of traffic flow was due to Nagel and Schreckenberg (1992). The simple model proposed was based on the Wolfram rule 184 (Wolfram, 1986) with stochastic elements incorporated into the update rules, as well as a permitted velocity maximum that can be greater than 1. The *Nagel-Schreckenberg (N-S) rules* can be summarised as follows:

1. Acceleration: if $v_i < v_{\mathrm{MAX}}$ and $v_i < d_i$, $v_i \to v_i + 1$

2. Slowing (due to cars ahead): if $d_i < v_i$, $v_i \to d_i$

3. Randomisation: if $v_i > 0$, with probability p_{R}, $v_i \to v_i - 1$

4. Vehicle motion: each vehicle is advanced v_i cells along the track

where v_i is the velocity of the i^{th} vehicle, v_{MAX} is the maximal velocity, d_i is the number of free cells between the i^{th} vehicle and the vehicle ahead of it and p_{R} is the randomisation parameter, used to introduce stochasticity into the model. The right arrow ($\rightarrow$) denotes a velocity update, to the value shown on the right-hand-side of the arrow.

In the paper, real-life equivalents of length, time and velocity units are discussed, with a length unit of 7.5m and time unit of 1s deemed to be reasonable. The basic model was examined in more detail in Schreckenberg et al. (1995), with the conclusion drawn that parallel updates and a maximum velocity of 5 produce the fundamental diagram closest to that observed for real traffic.

Building on the N-S rules Nagatani (1993b) presented a two lane model, with update rules including lane changing manoeuvres. Here the drop in the average velocity occurs earlier with lane changing allowed than without (possibly because of 'time wasting' on unnecessary transversal movements). In Rickert et al. (1996), more realistic results were achieved by incorporating 'look back' into the lane changing rule (as opposed to only checking space to the side). In Chowdhury et al. (1997) the effect of lane changing rules was investigated in a model with two maximum velocities, with the finding that asymmetrical rules allow higher flows. Similar results were reported in relation to aggressive lane-changing in Li et al. (2006). The initial two lane model was improved upon further in Wagner et al. (1997) to reproduce the *lane inversion* phenomenon, found in real traffic. A systematic study of lane-changing rules, in Nagel et al. (1998), also showed that the logical structure of the rules is more important in producing realistic results, than their microscopic details are.

Further models based on the Nagel-Schreckenberg template include that of Emmerich and Rank (1997), where the rules were modified so that deceleration was represented as gradual (to reproduce stop-and-go traffic and provide a better quantitative fit for field data) and that of Barlovic et al. (1998), who used velocity dependent randomisation (VDR) to reproduce the real-life phenomenon of metastability of traffic around the point of maximum flow and hysteresis, using a slow-to start model (first suggested in Takayasu and Takayasu, 1993). Simon and Gutowitz (1998) applied the Nagel-Schreckenberg rules to a two-way road, with one lane in each direction, and extended this to include interactions caused by passing manoeuvres, producing results that were a striking fit for historical real-life data of maximal flow as a function of flow in opposing direction. Helbing and Huberman (1998) described a CA-type model, involving mixed truck and car traffic. At high enough densities this displayed coherent states of flow, with all vehicles at approximately the same velocity and no lane changing, a property reported by the authors to be supported by data from the Netherlands. Barlovic et al. (2002) used their VDR model to study a road with open boundary conditions. In the same year, Kerner et al. (2002) described a CA model that reproduces three-phase traffic (see Section 2.2), while Zhao et al. (2009) described a CA model with an element of anticipation, which reproduces synchronised flow and produces spatio-temporal patterns on an 'on-ramp', i.e., entrance to a motorway (also consistent with three-phase theory). Yuan et al. (2009) developed a hybrid CA and car following model used to examine the effect of adaptive cruise control (ACC) on phase transitions and showed that traffic stability can be increased using ACC. A mix of vehicles with and without Advanced Vehicle Control and Safety System (AVCSS) were modelled in Lo and Hsu (2010) for a single-lane road, using different settings for the gap between vehicles and it was found that the gap imposed by AVCSS can have a positive impact on average flow and average velocity even if only some vehicles are equipped with the system.

While cellular automaton (CA) models are designed for simulation, theoretical derivation of macroscopic system parameters from CA rules is of interest for the verification of simulation output and comparison with other models. The Nagel-Schreckenberg rules were theoretically treated in Schadschneider and Schreckenberg (1993), Schreckenberg et al. (1995) and Schadschneider (1999), using mean-field theories. It was found that in the case of $v_{\mathrm{MAX}} = 1$ these provided exact matches for simulation results but that they became only approximations for higher maximal velocity values and that the quality of the approximations depended on the value of the randomisation parameter. Other analytical work is found in Krauss et al. (1996), where a continuous limit of the Nagel-Schreckenberg model was derived, and in Nagel (1996), where the it was concluded that, despite appearing to only roughly approximate traffic flows, CA models have the capability of reproducing the same range of dynamic phenomena as fluid-dynamic models. Daganzo (2006) specified a kinematic wave model, a car following model and two cellular automaton models that produce very similar vehicle trajectories. The kinematic wave model used was one with a triangular fundamental diagram, the car-following one was based on a linear dependency between car spacing and velocity, while the two CA models comprised a deterministic version of the Nagel-Schreckenberg rules and one with vehicle memory.

While heterogeneity of vehicle type is not given much attention in literature, regardless of model type, most examples are based around cellular automaton models, possibly because the additional complexity is most easily dealt with through microscopic specification followed by simulation. The CA models representing heterogeneous traffic must, in themselves, be more complex, for example using multiple cell occupancy to allow for different vehicle sizes, as in Lan and Chang (2005), Gundaliya et al. (2008) and Mallikarjuna and Rao (2009), where results were a good fit for empirical data. None of these models, however, would be easy to further integrate into a network simulation model. The work in this thesis addresses this gap by defining a comprehensive network model for heterogeneous traffic.

2.3.2 Urban models of traffic flow

Urban traffic has the additional complexity of topology and properties such as convergence and divergence of flow. The necessity for turning manoeuvres and conflict resolution actions affects microscopic properties of traffic and real-time flow dynamics. Route-related decision making by participants lies at a level higher than immediate vehicle control decisions and associated local effects, but does influence the whole-network dynamic state, including average flows and journey lengths. Routing decisions and the resulting network flows are studied both by planners and engineers as part of the assignment portion of traffic forecasting (this could be termed a top-down approach) and by scientists studying the physical properties of traffic, as these incorporate topology into their problem space (a bottom-up approach to network flow).

In the same way that the term “motorway traffic” is used to denote uninterrupted traffic on a stretch of road, the term “network traffic” will be used, for the sake of simplification, when referring to any traffic flows that involve conflicts or decisions, including at isolated intersections.

Macroscopic models

Determining *flows in network links*, a topic more often associated with transportation forecasting traffic assignment (Boyce et al., 2005), was addressed by Daganzo (1995a) as an expansion of an earlier, motorway-related model (Daganzo, 1994). Here the links of the network were viewed as *cells* and the flow between them is modelled macroscopically, disregarding the merge mechanism to focus on flows exclusively. In a subsequent paper, Daganzo (2007), this concept was taken further, as the network of streets is divided into *reservoirs*, these becoming the unit of moving traffic rather than the individual vehicle or even the group of vehicles occupying a single link in the network. A simple congestion relief strategy based on this model, that promotes movement into more sparsely occupied reservoirs, was also proposed and applied to multimodality in Gonzales et al. (2008). Helbing et al. (2007) used the vocabulary of traffic assignment to study flow through unsignalised, geometrically regular networks of different shapes and found that a pattern not unlike that existing under the control of traffic lights, develops through self-organisation.

Equivalent to the concept of fundamental diagram in the motorway context is the network *macroscopic fundamental diagram* (MFD), defined as the density-flow space diagram for global network density and flow averaged over all links in the network. Early macroscopic relations for networks, reviewed in (Helbing, 2009a), were empirically derived, with curves of average travel time as a function of inflow into the network. A combined theoretical and empirical approach was taken in Herman and Prigogine (1979), where a function that describes the relationship between the density (which they refer to as *utilisation*) and average speed in the city or area was derived using a two-fluid approach, in which the speed is deemed to directly depend on the proportion of cars that are not moving. The function has a constant parameter that expresses the quality of the network. The theory was tested with data measured during peak traffic in the business districts of four cities in Texas, USA, and elaborated on in Herman and Ardekani (1984) and Ardekani and Herman (1987). Daganzo and Geroliminis (2008) developed a theory for predicting the macroscopic (network) fundamental diagram and validated this using results from the empirical study of Geroliminis and Daganzo (2008). The authors advised that the macroscopic fundamental diagram is expected to be well defined only for some networks or parts thereof and stated, through further analysis of field data, in Geroliminis and Sun (2011), that a network will have a well-defined macroscopic fundamental diagram if it has the property that its density distribution is always the same for equal overall input flows, regardless of the origin-destination table. Gayah and Daganzo (2011) used a 'two-bin' network model to reproduce hysteresis observed in network data from Toulouse, France, caused by a disturbance and subsequent slow recovery. In Helbing (2009a), a theoretical relationship was developed between utilisation and *average travel time* (the function used in transportation forecasting and queuing theory), which is a better fit for the Yokohama data (Geroliminis and Daganzo, 2008) than the widely used formula published by the Bureau of Public Roads in the US. Daganzo et al. (2011) investigated macroscopic fundamental diagrams further and concluded that, due to the effect of cross-flows, the diagram undergoes a bifurcation at some critical density, causing much lower flows than expected for the global vehicle density, but that this can be avoided through driver adaptivity in congestion situations.

The macroscopic view is important for the understanding of network performance, providing a summary measurement of a complex entity. The existing literature on this topic is not abundant, mostly because whole-network measurements, which allow conclusions at this level, have not been available in the past. However, while models of an agreggate (macroscopic) nature, such as those reviewed in the previous paragraph, even include examples of multimodality (Gonzales et al., 2008) and of individual parameter analysis (Buisson and Ladier, 2009), the bottom-up approach to studying network performance dependency on detailed configuration, including heterogeneity, using microscopic simulation, is rare (e.g. Currie et al., 2004). It is possible that the difficulty of obtaining the required large amounts of real input for such studies, as pointed out by Gonzales et al. (2008), is the cause of their scarcity. Still, and in defence of microscopic simulation for the purpose of studying large network properties, it can provide sensitivity analysis to microscopically manipulated parameters, with the possibility of such manipulation being the main attraction of the method. This would have to include an 'engineering' (as opposed to 'scientific') utilisation, whereby configuration patterns (e.g. allocation of street space, road directionality and others) shown to be very favourable during simulation for a range of demand distributions are 'tried out' in a real network with demands known to be in that range.

Cellular automaton models

Given the simple representation of physical space for a cellular automaton model of motorway traffic, there is close correspondence to mathematical flow models of the same phenomena; analytical counterparts were described in, e.g., Schadschneider (1999) and Krauss et al. (1996). Urban CA models are different, in that the added complexity of the network with which the flows interact prohibits direct theoretical equivalence of the same kind. That departure from simple theory becomes even greater as elements of control and human decision making (for example, whether to turn or continue straight ahead at an intersection) are introduced into the models. As they evolve naturally, through consideration of diverse interactions, CA models acquire a stronger identity as simulation models of a direct applicative nature.

The first urban CA model of note was the one presented by Biham, Middleton and Levine (1992) (BML). A two-dimensional lattice of cells, with periodic boundary conditions, and two types of particles (vehicles), each occupying a single cell, were used (up- and right-moving vehicles). The illustration of this model, based on one in the paper, is shown in Figure 2.4a. The up-moving vehicles were permitted only to move up and the right-moving vehicles right. Simulations were carried out for different lattice sizes and particle densities. Unsurprisingly, at some critical density value a jam

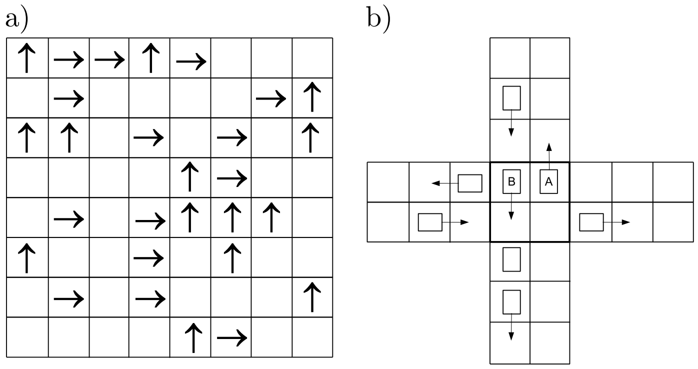


Figure 2.4: Illustration of network CA models: (a) Biham, Middleton and Levine (1992); (b) Chopard, Luthi and Queloz (1996)

arose in all cases. The interesting result was that the critical density was lower and the transition steeper, for larger lattices. In Nagatani (1993a), the BML model was modified to include two-level crossings (these emulate a fly-over, with paths crossing without interaction) as well as 'normal' intersections, in order to investigate the effect, on the critical density, of the proportion of two-level crossings in the system. The model described in Chopard et al. (1996) included finite length two-way roads between intersections, which were modelled as 4-cell rotaries. One such rotary, represented by a thick square line, with portions of surrounding roads, is illustrated in Figure 2.4b, where a vehicle in position A can go left or straight ahead and vehicle in position B must move ahead by one cell in order to be able to turn left. The conclusions were that this model, in terms of the jamming pattern, was closer to that of the one dimensional model than of the pure two-dimensional BML (Figure 2.4a) and that the two-way directionality of the system made it susceptible to deadlocks. It was also suggested that the fine-tuning of CA models to produce precise traffic phases and other effects, such as stop-and-go traffic, is of less value in the context of urban traffic, because these effects are much smaller than those at intersections. Fukui and Ishibashi (2010) performed a simulation experiment starting with the BML spatial model and gradually transforming it into that of Chopard et al. (1996), observing the corresponding transformation of the phase space.

A further refinement was introduced by Esser and Schreckenberg (1997), who included *signalisation* and a more detailed representation of intersections than that found in Biham et al. (1992), Chopard et al. (1996) or Fukui and Ishibashi (2010), while in Simon and Nagel (1998) various types of blockages (e.g. traffic lights with different sequences) were studied. Chowdhury and Schadschneider (1999) examined a model very similar to BML, but included traffic lights, under periodic boundary conditions. Brockfeld et al. (2001) and Huang and Huang (2003) studied the same model in more detail, to find that link lengths correlated better with the effect of the traffic light cycle than did the number of nodes in the grid and that traffic light delay adjustment made a difference for low and high densities but not for the intermediate case. Another model of an 'urban grid' with control elements included was studied in Tonguz et al. (2009) and it was found that the way in which control was applied has considerable effect on the flow through the grid.

Yet further complexity was added to CA urban models through the introduction of *intelligent agents* to handle additional, higher-level, update rules. In Wahle and Schreckenberg (2001) an on-line simulation and prediction system relies on agents to make decisions about turning, based on real-life data. In Scellato et al. (2010), *dynamic routing* was performed, where agents made turning decisions based on local traffic information, to improve the throughput of traffic in the network. It was also found, however, that more global information (even if available to agents) did not substantially improve the performance. Sanchez et al. (2010) also approached network traffic modelling using an agent-base and applied the concept of learning to routing decisions (a set of random parameters influencing those decisions being permitted to change with experience). The authors studied how levels of knowledge relating to congestion (based on a 150m radius vs. global picture) affected the average speed in a road network and found that complete knowledge results in lower average times.

The *isolated intersection* has been studied by many authors. Such work offers the opportunity for in-depth investigation of unit movements and manoeuvres and hence for refined parameterisation. An early cellular automaton model of an unsignalised intersection, using the asymmetric simple exclusion process (ASEP), which denotes a cellular automaton model with asynchronous updates (as detailed in Schadschneider, 2008), was presented in Nagatani (1993c). Foulaadvand and Neek-Amal (2007) reproduced this model but included a theoretical derivation of the model's phase diagram. An isolated intersection is also modelled in Ruskin and Wang (2002), specifically designed to investigate heterogeneity and inconsistency of driver behaviour. The method used, Mimimum Acceptable sPace (MAP) is defined for conflict resolution at the two-way intersection. A T-junction was modelled in Li et al. (2004), Wu et al. (2005) (single cell) and Li et al. (2009a) (multi-cell) for a similar purpose. Foulaadvand and Belbasi (2007) also use a multi-cell Nagel-Schreckenberg model applied to an unsignalised intersection with periodic boundary condition and a *safe distance* method of conflict resolution, providing detailed plots of dependency between the intersecting flows. This was extended (Belbasi and Foulaadvand, 2008) to include signalisation, while Foulaadvand et al. (2010) produced the phase diagram for this signalised intersection model. With a somewhat different focus, Spyropoulou (2007) developed and analysed in some detail a signal controlled single lane CA traffic stream.

Problems associated with multiple lanes and direction-changes were also addressed for other road features, designed to control flow. Thus, cellular automaton models have been developed for single lane (Wang and Ruskin, 2002) and two-lane (Wang and Ruskin, 2006) traffic on roundabouts. Implications of mixed traffic for flow have also been considered in connection with road control features. Heterogeneous traffic CA models including cars and trucks were described, Deo and Ruskin (2006b,a) and the same long/short vehicle mix is used in two-lane roundabout simulation in (Feng et al., 2007).

While the models described here deal with networks and network features of increasing complexity and realism, there is none that simulates - or would be easily adaptable to simulate - heterogeneous traffic with arbitrary vehicle types, a problem that our network design will address. The intersection, which is an important feature of network modelling, is dealt with in one of two ways: it is either treated as a special feature of the network, with cells operating differently from those in a road (e.g. Chopard et al., 1996; Chowdhury and Schadschneider, 1999) or its area is eliminated, as in Esser and Schreckenberg (1997). The first approach is non-universal, as it would require a new specification for any new type of modelled intersection. The second approach disregards the effect of the intersection space itself. The model described in in this thesis generalises the specification of intersections and allows for effects of the physical area of intersections on dynamics. Finally, the complexity involved in network-related behaviour seems to commonly result in textual descriptions that complement the CA update rules (e.g. Wahle and Schreckenberg, 2001) or the use of pseudocode (e.g. Tonguz et al., 2009). Our model specification partly overcomes this, providing comprehensive behaviour rules, which include intersection-related behaviour.

An interesting conceptual question relating to cellular automaton (CA) models of traffic is where the boundary lies between the *cellular automata* and *agent-based* paradigms in traffic models with a lattice-based space. How much intelligence must a vehicle-driver unit have in order for the representative model to be deemed agent-based? When 'intelligence' is viewed in the broadest sense, as a spectrum of capabilities (with mere 'existence' of an entity with a given lifetime at the lower end of the scale and the ability to 'learn' and 'remember' at the upper) and the "cellular automaton" designation of a system based exclusively on its lattice-based spatial aspect, the boundary becomes fuzzy and the paradigms coincide. This means that the criteria for determining the applicability of either paradigm can be quite vague, the choice largely depending on researcher preference and background to the work. For example, Nagel and Schreckenberg (1992) suggest that particles in a traffic CA simulation would be better characterised as 'intelligent agents', Balmer et al. (2004) term the vehicles in a traffic assignment model 'agents' because they have individually specified destinations, which contrasts the more usual origin-destination definition in terms of areas, McNally (2007), and Laemmer and Helbing (2008) refer to vehicles on a self-regulated signalised network as agents, based on their interaction with each other and with traffic lights.

An even more basic question is whether traffic cellular automata (TCA) qualify as CA models at all. With original CA each cell's state is updated according to some rules and based on the state of the cell's environment (consisting of a number of cells around it), in each time-step (Wolfram, 1986). With traffic cellular automata (TCA), system updates are performed by *iteration over vehicles*. Even if the velocity update, corresponding to rules 1, 2 and 3 of the Nagel-Schreckenberg model (see page 20), were considered to be implicit for cells that do not contain vehicles, making it a valid CA update, this is not the only update involved. The position of the vehicles must be changed (rule 4 on page 21) and this is a different CA update rule. This makes TCA a CA model with two different update rules which are applied in an alternating fashion. Taking into account that, in addition, the environments for these updates are both asymmetrical (in different directions) and that the set of states involved ('no vehicle', 'vehicle with velocity 0', 'vehicle with velocity 1' etc.) is complex when compared to the original 0 and 1, the similarity to basic CA becomes contrived. The divergence is even greater where vehicles are modelled to occupy multiple cells (e.g. Mallikarjuna and Rao, 2009).

It must be noted, however, that in spite of any conceptual ambiguities and as long as these anomalies are recognised, use of the 'CA' label for this large and diverse group of models remains helpful.

Other models

Sasaki and Nagatani (2003) studied the *isolated traffic light-controlled intersection*, using an *optimal velocity* model, both theoretically and via simulation to arrive at values for flow saturation and critical densities for different switching strategies and different cycle times. Toledo et al. (2004) and Nagatani (2006) both presented mathematical models of a single car moving through a sequence of traffic lights and described how the cycle length affects the pattern of average speeds and times between the sequence and found it to be surprisingly complex.

A dual representation (streets mapped to vertices, intersections mapped to edges) of the network is used for spatial realisation of the network model in (Hu et al., 2009). Simulations using average values for density and velocity produced results in agreement with empirical data and with the Biham-Middleton-Levine model.

Simulation

Simulation generally implies the reconstruction of a system *of some complexity* - and its behaviour - in another domain that, because of *smaller scale* or *repeatability* is useful to the observer, in terms of cost, manageability and capability. Computational simulation involves the building of an abstract model, of a real or proposed system, and the use of that model in computational experiments from which properties of the modelled system can be validly derived (Barceló et al., 1998). Simulation of traffic is particularly useful since, even though some real-life experiments have been carried out (Gazis, 2002; Sugiyama et al., 2008), these are not practical in the majority of scenarios and are virtually impossible on a large scale.

In consequence, there are a number of commercial or open-source traffic simulators that started as research projects but which, over a period of time, matured into complete products for practical application in planning, management and design. Some of these are:

- SATURN (Van Vliet, 1982), a traffic assignment software suite
- PARAMICS (Cameron and Duncan, 1996), today a comprehensive network simulator, with an underlying car-following model
- TRANSIMS (Nagel and Rickert, 2001), which was built upon cellular automaton dynamics
- SUMO (Krajzewicz et al., 2002), traffic simulator using the Krauss et al. (1996) CA model
- AIMSUN (Barceló et al., 2005), another comprehensive network simulator with an underlying car-following model
- VISSIM (Fellendorf and Vortisch, 2010), commercial simulator based on a car following psycho-physical model; spatially employs only links and connectors (no intersection elements)

On the other hand, the small-scale research-based simulation model, while usually focused on particular problems and not robust enough for general use, is a useful and widely used tool in traffic research, with numerous examples mentioned in this chapter. While in the product-level simulators many aspects of the underlying model are unchangeable, the potential flexibility of the research-based model, unburdened by the requirements of usability and high quality with respect to the computing aspect (e.g., providing a sophisticated user-interface and handling numerous exceptional scenarios), act as test-beds for new models and simulation techniques.

Due to the complexity of the traffic problem space, computer simulations, especially of large networks, require a lot of processing power. The *parallelisation* of such simulations across many processors allows for shorter running times and is a way to enlist the computing horse-power available to researchers today. Publications in this space are not numerous but there are a few.

Barceló et al. (1998) described the parallelisation of their AIMSUN simulator, which is based on shared memory and spatial-domain decomposition, with non-bordering portions of the network assigned to different processors to avoid the need for *memory access synchronisation*[6]. Nagel and Rickert (2001) offered a review of older parallelisation work with micro-simulation of traffic. They also described their own approach, with spatial-domain decomposition and Message Processing Interface (MPI) for communication between processes. They applied both *orthogonal recursive bisection*[7] and a library for graph partitioning, METIS, to achieve breakdown into components and found that the optimisation built into the library provided better 'load balancing'. They also applied some off-line (between-simulations) load balancing, based on previous runs. In Rickert and Nagel (2001) the same system, but including a traffic assignment algorithm with route re-calculation, was described. Lee and Chandrasekar (2002) used a multi-processor machine to balance simulation loads and domain decomposition, which is done dynamically, from shared memory. Klefstad et al. (2005) produced a framework for parallel simulation with the PARAMICS (Cameron and Duncan, 1996) simulator, facilitating automated network division and across-system routing using a centralised origin-destination matrix, with CORBA[8] for real-time communication. Dai et al. (2010) used MPI with a simplified car following model to simulate the traffic of Shanghai on

[6] Protection against simultaneous access, by more than one thread of execution, to shared memory.

[7] This refers to dividing a space in two, then dividing the resulting spaces each in two, with lines perpendicular to the line used in the first section, and so on.

[8] Common Object Request Broker Architecture, specified by the Object Management Group.

multiple processors and also used automated network decomposition.

No studies have been found where the two parallel processing paradigms - shared memory and message passing - are compared for the specific problem space of traffic simulation. Our work fills this gap, approaching parallelisation of traffic simulation code not only as a means to faster subject matter results but as an experiment in itself.

2.4 Bicycle traffic

This section focuses specifically on bicycle-related research, including a short overview of policy and planning but predominantly looking at bicycle-related traffic models known to date, irrespective of type. Most of these are either based on, or extend, the modelling methods and approaches reviewed so far.

Successful incorporation of alternative modes of transportation into urban traffic systems has become an important part of contemporary transport policy-making (Department of Transport, National Sustainable Transport Office, Ireland, 2009), owing to the benefits of using motorised vehicle alternatives, which range from the social (Geurs et al., 2009), through health (Pucher et al., 2010), to consideration of sustainable use of resources (Wright and Fulton, 2005; Massink, 2009). While these are universal, however, existing conditions for realisation of transport options, including infrastructure, attitudes in society and volume, differ greatly from city to city (Zacharias, 2002; Pucher and Buehler, 2008; Cervero et al., 2009). While in developed countries the trend is towards measures to make transportation more sustainable (Martens, 2007; Shay and Khattak, 2010), changes are also taking place in the opposite direction (Zacharias, 2002). Also, for example, Ortuzar et al. (2000) examined the viability of introducing a complete network of cycleways and cycling-related facilities in Santiago, Chile, and concluded that, while such measures would increase the uptake of bicycling as a means of transport, major changes to land use policies and urban planning would be required and need to be facilitated for a significant modal shift.

Models of uninterrupted flow

Bicycle and bicycle-motorised mixed traffic exhibit several aspects of interest to researchers. The characteristics of bicycle-only flow and related road capacities have been studied since the 1970s. A wide range of values for bicycle flow capacity (from about 700 to 4000 bicycles per meter of width per hour) has been reported, differing by author and country (Allen et al., 1998b; Raksuntorn and Khan, 2003; Gould and Karner, 2009). Gould and Karner (2009) provided a relatively recent review of work on the bicycle flow fundamental diagram and levels of service. The authors presented a cellular automaton bicycle-only flow model of their own, verified by means of empirical data, which they collected at the University of California, Davis campus over a period of a week in December 2007. The Burgers CA (a model based on the Burgers equation, which is essentially the continuity equation, with an additional term for viscosity) of Nishinari and Takahashi (1998) was used in Jiang et al. (2004), to reflect the 'in bulk' movements of bicycles, with several bicycles permitted inside a single cell in the same time-step. That model was improved upon by Jia et al. (2007) who make a slow/fast distinction between cyclists. The Burgers CA was also used in Xie et al. (2010), but for combined motorised-non-motorised flow, to determine the effects between the two vehicle types and conditions of optimal flow.

Heterogeneous flow that includes bicycles can differ widely, depending on infrastructure and on locality-specific rules and customary behaviour. Corresponding models are, consequently, equally diverse. Khan and Maini (1999) reviewed a number of models aimed at representing the broadly heterogeneous traffic of Indian cities, all of which employed space-continuous simulation, e.g. Hossain and McDonald (1998), whether for a stretch-of-road or in a network context. Faghri and Egyháziová (1999) developed a feature-rich lane-based network model using car-following rules, different for cars and bicycles, and validated this in terms of volumes at multiple points in a sample network. The authors introduce the idea of bicycle-motorised interaction event counters, but do not report on conclusions drawn from any such data. More recent models of highly mixed traffic flows, due to Gundaliya et al. (2008) and Mallikarjuna and Rao (2009), employed multi-cell occupancy cellular automata specifically targeted to India and conditions which apply. These models, while not applied by the authors to traffic including pedal-cycles, do take into account motorcycles and three-wheelers and could be extended to include non-motorised traffic categories in the modelled mix.

All the reviewed studies simulate bicycle-only flow on wide lanes, where more than one bicycle can travel abreast, separately from cars, or traffic of pervasive heterogeneity, such as found on Indian roads. With the exception of the car-following-based simulation model of Faghri and Egyházinová (1999), the scenario where a lane is shared, with positional discipline, between bicycles moving in single file and cars, does not feature in work done so far. Faghri and Egyházinová (1999) did not provide any data, collected or simulated, on specific interaction effects.

Models of bicycle-motorised interaction

Interactions between bicycles and motorised traffic can be classified broadly into *lateral interference* and *cross-flow* interactions. The former occur where bicycles and motor vehicles are moving side-by-side and affect each other in some way, primarily by inducing deceleration in the other vehicle type. The latter refer to interactions arising from bicycle flows intersecting with motorised flows and in circumstances created specifically by the presence of bicycles. A typical instance of this is the situation where cars turning off to the near side of the road[9] are in conflict with bicycles sharing the road and continuing straight ahead.

The technical report by Wilkinson et al. (1994) gives the Highway Capacity Manual recommended width for lanes that are shared with bicycles as 3.7m, also indicating with of less than 3m as entirely unsuitable. Allen et al. (1998b) suggested a method for determining on-street bicycle level of service (LOS), based on the approach by Botma (1995) for separate bicycle facilities. This accounts for *lateral interference* by reducing LOS where car impingement into bicycle space occurs. Lateral interaction between bicycles and cars in side-by-side lanes was measured by Jia et al. (2008) and average velocity values for both cars and buses shown to fall if bicycles were moving alongside them or, with even greater impact, impinging on their space. Cheng et al. (2008) used

[9] This is a left turn in Ireland, UK, etc. but a right turn in France, US etc.

a spatially fine-grained cellular automata to model this type of interactions between a multi-lane bicycle stream and an adjoining car stream. The interference of bicycles with the car flow was expressed through a higher probability of cars slowing down when faced with 'friction' or 'blockage'. Chen et al. (2009) took similar measurements and derived a well-fitting linear equation for car velocity, as a funciton of distance from the bicycle lane and bicycle volumes. The optimal velocity model of Tang et al. (2010) included a friction component, which accounted for the effect of pedestrians on cyclists and cyclists on motorised vehicles, each mode belonging to a distinct but spatially adjacent flow. In their combined forecasting model, Si et al. (2008) account for interactions through *link impedance functions* whereby *higher* bicycle flows *increase the impedance* of motorised flows and vice versa, with dramatic effect in terms of general flows in the network.

Cross-flow interaction between right-turning cars and straight-moving bicycles was examined by many authors. Allen et al. (1998a) derived a formula, based on collected data, for a car flow reduction factor as a function of bicycle occupancy of the conflict zone. Chen et al. (2007) developed a mathematical model based on collected car mean speeds for different circumstances relating to bicycles, with slightly different results. In Zhang et al. (2007) empirical data was used to quantify the bicycle distribution in a wide lane in terms of distribution, critical gap and follow-up times. A logit model [10] was suggested for the derivation of right-turning car flows as a function of the bicycle arrival rate, starting with the probability of cars turning into the gaps between bicycles. Yet the same scenario was investigated by Li et al. (2009b) with the combined cellular automaton model of Jia et al. (2007). While simulation updates here were synchronous in general, instances of conflict between flows were resolved using a sequential subgrouping of updates affected: the update order among these was decided using probability weighting that was an integral part of the behaviour model. A similar problem to that studied in Li et al. (2009b) was examined using a two-dimensional car-following model by Xie et al. (2009). This model allowed for interactions between motorised and non-motorised vehicles in two dimensions and included impact of other

[10] A logit model is one where a group of independent variables, combined using linear coefficients, are transformed into a probability that some event dependent upon the variables will happen.

vehicles positioned laterally and behind, when determining the behaviour of any individual unit. A signalised intersection with and without conflicts between bicycles and cars is modelled macroscopically and compared with empirical results in Xuan et al. (2009).

The amount of literature found on the two main types of bicycle interaction with motorised flow is an indication of the significance of these effects where bicycles are present in traffic. None of the reported studies matches exactly the conditions of interest in this thesis, mainly because they all deal with wide bicycle lanes. However, the results, particularly where they are associated with collected field data, are useful since they should be comparable with our results. Also, while a lot of work is being done on individual features of bicycle-car dynamics, there do not seem to be any efforts to define comprehensive models that would include bicycles and relevant interactions in mixed network traffic.

2.5 Summary

This chapter reviewed literature relating to traffic research. It first provided a general historical and disciplinary setting, including background sections that reviewed sample writings on control, routing and empirical data analysis, then narrowing its focus to models of traffic flow. These were looked at in more detail, specifically work on each of the modelling approaches: macroscopic, mesoscopic and microscopic, both for uninterrupted (motorway) traffic and for networks.

A scarcity of reported work on heterogeneous traffic was noted and, in particular, the absence of any cellular automata network model that would incorporate vehicle types as different as bicycles and cars and handle their interactions, both on streets and at intersections. For example, while Gundaliya et al. (2008) and Mallikarjuna and Rao (2009) present heterogeneous flow models that successfully reproduce the fundamental diagram of real traffic, the models are of a type that would be difficult to connect into a network, indicating that perhaps an entirely different approach is needed if such heterogeneity were to be transferred into a network model. The model presented in the following chapters introduces such an approach, after establishing the suitability of CA for the task. Most of the literature was found to have mathematically formulated models for basic movement, while resorting to verbal description of interactions at intersections. Our model has been defined in a way that allows comprehensive specification of dynamics, including intersection impact.

The review of literature on traffic simulation parallel implementations found that, while there were a number of authors that carried out such work, no case was found where the same simulator was parallelised both through memory sharing and with the use of messaging to allow the comparison in performance between the two paradigms. The work presented in the thesis includes a study of this kind.

A number of papers, in particular by Chinese authors, were found to study effects of specific interactions between bicycles and motorised traffic. The abundance of this work provided confirmation of the importance of the specific bicycle-car interactions and of the need for their inclusion in a network model. While the studied scenarios do not exactly match the bicycle-car lane sharing of interest here, there is enough similarity for the collected data to be useful for comparison and validation.

Chapter 3

A Time-Space Discrete Model of Network Traffic Including Bicycles

The focus of the research for this thesis, and of this chapter, is a model of network traffic including bicycles. It is described in sections that correspond to the components of network traffic modelling identified in the reference model in Table 2.1. These are followed by a short section that analyses the relationship between some of the modelling components and a final section taking a look at parallelisation.

3.1 The spatial meta-model

The main challenges of modelling heterogeneous traffic networks, from a spatial viewpoint, are the accommodation of differently sized vehicles and the representation of contention such as that which exists at intersections and entry points onto roads (from parking areas or buildings). The traffic patterns of specific interest, i.e. those characteristic of Dublin and similar cities, typically involve sparse bicycle flows and road sharing between bicycles and motorised vehicles based on *positional discipline.* This means that for any stretch of road there is no 'hard' contention for space but there can be influence between vehicle types as the spaces they occupy on the road are not physically separated. *Positional discipline* implies that bicycles keep to the side of the road nearest the kerb, while cars allow space for any bicycles present, by staying as far away from the kerb as possible. We shall refer to the sides of the road occupied by bicycles and cars as left and right, respectively, which corresponds to Irish driving, but without loss of generality for right-left occupancy as applies in other European countries, the U.S. and others.

In constructing the spatial model, we start with the already well tested assumption (Maerivoet and De Moor, 2005) that the basic movements of vehicles on a single-lane stretch of road can be approximated using a one dimensional cellular automaton model and try to address the representation of bicycle-inclusive network traffic based on lane-sharing, the features of which can be summarised as follows:

(i) differences in vehicle size

(ii) contention for space and passage at intersections and merge points (conflicts)

(iii) lateral interactions between cars and bicycles sharing a lane.

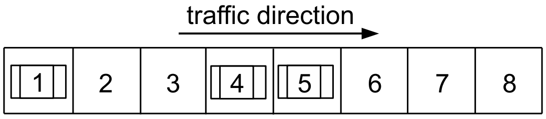


Figure 3.1: One-dimensional CA lattice with vehicles.

The model presented here uses one-dimensional cellular automaton spaces (like that shown in Figure 3.1) as building blocks and addresses the listed model requirements with some *novel modelling elements*, namely, by:

(i) allowing cellular automaton (CA) spaces with *differently sized cells*,

(ii) representing intersections as a collection of cellular automaton spaces, each shaped to represent a route taken through the intersection by a vehicle flow (for example, the cellular automaton space representing the trajectory of a left turning vehicle through and intersection will be curved, representing roughly a quarter of a ring); this allows for the *extraction of the complexity of interactions from the geometry* *in a systematic way*, as opposed to the more usual superimposition of complex interactions onto a relatively simple spatial representation of the intersection space (e.g. Deo and Ruskin, 2006b) and

(iii) including the *relationship between two CA spaces* as a type of model element and allowing the definition of relationship type values, such as "unrelated", "sharing a wide road" or "sharing a narrow road", so that, for example, the presence of the latter value in the spatial model would invoke lateral interaction rules for bicycles and motorised traffic on the relevant CA spaces.

The spatial aspect of an actual simulation model, which may include any kind of infrastructural element, can be built from pieces of information (*spatial model information items*) that define the space in terms of *spatial modelling constructs* derived from the three novel modelling elements described in the previous paragraph. Such a model is realised through transposition of the infrastructure's natural geometry into a set of spatial model information items, which are instances of the constructs. Because these spatial modelling constructs do not form a model in themselves but are, rather, definitions of the building-block types for a spatial model, they can be said to constitute a *spatial meta-model.* The one-dimensional cellular automaton space will be called a *track* and since the meta-model prescribes a way of putting together one-dimensional cellular automaton spaces to form a complex model, it will be called the *track meta-model* (TMM).

The spatial modelling constructs of the TMM can be defined as follows:

I. The **track** is a one-dimensional cellular automaton space and is the basic building-block in the meta-model. It consists of a sequence of cells of a certain size, numbered 1, 2 etc. This ascending sequence of cell indices determines the direction of travel on the track. Each track is associated with one or more *vehicle types* of a size corresponding to the track's cell size. All cells in a track are of the same size and a track's cell size can vary from track to track. A track descriptor can be defined as:

$$D_{\mathrm{T}i} := \{V_{\mathrm{T}i}, l_{\mathrm{T}i}\} \tag{3.1}$$

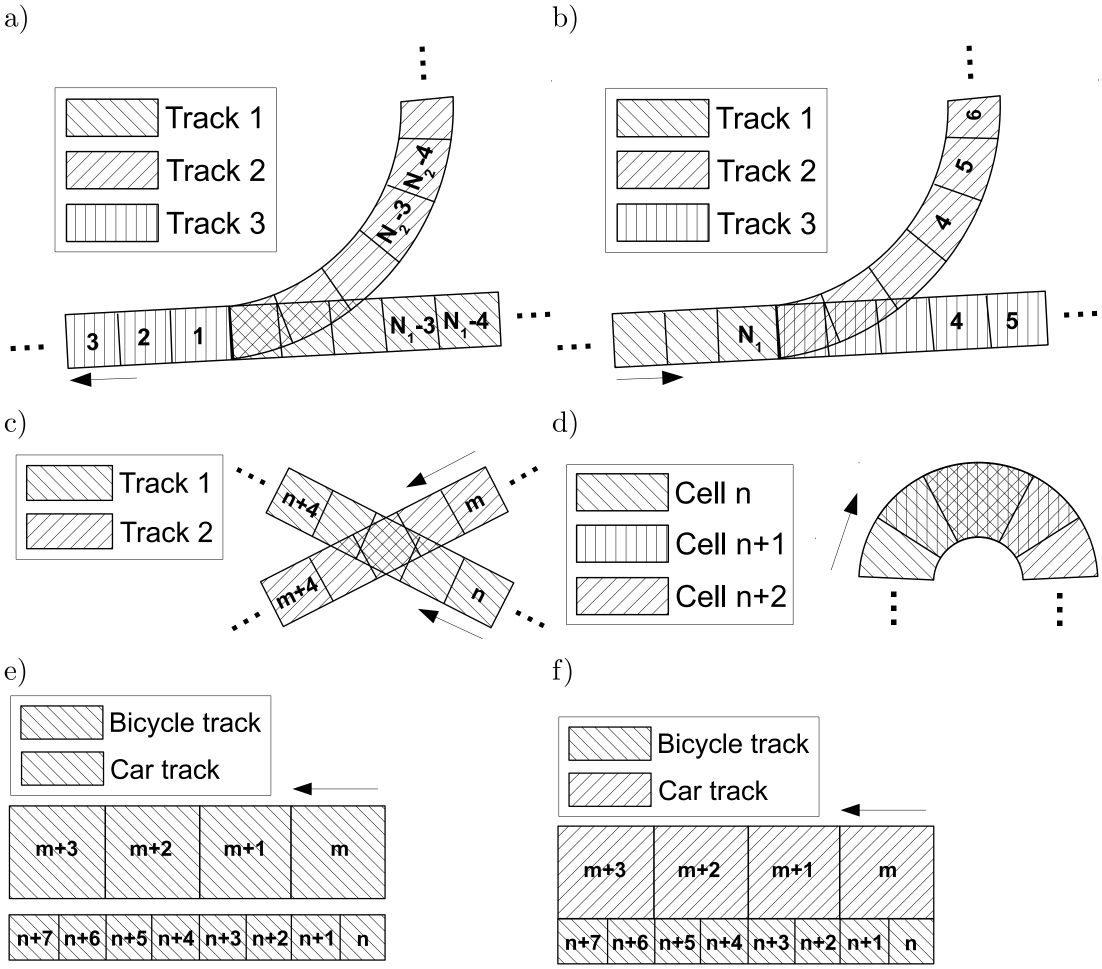


Figure 3.2: Spatial modelling construct graphical representation: (a) a merge of two one way streets, (b) a divergence of two one-way streets, (c) an intersection of two one way streets, (d) a sharp bend in a one-way street, (e) a portion of a wide one-way street shared between bicycles and cars with positional discipline and (f) a portion of a narrow one-way street shared in the same way. The arrows show track orientation. n and m are, in all cases, positive whole numbers, the value of which depends on where in the tracks the illustrated construct occurs.

where $D_{\mathrm{T}i}$ is the track descriptor for the i^{th} track in the model and $V_{\mathrm{T}i}$ and $l_{\mathrm{T}i}$ are the set of vehicle types that can be accommodated and the length, respectively, of that track (in cells). Let us say that Track 1 in Figure 3.2b is a car track. Its descriptor would then be $((car), N_1)$.

II. **Turns** on a route have bearing on the dynamics of a vehicle's movement, hence are included as a spatial modelling construct. [1]. A descriptor for a turn is also defined as a pair of values:

$$D_{\mathrm{U}i} := \{t_{\mathrm{U}i}, c_{\mathrm{U}i}\} \tag{3.2}$$

where $D_{\mathrm{U}i}$ is the descriptor for the i^{th} turn in the model, $t_{\mathrm{U}i}$ is the identifier of the track that the turn is on and $c_{\mathrm{U}i}$ is the index of the first cell of the turn. For example, the turn in Figure 3.2b has the descriptor $(2, 1)$, i.e., the turn is on the track with identifier 2 and starts at cell 1 of that track.

III. As this is a model where tracks are based on the natural geometry of the modelled infrastructure, cells of different tracks can overlap. Each pair of overlapping cells are recorded in the model as an **instance of cell overlap**. For example, in Figure 3.2a, cell $n+5$ of Track 1 and cell $m+5$ of Track 2 overlap. The model also allows for overlap between cells on the same track. This is used to model "slowing down" features of the road network, such as a turn[2]. An acute bend modelled using three overlapping cells is shown in Figure 3.2d. Overlaps can be seen in the sections of that figure where more than one type of hatch line appears. An overlap can be described as:

$$D_{\mathrm{O}i} := \{t_{\mathrm{O}1i}, c_{\mathrm{O}1i}, t_{\mathrm{O}2i}, c_{\mathrm{O}2i}\} \tag{3.3}$$

where $D_{\mathrm{O}i}$ is the descriptor for the i^{th} overlap in the model, $t_{\mathrm{O}1i}$ and $t_{\mathrm{O}2i}$ are the overlapping tracks and $c_{\mathrm{O}1i}$ and $c_{\mathrm{O}2i}$ are the particular cells in those tracks, the overlap of which is being described. An example is the overlap between cell 1 of track 2 and cell 1 of track 3 in Figure 3.2b, for which the descriptor is $(2, 1, 3, 1)$.

IV. Tracks can **connect** to other tracks, so that the first cell of one track follows the last cell of the other track, forming a longer track. If there are several connections

[1] A turn here is not defined in terms of its geometry or dimensions in relation to vehicles but rather in terms of the effect it has on vehicles, which may well be related to dimensions. A process of calibration could result in several turn types, with differ in the amount of deceleration they cause.

[2] Indeed, if the intra-track overlap were to be used in a free-standing track it would have no effect, however, if the track is inside an intersection and overlaps with conflicting tracks, the intra-track overlap has the effect of forcing the vehicle to 'linger' in the conflict zone of the intersection, causing delay to vehicles on the conflicting track, as might happen in real traffic.

from a single track to other tracks, those connections form a "divergence" and, similarly, if there are several connections into a single track, they form a "merge" in the model[3]. Apart from being the mechanism for building a network structure, connections can be used to facilitate the split of a longer track, if it has two parts with different properties. A connector can be described formally by the following:

$$D_{\mathrm{C}i} := (t_{\mathrm{C}1i}, t_{\mathrm{C}2i}) \tag{3.4}$$

where $D_{\mathrm{C}i}$ is the connection descriptor and $t_{\mathrm{C}1i}$ and $t_{\mathrm{C}2i}$ are the first and second tracks of the connection, the first cell of the second track following immediately on to the last cell of the first track. An example, in Figure 3.2b, is the connection between Track 1 and Track 2: (1, 2).

V. A **track relationship type** is defined for each specific way in which relative geometric positioning of two tracks can be the cause of some behaviour on the part of vehicles on either or both of those tracks. For example, Figures 3.2e and f show spatial models for road sharing between bicycles and cars, based on positional discipline. No track relationship types are involved in the former, even though the two tracks belong to the same road, as there is enough space for comfortable side-by-side movement of all vehicles. In the latter, however, a relationship of type "sharing a narrow road" is assigned to the car and bicycle track pair and this will affect the dynamics of the vehicles on the two tracks. Another example of a relationship would be that between two adjacent car tracks representing lanes on the same road, which could be labelled "adjacent lanes" and which would allow for lane changing by means of vehicles moving from one track to another. A relationship between tracks can be described as:

$$D_{\mathrm{R}i} := (t_{\mathrm{R}1i}, t_{\mathrm{R}2i}, r_i) \tag{3.5}$$

where $D_{\mathrm{R}i}$ is the descriptor for the i^{th} track relationship in the model, $t_{\mathrm{R}1i}$ and

[3] A merge or a divergence is described fully with a connectors and overlap instances.

$t_{\mathrm{R}2i}$ are the two tracks in that relationship and r_i its type. The descriptor for the relationship in Figure 3.2f is (bicycle track, car track, sharing a narrow road). Unlike the other modelling constructs, where the graphical representation is intuitively interpretable, for the relationship we introduce the convention that adjacently drawn tracks are sharing a narrow lane and those separated by a space are sharing a wide lane (with no interaction between tracks)[4].

Spatial modelling construct	**Description**	**Example in Figure 3.5 or indicated**
track specification	description of tracks in terms of cell size, cell count and direction of cell numbering (corresponding to that of vehicle movement)	there are 9 tracks (labelled *a*-*i*), with sizes and directions of movement as in the picture
turns	points at which a turn (left or right) starts in the track (this may affect speed, so is a relevant feature of a road model)	there is a turn at the first cell of track *f* (to the left)
track connections	where one track extends another one so that the first cell of the extending track follows the last cell of the extended one	the following track connections exist: *d-f, d-e, f-h, g-h*
cell overlap instances	these express conflict between tracks	for example, first cell of track *a* overlaps with first cell of track *b* etc.
track relationship type	indication if two tracks are geometrically positioned so as to cause inter-track interaction (other than that at conflicts) between vehicles and if yes, what kind of interaction	"sharing narrow lane" and "sharing wide lane", respectively, in Figure 3.2e and f

Table 3.1: The track meta-model: spatial modelling constructs and examples

Using only these constructs, an entire spatial model can be described as a set of

[4] If the model is extended to include other relationships, such as the adjacency of lanes in a multi-lane road, graphical correspondents can be introduced for these.

instances of those constructs:

$$D_{\mathrm{M}} := \{D_{\mathrm{T1}}, ..., D_{\mathrm{TN_T}}, D_{\mathrm{U1}}, ..., D_{\mathrm{UN_U}}, D_{\mathrm{O1}}, ..., D_{\mathrm{ON_O}}, D_{\mathrm{C1}}, ..., D_{\mathrm{CN_C}}, D_{\mathrm{R1}}, ..., D_{\mathrm{RN_R}}\} \tag{3.6}$$

where D_{M} is the descriptor of the spatial model, D_{T1} is the first track descriptor and $D_{\mathrm{TN_T}}$ is the descriptor with ordinal number N_{T}, which is the number of tracks in the model. Similarly, the pairs of values $(D_{\mathrm{U1}}, ..., D_{\mathrm{UN_U}})$, $(D_{\mathrm{O1}}, ..., D_{\mathrm{ON_O}})$, $(D_{\mathrm{C1}}, ..., D_{\mathrm{CN_C}})$ and $(D_{\mathrm{R1}}, ..., D_{\mathrm{RN_R}})$ are the first and last descriptors of turns, overlaps, connectors and track relationships, respectively.

Apart from the defined spatial modelling constructs (summarised in Table 3.1 for convenience), there are a number of derived constructs that will be used in the definition of the other parts of the model. These are:

i. A **track entrance point** is the entry side, with respect to the direction of movement, of the first cell of that track.

ii. A **network entrance point** is the entry side, with respect to direction of movement, of the first cell in a track that is not back-connected i.e. a track that is not the second track in any connection.

iii. A **track exit point** is the exit side, with respect to the direction of movement, of the last cell of that track.

iv. A **network exit point** is is the exit side, with respect to direction of movement, of the last cell in a track that is not forward-connected i.e. a track that is not the first track in any connection.

v. A **route** is a collection of tracks, starting with an entrance point and ending with an exit point, defined as:

$$D_{\mathrm{R}i} := \{t_{\mathrm{R}i1}, t_{\mathrm{R}i2}, ..., t_{\mathrm{R}i\mathrm{N}_{\mathrm{TR}i}}\}, \tag{3.7}$$

$$\forall k \in \{1, ..., N_{TRi} - 1\} \;\; \exists\, c_{\mathrm{R}ik} = \{t_{\mathrm{R}ik}, t_{\mathrm{R}i(k+1)}\}, \tag{3.8}$$

$$\nexists\, c = \{t, t_{\mathrm{R}i1}\}, \;\; \nexists\, c = \{t_{\mathrm{R}i\mathrm{N}_{\mathrm{TR}i}}, t\} \tag{3.9}$$

where $D_{\mathrm{R}i}$ is the descriptor of the i^{th} route in the network, $t_{\mathrm{R}i1}$, $t_{\mathrm{R}i2}$... are tracks in that route, $\mathrm{N}_{\mathrm{TR}i}$ is the number of tracks in route i, $c_{\mathrm{R}ik}$ is the k^{th} connector in route i and c and t are any connector or track, respectively. Overlap instances can be defined in terms of route cells rather than track cells:

$$D_{\mathrm{O}i} := \{r_{\mathrm{O}1i}, c_{\mathrm{O}1i}, r_{\mathrm{O}2i}, c_{\mathrm{O}2i}\} \tag{3.10}$$

where the definition is the same as in Equation 3.3, except that routes and cell indices with reference to the beginning of the *route* are used.

vi. **Conflicts** are caused by overlap between tracks, causing contention for passage and space between vehicles moving on the conflicting tracks. Information about conflicts can be extracted from the spatial model and expressed in the form of the two descriptors defined below under the following condition: *tracks must be chosen so that all entrance points of individual tracks in the model do not intersect with any other tracks in the model.* The absence of this condition would unnecessarily complicate the model as described in the next paragraph.

A conflict consists of two **views**, one for each track involved, and can be described as follows:

$$D_{\mathrm{K}i} := \{v_{i1}, v_{i2}\} \tag{3.11}$$

where $D_{\mathrm{K}i}$ is the conflict descriptor for the i^{it} conflict in the model and v_{i1} and v_{i2} are the two views of the conflict. The descriptor of a view is:

$$D_{\mathrm{V}iv} := \{t_{\mathrm{V}iv}, c_{\mathrm{V}iv}, l_{\mathrm{V}iv}\} \tag{3.12}$$

where $D_{\mathrm{V}iv}$ is the view descriptor for view v of the i^{th} conflict, $t_{\mathrm{V}iv}$ is track v in the conflict, $c_{\mathrm{V}iv}$ is the starting cell of the *conflict zone* on track v of the conflict and $l_{\mathrm{V}iv}$ is the length of the *conflict zone* on track v of the conflict; $v \in \{1, 2\}$. As an example, conflict view descriptors for the conflict in Figure 3.2c, are $\{1, n+1, 3\}$ and $\{2, m+1, 3\}$. The conflict zone on each track has 3 cells.

The extraction of information about a conflict from the TMM-based spatial model can be performed by identifying the *conflict zones*. A conflict zone for a track with respect to another track is a collection of contiguous cells of the track that each overlap with at least one cell in the other track. The conflict zone either starts with the first cell of the track or is preceded by a cell of the track that does not overlap with any cells in the other track. The conflict zone also either ends with the last cell of the track or is followed by a cell of the track that does not overlap with any cells in the other track. In order for the geometrical conflicts to be extracted correctly in this fairly simple manner, the model must fulfil the condition of clear track entry points, stated above. If the condition were not imposed on the model, situations like that shown in Figure 3.3a would be possible. Here profiles $p1$ and $p2$ represent connectors between pairs of tracks $(t3, t4)$ and $(t1, t2)$, respectively, and both lie in the middle of a conflict. Allowing these configurations would require a more complex conflict extraction approach. It should be noted, however, that the conflicts extracted from the model through the identification of conflict zones, as described above, do not in all cases coincide with the geometric overlaps of the tracks. For example, the conflict sketched in Figure 3.3b consists of a two-cell conflict zone in track t_1 and a two-cell conflict zone in track t_2. The geometrical crossings, of which there are two in the picture, one between cells c_{12} and c_{21} and the other between cells c_{11} and c_{22} are not identifiable in our model[5] since the shape of the overlap is not a pertinent feature. Finally, Figure 3.3c shows an anomalous case in terms of conflicts resulting from conflict extraction performed in the described way (using contiguous groups of cells overlapping with the other track). Here two conflicts are extracted, one between track t_1 (cells c_{11} and c_{12}) and track t_2 (cell c_{23}) and the other one between track t_1 (cells c_{11} and c_{12}) and track t_2 (cell c_{21}). This result does not make intuitive sense although it would not cause problems in a simulation. If track t_2 were much longer, with many cells between c_{21} and c_{23}, it would be a better design to split track t_1 into two tracks.

[5] This is just as well in the particular case of Figure 3.3b, since it is a configuration that would cause deadlocks if interpreted as two separate conflicts.

In the case in Figure 3.3b, however, a split would in fact be undesirable as it would cause deadlocks during simulation. Ultimately, some heuristics have to be applied in the design of the tracks and their connections.

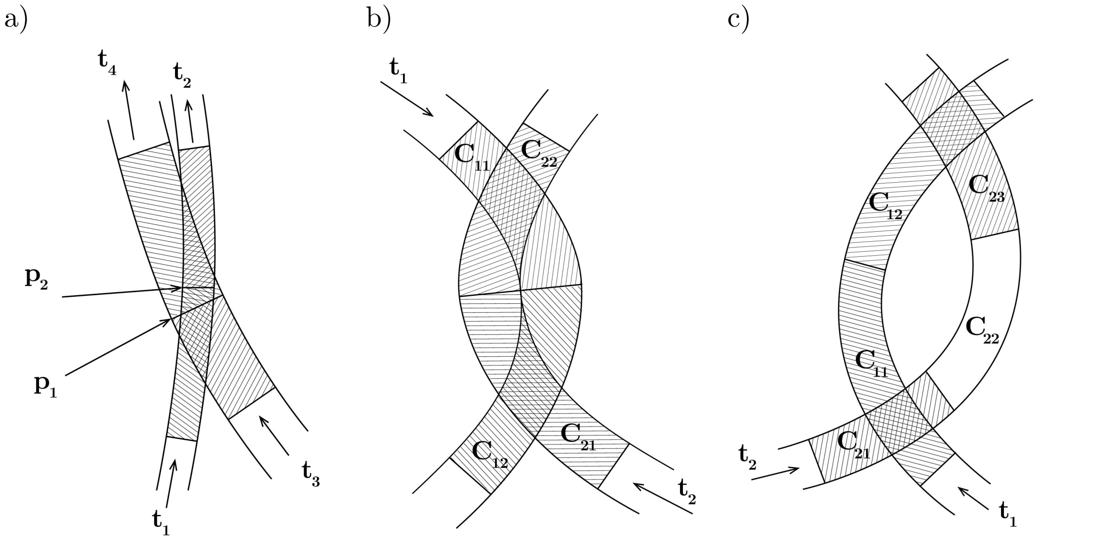


Figure 3.3: Sketches of special conflict cases: a) cell overlap does not entirely define conflict; b) two conflicts with the same conflict zone in track 1; c) two conflicts that must be treated as 1 to avoid deadlocks

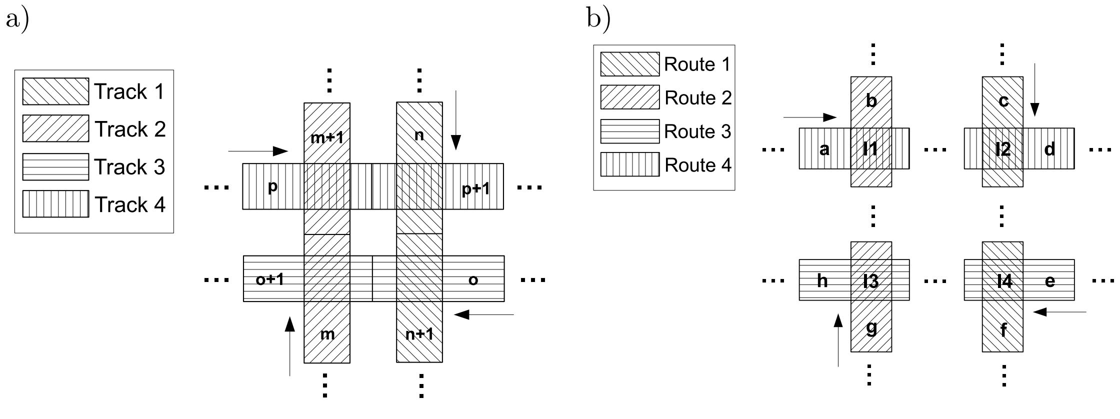


Figure 3.4: The graphical representation of an intersection deadlock (a) and an internode deadlock (b)

A case in point is a slightly different configuration, which has the potential to cause deadlocks, and is shown in Figure 3.4. If the deadlock-causing ring of overlaps belongs entirely to a single TE, like the one in Figure 3.4a might, deadlocks can be avoided by extending the conflict zones so that each spans two of the overlaps on the ring. Then, for example, the conflict between Track 2 and Track 3 in Figure 3.4 would be transformed from $\{\{2, m, 1\}, \{3, o+1, 1\}\}$ into $\{\{2, m, 1\}, \{3, o, 2\}\}$ and would include some non-physical cell overlaps. Internode deadlocks, such as the one in Figure 3.4b, are not as easy to eliminate.

While this mathematical description of the TMM is useful for the purpose of clearly defining the way in which information about a network space is organised, a graphical representation of construct instances is much more immediately informative of specific topologies. An example of a graphical representation of a topology, in terms of the TMM constructs, is shown in Figure 3.5. Table 3.1 contains a summary of the modelling constructs, with reference to graphical representation in the example topology.

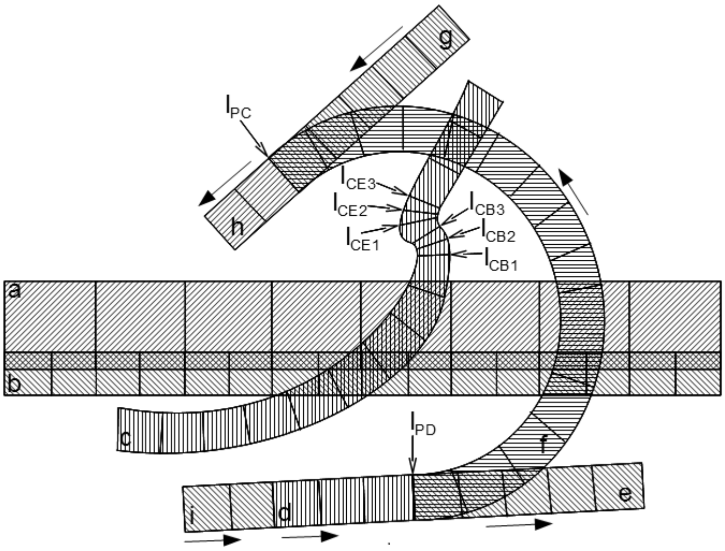


Figure 3.5: An example infrastructure graphically transposed into track meta-model constructs: tracks are labelled *a*-*i*; position pairs $(l_{\text{CB1}}, l_{\text{CE1}})$, $(l_{\text{CB2}}, l_{\text{CE2}})$ and $(l_{\text{CB3}}, l_{\text{CE3}})$ are beginning-end pairs of three overlapping cells; l_{PC} is a point of convergence, or merge (tracks f and g merge into track h); l_{PD} is a point of divergence (track d diverges into tracks f and e)

The example in Figure 3.5 is deliberately made particularly artificial and is unrealistic in order for the focus to remain with the modelling constructs. Subsection 3.1.1, however, describes the topological element models actually used in simulations.

3.1.1 Topological element models

The topological element models (TEM) presented here are elemental spatial models based on the TMM that can be used to represent the space in a traffic simulation, either in a stand-alone manner or as part of a bigger space constructed from several topological elements. The choice of infrastructure elements to represent as TEMs was based on maximising re-usability in different network configurations but also on implementation considerations. The latter were essentially the requirements that no cell should span TEMs and that any two overlapping cells should be in the same TEM.

For TEMs to be connectible into a larger network, they must expose some connectors of the tracks that they contain. Places where TEMs expose tracks are *called topological element (TE) connectors*. A TE connector may be interchangeable with a track connector but often it exposes a group of track connectors forming a merge or a divergence.

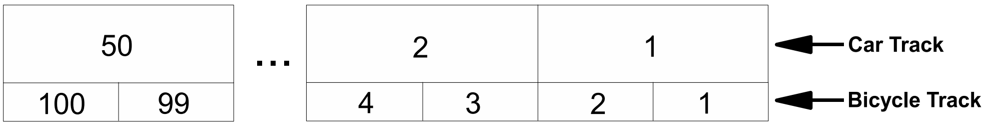


Figure 3.6: Topological element model of a single-lane straight road stretch shared by bicycles and cars on the principle of *positional discipline*.

Cell size. At this stage, i.e. during the design of the actual spatial model or model elements, the relative size of bicycle and car cells must be decided upon, with consideration given to real-world correspondence. Currently, a single mapping scheme is being used, with car cells 5m long. This cell length is shorter than the 7.5m of the original Nagel-Schreckenberg model but is better suited to exclusively urban traffic, since it corresponds to lower velocities (1 cell per second is equivalent to 18 km/h with 5m cells). We use a bicycle cell of length 2.5m, which is half the length of a car cell. Also, we specify the width of the car cell as 1.67m and that of the bicycle cell as 1.25m. Because tracks cross each other in the model, the width plays a role, which did not need to be accounted for in (Nagel and Schreckenberg, 1992).

The topological element (TE) model for a straight road stretch with one narrow

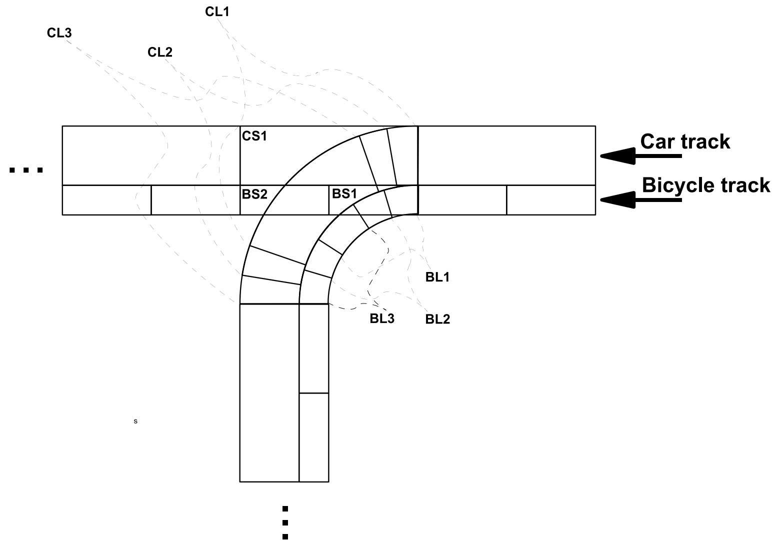


Figure 3.7: Topological element model of a left turn; the picture includes 1 car cell and 2 bicycles cells of each track connected to the left turn topological element; the cells of all the tracks in the turn TE are marked, for example, BL1 is the first cell in track “bicycle left” and BS2 is the second cell in track “bicycle straight”)

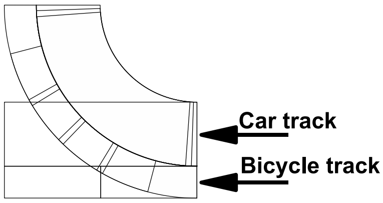


Figure 3.8: Topological element model of a right turn; the straight tracks are identical to those in Figure 3.7, while the right-turning car and bicycle tracks contain 3 and 5 overlapping cells, repsectively. There is some overlap between cells withing the tracks, i.e. all the cells in the car track overlap with each other, while each cell in the bicycle track overlaps with the next cell in the same track.

lane is shown in Figure 3.6. It has two tracks, a track that accommodates cars and a track that accommodates bicycles. By convention explained earlier, the bicycle and car track shown as sharing a side means that they are sharing a *narrow* lane. The cells are numbered and show that the particular track represented has 50 car cells and 100 bicycle cells, but the length is configurable as it can be modified without changing the semantics of the model. All the other topological element models introduced here consist of lanes that are the same as this one in that it is shared by bicycles and cars

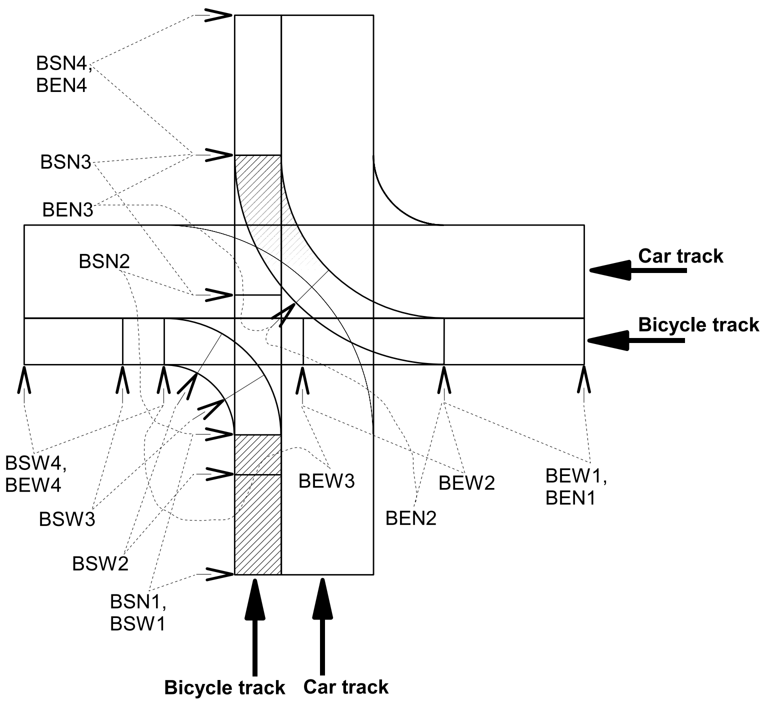


Figure 3.9: Topological element model of an intersection of two one-way roads; here all the cells are marked with four character names consisting of indicators for vehicle type, track starting point, track end point and ordinal number of cell (for example BEW3 is the 3rd cell in the bicycle track running from east to west (right to left). Car cells are not shown for the east-north and south-west car tracks, but they, like the east-west and south-north car tracks, contain 2 cells each.

and that it is narrow.

Figures 3.7 and 3.8 shows the topological element models for a left turn and for a right turn. The topological element models of intersections of two one-way roads and two two-way roads, respectively, are shown in figures 3.9 and 3.10. Four different configurations of the latter are used in simulations: 1) *basic*, which corresponds to the picture in the number of tracks and how they overlap, 2) *one-view*, whereby all the conflicts as seen from the point of view of any vehicle entering the intersection are rolled into one, 3) *cell-less*, which is the same as one-view but with tracks tha thave 0 cells each and 4) *phantom*, where cross-conflicts are ignored and only merge conflicts are heeded (a vehicle passing through an intersection is exclusively interested if the first cell on the other side can accommodate it)

Finally, Figure 3.11 is an example of a road stretch with entrances and exits (this

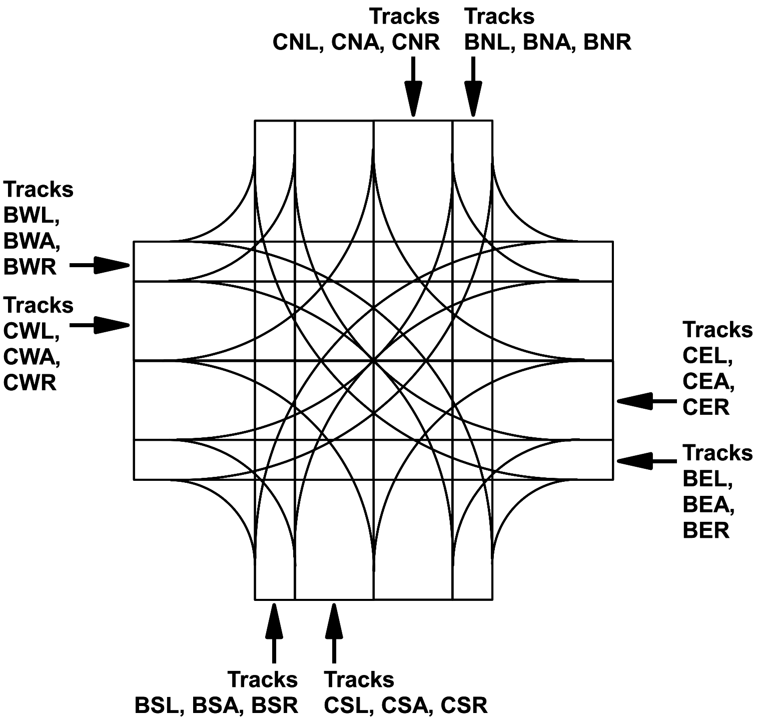


Figure 3.10: Topological element model of an intersection of two two-way roads; the cells are not shown to avoid clutter in the picture, but the tracks are marked with three character names consisting of indicators for vehicle type, track starting point and the direction (left, straight ahead or right) in which the track runs from its starting point (for example BEA is the name of the bicycle track running from the east and ahead.

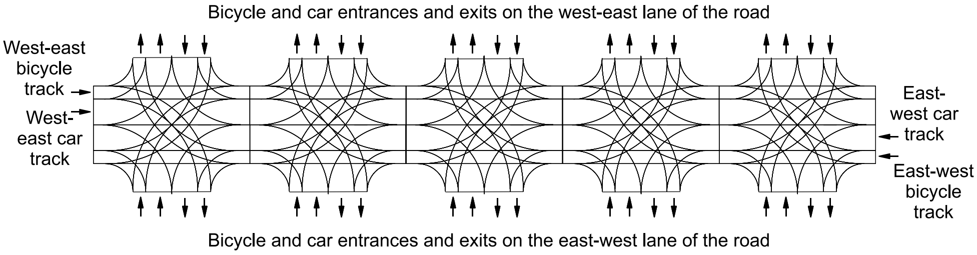

Bicycle and car entrances and exits on the west-east lane of the road
West-east
bicycle
track
West-
east car
track
East-
west car
track
East-west
bicycle
track
Bicycle and car entrances and exits on the east-west lane of the road

Figure 3.11: Topological element model of a two-way road with entrances and exits. This topological element model is composed of a configurable number of sections similar to the intersection of two two-way roads.

one has 5 entrances/exits on each side of the road).

While topological element models are not intrinsically part of the meta-model and not essential for the description of a simulation system based on the TMM, they ease network description and will also be the basis for modular implementation of the simulation program. In the context of TMM-based models they will also be referred to as simply topological elements (TEs).

3.2 Control model

In the context of the reference model shown in Table 2.1 the control aspect of a traffic model has the aim of resolving conflicts in the network. It is, thus, related to the overlap constructs of the network, as opposed to routing, which deals with divergence points. The concepts it models are effectively part of the infrastructure, albeit extended to include the entire environment within which vehicles move. While traffic lights are physically present, the effect of priority and other rules on shaping that environment is no less important.

Each conflict of the spatial model (i.e. each instance of the construct defined by Equation 3.11) must have a means of resolution in the simulation model. This may be a traffic light or a priority value.

A priority value specifies whether a route has priority over the other route in a conflict and is associated with a *conflict view* (see Equation 3.12). The priority values are thought of as configurable in the model. The priority values defined are:

- Priority
- Yield
- Both

Normally, “Priority” and “Yield” will be assigned to a conflict as a pair, i.e. each value will be assigned to one of the views in a conflict. For some special applications, such as investigating blockages this may not be the case. The value “Both” is assigned to a conflict as a whole and denotes a situation where priority is assigned dynamically on a random basis, and is meant to model situations in traffic where priority is not clear or unknown to the drivers.

Only a very simple model of traffic lights is included, with each track assigned a separate light and only two configurable phases (green and red).

3.3 Movement model

In our model for network traffic including bicycles, vehicles move along routes (see Equation 3.5) defined using the TMM. These routes are in fact one-dimensional cellular automaton spaces themselves, albeit within a framework of different model constructs that can also affect the movement dynamics. We build the extended movement rules, which include TMM-specific elements, around the original rules by Nagel and Schreckenberg (1992). Even though these do not reproduce all the properties of traffic flow on motorways, they are sufficient as a base for the purpose of modelling movement in an urban network (Maerivoet and De Moor, 2005). The rules, as formulated by Nagel and Schreckenberg (1992), are:

1. Acceleration: if $v_i < v_{\mathrm{MAX}}$ and $v_i < d_i$, $v_i \rightarrow v_i + 1$
2. Slowing (due to cars ahead): if $d_i < v_i$, $v_i \rightarrow d_i$
3. Randomisation: if $v_i > 0$, with probability p_{R}, $v_i \rightarrow v_i – 1$
4. Vehicle motion: each vehicle is advanced v_i cells along the track

where v_i is the velocity of the i^{th} vehicle, v_{MAX} is the maximal velocity, d_i is the number of free cells between the i^{th} vehicle and the vehicle ahead of it on the track and p_{R} is the randomisation parameter, used to introduce stochasticity into the model. The right arrow ($\rightarrow$) denotes a velocity update to the value shown on the right-hand-side of the arrow. All the variables are dimensionless, the velocities representing the number of cells traversed per time unit. At each time step of the simulation the rules are applied to each vehicle in the system.

In order for these rules to be used with the TMM, they need to be converted to a single-limit form. This and some other relevant concepts are discussed before the actual rules for network traffic with bicycles are specified in Section 3.3.7.

3.3.1 Synchronous vs. asynchronous update

The first three rules of the Nagel-Schreckenberg model have the purpose of velocity update, while the last rule updates the vehicle position, i.e. the cell that it occupies. Position update can be performed (i) in parallel for each time step, where first the velocities for all the vehicles in the system are determined, then all the vehicles are moved or (ii) in sequence, where each vehicle moves immediately upon determining its velocity for the time step. The movement rules have to be adapted to the update sequencing approach (*synchronous* vs. *asynchronous*), i.e., not all movement rules would work with both approaches. This means that the type of update sequencing has to be fixed in advance. We chose the more commonly used *parallel update*, because it is "less safe" from the point of view of conflict handling in that, during an update, potential changes to other vehicles (for example increase in velocity), taking place in parallel, must be taken into account as well as the current state of those other vehicles. This more closely models the problems of interest here than asynchronous updates, with which vehicle state take place sequentially.

3.3.2 Heterogeneity considerations

The same movement rules are applied to bicycles and cars, with one exception, which is discussed in Section 3.3.6. The difference in relative velocity stems from the difference in cell size (a bicycle cell is half the length of a car cell) and the different values for v_{MAX}. While the model is designed to 'work' with any values, $v_{\mathrm{MAX}} = 3$ for cars and $v_{\mathrm{MAX}} = 2$ for bicycles is reasonable in the context of urban traffic, since, with a time step of 1s, these velocities correspond to 81km/h and 27km/h, respectively, which is fairly realistic (for bicycles, see El-Geneidy et al., 2007).

3.3.3 A single-limit form of the Nagel-Schreckenberg (N-S) rules

The first two Nagel-Schreckenberg rules can be viewed as if based on a combined limiting value, contributed to by the maximal velocity and the distance to the vehicle ahead. With this in mind the N-S rules can be re-formulated as follows:

1. Acceleration: if $v_i < v_{\mathrm{L}i}$, $v_i \to v_i + 1$
2. Slowing: if $v_i > v_{\mathrm{L}i}$, $v_i \to v_{\mathrm{L}i}$
3. Randomisation: if $v_i > 0$, with probability p_{R}, $v_i \to v_i$–1
4. Vehicle motion: each vehicle is advanced v_i cells along the track

where, $v_{\mathrm{L}i} = \min(v_{\mathrm{MAX}}, d_i)$.

This form of the rules, which can be termed the *single-limit* form of the N-S rules, is more suitable than the original form as a basis for building a set of movement rules for network traffic including bicycles.

3.3.4 The deceleration table

A *deceleration table* contains lists of distance-dependent maximal velocity values allowed for a vehicle. The distance is to something that the vehicle encounters on its path and can be another vehicle or either a static or a dynamic feature of the infrastructure. The table serves as a model-specification tool but also provides lookup data, which will ultimately make for a faster implementation than if velocity had to be calculated. A similar concept, but with data for a single deceleration cause placed in a matrix, was used in Emmerich and Rank (1997).

Table 3.2, is the actual deceleration table used in our simulations. The data contained in the deceleration table, however, are a configurable aspect of the movement rules.

In Table 3.2, $v_{\mathrm{L}}^{\mathrm{T}}(d_i^{\mathrm{T}})$ are the velocity limits imposed by the model on a vehicle approaching a turn and they depend on the distance from the vehicle in question, with identifier i, to the turn. Similarly, $v_{\mathrm{L}}^{C}(d_i^{\mathrm{C}})$ is the velocity limit for a vehicle approaching a conflict, as a function of distance between that vehicle and the conflict. Finally, the maximal velocities $v_{\mathrm{L}}^{B}(d_i^{\mathrm{B}})$ apply only in the case that the inspecting vehicle is a car and the approached feature is a bicycle in the adjacent track. The numbers in the top row represent the distance, in cells. The table applies to the case of $v_{\mathrm{MAX}} = 3$ for cars and $v_{\mathrm{MAX}} = 2$ for bicycles. For distances not shown in the table, as for the

$\boldsymbol{d_i^{\mathrm{T}}}$ **or** $\boldsymbol{d_i^{\mathrm{C}}}$ **or** $\boldsymbol{d_i^{\mathrm{B}}}$	**6**	**5**	**4**	**3**	**2**	**1**	**0**
$\boldsymbol{v_{\mathrm{L}}^{\mathrm{T}}(d_i^{\mathrm{T}})}$**, where** $\boldsymbol{i^{\mathrm{th}}}$ **vehicle is a car**	∞	2	2	2	1	1	∞
$\boldsymbol{v_{\mathrm{L}}^{\mathrm{C}}(d_i^{\mathrm{C}})}$**, where** $\boldsymbol{i^{\mathrm{th}}}$ **vehicle is a car**	2	2	2	1	1	0	∞
$\boldsymbol{v_{\mathrm{L}}^{\mathrm{T}}(d_i^{\mathrm{T}})}$**, where** $\boldsymbol{i^{\mathrm{th}}}$ **vehicle is a bicycle**	∞	∞	∞	∞	1	1	∞
$\boldsymbol{v_{\mathrm{L}}^{\mathrm{C}}(d_i^{\mathrm{C}})}$**, where** $\boldsymbol{i^{\mathrm{th}}}$ **vehicle is a bicycle**	∞	∞	∞	1	1	0	∞
$\boldsymbol{v_{\mathrm{L}}^{\mathrm{B}}(d_i^{\mathrm{B}})}$**, where** $\boldsymbol{i^{\mathrm{th}}}$ **vehicle is a car**	∞	2	2	2	1	1	1

Table 3.2: Deceleration table: contains maximal velocities allowed for vehicles as a function of distance to the limit-imposing environment feature. Distances d_i^{T} and d_i^{C} for vehicle i are the distance to the nearest turn and the distance to the nearest unresolved conflict, respectively.

cases indicated by the symbol ∞, no limiting value applies. These velocity limits were chosen so as to allow a vehicle to reach the velocity of 1 at the cell before a turn or before reaching a bicycle in the adjacent track and the velocity of 0 at the cell before an unresolved conflict, by decelerating, at most, by 1. The distance values are expressed in number of cells, as the distance that a vehicle needs to cover in order to enter the first cell of the feature in question, such as a turn or a conflict. This means that the distance is equal to the number of cells counted between the observing vehicle and the encountered network feature plus 1:

$$d_i^{\mathrm{F}} = c_{\mathrm{F}} - c_i \tag{3.13}$$

where d_i^{F} is the distance value, c_{F} is the number of the cell (with respect to the route of the vehicle) at which the observed network feature is positioned and c_i is the number of the cell (also with respect to the route of the vehicle) at which the vehicle is positioned. For a vehicle positioned in cell m of track t_2 in Figure 3.12, the distance to the conflict is 1, since the conflict starts at cell $m+1$.

3.3.5 Conflict handling

When approaching a conflict, a vehicle must inspect it in order to determine whether, from its point of view, the conflict is resolved. Determining this is part of the behaviour of the vehicle and the conflict status affects whether a conflict needs to be taken into account by the movement rules. A resolved conflict will be ignored by the movement rules.

Whether a conflict is resolved or not is clear in cases where traffic lights are present. In the case that the conflict handling depends on priority values, conflict handling rules with reference to the TMM must be defined. If the conflict view for the track on which the inspecting vehicle is moving is labelled the *current view*, the conflict handling rules can be described as follows:

- If the current view is associated with the resolution rule "Both", this rule is converted to a temporary value of "Priority" or "Yield", effective for one time step and the following rules applied with the new value.
- If the current view is associated with resolution rule "Priority", the conflict is deemed resolved.
- If the current view is associated with resolution rule "Yield" and the vehicle is at a distance greater than 1 from the beginning of the conflict zone in the current view, the conflict is deemed unresolved, as the vehicle is considered unable to 'see' what is happening at the conflict.
- If the current view is associated with resolution rule "Yield" and the vehicle is at the cell right before the first cell of the conflict zone in the current view (at distance 1), an inspection must take place on the other conflicting track 'upstream' from the conflict zone on that other track, for approaching vehicles. The presence of such a vehicle and its current parameter values are used to determine whether the conflict should be deemed resolved or not. The details of this are discussed below.

Figure 3.12, shows a situation that will be used to illustrate conflict inspection, without loss of generality. If Track 2 has priority, a vehicle approaching on Track 1 cannot go beyond cell n without entering the conflict zone. It must inspect Track 2 backwards for a vehicle that might enter the conflict zone in the next time step. If

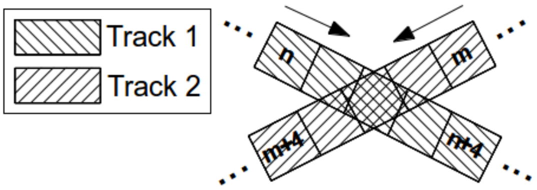


Figure 3.12: The graphical representation of a conflict between two tracks

the inspecting vehicle on Track 1 finds a vehicle on Track 2 satisfying this condition, it sees the conflict as unresolved, otherwise it sees it as resolved. In order to decide if the approaching vehicle will enter the conflict zone, the inspecting vehicle must know the velocity and intended route of the approaching vehicle. This knowledge has equivalents in real traffic, since drivers see how fast other vehicles are moving and know of their intended routes based on indicators (or on cyclists extending their arms to show that they are turning). The conflict is resolved from the point of view of the inspecting vehicle if **A and (B or C)** are true, where:

A) there is no vehicle on Track 2 that is in a conflict zone *blindspot*

B) $v_{\text{app.}} \times g_{\text{A}} \leq d_{\text{free}}$, where $v_{\text{app.}}$ is the velocity of the approaching vehicle, g_{A} is the accepted gap (Brilon et al., 1999) and d_{free} is the number of free cells from the first cell inspected and up to the approaching vehicle

C) the approaching vehicle's route is not bringing it into the conflict zone

Otherwise the conflict is not resolved. A blindspot occurs where the entry cells into the conflict zone on the two involved tracks do not overlap. For example, in Figure 3.12, if Track 2 has priority, in the situation where a vehicle on Track 2 enters cell $m + 1$, a vehicle approaching the conflict on Track 1, if checking only backwards for the accepted gap and inspecting for impingement, may move into cell $n + 1$, causing a deadlock. Hence cell $m + 1$ is called a blindspot, which must be checked from Track 1 during conflict inspection. In general, a blindspot consists of a sequence of cells at the beginning of the conflict zone in the priority Track that do not overlap with at least one cell at the beginning of the conflict zone on the non-priority track. An illustrative example of the fulfillment of condition (b), i.e. of a situation where the route of the approaching vehicle is decisive in the appraisal of the conflict, can be made using Figure 3.9. If a bicycle is in cell BSN1, intending to go straight, it must inspect backwards, into the car track outside of the intersection that is connected into the south-west car track of the intersection. If it finds an approaching car there with sufficient speed to enter the conflict between the south-north bicycle track and south-west car track, i.e., to enter the first cell of the south-west car track, it must check where the car is going. If the car is intending to turn left, the conflict is not resolved for the bicycle, but if the car is intending to go straight, the conflict is resolved.

3.3.6 Car-bicycle interaction due to lane sharing

Car-bicycle interaction due to lane-sharing, in addition to the already defined impingement and conflict-related interactions, will be taken to exist in the cases where cars and bicycles move on tracks that share a narrow lane. The interaction is considered one-directional, in that bicycles affect the movement of cars but not the other way around. A car, if faced with a bicycle ahead on the same narrow lane, slows down so as to have more control over its manoeuvres when overtaking the bicycle. For this behaviour we define two different models, one based on a deceleration table and the other on randomisation parameter variation.

In the case that the *deceleration table* is used, the distance in the table is the shortest distance that the car needs to travel forward in order to end up in a cell alongside a bicycle in the adjacent track. In the case that a bicycle is already in a cell alongside the car's cell, the distance is 0.

The *randomisation parameter variation* model incorporates the effect of the presence of bicycles on cars in the randomisation rule. This involves a second randomisation parameter value, the *interaction randomisation parameter*, p_{RI}, which is lower than the general one applied for movement rules. This randomisation parameter is applied to cars in the case that the distance to a bicycle in the adjacent track is within a certain time gap, i.e. the *interaction time gap* t_{GI}, meaning that at the current velocity the car would reach the cell alongside the bicycle's current cell in t_{GI} seconds.

3.3.7 Movement rules for network traffic including bicycles (NTIB)

The movement rules for network traffic including bicycles have three variations: rules for bicycles, rules for cars with deceleration table-based interactions and rules for cars with interactions represented by variation of the randomisation parameter. In all these, the conflict handling takes place after all the other velocity update rules, applied synchronously across all vehicles in the system. In other words, the first three rules (acceleration, slowing and randomisation) are applied, then conflict handling and, finally, motion, to each vehicle in the simulation. The three movement rule application stages will be called *velocity update*, *conflict handling* and *motion*.

The reason for separating conflict handling into a separate stage of rule application is that the result of conflict inspection is dependent on the velocity of other vehicles, which is not guaranteed to be known for the next time step during velocity update (since the particular other vehicle being inspected may not have been updated before the current vehicle). What follows is that bundling together all the update rules, including conflict handling, would cause inconsistency in how the rules are applied to different vehicles. It would also make implementation more complex for two reasons: a) the possibility of the other vehicle's velocity increasing would have to be taken into account when determining the available time gap and b) the fact that a vehicle on a "Priority" track is blocked would never be detected by a vehicle on the "Yield" track of the same conflict (as it would always have to allow for the possible increase of velocity by 1) and this would cause deadlocks[6].

The network traffic including bicycles (NTIB) model rules are fully defined as follows:

0. Determine route: for any divergences with $d_{\mathrm{D}} < d_{\mathrm{I}}$ pick direction to take (based on routing model)

1. Acceleration: if $v_i < v_{\mathrm{L}i}$, $v_i \to v_i + 1$

2. Slowing: if $v_i > v_{\mathrm{L}i}$, $v_i \to v_{\mathrm{L}i}$

[6] This was only discovered during simulations.

3. Randomisation: if $v_i > 0$, with probability p_{R}, $v_i \rightarrow v_i{-}1$

4. Conflict handling: if $v_i > v_{\mathrm{L}}^{\mathrm{C}}(d_i^{\mathrm{C}})$, $v_i \rightarrow v_{\mathrm{L}}^{\mathrm{C}}(d_i^{\mathrm{C}})$

5. Vehicle motion: each vehicle is advanced v_i cells along the track

where:

- d_{D} is the distance to a divergence

- d_{I} is the inspection distance i.e. the distance up to which inspections will be made in the following 4 steps

- v_i is the velocity of the i^{th} vehicle.

- p_{R} is the randomisation parameter.

  In the case of a bicycle or a car in a simulation with *deceleration table-based interaction model* (see Section 3.3.6), the randomisation parameter is a probability, and hence has a fixed value that fulfils: $0.0 \leq p_{\mathrm{R}} \leq 1.0$

  In the case of a car in a simulation with *randomisation parameter variation-based interaction model* (see Section 3.3.6), there are two fixed probability values for the randomisation parameter: p_{RG} and p_{RI}, both between 0.0 and 1.0, inclusive. The value to use at a particular time step is determined in the following way:

$$\text{if} \quad v_i \times T_{\mathrm{GI}} \geq d_i^{\mathrm{B}} \quad \text{then} \quad p_{\mathrm{R}} = p_{\mathrm{RI}} \quad \text{otherwise} \quad p_{\mathrm{R}} = p_{\mathrm{RG}} \tag{3.14}$$

- $v_{\mathrm{L}}^{\mathrm{C}}(d_i^{\mathrm{C}}))$ is the limit imposed by the closest unresolved conflict and its value is determined from the appropriate row in the deceleration table (either conflict limit row for bicycles or conflict limit row for cars). A detailed description of how conflict status is determined can be found in Section 3.3.5.

- $v_{\mathrm{L}i}$ is the velocity limit for the i^{th} vehicle.

  The formula to determine this for *bicycles* is:

$$v_{\mathrm{L}i} = \min(v_{\mathrm{MAX}}, d_{\mathrm{U}i}, v_{\mathrm{L}}^{\mathrm{T}}(d_i^{\mathrm{T}}), v_{\mathrm{L}}^{\mathrm{S}}(d_i^{\mathrm{S}})) \tag{3.15}$$

where:

- v_{MAX} is the maximal velocity for bicycles.
- $d_{\mathrm{U}i}$ is the number of unimpinged cells ahead of the bicycle. The checking for *impingement* relies on information about instances of cell overlap.
- $v_{\mathrm{L}}^{\mathrm{T}}(d_i^{\mathrm{T}})$ is the limit imposed by the closest turn ahead of the bicycle and its value is determined from the row for bicycles approaching turns in the deceleration table.
- $v_{\mathrm{L}}^{\mathrm{S}}(d_i^{\mathrm{S}})$ is the limit imposed by the closest red traffic light (signal) ahead of the bicycle and its value is determined from the row for bicycles approaching an unresolved conflict in the deceleration table (see Section 3.3.4).

The formula to determine this for *cars* in the case of *deceleration table-based interaction model* is:

$$v_{\mathrm{L}i} = \min(v_{\mathrm{MAX}}, d_{\mathrm{U}i}, v_{\mathrm{L}}^{\mathrm{T}}(d_i^{\mathrm{T}}), v_{\mathrm{L}}^{\mathrm{S}}(d_i^{\mathrm{S}}), v_{\mathrm{L}}^{\mathrm{B}}(d_i^{\mathrm{B}})) \tag{3.16}$$

and in the case of *randomisation parameter variation-based interaction model* is:

$$v_{\mathrm{L}i} = \min(v_{\mathrm{MAX}}, d_{\mathrm{U}i}, v_{\mathrm{L}}^{\mathrm{T}}(d_i^{\mathrm{T}}), v_{\mathrm{L}}^{\mathrm{S}}(d_i^{\mathrm{S}})) \tag{3.17}$$

where:

- v_{MAX} is the maximal velocity for cars.
- $d_{\mathrm{U}i}$ is the number of unimpinged cells ahead of the car. The checking for *impingement* relies on information about instances of cell overlap.
- $v_{\mathrm{L}}^{\mathrm{T}}(d_i^{\mathrm{T}})$ is the limit imposed by the closest turn ahead of the car and its value is determined from the row for cars approaching turns in the deceleration table.
- $v_{\mathrm{L}}^{\mathrm{S}}(d_i^{\mathrm{S}})$ is the limit imposed by the closest red traffic light (signal) ahead of the car and its value is determined from the row for cars approaching an unresolved conflict in the deceleration table.

- $v_{\mathrm{L}}^{\mathrm{B}} d_i^{\mathrm{B}}$ is the limit imposed by the closest bicycle ahead on the adjacent bicycle track and its value is determined from the row for cars approaching adjacent-track bicycles in the deceleration table.

Note: Finally, a note on preserving the synchronous nature of updates with the described movement rules. Step 4 (conflict handling) in fact violates this, since it includes inspection and update of velocities. However, in practice, all priorities happen to be set so that no vehicle that has its velocity updated during conflict handling actually affects the conflict handling of another vehicle. If a solution were required a buffering variable could be introduced to hold the velocity between Conflict handling and Vehicle motion and the 'real' velocity variable would be updated just before vehicle motion takes place.

3.4 Routing and vehicle generation

Two *routing models* have been used with the NTIB simulation model:

- A macroscopic stochastic routing model, which is based on system-fixed probabilities of taking various routes at divergences.

- A zone-based destination model, where each destination zone is associated with a probability and a particular destination within the zone is picked randomly.

In the case of the zone-based destination model, once the destination is known, a route to it is chosen as follows:

- the maximally allowed length for the route is: $d_{\mathrm{A}ij} = d_{\mathrm{S}ij} \times (1 + f_{\mathrm{RLT}})$, where $d_{\mathrm{S}ij}$ is the length of the shortest route from origin to destination point, f_{RLT} is a route length tolerance factor and $f_{\mathrm{RLT}} > 0$

- at each divergence, a direction is chosen randomly from among the out-directions which, if taken, can guarantee a route shorter than $d_{\mathrm{A}ij}$

The described algorithm introduces some stochasticity into the process of route choice, while allowing advance calculation of shortest paths in the network, which speeds up the process of choosing a random route. A larger tolerance factor f_{RLT} implies a greater number of routes to chose from, allowing routes to deviate more from the shortest one. However, while it should have a value greater than 0, very large values would produce unrealistically indirect routes.

Vehicle generation is based on a Bernoulli process[7] at each entrance point. Individual entrance points are associated with an insertion probability per time-step. Vehicles that fail to be inserted at an entrance point are queued for possible insertion at a later time-step.

3.5 Relationship between model components

An important characteristic of the spatial meta-model is that it facilitates a *general* behaviour model with respect to topological features. The behaviour model does not need to have understanding of anything regarding the spatial model apart from a finite number of modelling constructs, i.e. the meta-model. The knowledge of permitted movements and handling methods for these allows a vehicle to traverse *arbitrary networks* modelled in this way. The route taken by a vehicle in the network is a one-dimensional cellular automaton (CA) space (consisting of one or more tracks connected together) and the vehicle travels along it as the NTIB rules, defined around the meta-models, are applied in a sequence of time-steps.

The presence of cell overlap in the spatial model allows for actual velocities in-between the discrete values, enabling the representation of below-1 velocities during turns at intersections, which would otherwise have to be specially incorporated into the behaviour rules. This, like the use of the meta-model for the definition of simulation spaces, could have been assigned to the behaviour part of the model with equal validity, but was assigned to the spatial part.

While this fluidity of the spatial-behaviour model boundary is an interesting example of how the same system could be modelled in strikingly different ways, it is important to understand that even with the chosen assignment of roles, the complexity

[7] This is the discrete counterpart of the Poisson process.

has not been wiped out of the system, but largely moved from the behaviour domain to the spatial one, which must include all the *spatial model information items* that define the simulation space. The advantage of that location for the complexity is that it has the potential for automatic generation from geographic information, e.g. a map.

3.6 Parallel processing design considerations

A parallel implementation of a computer program, whereby the program can be executed in more than one processor in parallel, requires the program's computational problem space to be decomposed in a suitable way for assignment to multiple processors. As with most applications, the way the work is divided in traffic simulations may depend on the available means of inter-processor communication, i.e. on whether the available processors share memory or rely on messaging to 'talk'. In this section the possible approaches to parallelisation of traffic simulation are discussed for these two cases.

3.6.1 Parallel processing with shared memory

In the shared memory case, each processor sharing the load has a view of all the available memory and hence of the entire modelled system. The workload can be split without any changes to the view, by dividing the main task of the simulation, i.e., vehicle update processing. Because the rule application is synchronous, the splitting must take place three times: for the velocity updates, for conflict handling and for vehicle motion. This is shown schematically in Figure 3.13.

Apart from synchronisation, which is a concern with any kind of parallel computing, data access must be lock-protected if there is a chance of two processes accessing the same piece of data simultaneously. In the case of our traffic model locking can be avoided entirely, owing to the synchronous nature of the updates and the fact that the model is designed to facilitate those.

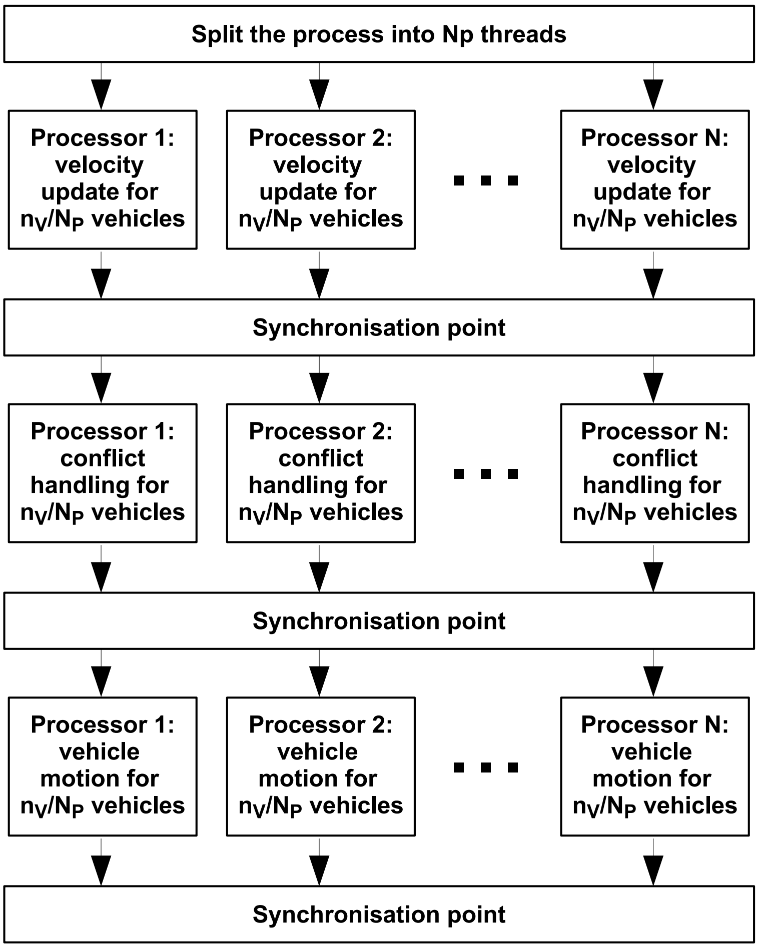

Split the process into Np threads
Processor 1: velocity update for nV/NP vehicles
Processor 2: velocity update for nV/NP vehicles
Processor N: velocity update for nV/NP vehicles
Synchronisation point
Processor 1: conflict handling for nV/NP vehicles
Processor 2: conflict handling for nV/NP vehicles
Processor N: conflict handling for nV/NP vehicles
Synchronisation point
Processor 1: vehicle motion for nV/NP vehicles
Processor 2: vehicle motion for nV/NP vehicles
Processor N: vehicle motion for nV/NP vehicles
Synchronisation point

Figure 3.13: Parallel simulation with shared memory: the diagram represents the flow of processing for a single time-step, n_{V} is the number of vehicles in the system at the current time step and N_{P} is the number of available processors.

3.6.2 Parallel processing with inter-process messaging

If communication between processes of a program execution instance relies on message passing for communication, the view of the system is separate for each processor, resident in a memory space associated with that processor. This poses the requirement for the modelled system to be partitioned between processes. Since message exchange is expensive in terms of time and processing power, the partitioning must be designed so as to minimise the amount of communication that takes place during a simulation.

The movement rules defined as part of the model each involve access to both spatial constructs and vehicle properties. This means that a partitioning approach that would decouple vehicles from occupied spatial elements would not perform very well in terms of minimising communication between processes. An optimal solution has to keep the spatial elements and the vehicles that occupy them together, hence must be based on spatial domain decomposition. This, unsurprisingly, is the approach taken by most authors (Klefstad et al., 2005; Dai et al., 2010). An algorithm we have designed for network partitioning is described in the following paragraphs.

Domain decomposition

For best performance of parallel execution of a program, the entire processing load should be evenly distributed between the processes. Otherwise the processors to which smaller loads are allocated would end up under-utilised and the running time would be longer than optimal. Where synchronisation is employed, the goal for optimal performance is to have equal loads on all processors, between any pair of synchronisation points. Also, communication between processes should be minimised, since it is expensive in terms of processing power.

With this aim, we define an algorithm for *occupancy-weighted topology-based partitioning* of networks. In the description of the algorighm, a uniform distribution of vehicles is assumed across the network, however, for cases where vehicle distribution can be predicted or where the load is balanced dynamically and the real-time vehicle distribution is consequently known, weights are assigned to each Topological Element based on the number of vehicles it contains. The

The abstract view of the TMM-based spatial model used in the partitioning algorithm is a network that has all the topological elements (TEs) of the spatial model as vertices and all topological element connectors as edges. This will be referred to as the 'partitioning network'. The following steps are applied $N_{\mathrm{P}} - 1$ number of times, where N_{P} is the number of partitions to be created.

The algorithm steps are:

- Find a non-leaf vertex, V_{R}, connected to the smallest possible number of other non-leaf vertices in the network. This non-leaf vertex will for most traffic networks be situated at the edge or a corner in the graphical representation of the network in question

- Using V_{R} as root, perform a breadth first search (e.g. Even (2011)) on the partitioning network, adding vertices to the tree until the aggregate vehicle cound in the TEs corresponding to the vertices in the discovered tree becomes $V_{\mathrm{Tree}} > V_{\mathrm{N}}/N_{\mathrm{P}}$, where V_{Tree} is the number of vehicles in the discovered tree and V_{N} is the number of vehicles in the entire network.

- Assign the TEs corresponding to the vertices in the discovered tree to the next partition to be created.

- Remove the vertices in the discovered tree from the partitioning network.

Once these steps have been applied $N_{\mathrm{P}} - 1$ number of times, the TEs corresponding to the remaining partitioning network are assigned to the last partition.

With the choice of root and the breadth-first search an effort is made to minimise the number of TE connectors that will cross inter-process boundaries. Starting the search at an edge or a corner makes use of the existing boundaries of the network for partition boundaries, reducing the inter-process boundary length. The breadth-first search minimises the perimeter of the traversed tree.

A step further from static load balancing are measures that are taken to re-balance processing loads during simulation, as conditions in the system change, or become known. The same algorithm could be used at each re-balancing point, with the dynamic re-distribution of the network becoming the main problem.

Communication

Spatially-based partitioning of the model means that vehicles and inspection will, in certain instances, have to cross boundary lines of the partitions and this represents the messages exchanged between processes. The direct way of tackling this is to exchange messages on an 'as required' basis, when processing the particular vehicles close to process boundaries. This would involve sending an inspect request and receiving a response or sending an 'actual vehicle' by the upstream process (as illustrated in Figure 3.14). This messaging pattern would have to include explicit exchange of synchronisation messages just before the application of any of the rules (implying three synchronisation points for each time-step), in order for all the processes to 'know' when the rule is ready to be applied in the entire system.

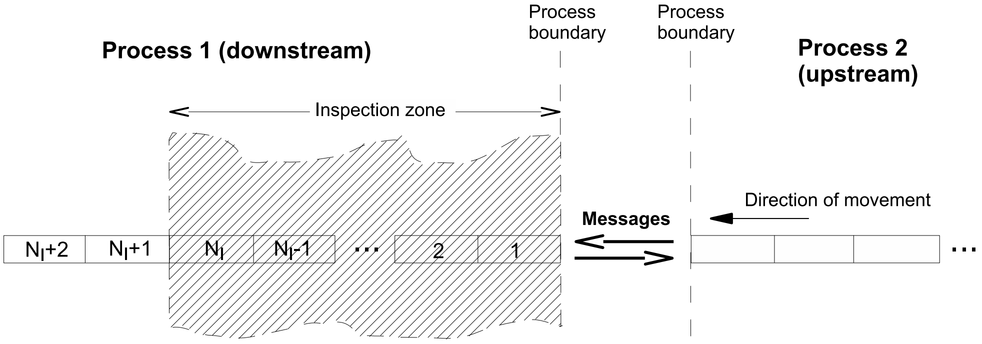


Figure 3.14: Parallel simulation with message exchange: message exchange as an extension of traffic flow on a single cross-boundary track connection.

An alternative design takes into account the fact that the exchange of fewer messages that are greater in size is more cost-effective in terms of message processing overheads than the exchange of numerous smaller messages (Livermore Computing Center, 2013). Here, synchronisation messages are used to carry all the data that might be needed by the subsequent rule. A synchronisation message marking the start of an update rule is sent at each time-step from one process to another for each cross-boundary connector in which the first process is in the downstream role and the second in the upstream role. The data placed in the synchronisation message at the start of an update is obtained by necessary inspection of the entire *inspection zone* (Figure 3.14). The width of the inspection zone, N_{I} is the maximal distance that might need to be inspected during the relevant update rule and will depend on simulation parameters. All vehicles that need to cross process boundaries at "Vehicle motion" rule application time are also packed into a single message by every process of upstream type, and sent after the rule has been applied locally. This design reduces the number of messages exchanged during a simulation.

Summarising the described communication scheme, Table 3.3 shows synchronisation points, message names, message directions and the data that the messages contain.

Synchronisation point	**Message name**	**Message direction**	**Content**
Before velocity update	inspection data for "Velocity update" rule	downstream → upstream	results of inspecting inspection zone
Before conflict handling	inspection data for "Conflict handling" rule	downstream → upstream	results of inspecting inspection zone
After vehicle motion	vehicles	upstream → downstream	vehicles crossing process boundaries

Table 3.3: Messages for inter-process communication: each message serves as a synchronisation message and carries information needed by the following movement rule.

The application of movement rules in-between message exchange takes place in the normal way, except that where calls made during "velocity update" or "conflict handling" would have to cross a process boundary they use information received earlier and where "vehicle motion" would have to cross a boundary it stores the vehicle, to be sent later, with the appropriate message.

3.7 Summary

In this chapter we have presented a generally applicable simulation model for network traffic based on an elemental cellular automaton (CA) model. A general spatial metamodel is described, which facilitates heterogeneity of vehicle type through the use of one-dimensional CA spaces with differently sized cells. This is a novel approach, not encountered in literature so far. The set of Nagel and Schreckenberg (1992) cellular automaton update rules is extended with rules for network negotiation (consisting of conflict handling and deceleration at turns), which rely on the simulation space being built exclusively from connected one-dimensional CA systems, or *tracks*. The track-based approach to building a network has been used before, for example in the commercial traffic simulator VISSIM (Fellendorf and Vortisch, 2010), but has not been proposed in the same context with CA. In order to allow arbitrary systems to be built in this way, using elemental CA spaces, our model diverges from existing practice of limiting the CA space with the requirement for planar arrangement of cells. The model allows overlaps, which simplify the representation of intersections and the specification of conflicts, while not complicating the implementation. Also, the CA rule-based formulation of comprehensive network behaviour, which is part of our model, is not usual. The general model is made specific by the addition of rules for car-bicycle interaction in a shared lane. We introduce two different models of interaction, the *deceleration table*-based model and the model with *interaction randomisation parameter*. To complement the track-based meta-model, and in a step closer to implementation, the chapter also describes *topological elements built from tracks*, which are conveniently re-usable both for scenario description and for implementation.

In addition to the model for network traffic including bicycles (NTIB), the chapter includes a section on parallelisation of the traffic simulation implementation. Consideration is given to synchronisation for the case of parallel processing with shared memory and to domain decomposition and communication for the case of parallel processing with message passing. An algorithm for domain decomposition with load balancing is presented.

Chapter 4

Implementation

The simulation model described in Chapter 3 was realised through the development of an executable simulation program. This chapter describes the design features of the program itself and of adaptations made to it for parallel processing. The details of the development and run environments are given in Appendix C.

4.1 Simulation model implementation

The structure of the C++ code that implements the simulation model is shown in Figure 4.1. All the classes shown inside the broken rectangular line constitute the *simulation framework*, which can be used for the simulation of various topological configurations. Through its spatial aspect, the framework provides the basis for the design of concrete topological elements, in the form of Topological Element sub-classes, and as such implements the *track meta-model.* The bulk of this section examines the functionality of individual elements of the design, implemented as classes, and the way they fit into the overall design of the simulation framework. A number of smaller sub-sections deal with 'special implementation topics', i.e., aspects of the implementation that are of interest at a more detailed level.

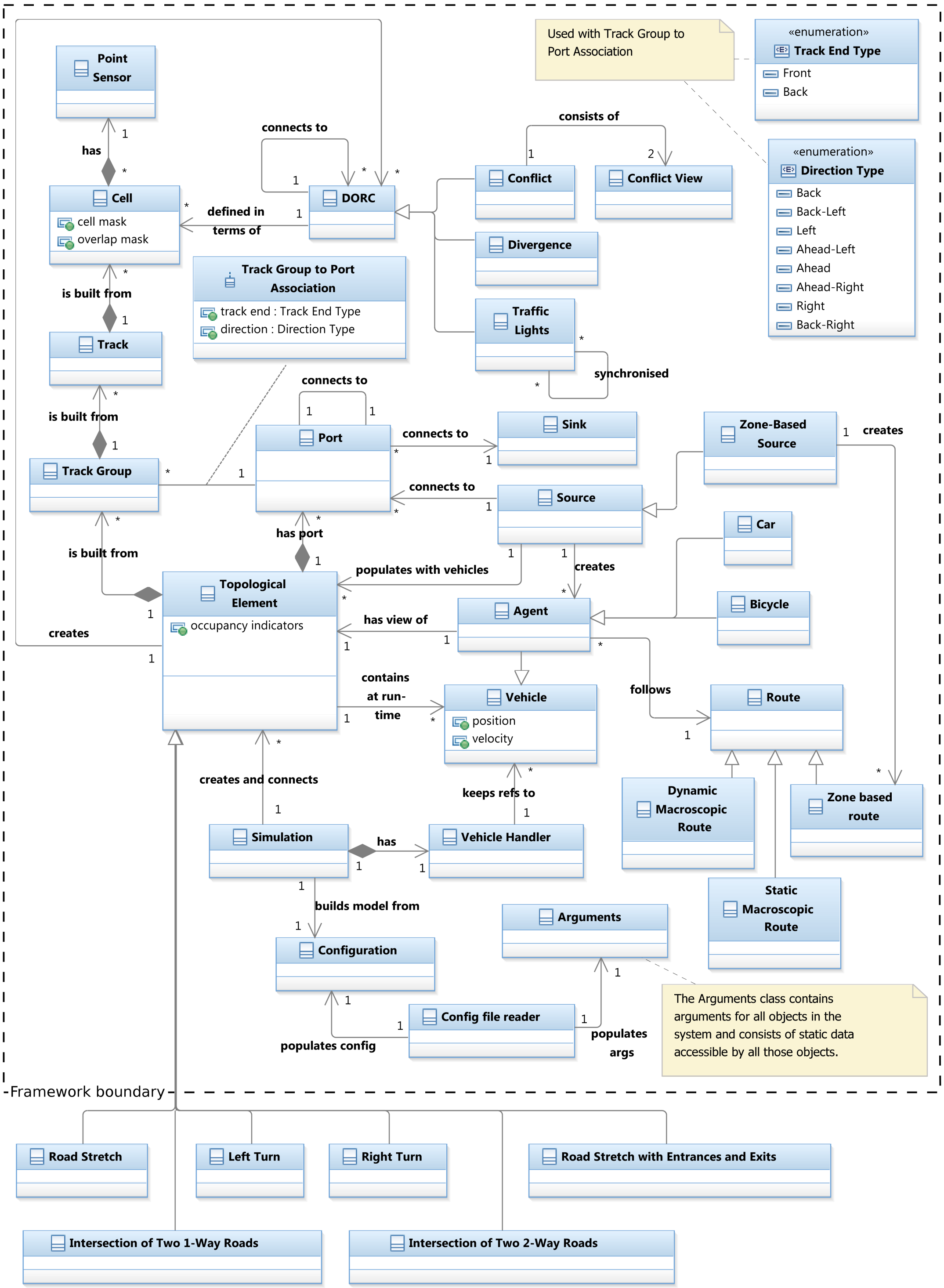


Figure 4.1: UML diagram of simulation model implementation

4.1.1 The simulation framework

Spatial: Central to the track meta-model implementation is the **Topological Element**. This class embodies the spatial track meta-model and provides the relevant building blocks for the realisation of topological elements through sub-typing. Classes **Track Group**, **Track** and **Cell**, defined within the Topological Element class, are available to sub-types of Topological Element for definition of the concrete model. Track Group represents groups of Tracks and implies track meta-model *track relationship* constructs, such as lane sharing and adjacency of multiple lanes. The Track class represents the track meta-model *track* construct, with each Track object consisting of Cell objects. Overlap is handled using Cell attributes **cell mask**, **overlap mask** and the Topological Element attribute **occupancy indicators**. The latter holds dynamic information about the cells that are occupied in the Topological Element and is used in conjunction with the attributes of Cell for impingement inspection. The attributes are all shown in Figure 4.1.

Connections between Topological Elements are formed through their **Port** objects, which provide an external plug-like interface to the starting or ending points of tracks. Thus, the connection between two Ports realises multiple inter-TE track connections. Tracks that are grouped together through Track Groups and, consequently, through Ports, take part in connections to other tracks, dictated by the tracks' transversal position in the Port. A **Track Group Association** with a Port is characterised by two attributes:

a) **connected end**, which can have values

- *end*, if the track is connected forward from the last (when counting in the direction of movement) cell or
- *begin*, if the track is connected backwards from the first (when counting in the the direction of movement) cell

b) **direction**, which can have values:

- *Back*

- *Back-Left*, a direction somewhere between backwards and left, at 225°
- *Left*
- *Ahead-Left*, a direction somewhere between ahead and left, at 135°
- *Ahead*
- *Ahead-Right*, a direction somewhere between ahead and right, at 45°
- *Right*
- *Back-Right*, a direction somewhere between ahead and right, at 315°

Connections between Topological Elements, through Ports, are illustrated in Figure 4.2.

A Port can be *in-connected*, *out-connected* or both. If a Port has at least one Track Group Association with "end" for "connected end", then it is able to form an out-connection; conversely, if it contains at least one Track Group Association with "begin" for "connected end", it is able to form an in-connection. A Port object can also connect to **Source** (by in-connections) and **Sink** objects (by out-connections), allowing the modelling of *open boundary conditions*. A network model with no such connections, where all Port objects are connected to other Port objects, either in the same Topological Element or in different Topological Elements, has *closed boundary conditions*.

A Source object creates and inserts Agents onto a specified Track for each Port that it is connected to, using a Port-specific generation probability at each time-step. A single general Source may be connected to any number of Ports. A **Zone-Based Source** is a specialised Source, associated with a zone, which only generates Agents for entrances (Ports) in that zone. A Sink object provides a seamless exit from the network. It provides the same interface as a Port but emulates a track that is perpetually empty and returns corresponding information when interrogated during simulation. Many TE Ports can be out-connected to a single Sink object, but in a scenario with zone-based routing a Sink must be provided for each car zone and bicycle zone pair in the simulation that appear at the same exit point, the Sink having been implemented to represent a single abstract destination per vehicle.

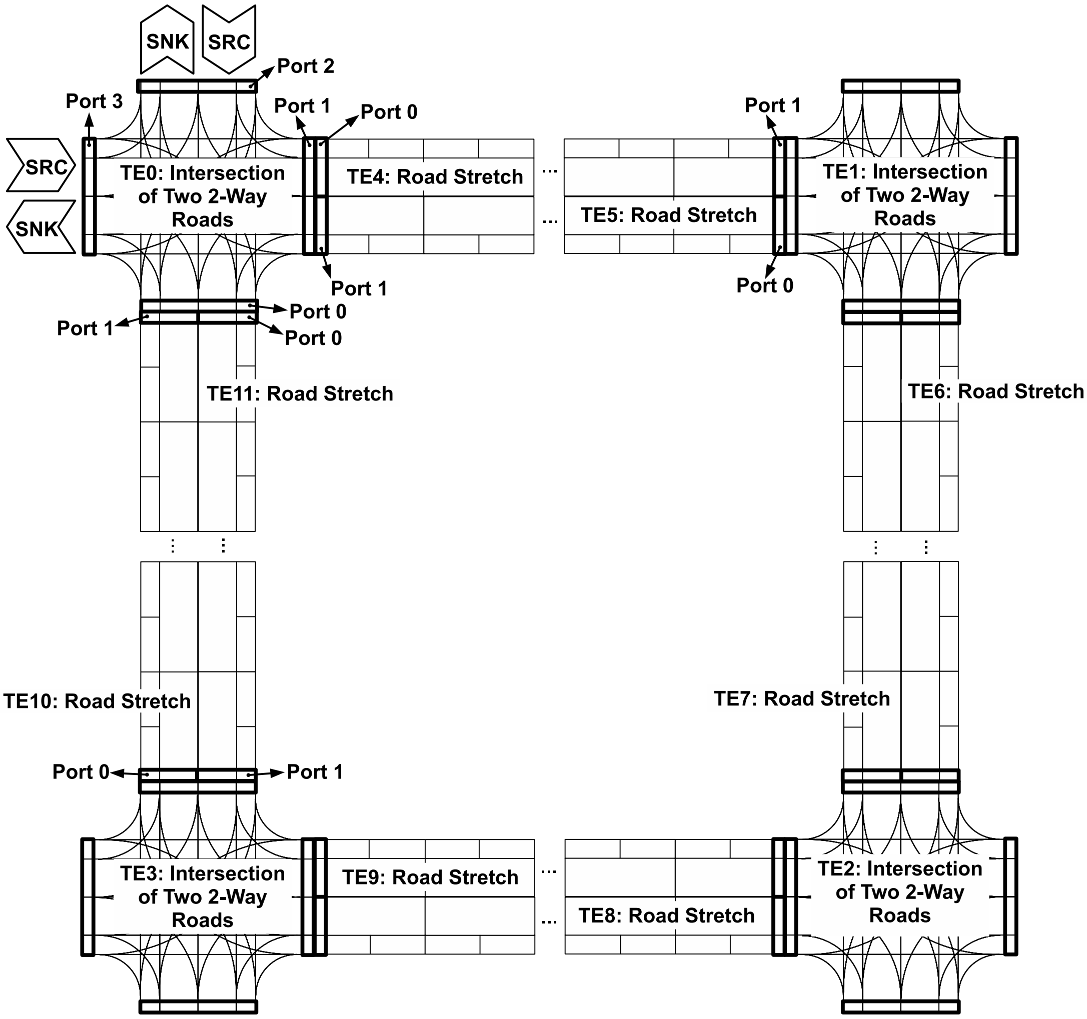


Figure 4.2: Illustration of connections between Topological Elements. Connections are shown for Ports 0 and 1 of TE0. Ports are shown as elongated rectangles with thick edges. Port 0 of TE0 is in-connected to Port 1 of TE10 and Port 0 or TE0 is out-connected to Port 0 of TE11. Port 2 of TE0 is out-connected to a Sink (labelled SNK) and in-connected to a Source (labelled SRC). The other connections are analogous to these.

Network view: The Topological Element class provides functionality for the generation of a *network view* from specified Track Groups, Tracks, Cells and overlap information. The network features, **Divergence**, **Conflict** and **Traffic Lights**, all have the same superclass, the **DORC** (standing for "Divergence or Conflict"). An illustration of an network's implementation, with structural and network-level elements included to form the entire infrastructure is shown in Figure 4.3.

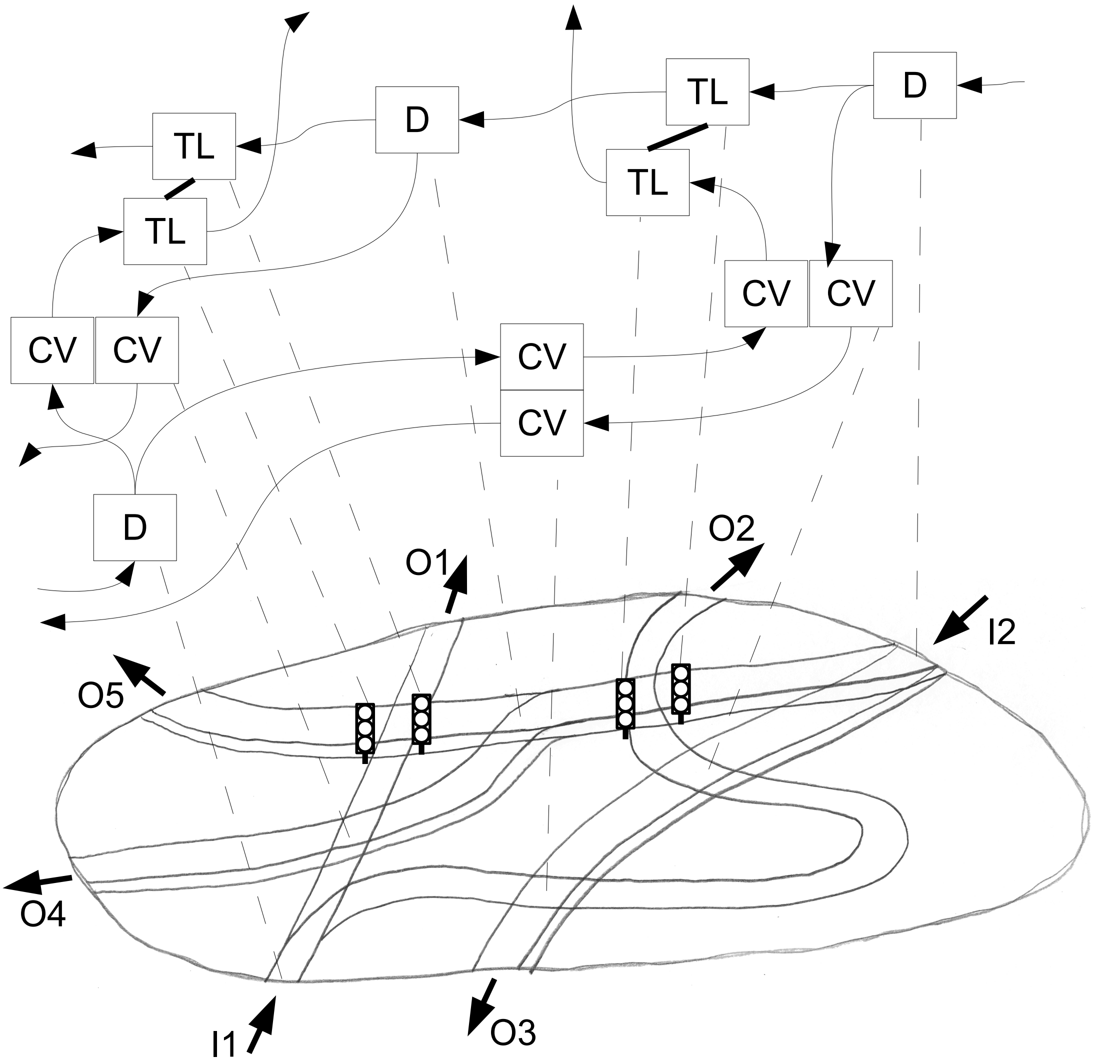


Figure 4.3: Structural and abstract views of network in simulation: for clarity of illustration, the abstract view shown is only that for car tracks; a separate view for the bicycle tracks and the addition of conflicts between different track types would complete the picture. TL: traffic lights, CV: conflict view, D: divergence; O1, O2, O3, O4, O5: exits from the network, where network tracks form out-connections to sinks, I1, I2: entrances to the network, where network tracks form in-connections from sources.

Configuration: The entire spatial structure of any simulation instance is created by an object of type **Simulation**, with the help of the **Configuration**, **Configuration File Reader** and **Arguments** classes.

Movement: While the spatial aspect of a simulation model is built using Topological Element objects, the movement rules are embodied in the **Agent** class, which has two sub-classes implemented at the moment: **Car** and **Bicycle**. The Agent has acces to information about its structural and network environment (Figure 4.3) through interrogation of the Topological Element that is *hosting* it at the given time-step (see Appendix A for specification of the interface). On the other hand, the Agent class has a subclass called **Vehicle**, which houses the externally visible information about the Agent, e.g. **position** and **velocity**. This information forms part of the inspected environment. Armed by structural, network and other vehicle information, the **Agent** class implements the movement rules described in Chapter 3.

Routing: The routing aspect of agent behaviour has been separated into the **Route** class, which provides Agents with a route, employing a number of different mechanisms, externalised with respect to the Agents. In some cases (implemented by the **Dynamic Stochastic** or the **Static Stochastic Route** class), routes are based on macroscopic requirements (for example, on the percentage of left turns that should take place during the simulation or the average number of turns that an agent should take during its lifetime); in the case of zone-based routing, as described in Chapter 3, the route is individually assigned to the Agent, which follows that route until it reaches the destination (this is implemented by the class **Zone Based Route**).

Measurements: The **Point Sensor** class provides data collection functionality that imitates a real-life point sensor on a road. It collects flow and local occupancy data. It can be placed at any Cell in the model. The **Vehicle Handler**, apart from holding references to all Vehicle objects and handling their destruction when they exit the network, collects macroscopic information on simulation dynamics, such as global occupancy of cells, global outflow, average journey length and average journey time.

Finally, apart from its role in constructing and initialising the network, the Simulation class drives the time-step to time-step running of the simulation, by performing the following in a loop:

- sends a 'tick' to all **Source** objects, so that they insert any new **Agents**
- requests a list of vehicles from the **Vehicle Handler** then iterates through all the vehicles, invoking the *velocity update* rule

- iterates through all the vehicles, invoking the *conflict handling* rule
- iterates through all the vehicles, invoking the *vehicle motion* rule

4.1.2 Implemented topological elements

The Topological Element sub-classes that have been implemented so far are also shown in Figure 4.1. These are: **Road Stretch**, **Left Turn**, **Right Turn**, **Intersection of Two 1-Way Roads**, **Intersection of Two 2-Way Roads** and **Two-Way Road with Entrances and Exits**. The Topological Element sub-classes primarily consist of structural information specific to the topological element they represent. They specify the instances of *track meta-model* constructs, described in Chapter 3, through which the shape, size, conflicts and divergences of the represented topological feature are expressed.

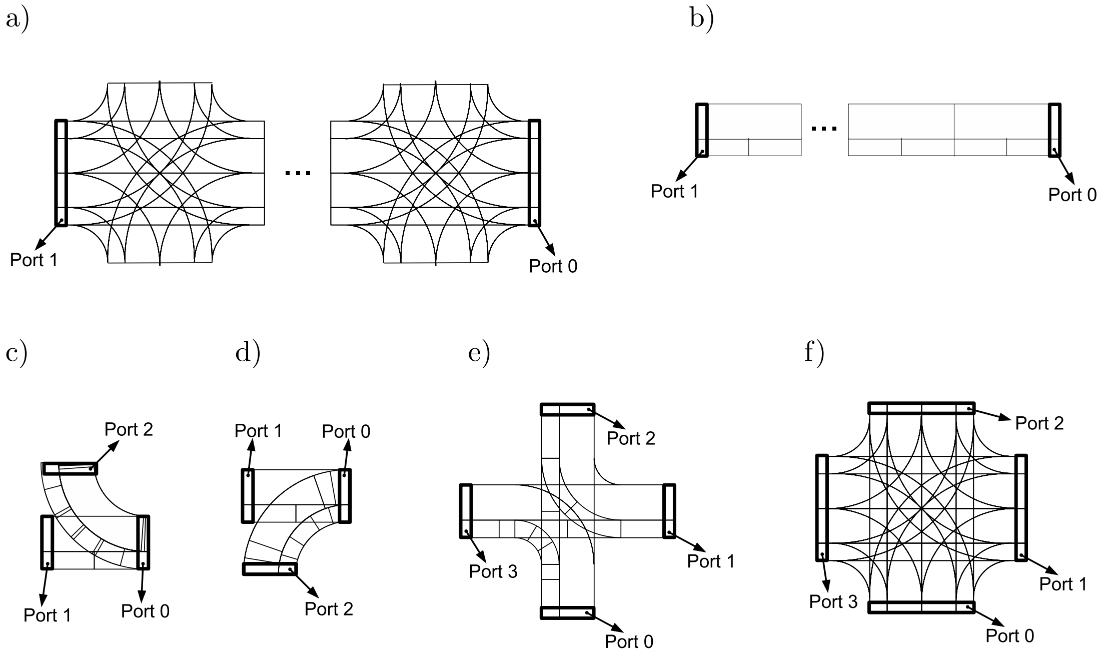


Figure 4.4: Port numbers for Topological Element sub-classes: a) two -way road with with entrances and exits, b) road stretch, c) left turn, d) right turn, e) intersection of two 1-way roads and f) intersection of two 2-way roads.

Figure 4.4 shows the port numbers for the different Topological Element sub-classes. This information will make referencing easier in scenario descriptions.

4.1.3 Cell overlap

The information about overlap between cells is provided by the sub-classes of Topological Element (TE) and is in the form of an array of binary values for every cell. An occupancy array of binary values for the TE can be checked efficiently, using the overlap binary array for the relevant cell.

4.1.4 Sources

The most straightforward way of implementing the Bernoulli process of vehicle insertion into the network is by generating a random number for each entrance at each time-step and comparing that number with the insertion probability for the entrance, normalised to the maximal possible random number. This approach works well in the case of a network with open boundaries that represents a cut-out section of a larger network (where the number of entrances is low, the flow into the network relatively high and 'return on investment' in terms of processing time is good). However, when the network has large numbers of entry points, the demands on processing resources imposed by this part of the simulation can become very high, regardless of the insertion probability and the actual number of generated vehicles. This is the case with spatial configurations that include many instances of TE "Road Stretch with Entrances and Exits", since this Topological Element (see Section 3.1.1) introduces an entrance point for every two car cell lengths of road, mimicking a city street with buildings alongside it that are all potential sources/sinks of vehicles.

In the case that the insertion probability for the entrances is very low, as is the case with the entrances modelled by **Road Stretch with Entrances and Exits**, and that the number of entrances is very high, a 'trick' can be applied to avoid the numerous random number generations required by the straightforward method. We take advantage of two properties of the described scenario, namely, 1) the *ergodicity* of Bernoulli processes and 2) the fact that the overall network arrival rate per time-step is low in comparison with the number of entrances. A Bernoulli process is said to be *ergodic*, as its *time averages* equal its *ensemble* averages over long enough periods of time and large enough ensembles (e.g. Gray, 2009). This, in turn, means that the average number of arrivals per time-step on N_{E} entrances is:

$$n_{\mathrm{Aavg}} = N_{\mathrm{E}} \times p_{\mathrm{Asng}} \tag{4.1}$$

where n_{Aavg} is the average number of arrivals per time-step on all entrances together and p_{Asng} is the probability of arrival in a time-step at a single entrance. In a very large network and with realistic arrival rates, n_{Aavg} is very low. For example, for a network of 1000 nodes with nodes about 150m apart and 10 entrances on each side per link (an area of approximately 5km in diameter) the number of entrances would be 40000. Let's say that in the worst case scenario a vehicle emerges from an entry point with a frequency of once every 20 minutes or less, so that the insertion probability is approximately 0.001. This would correspond to the arrival, on average, of 40 vehicles in every time step. Since this input rate is very small with respect to the size of the entire system, its effect, if any, is based on cumulative values (over time) in the simulated space and its variance can be disregarded.

In this, and similar cases, therefore insertion can be based on a *constant* number of vehicles per time-step (spatially distributed depending on origin and destination probabilities for the particular simulation scenario), provided the number of insertions remains relatively small in relation to the network area in which the insertions are taking place.

The Zone-Based Source class implements both the direct and *mass-generating* approaches to vehicle insertion, with a single source object handling all the entrances belonging to a single zone.

4.1.5 Zone-based routing implementation

This section deals with two aspects of zone-based routing and their implementation. One is the definition of zones and their precise mapping to spatial model structures during the initialisation phase of simulation. The other is the actual route choice process during simulation.

4.1.5.1 Zone definition

The large numbers of entrances and exits in a simulation are a problem also in the case of zone-based routing. Marking each individual entrance and exit in the network as belonging to a particular zone is not straightforward given their large numbers and the fact that the spatial model does not reference a coordinate system, being track-based. When coordinate system information is available, as it is in simulation models that derive spatial information from a map (e.g. Klefstad et al., 2005; Dai et al., 2010), zones are easily defined in terms of coordinates and can be directly assigned. In contrast, our meta-model, while it does not exclude the use of a map as starting point, is also employed for the construction of theoretical traffic networks, directly defined in terms of topology and link lengths.

We use the network itself to 'spread' zone information. The Topological Element class is given the task of seeding a 'circular wave' that associates entrance and exit points in the network with zones based on the distance from the 'seed point'. The spreading can be configured to have one or more seed points and each *seed point* can be the centre of one or more zones delimited by concentric circles that are defined by a distance (along roads) from the seed. A zone with an infinite radius is used as a default zone to 'catch' any entrances or exits that are not part of other specified zones. The simplest zone mapping is that with a single seed point and single default zone. Zones have to be mapped for 4 independent cases: for car entrances (*car in-zones*), bicycle entrances (*bicycle in-zones*), car exits (*car out-zones*) and bicycle exits (*bicycle out-zones*).

An example of zone mapping is shown in Figure 4.5. This figure shows a grid-shaped network of 16 nodes, with links that have 5 entrance and exit connections on each side of the road. Each one of these is modelled as a 'mini intersection' (see Figure 3.11) and is of length 2 car cells along the road. This means that the links of the network are 10 car cells in length. Seed points are placed at road cross-section lines, which are also route cross-section lines, and, for ease of implementation in the cell-based space of the model, run along cell boundaries. Zones are 'spread out' forward and backwards along

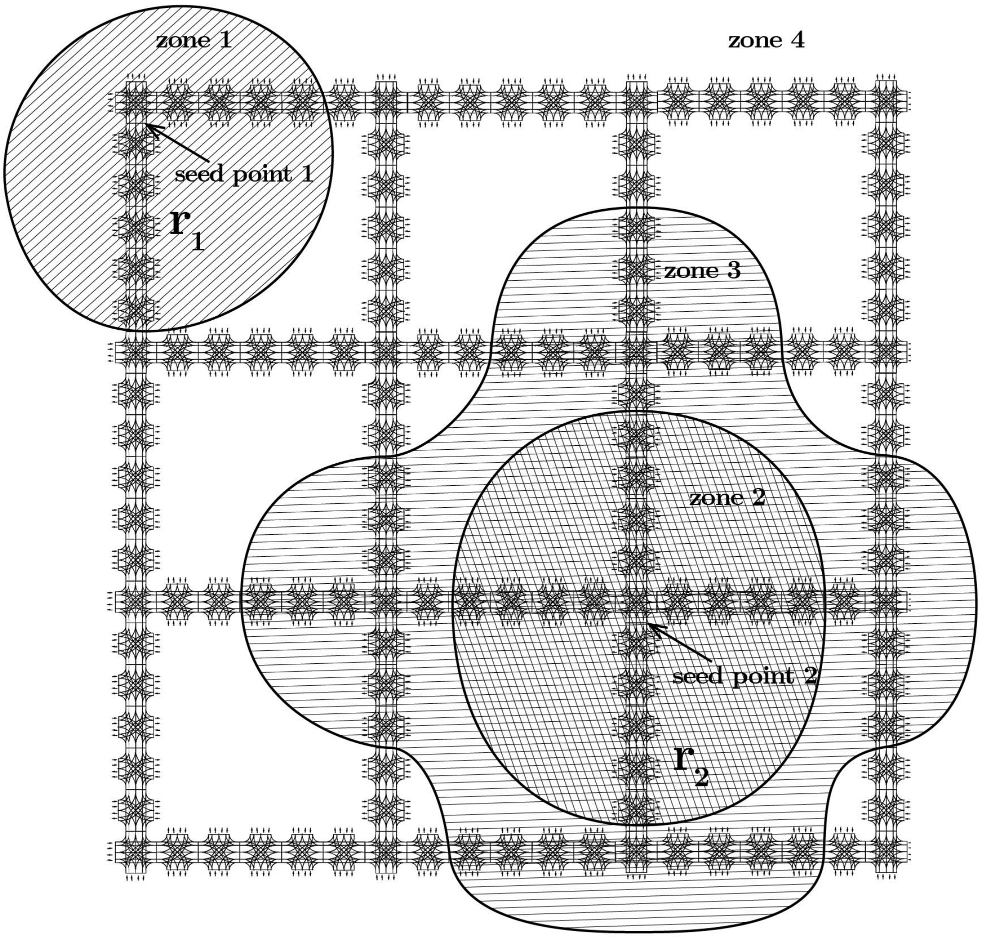


Figure 4.5: Zones in a network defined using the track meta-model

each route that is cross-sectioned by the seed point. Figure 4.5 shows three zones, each of them, apart from the default zone (4), 10 car cells in width. Zone 1 is the only one that spreads from seed point 1, which is at the cross-section of road r_1 after the 10th car cell (when counting in the direction shown as upward in the picture). Zones 2 and 3 both spread from seed point 2, which is at the cross-section of road r_2 after the 10th car cell also when counting upward in the picture.

4.1.5.2 Zone-based route assignment

Assignment of a zone-based route is implemented in the Zone-Based Route class and consists of a choice of destination and assignment of route:

1. A request for a route by the Zone-Based Source, on behalf of a newly created Agent, results, first, in a destination being picked by Zone-Based Route functionality. The choice of exit point is based on out-zone probabilities but is random within the chosen out-zone. Destination choice has not been made in-zone dependent, although such an extension to the current implementation would be fairly easy to realise, as the next level of complexity in the routing design.

2. Once the exact destination is chosen, a route must be determined. The route choice algorithm described already in Section 3.4 was implemented directly, with the shortest paths between all nodes determined in advance using the Dijkstra (1959) algorithm, and stored for use during simulation.

4.2 Parallel processing

Both parallel processing with *shared memory* and with *message passing* have been implemented. A *hybrid parallelisation model*, which combines the shared memory and message passing models, can also be realised since, in combination, the shared memory and message passing implementations constitute the implementation of the hybrid model. While this does not hold true in general, no additional adaptation of the two combined implementations is required in our case.

Terminology used here corresponds to the High Performance Computing (HPC) systems commonly deployed today (see Appendix C), which consist of a large number (e.g. thousands) of CPUs, referred to as *cores*. In these systems, the cores are grouped into *nodes* of, e.g. 8 or 12, which share memory and are located on the same integrated circuit board, not requiring a network to communicate between them. All the nodes in the system are connected by a network. This kind of system supports the three parallel processing paradigms: (1) shared-memory, (2) message passing and (3) hybrid of the two. The shared-memory parallelisation model can be mapped only onto a single node, utilising the multiple cores sharing memory within it. In this case, a process running on a node gets split into processing *threads*, each of which runs on a single core. The system is equipped with messaging software that facilitates parallelisation of programs across many nodes using message passing. Hybrid parallelisation uses both message passing and memory-sharing models in the same simulation. It is based on multiple processes, communicating by passing messages to each other, each split into several threads, communicating through shared memory. While the simplest hardware mapping of a hybrid simulation is to have the multiple processes run on distinct nodes and each of those split within a node into threads, any other mapping configuration where a single process is not split between nodes is possible. An example would be six processes on 3 nodes, 2 processes per node, each process split into 6 threads following the shared memory model: this configuration uses 36 cores on 3 nodes (12 cores per node).

4.2.1 OpenMP: parallel processing with shared memory

For the shared memory approach we used the OpenMP Application Program Interface (OpenMP Architecture Review Board, 2011), with which relatively simple adjustments to the implementation resulted in a working parallel version of the simulation program. In each of the three iterations through Agent objects, corresponding to *velocity update*, *conflict handling* and *vehicle motion* rule application, the load is split between processors sharing memory. OpenMP achieves this by assigning groups of iterations of a repetitive task to different processing *threads*, which provide a mechanism for process instructions to be split and executed in more than one simultaneous sequence.

Synchronous application of the movement rules, with sync points as shown in Figure 3.13,his ensured implicitly, since OpenMP creates a synchronisation point, i.e., a point at which the process waits for all threads to finish their respective work, after the split-processing of the iteration loop.

The requirements for data access locking are as follows:

- the velocity and position, which are both attributes of the Vehicle class, are inspected (by other Agents) and updated (by self in the form of an Agent) but never in the same movement rule, which means that ***locking is not needed***;
- vehicle position information is also held in the Cells of the Topological Element

(TE) class, i.e. a cell occupied by a Vehicle holds a reference to it, which may change during the application of the *vehicle motion* rule; however, since the rules are such that they never include one vehicle leaving and another entering the same cell in the same time step, this value change in a particular Cell is exclusive to a single Agent during one time-step and ***does not require locking***

- the *occupancy indicators* attribute of the Topological Element (TE) class, which contains summary information of Cell occupancy in the entire TE and is used for cell impingement checking during the application of *velocity update* and *conflict handling* rules, may be changed multilaterally during the *vehicle motion* rule, as more than one Vehicle may change its position in the TE (by entering it, leaving it or just changing the cell that it occupies within it); this indicates ***a need for data access locking***.

The access lock on the *occupancy indicators* attribute could be removed artificially, by delaying its update until the *vehicle motion* rule is applied to all vehicles. Performance advantage of this approach is not certain and has not been investigated. Rather, OpenMP *mutual exclusion locks* have been used to protect access to this attribute, since a currently up-to-date occupancy bit-mask for groups of overlapping cells is a very useful aid in detecting problems with the implementation, since it facilitates easy detection of 'crashes'.

While relatively easy to implement and maintain, the use of parallelisation with shared memory is limited in terms of performance improvements that it can offer, since, in practice, hardware architectures include up to 12 memory sharing processor cores (www.ichec.ie).

4.2.2 MPI: parallel processing with message passing

Parallel processing with message passing does not depend on the tight coupling between processors that is provided by memory sharing. This means that it is characterised by greater flexibility with respect to hardware configurations it can run on, including clusters of thousands of processing nodes. On the other hand, it is also more difficult to realise, requiring *domain decomposition*, *design of synchronisation schemes* and *implementation of dedicated communication functionality.*

The actual message passing is implemented using MPI (MPI Forum, 2009), a de facto standard for inter-process communication, which supports a number of programming languages, including C++. The following sections look at the individual aspects of a parallel program implementation, as they apply to the implementation described in Section 4.1.

4.2.2.1 Domain decomposition

Domain decomposition is done separately, in advance of simulation, following the algorithm described in Section 3.6.2. However, we applied the algorithm to a network in which intersection-type Topological Elements (Left Turn, Right Turn, Intersection of Two 1-Way Roads, Intersection of Two 2-Way Roads) are *nodes* and road-type Topological Elements (Road Stretch, Road Stretch with Entrances and Exits) are *edges.* The reason for this was to reduce the amount of necessary communication between processes. Balance of portion sizes was still achieved, owing to the roads all being the same length in the current network simulations. Once the nodes of this network are assigned to partitions using the algorithm, any road-type Topological Elements (TEs) lying between two nodes that have been assigned to different partitions are split into two parts of equal length. Each half is assigned to the partition containing the node to which it (the road part) is connected. In this way, the process boundaries are removed from intersections by a number of cells, which means that the transfer, between processes, of information relating to conflicts can be avoided entirely if the edges (i.e. roads) are long enough. Figure 4.6 shows a small network split in the described manner. Here the network consists of 4 intersection-type nodes, TE0, TE1, TE2 and TE3, which are connected by Road Stretch elements. In the diagram, the cells of Road Stretch elements TE4, TE5, TE6 and TE7 are marked with numbers of the form ⟨road stretch index*rangle*.⟨cell index⟩ (so, for example, the first cell of Road Stretch TE4 is 4.1). If the network partitioning algorithm from Section 3.6.2 is applied in the manner described earlier in this paragraph, the resulting partitions each contain

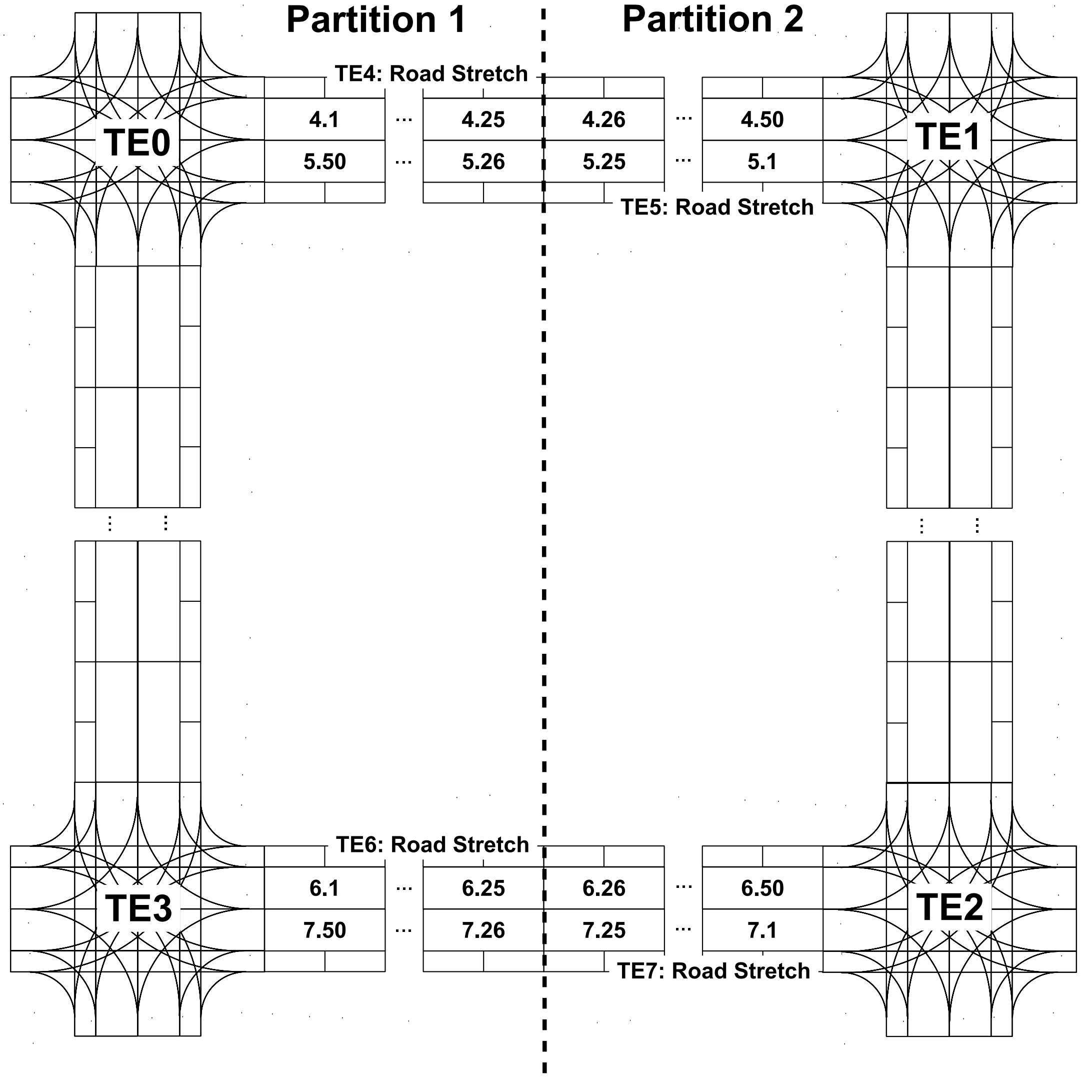


Figure 4.6: Domain decomposition for a small network defined using the *track meta-model*: Partition is performed on the set of nodes represented by intersections, i.e. TE0, TE1, TE2 and TE3. Once these were allocated to partitions, any cross-boundary road stretches were 'cut in two'. The horizontally shown road stretches have 50 cells each, labelled *t.c*, where t is the number of the track and c is the number of the cell.

two intersection-type TEs: Partition 1 contains TE0 and TE3 and Partition 2 has TE1 and TE2. While the partitioning in this case would be trivial if done manually since it has an intuitively 'obvious' solution, there are other possible solutions, e.g. grouping TE0 with TE2 and TE1 with TE3, which, correctly, do not emerge from application of the algorithm, since they would involve more inter-process connections. The example

```
7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 6 6 6 6 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 6 6 6 6 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 6 6 6 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 6 6 6 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 6 6 6 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 6 6 6 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 6 6 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 6 6 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 6 6 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 6 6 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 6 6 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
4 4 4 4 4 4 4 4 4 4 4 4 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
4 4 4 4 4 4 4 4 4 4 4 4 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
4 4 4 4 4 4 4 4 4 4 4 4 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
2 2 2 2 2 2 2 4 4 4 4 4 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Figure 4.7: Domain decomposition example showing the automated assignment of intersection-type nodes to 8 partitions: Each number indicates the partition id (from 0 to 7), to which a node in the grid is assigned, the grid having 60 x 60 two-way intersections. Each pair of different adjacent numbers indicates a road split between partitions.

in the picture also shows the splitting of the Road Stretch TEs. Each is 50 car cells long and split into two parts that are, in turn, each 25 car cells long. Each part is assigned to the same partition as the intersection-type TE to which it is connected.

Figure 4.7 shows the results of the algorithm applied to a 60×60 grid of two-lane two-way intersections. Similar results were obtained with the 111×111 node network, dividing into a number of processes that ranged from 2 to 384. In all cases the sections were 'in one piece' but not always with minimal perimeters. E.g. the assignment of network nodes to processing nodes in Figure 4.7 shows that the algorithm could be improved with respect to minimising the number of connections (for example, close to the middle of the matrix there is a two-number-wide 'tail' of 6s that introduces redundant connections). Currently nodes are selected one by one for a partition based on a *breadth first search.* One idea for improvement would be to prioritise picking nodes within 'tails', as these start forming in the unselected portion of the network, and apply the *breadth-first search* as second-in-line only after any 'tails' are eliminated. We have not pursued this improvement, because the performance impact is not expected to be high, since the number of connections does not change the number of exchanged messages, and because domain decomposition was not the main focus of the project.

4.2.2.2 Synchronisation

The implementation of synchronisation is based on the design described in Section 3.6.2, which will be called *preemptive messaging.* An initial attempt was made to implement communication on an 'as required' basis, however, this was found to be several times slower (a particular simulation took between 5 and 10 minutes on two processors on the same node with 'as required' communication; it took about 1 miute with a single process and about 30 seconds with preemptive messaging). To implement the exchange messages at the designated synchronisation points, identified in Table 3.3, we use the MPI *send-receive* function, which handles the sending of one message and the receipt of another as if they were taking place simultaneously. This removes the need for deadlock avoidance considerations. Because each process knows exactly how many connections it has and whether, for those connections, it plays the role of "upstream" process or "downstream" process (see Section 3.6.2), it knows how many messages to expect at each synchronisation point. The following steps represent the communication procedure at each time step:

- perform inspections for any data that may be needed in a connected process for the next movement rule (see Figure 3.14
- call *MPI send-receive* $min(n_{\mathrm{R}}, n_{\mathrm{S}})$ times
- if $n_{\mathrm{R}} > n_{\mathrm{S}}$ call *MPI receive* $n_{\mathrm{R}} - n_{\mathrm{S}}$ times
- if $n_{\mathrm{S}} > n_{\mathrm{R}}$ call *MPI send* $n_{\mathrm{S}} - n_{\mathrm{R}}$ times

where n_{R} is the number of expected messages, n_{S} is the number of messages to send. "MPI send" involves sending a message to a single process, including all the data that needs to be sent to that process at the given time step. Similarly "MPI receive" involves receiving a message from a single process, including all the data that needs to be received from that process at the given time step.

4.2.2.3 Communication

The implementation of communication effectively involves coding for marshalling and de-marshalling of data structures into messages. These are either messages relating to movement rules, messages sent at the very end of the simulation with collected data so that it can be processed and written into a file at a single node or, if zone-based routing is in effect, messages for the requesting and provision of routes.

Inspection across process boundaries for the two update rules, *velocity update* and *conflict handling* (if need for the latter was not eliminated during domain de-composition, as explained in Section 3.6.2), is based on a pattern that could be named 'push-retrieve'[1], while *vehicle motion* across process boundaries is in the form of a single forward message. These patterns, and how they are realised using the Call Interceptor class, specially implemented for parallelisation with MPI, is shown in Figure 4.8.

[1] Whereby a large message is sent with state information from one process to another and stored in the receiving process until such a time as the actual users in the receiving process need it and access it by interrogating the stored message, which transparently provides a response, as if it were arriving from the process that sent the message earlier.

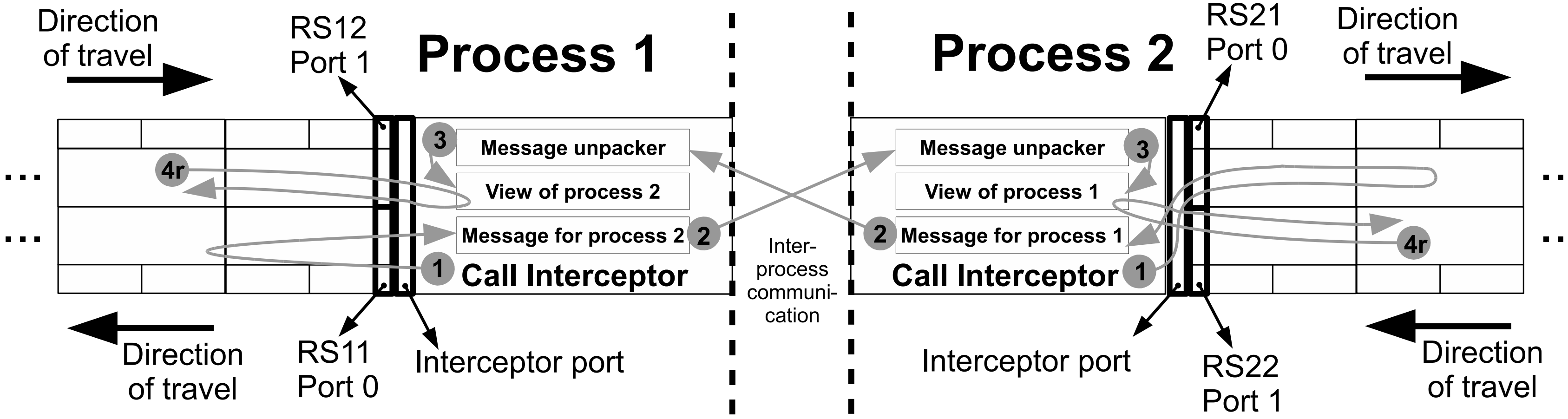


Figure 4.8: Communication patterns with MPI, a) Sequence of events for the *velocity update* message, shown for a two-way road split between processes: (1) the Call Interceptor in each process makes calls to forward inspection functions on any Ports that it is out-connected with - Port 0 of Road Stretch RS11 in Process 1 and Port 0 of Road Stretch RS21 in Process 2 - and marshals the results into the Message for process 2 and 1, respectively; (2) message exchange takes place - the Call Interceptor in Process 1 knows that it must send a message to Process 2 in this stage, because it is out-connected into a port (Port 0 of Road Stretch RS11) that is associated with Process 2; it also knows that it must receive a message from Process 2, because it has an in-connection from a port (Port 1 of Road Stretch RS12) that is associated with Process 2 (these associations will have been set-up at initialisation time); the Call Interceptor in Process 2 has analogous knowledge; (3) the Message unpacker in Process 1/2 reads the message and stores extracted result data into appropriate structures in the View of process 2/1; (4r) while the previous steps of the sequence take place between rule applications (after *vehicle motion* and before *velocity update*), the fourth step takes place during the application of rule *velocity update* (hence the r) - a normal forward-inspection call is made through Port 1 of the Road Stretch TE RS12/RS22 and a result transparently returned by the Call Interceptor, using the data stored in View of process 2/1.

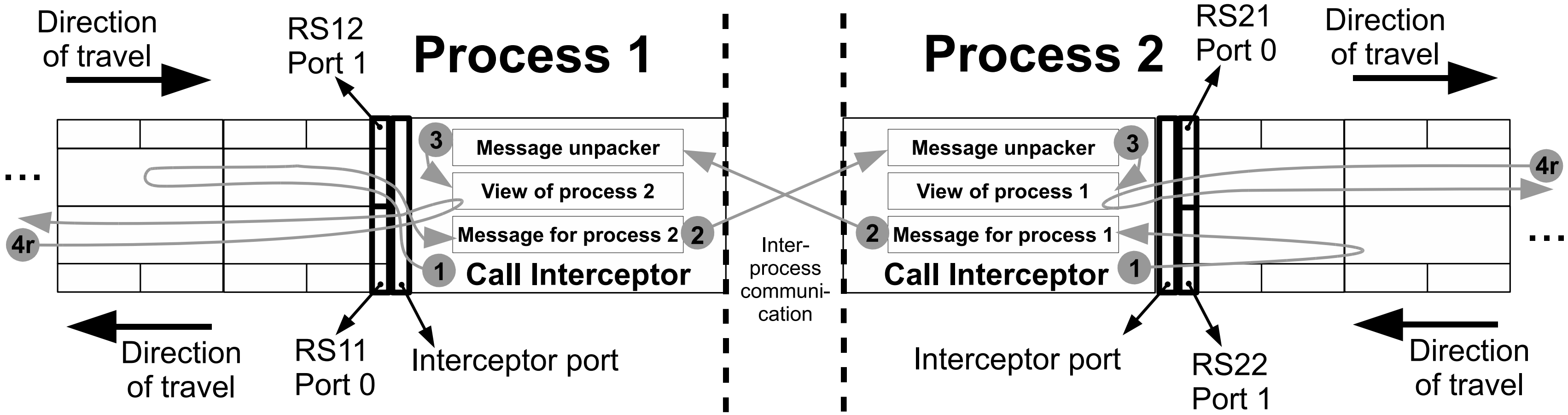


Figure 4.8 (continued): Communication patterns with MPI, b) Sequence of events for the *conflict handling* message, shown for a two-way road split between processes: (1) the Call Interceptor in each process makes calls to back-inspection functions on any Ports that it is in-connected with - Port 1 of Road Stretch RS12 in Process 1 and Port 1 of Road Stretch RS22 in Process 2 - and marshals the results into the Message for process 2 and 1, respectively; (2) message exchange takes place - the Call Interceptor in Process 1 knows that it must send a message to Process 2 in this stage, because it has an in-connection from a port (Port 1 of Road Stretch RS12) that is associated with Process 2; it also knows that it must receive a message from Process 2, because it is out-connected to a port (Port 0 of Road Stretch RS11) that is associated with Process 2 (these associations will have been set-up at initialisation time); the Call Interceptor in Process 2 has analogous knowledge; (3) the Message unpacker in Process 1/2 reads the message and stores extracted result data into appropriate structures in the View of process 2/1; (4r) while the previous steps of the sequence take place between rule applications (after *velocity update* and before *conflict handling*), the fourth step takes place during the application of rule *conflict handling* (hence the r) - a normal back-inspection call is made out through Port 0 of the Road Stretch TE and a result transparently returned by the Call Interceptor, using the data stored in View of process 1/2; the originating/receiving point for this call is shown to be somewhere outside of the picture, as it must be at a conflict and one is not shown in the picture.

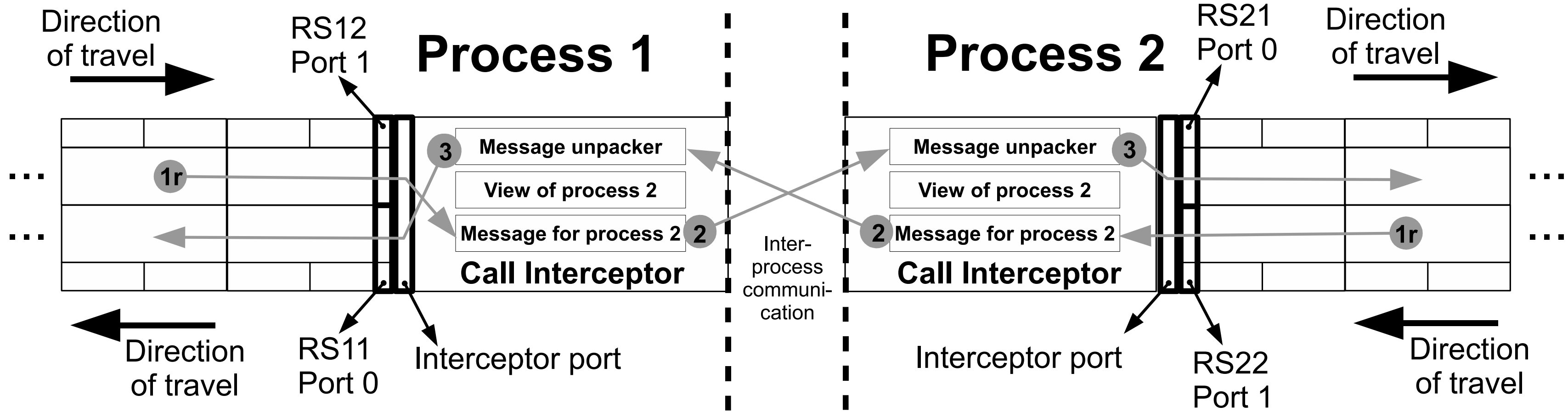


Figure 4.8 (continued): Communication patterns with MPI, c) Sequence of events for the *vehicle motion* message, shown for a two-way road split between processes: (1r) the first step takes place during the application of rule *vehicle motion* (hence the r); a call for vehicle motion out through Port 1 of TE Road Stretch RS12 or through Port 1 of TE Road Stretch RS22 results in the Call Interceptor marshalling the Agent into a Message for Process 2/1, respectively; (2) once the *vehicle motion* rule has been applied to all Agents in a process, message exchange takes place - the Call Interceptor in Process 1 knows that it must send a message to Process 2 in this stage, because it has an in-connection from a port (Port 1 of Road Stretch RS12) that is associated with Process 2; it also knows that it must receive a message from Process 2, because it is out-connected to a port (Port 0 of Road Stretch RS11) that is associated with Process 2 (these associations will have been set-up at initialisation time); the Call Interceptor in Process 2 has analogous knowledge; (3) the Message unpacker in Process 1/2 reads the message, un-marshals the Agent, creating a clone of the one that was marshalled in the other process (2/1) and places it onto the appropriate track, which will have been specified as part of the message.

In the case that zone-based routing is in effect, and because the route-generator functionality must have a view of the entire network, functional decomposition must take place, with one process (the route-generator process) computing and providing routes to Agents in all the other processes (traffic processes). All inter-process communication takes place within the Zone-Based Route class, which has its own call interception functionality. Route requests are made in two stages. Initial requests are sent in the first stage, using one message per traffic process and the pattern shown in Figure 4.8c. These messages are sent from a traffic process to the route-generator process. In the second stage messages containing the routes travel in the opposite direction, i.e., from the route-generator process to the traffic processes. This stage uses the push-retrieve pattern, like that presented in Figure 4.8a. The request/retrieval of routes is done in two stages for two reasons: 1) because this allows the collection of all requests from a traffic process and the sending of a single message, reducing overheads; 2) for better time-slotting of loads: if a request for all the routes needed by a traffic process is sent a time-step 'early', the route-generator process can work on producing the routes while, the other processes are applying the movement rules, avoiding the delay that would ensue if the traffic processes were to wait on the route-generator process to receive routes synchronously.

4.2.3 Running times and speed-up with parallel implementations

Experiments were conducted to investigate the effectiveness of the parallel-processing adaptations made to the simulation model implementation. These consisted of running simulations with the same network configuration for each of the different parallel processing configurations enabled by the adaptations. The speed-up has been plotted for a number of different scenarios, all of which are either (i) a Road Stretch with closed boundary conditions or (ii) a grid of Intersections of Two 2-Way Road elements, connected between them by Road Stretch elements of length 20 (also with closed boundary conditions). The grids have been chosen so that they have approximately the same overall size, in terms of car cells, as the elements used in the Road Stretch experiments. Grid sizes 4 × 3, 11 × 11, 40 × 31 and 111 × 111 approximately correspond to Road

Stretch sizes 10^3, 10^4, 10^5 and 10^6, respectively. All processing time and speed-up data in this section is for the ICHEC[2] Stokes system, described in Appendix C.

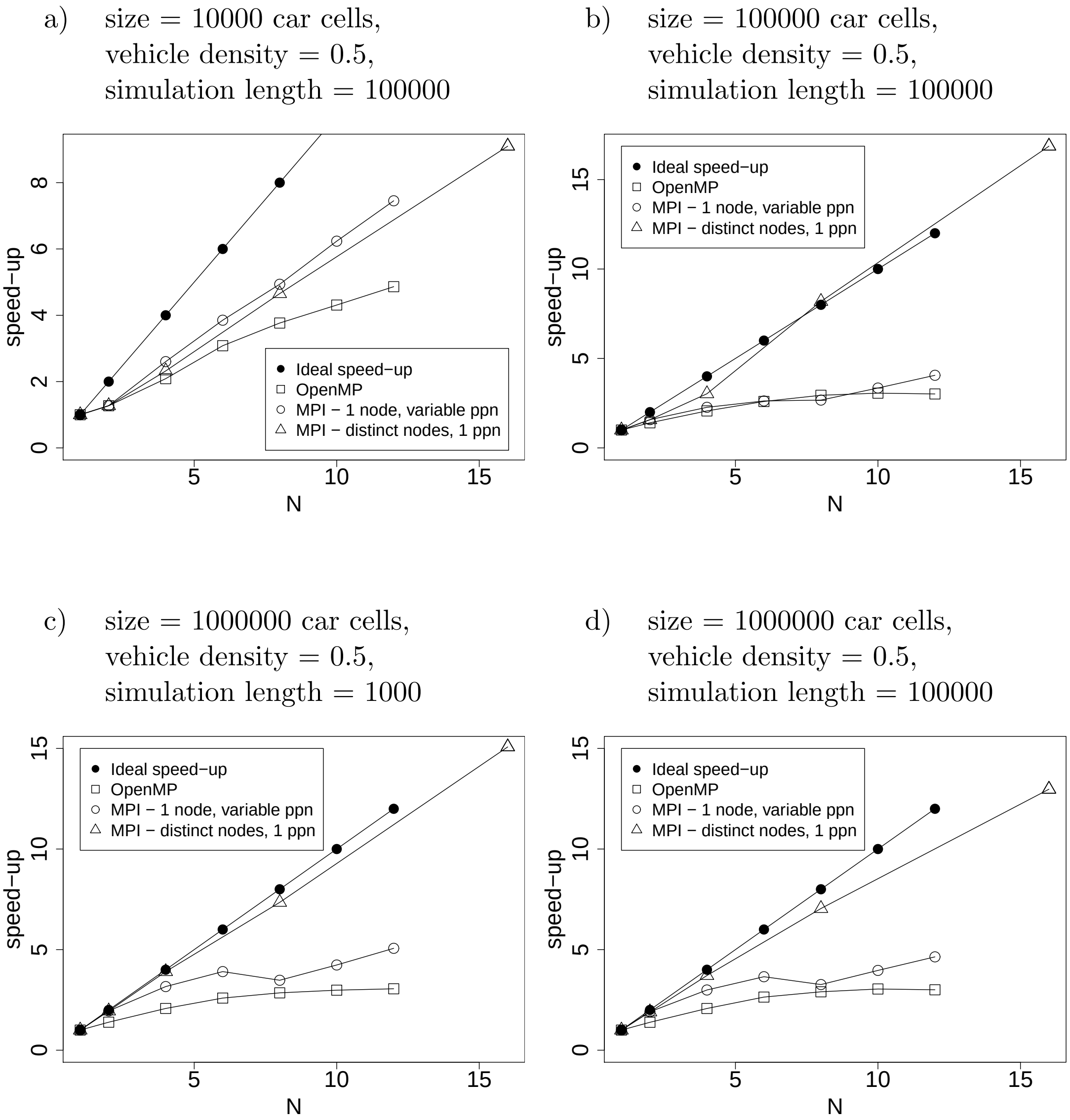
Size in car cells	**Vehicle density**	**Road Stretch**			**Grid of intersections**		
		10	**1000**	**100000**	**10**	**1000**	**100000**
10^3	0.2	270	141	13810	37	289	27734
	0.5	5	270	26533	17	429	41613
	0.8	8	294	28651	18	521	50865
10^4	0.2	23	1484	144485	137	3459	332741
	0.5	41	2783	273733	175	5448	538646
	0.8	50	3127	317118	196	8038	794461
10^5	0.2	369	27890	2756598	1742	85634	8530771
	0.5	600	46184	4569228	2115	116249	11828267
	0.8	759	54892	5442484	2303	129995	13702232
10^6	0.2	3943	279215	27735581	17799	903808	90953187
	0.5	6444	461121	45891610	21713	1216141	125032339
	0.8	7899	549499	54886261	23499	1388119	145222118

Table 4.1: Running times for various simulation configurations: The six columns of the table correspond to road stretch with periodic boundary conditions, simulation lengths of 10, 1000 and 100000 time-steps, and for a grid of intersections of type "Intersection of Two 2-Way Roads", also for three cases of simulation length: 10, 1000 and 100000 time-steps. The rows correspond to different combinations of aggregate road lengths in car cells and vehicle densities. The running times are in milliseconds, for execution on Intel Xeon Processor E5650 (details in Appendix C).

The running times, for a single node, are shown in Table 4.1. As expected, larger areas (in car cells) and greater vehicle densities command more processor time. The grid simulation processing times are greater than those for corresponding road stretch sizes, since the former include turn and conflict processing.

Speed-up graphs are shown in Figures 4.9, 4.10, 4.11 and 4.12. Figures 4.9 and 4.10 show speed-ups by core with OpenMP, MPI on a single node and MPI across multiple nodes (for road stretch and grid simulations, respectively). For some of the effects visible in these graphs to be understood, the hardware configuration of a processing node needs to be explained in more detail. The processing cores on a node of a HPC system share memory, but also share the *memory cache*, which is a small amount of memory accessible at high speed to the cores. In the system used for the simulations (ICHEC, details in Appendix C), the cores in each node are grouped further into two

[2] Irish Centre for High-End Computing.



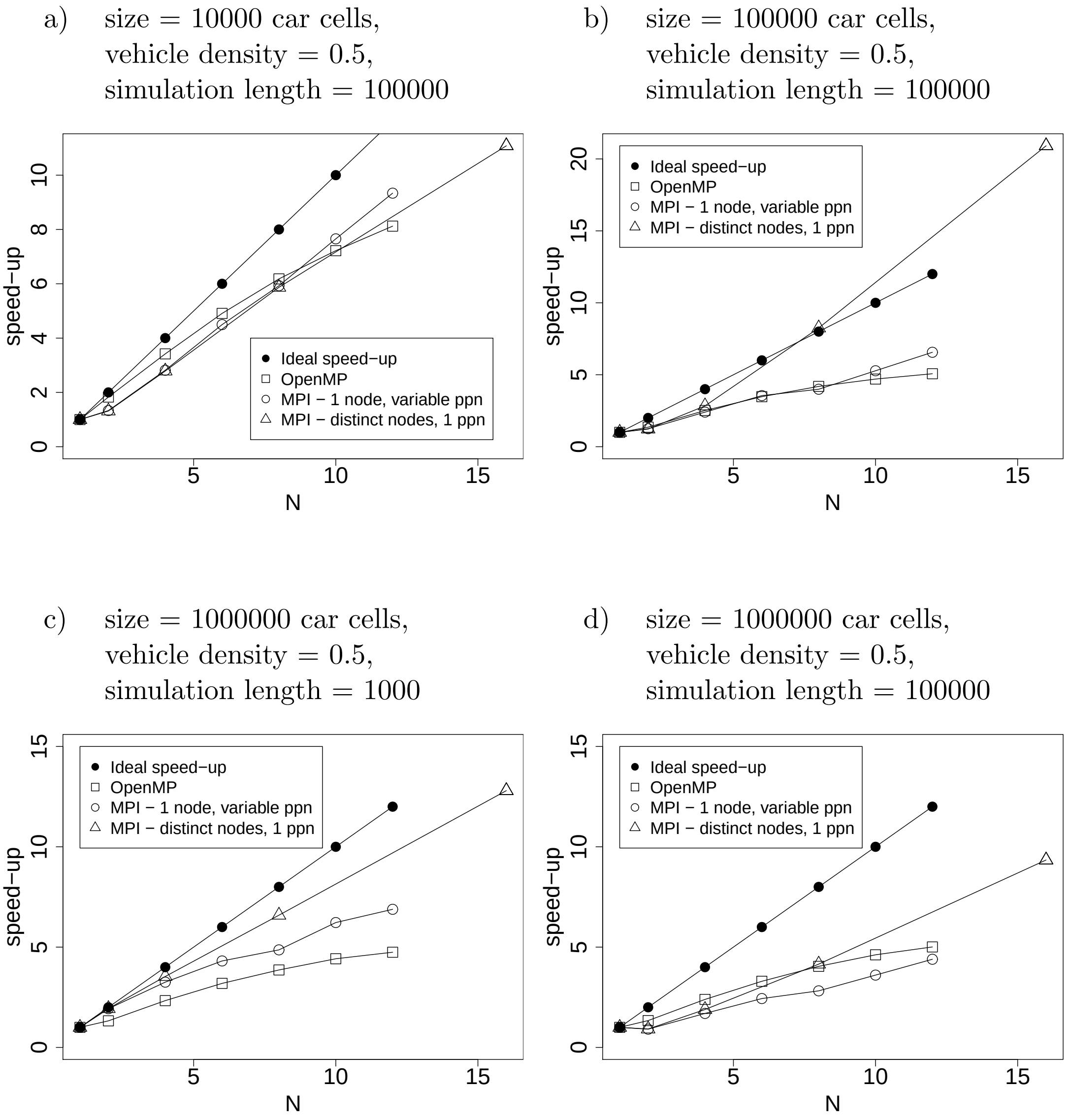
a) size = 10000 car cells, vehicle density = 0.5, simulation length = 100000

b) size = 100000 car cells, vehicle density = 0.5, simulation length = 100000

c) size = 1000000 car cells, vehicle density = 0.5, simulation length = 1000

d) size = 1000000 car cells, vehicle density = 0.5, simulation length = 100000

Figure 4.9: Speed-up in simulation of *road stretch* with periodic boundary conditions - **OpenMP** and **MPI**: Each sub-figure shows (i) ideal speed-up, (ii) speed-up with OpenMP (N representing the number of memory-sharing processor cores); (iii) speed-up with MPI on memory-sharing cores, i.e., on the same hardware configuration as for the OpenMP case, except that memory sharing is not utilised (MPI - 1 node, variable ppn) and (iv) speed-up with MPI across networked nodes, with 1 process per node (N representing the number of networked nodes, MPI - distinct nodes, 1 ppn). Different sub-figures show speed-ups for different sets of parameters.

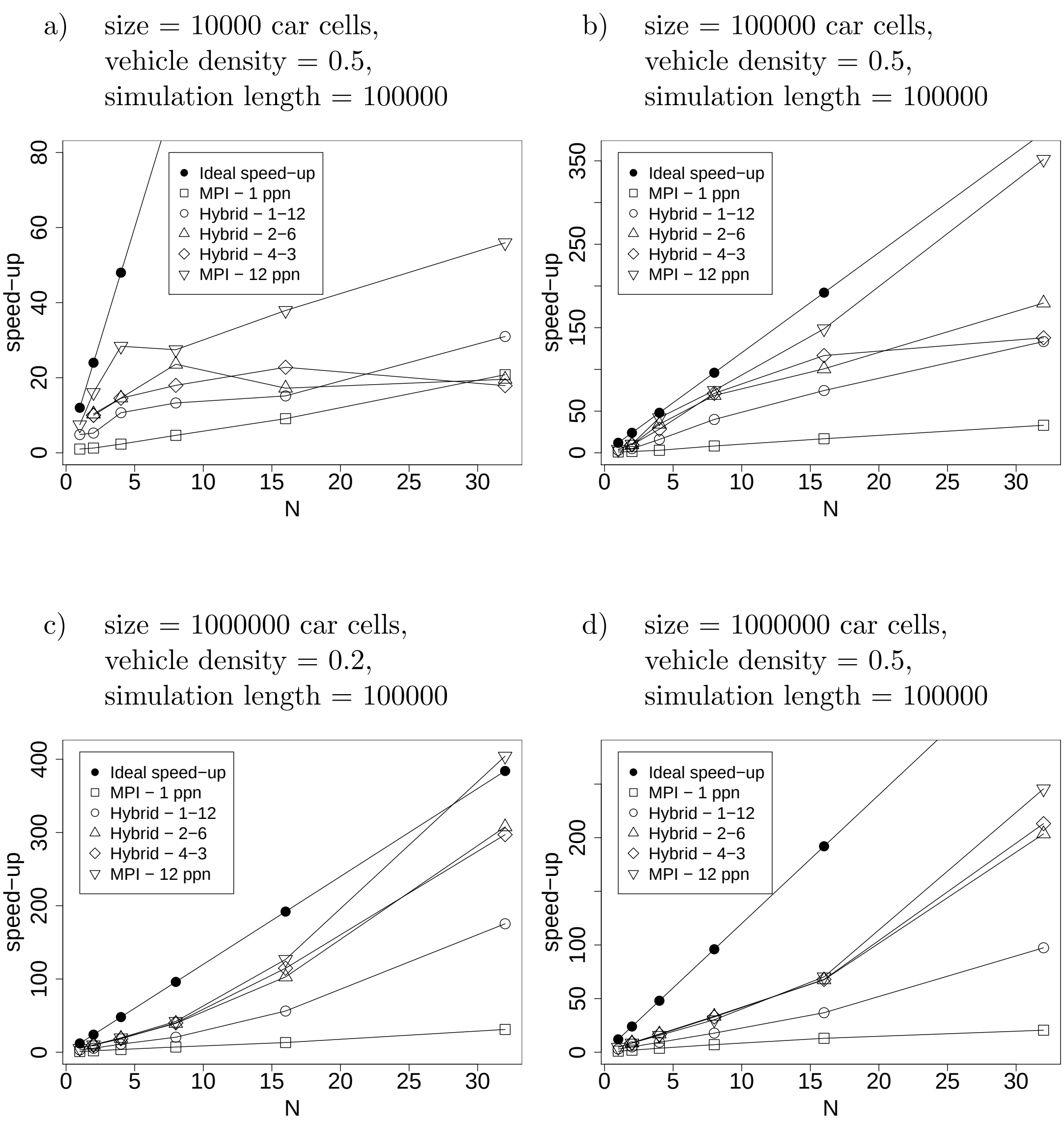

a) size = 10000 car cells, vehicle density = 0.5, simulation length = 100000

b) size = 100000 car cells, vehicle density = 0.5, simulation length = 100000

c) size = 1000000 car cells, vehicle density = 0.5, simulation length = 1000

d) size = 1000000 car cells, vehicle density = 0.5, simulation length = 100000

Figure 4.10: Speed-up in simulation of a *grid of two-way two-lane road intersections* with periodic boundary conditions - **OpenMP** and **MPI**: Each sub-figure shows (i) speed-up with OpenMP, N representing the number of memory-sharing processor cores; (ii) speed-up with MPI on memory-sharing cores (i.e., on the same hardware configuration as for the OpenMP case, except that memory sharing is not utilised, MPI - 1 node, variable ppn) and (iii) speed-up with MPI across networked nodes, with 1 process per node (N representing the number of networked nodes, MPI - distinct nodes, 1 ppn). Different sub-figures show speed-up for different sets of parameters.



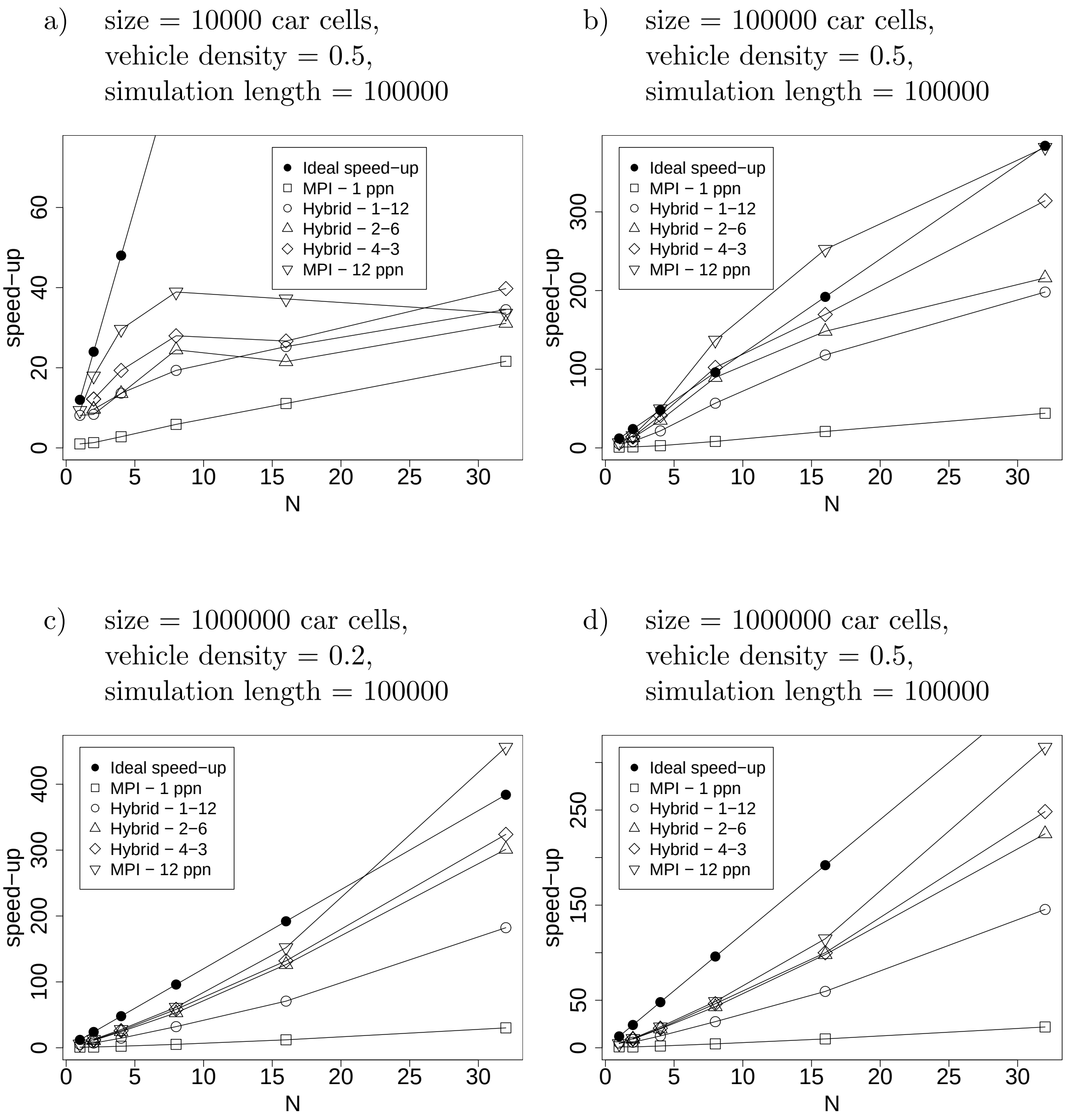
a) size = 10000 car cells, vehicle density = 0.5, simulation length = 100000

b) size = 100000 car cells, vehicle density = 0.5, simulation length = 100000

c) size = 1000000 car cells, vehicle density = 0.2, simulation length = 100000

d) size = 1000000 car cells, vehicle density = 0.5, simulation length = 100000

Figure 4.11: Speed-up in simulation of *road stretch* with periodic boundary conditions - **hybrid parallel processing schemes**: Each sub-figure shows (i) ideal speed-up (ii) speed-up with MPI across networked nodes, with 1 process per node (MPI - 1 ppn) - repeated from Figure 4.9, for comparison with hybrid schemes; (iii) speed-up for hybrid 1-12 scheme, where there is 1 MPI process per node and 12 OpenMP threads per MPI process; (iv) speed-up for the hybrid 2-6 scheme, where there are 2 MPI processes per node and 6 OpenMP threads per MPI process; (v) speed-up for the hybrid 4-3 scheme, where there are 4 MPI processes per node and 3 OpenMP threads per MPI process and (vi) speed up for an all-MPI simulation that utilises all cores in each node, equivalent to hybrid 12-1, where there are 12 MPI processes per node, each consisting of a single thread (MPI - 12 ppn). In all cases, N is the number of nodes and the number of utilised cores within each node is 12. Speed-up is with respect to to a purely serial run, which means that the ideal speed-up for 1 node is 12.



a) size = 10000 car cells, vehicle density = 0.5, simulation length = 100000

b) size = 100000 car cells, vehicle density = 0.5, simulation length = 100000

c) size = 1000000 car cells, vehicle density = 0.2, simulation length = 100000

d) size = 1000000 car cells, vehicle density = 0.5, simulation length = 100000

Figure 4.12: Speed-up in simulation of a *grid of two-way two-lane road intersections* with periodic boundary conditions - **hybrid parallel processing schemes**: Each subfigure shows (i) ideal speed-up (ii) speed-up with MPI across networked nodes, with 1 process per node (MPI - 1 ppn) - repeated from Figure 4.9, for comparison with hybrid schemes; (iii) speed-up for hybrid 1-12 scheme, where there is 1 MPI process per node and 12 OpenMP threads per MPI process; (iv) speed-up for the hybrid 2-6 scheme, where there are 2 MPI processes per node and 6 OpenMP threads per MPI process; (v) speed-up for the hybrid 4-3 scheme, where there are 4 MPI processes per node and 3 OpenMP threads per MPI process and (vi) speed up for an all-MPI simulation that utilises all cores in each node, equivalent to hybrid 12-1, where there are 12 MPI processes per node, each consisting of a single thread (MPI - 12 ppn). In all cases, N is the number of nodes and the number of utilised cores within each node is 12. Speed-up is with respect to to a purely serial run, which means that the ideal speed-up for 1 node is 12.

groups (12 cores in a node, with 6 cores in each of the two groups), each tightly associated with half the node's memory and half the node's memory cache, realising a Non-Uniform Memory Access (NUMA) architecture. This means that access to half of the available memory to a core in the node (while transparent to applications) is in fact slower, since that memory is associated with the 'other' group of cores. A further issue is the *memory cache effect*. This is the impact on performance that stems from program data size relative to size of the available memory cache, which can lead to non-linear relations between program data size and running times, both in serial and parallel programs. For example, if a program data are roughly bigger than a single cache unit but smaller than two cache units, running such a program with OpenMP on two processing cores displays dramatic speed-up in comparison with the single-core case.

Figures 4.9 and 4.10 allow a 'like-for-like' comparison of OpenMP and MPI. The MPI times are shown both for the same configuration as that used for OpenMP (several cores on a single node) and for a single process-per-node configuration. The latter is not a configuration that would actually be used for content-focused simulations (as opposed to the experimental ones discussed here, which have investigation of speed-up as their focus), since it would mean under-utilisation of cores on a node. However, direct comparison of MPI performance with that of OpenMP is useful. In the case of a road stretch, where MPI communication is 'light' since it occurs only for two connections per process, MPI performance is generally much better than for OpenMP (all diagrams in Figures 4.9 and 4.10 except 4.10d). In the case of small simulation size in terms of modelled area (measured in car cells counted in the system), the speed-up with OpenMP is fairly high (Figures 4.9a and 4.10a) compared to the other cases as a *memory cache effect* is in action. The general speed-up in the case of a road stretch simulation with OpenMP is worse than with a grid and is only about ×2, even with 12 cores, in contrast to about ×5 for the grid simulations. While both values are far from the ideal value of ×12 due to the non-parallel parts of the implementation and OpenMP overheads such as locking and context-switching (assignment of threads to cores), the overheads have a greater effect in the road stretch case, because of the relatively lower processing requirements of that case (see Table 4.1). The 'dip' in speed-up that can be seen at 8 cores in all cases of simulation with MPI on a single node is due to the number of cores exceeding half those in a node (6 in the case of these simulations). When this happens, the cores involved belong to two of the core sub-groupings in the node, making communication slower.

Speed-up with MPI with 1 process per node, also shown in Figures 4.9 and 4.10, is close to ideal for most road stretch simulations. In this case the portions of the spatial model have very little inter-communication, imposing very little overhead, which results in almost 'purely parallel' simulations. In fact, in the cases shown in Figure 4.9b the speed-up is *super-linear* (above ideal), owing to a cache effect, but which manifests itself when the portions of data in different nodes fall beneath the memory cache size. The grid simulations with MPI (Figure 4.10) show lower speed-ups, due to the messaging overheads, which are much higher in the case of a 'cut-up' grid, where many roads are split and each split requires inter-process data exchange. Caused purely by simulation data size, super-linearity is manifested here also (in the grid case, in Figure 4.10b). In Figure 4.10d, the speed-up is lower than 1 for two nodes with MPI. The same problem size, with regard to spatial model and vehicle density does not, however, produce this recognisable 'dip' for a 1000-step simulation (Figure 4.12c). While this quirk in the graph is not easily explained, it seems due to a load that increases with time (i.e. occurs for a very long simulation but not for the short one) and is less pronounced for smaller sizes. If it were simply the message passing overhead (which does decrease as the number of nodes increases, since the number of connections for each process becomes smaller, and would not, therefore, prevent observed high speed-ups at higher node counts), it should then also be present in the 1000-step case in Figure 4.12c. One possible explanation is that the 'dip' is related to inadequate message buffering, or other message exchange-related factors contributing to unevenness of delay at each time step, which would be worse for larger messages and compounded by greater simulation length. Further experiments will be needed to investigate this.

Speed-up with hybrid parallelisation for road stretch simulations and grid simulations are shown in Figures 4.11 and 4.12, respectively. Each of the sub-figures in these includes the speed-up curve for 1 process-per-node MPI simulations, for comparison. This curve should, ideally, consist of values that are 12 times lower than those in the other curves, since the hybrid runs include all the cores of every node. Figures 4.11a and 4.12a display a situation where the number of cores becomes too big for the problem size (consisting of spatial elements and vehicles), with the combined overheads of OpenMP and MPI causing irregular speed-up curves. Notably, the "MPI - 12 ppn" curve in Figure 4.12a takes a downward trend after 8 nodes (which means $8 \times 12 = 96$ processes). The next point in the curve is for 16 nodes (192 processes), which is larger than the number of intersections in the network (121), causing hindrance rather than speed-up. Owing to the partition approach (i.e. at least one intersection in each process) the possible speed-up for a small network is limited, but even for the road stretch of the equivalent size the maximal speed up is only about 50 (as opposed to 40 for grid), because of the relatively high overheads/problem size ratio. With heavier loads (Figures 4.11b, c and d and 4.12b, c and d), speed-up is reasonably good for all schemes. Nevertheless, the all-MPI scheme (where all cores are utilised and all processes communicate via messages), performs best. This might be expected, since at lower loads MPI performs better between cores than OpenMP (Figures 4.9a, b and c and 4.10a b and c). In fact, super-linear speed-up can be seen in Figures 4.11c and 4.12b and c, starting at different node counts for different processing loads. Figures 4.12c and d refer to very similar scenarios, which differ only in the vehicle density: 0.2 versus 0.5. The lower load brings about super-linearity somewhere between 16 and 32 nodes for MPI with 12 processes per node, while for the higher load of 0.5 the trend of the speed-up line would suggest that super-linearity will occur but at a considerably higher number of nodes.

The conclusion inevitably drawn from the above is that in the case of parallel implementations of our model, MPI performs better in all cases and that the highest performing configuration is that of several nodes with all cores utilised, running separate processes that communicate via MPI. Further, node utilisation can be optimised with knowledge of load-dependent speed-up, in particular that for lower processing loads, where large numbers of nodes can be counter-productive.

4.3 Summary

This chapter has looked at two different aspects of the implementation: the functional and the performance aspect.

The functional aspect is described using a UML class model, which distinguishes between the *simulation framework* and *topological elements*. The framework, in part, mirrors the *track meta-model* constructs on a one-to-one basis and implements the movement rules as part of the *Agent* class. It also includes classes that represent abstract network structures, including *conflicts* and *divergences*, and *routing activities*. As well as an overall view of the implementation, contained in the UML class diagram, this chapter focuses on some specific implementation features, namely, *overlap tables* to check cell occupancy; *mass sources*, which handle the creation of vehicles in large networks with many *internal entrances*; and *zone-based routing*.

Turning to the performance aspect of the implementation, the chapter then discusses the extensions made for parallel processing, using OpenMP and MPI. Speed-up diagrams are shown for a number of parallel configurations and analysed as to their performance, which was found to be best for pure messaging-based parallel configurations (i.e. those using MPI), while hybrid configurations (with processes sharing memory only within a node) achieve similarly high speed-ups only if applied to large network simulations.

Chapter 5

Simulation and Results

This chapter gives an account of simulations that were performed using the implemented system and analyses the output produced. The system is based on the *track meta-model* described in Chapter 3. Its architecture and important features are described in Chapter 4. This chapter begins with an overview of the traffic measures used and their formats. Simulation reports are organised into sections by spatial configuration.

5.1 Measures of traffic

This section is dedicated to describing the types of diagrams used to present simulation results. The list is not exhaustive, but includes data presentation types of importance in traffic science in general or specific to the research underlying this thesis.

5.1.1 Fundamental diagram (FD)

The fundamental diagram (FD) is one of the primary means of characterising traffic flow dynamics and is parameterised by properties of infrastructure and driving conditions. It is defined for a point on a road, as the diagram of traffic flow values (y-axis) against the corresponding average vehicle densities at that point on the road (x-axis). The traffic flow is defined as the number of vehicles that passes the point in the road per lane per time step. The fundamental diagram is illustrated in Figure 2.1. It consists, roughly, of a) a free-flow region, from point (0,0) to a point close to the maximal flow value, b)

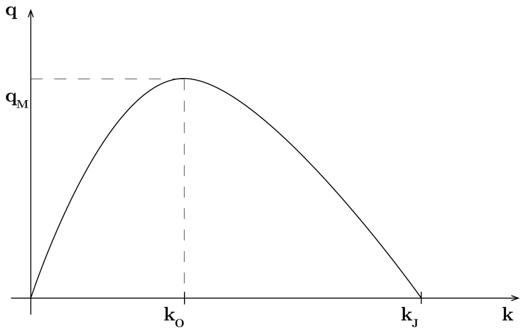


Figure 5.1: Illustration of fundamental diagram shape, where q_{M} is the maximal flow, k_0 is the traffic density at which the maximal flow is achieved and k_{J} is the jam traffic density.

the region of capacity flow, around the maximal flow point and c) the congested region, in which the flow curve has a downward slope.

Measured traffic data often do not form a well-defined fundamental diagram (Kerner and Rehborn, 1996b). The question is thus whether known irregular features of such data, for example wide scatter or hysteresis, are a consequence of external factors that essentially modify the FD dynamically (Daganzo et al., 1999) or an inherent property of traffic (Kerner and Rehborn, 1997). While a point of contention in literature, this has also been deemed to be simply a matter of interpretation (Treiber et al., 2010). In any case, a diagram of data in the FD dimension space, whether well-defined or not, is a good place to start in the assessment of a traffic system.

5.1.2 Input-flow diagram

The real-life equivalent of input is expressed in the same units as flow, i.e. in vehicles per unit of time per lane. In the cellular automata context this value is dimensionless and represents the number of vehicles that arrive at an entrance into a traffic system per time-step (where an entrance is a single lane). More precisely, for our model and implementation, it represents the number of vehicles that arrive at a *track in-connector* exposed by the simulated system. An input value translates into an insertion probability in the implementation. In the case of a network, the input-flow diagram shows the average out-flow for the entire network as a function of the average input.

5.1.3 Macroscopic fundamental diagram (MFD)

The re-use of the dimension space of the fundamental diagram in the context of network traffic results in the macroscopic fundamental diagram. The MFD shows the relationship between the average occupancy in a network and the average outflow from the network and has been shown to be well defined under given conditions (Geroliminis and Sun, 2011), as explained in Section 2.2.

5.1.3.1 MFD for simulations with track meta-model

In order to determine the MFD of a simulated system the average flow and occupancy of the network must be measurable. Measuring flow at each network link can be avoided since the average flow in an arbitrary network can be obtained by normalisation of the overall outflow from the network. In the case of in- and outflow *only through network boundaries*, the following normalisation factor can be used:

$$f_{\mathrm{NB}} = \frac{1}{(N_{\mathrm{GLeq}} \times N_{\mathrm{TPL}})} \tag{5.1}$$

where f_{NB} is the normalisation factor for the case with only boundary exit flows, N_{GLeq} is the number of grid lines in a regular grid with the same number of geometric links as the given network and N_{TPL} is the average number of tracks per geometric link in the network. The two latter values can be calculated as:

$$N_{\mathrm{GLeq}} = 2\sqrt{N_{\mathrm{L}}/2} \tag{5.2}$$

where N_{L} is the number of links in the network, and:

$$N_{\mathrm{TPL}} = N_{\mathrm{T}}/N_{\mathrm{L}} \tag{5.3}$$

where N_{T} is the number of tracks in the network and N_{L} is the number of geometric links (representing roads with both directions, if present) in the network. The normalisation factor, after substituting the two last into Equation 5.1, the normalisation factor becomes:

$$f_{\mathrm{NB}} = \frac{1}{N_{\mathrm{T}}}\sqrt{\frac{N_{\mathrm{L}}}{2}} \tag{5.4}$$

For the case where *outflow occurs throughout the network*, including exits along links, the journey length needs to be taken into account, because vehicles contribute to average link flow in proportion with the length of their journey, so the normalisation factor is:

$$f_{\mathrm{N}} = f_{\mathrm{NB}} \times \frac{l_{\mathrm{Javg}}}{L_{\mathrm{avg}}} \times \sqrt{\frac{N_{\mathrm{L}}}{2}} = \frac{1}{N_{\mathrm{T}}} \times \frac{l_{\mathrm{Javg}}}{L_{\mathrm{avg}}} \tag{5.5}$$

where f_{N} denotes the normalisation factor for the general case, f_{NB} is the normalisation factor derived earlier for a network with outflow only at boundaries, l_{Javg} is the average journey lenght in the network, L_{avg} is the average link length in the network and N_{T} is the number of tracks in the network.

So, the normalised flow, for the two cases (with outflows only at boundaries and with outflows anywhere in the network):

$$q_{\mathrm{NB}} = q_{\mathrm{out}} \times f_{\mathrm{NB}} \quad , \qquad q_{\mathrm{N}} = q_{\mathrm{out}} \times f_{\mathrm{N}} \tag{5.6}$$

where q_{NB} denotes the normalised flow for the case of only boundary traffic, f_{N} is the normalisation factor, and q_{out} is the overall outflow of vehicles from the network. The idea behind this formula is that it idealises the geometry of the network into a grid and ignores the impact of intersections. The number of tracks in that scenario is used to normalise the flow to a single track. The impact of the simulated network's topology and intersections will be quantified against this ideal.

The global density is:

$$\rho = N_{\mathrm{V}}/N_{\mathrm{C}} \tag{5.7}$$

where ρ denotes the network-wide vehicle density, N_{V} is the number of vehicles in the network and N_{C} is the number of cells in the network.

The MFD is acquired by plotting points (ρ, q_{NB}) or (ρ, q_{N}) in the density-flow space for a large number of simulations with different vehicle insertion rates.

5.1.4 Realisation diagrams

One of the challenges posed by the availability of a simulation model and large amounts of generated output is to find a way of representing these data that is clear and provides an informative summary view of the properties of the simulated system. While for a larger network aggregation of data is the only possibility, we have devised, for intersections with a limited number of identifiable individual flows, a single diagram that gives some idea of the relationships between the flows and the performance of the intersection. The process of producing a realisation diagram is described below. It is best followed with an example, such as Figure 5.16, at hand.

The measure called *flow realisation* can be defined as:

$$R = \frac{\sum\limits_{t \in S_T} q_t}{\sum\limits_{t \in S_T} p_{\mathrm{I}t}} \tag{5.8}$$

where S_T is the set of tracks for which the realisation, R, is being calculated, t is a track identifier, q_t is the flow on track t and $p_{\mathrm{I}t}$ is the insertion probability for track t. While this value does not have a physical meaning, it can aid the understanding of how successful a scenario is in managing the arriving vehicles. Higher values of R are more favourable.

The realisation diagram is produced in a number of steps:

- The results used are the average flow values on each track of an intersection, obtained from a large number of simulations with varied values of $p_{\mathrm{I}t}$ for each track.

- Each simulation instance, based on its calculated R, is assigned to a 'slot', $s_V \in N_0$, such that $0.1 s_V \leq R < 0.1(s_V + 1)$.

- The insertion probabilities for each track are averaged over the simulation instances in each slot.

- The diagram is drawn so that a column over each slot shows the corresponding average insertion probabilities calculated in the previous step. The column is divided into sections, one for each involved track, and its depth of shading indicates the number of simulation instances assigned to the slot (darker for higher numbers).

Realisation diagrams can give a quick indication of 'problematic' tracks in an intersection, i.e., those for which flows dominate in the lower realisation slots. Comparison of realisation diagrams among subsets of tracks, particularly those associated with a vehicle type, can show how different subsets fare in a given scenario.

5.1.5 Flow effects sequence

In order to analyse the effects of various stream combinations at an intersection, we focus on some intersection flow value of interest, which may be either a single stream, i.e. the flow on a *track*, or the sum of flows on a number of tracks. In the simplest case, and the only one we deal with here, the flow value is examined at inputs above capacity of the track (this ensuring capacity flow on the track if only input is considered). For the flow on a single track, t_1, the negative effect on this flow produced by flow on track 2 (t_2), is calculated as:

$$f_{\mathrm{E}1(2)} = f_{\mathrm{C}1} - f_{1(2)} \tag{5.9}$$

where $f_{\mathrm{C}1}$ is capacity flow on track 1 and $f_{1(2)}$ is the flow on track 1 when inputs exist on track 1 and track 2 and there are no other inputs. All inputs are such as would result in capacity flow on tracks in isolation.

Now, the negative effect on track 1 flow produced by simultaneous flows on tracks 2 and 3 can be calculated as:

$$f_{\mathrm{E}1(2,3)} = f_{\mathrm{C}1} - f_{1(2,3)} + f_{\mathrm{E}1(2)} + f_{\mathrm{E}1(3)} \tag{5.10}$$

where $f_{1(2,3)}$ is the flow on track 1 when inputs are present on tracks 1, 2 and 3 only and $f_{\mathrm{E}1(2)}$ and $f_{\mathrm{E}1(3)}$ are the negative effects on track 1 flow produced by track 2 flow and track 3 flow, respectively.

By applying these calculations recursively, the effect of any combination of track flows on a single track flow can be calculated from measured flow values. If the effect on a combination of track flows is of interest, this can be calculated by addition as, for example:

$$f_{\mathrm{E1+2(3)}} = f_{\mathrm{E1(3)}} + f_{\mathrm{E2(3}} \tag{5.11}$$

where the negative effect of track 3 flow on the sum of track 1 and track 2 flows, $f_{\mathrm{E1+2(3)}}$, is calculated as the sum of the individual negative effects of track 3 flow on track 1 flow and on track 2 flow ($f_{\mathrm{E1(3)}}$ and $f_{\mathrm{E2(3}}$ respectively).

All inputs are such as would result in capacity flow on tracks in isolation. It should be noted that the so-called "negative effect" could itself have a negative value (in effect, therefore, being positive). Let us say that the flows on tracks 2 and 3 in Equation 5.10 separately have a negative effect on track 1 flow. However, when present together, track 2 and 3 flows may interfere with each other in a way that reduces the negative effect of each on track 1 flow, givint $f_{\mathrm{E1(2,3)}}$ a positive value.

5.2 Field data

Field data were collected with the purpose of comparing the quantitative properties of actual traffic with those of traffic in the simulated scenarios. While the number of scenarios covered by the field data is limited, the measurements have enabled the validation of basic aspects of the simulation model and the analysis of calibration needs and scope.

5.2.1 Data collection

Since the data were to be collected manually, by video recording, it was important to identify the location and time that were most likely to result in high bicycle volumes as part of the traffic flow. Available Canal Cordon Count Data for Dublin (http://dublinked.com/datastore/datasets/dataset-163.php) showed that of all the intersections along the two main canals in the city, the one at Rathmines Rd. Lower has the highest bicycle flows, both in the direction along the canal and that crossing it. This location was, hence, chosen for data collection.

The video camera positions from which traffic was filmed at the chosen intersection are shown in Figure 5.2. In this figure, each pair of drawn-in lines forming a 'V' shows the approximate angle of filming at a camera position. The position marked A in the picture looks onto traffic approaching the intersection from Rathmines Road Lower, while the position marked B in the picture has view of the entire intersection. The video recordings obtained are about 45 minutes long for each position and were made on the 2$^{\text{nd}}$ of December 2013, between 7:30 and 9:00 am.

Lane widths were measured using the Google Maps Distance Measurement tool and are shown in Table 5.1.

Lane (see Figure 5.2)	1	2	3	4
Width (m)	2.2	2.2	1.5	3

Table 5.1: Measured lane widths at the data collection location

5.2.2 Data processing

Traffic flow information was acquired from the footage by visual inspection of a sequence of still images extracted from the video at 1s intervals. The interval of 1s was chosen as it is short enough for all vehicles passing through the space visible by the video camera to be observed but long enough to reduce the size of the processing task, which, in the event, took 20h for the 1h 30min of footage.

The traffic flow and density were recorded manually, per second, for 4 lanes, which are indicated with arrows and marked with numbers in Figure 5.2. Data for lanes 1 and 2 were recorded from footage corresponding to both camera positions, while data for lanes 3 and 4 were recorded from the footage corresponding to camera position B only. The intersection is controlled by traffic lights, hence average flow and density values were calculated for green phases, as applicable to the four lanes, excluding values recorded in the first 10s of the green phase, to allow for a transition period (May, 1990). The flow and density values thus obtained are listed in Appendix B.

Use is made of these data in the following sections of this chapter, during the

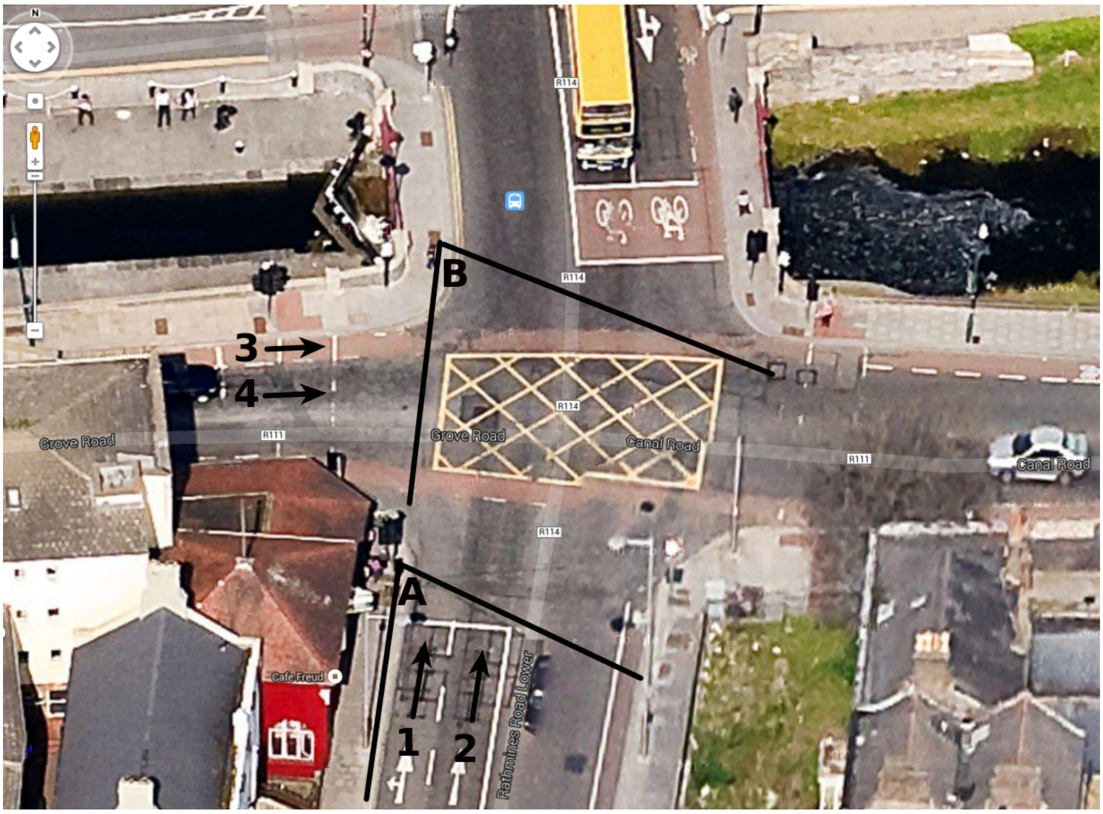


Figure 5.2: Field data collection location (based on a Google Maps satellite picture). The pairs of lines forming 'V' shapes, marked A and B, indicate the two camera positions and viewing angles that were used to record video footage. The arrows and accompanying numbers mark the traffic lanes of interest, lane 1, 2 and 4 being motorised traffic lanes and 3 a bicycle lane.

quantitative appraisal of the different types of simulation output.

5.3 Road stretch

The aim of simulating a road stretch is to produce a reference fundamental diagram for bicycles, confirm that the basic movement model behaves as expected and to investigate the properties of the part of the model that deals with car-bicycle interaction due to lane sharing. The spatial aspect of this scenario is built using a road stretch topological element (TE), the model for which is shown in Figure 3.6. The Road Stretch is configured to be of length 100 in car cells and periodic boundary conditions are realised by connecting Port 1 of the Road Stretch TE to Port 0 of the same TE (see Figure 4.4b for TE ports). Table 5.2 gives a summary of simulation parameters and their values

Parameter	**Value**
Configuration elements	road stretch (RS1)
Boundary conditions	closed
Connections (see Figure 4.4 for port numbers)	RS1:1[1]→ RS1:0
Simulation length per distinct parameter set	10^5 time-steps (corresponds to about 28 hours)
Randomisation parameter	0.1
Interaction parameters	$p_{\mathrm{RI}} \in 0.3, 0.9$ $t_{\mathrm{GI}} \in 0, 2$
Road length	100 car cells
Car density	$0.0 \leq \rho_{\mathrm{C}} \leq 1.0$, with step 0.05
Bicycle density	$0.0 \leq \rho_{\mathrm{B}} \leq 1.0$, with step 0.05

[1] The notation that is used for ports in parameter tables is <TE name>:<port number>.

Table 5.2: Road stretch simulation parameters. Each combination of the listed values represents a distinct simulation instance.

for the conducted *road stretch* simulations.

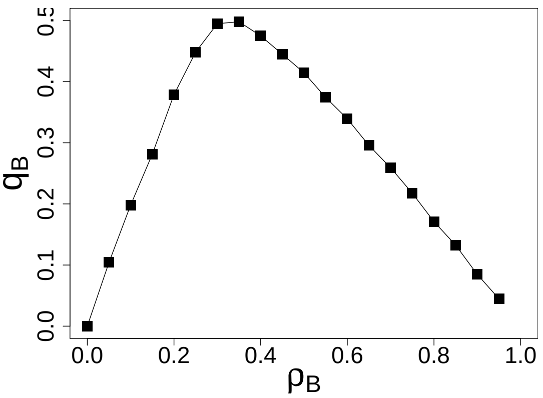


Figure 5.3: Fundamental diagram for bicycles, simulated

Figure 5.3 shows the fundamental diagram for bicycles alone. The maximal flow in this figure is 0.5, which corresponds to 1800 bicycles/hour. This value lies between the various bicycle capacity measurements reported in literature (ranging from 1500 bicycles/hour, as discussed in Zhang et al. (2007), under real traffic conditions, and 2100 bicycles/hour or more under experimental conditions, described in Gould and Karner (2009)). The randomisation parameter value producing this maximal flow is $p_{\mathrm{R}} = 0.1$ and this is the value used for bicycles in all simulations reported here.

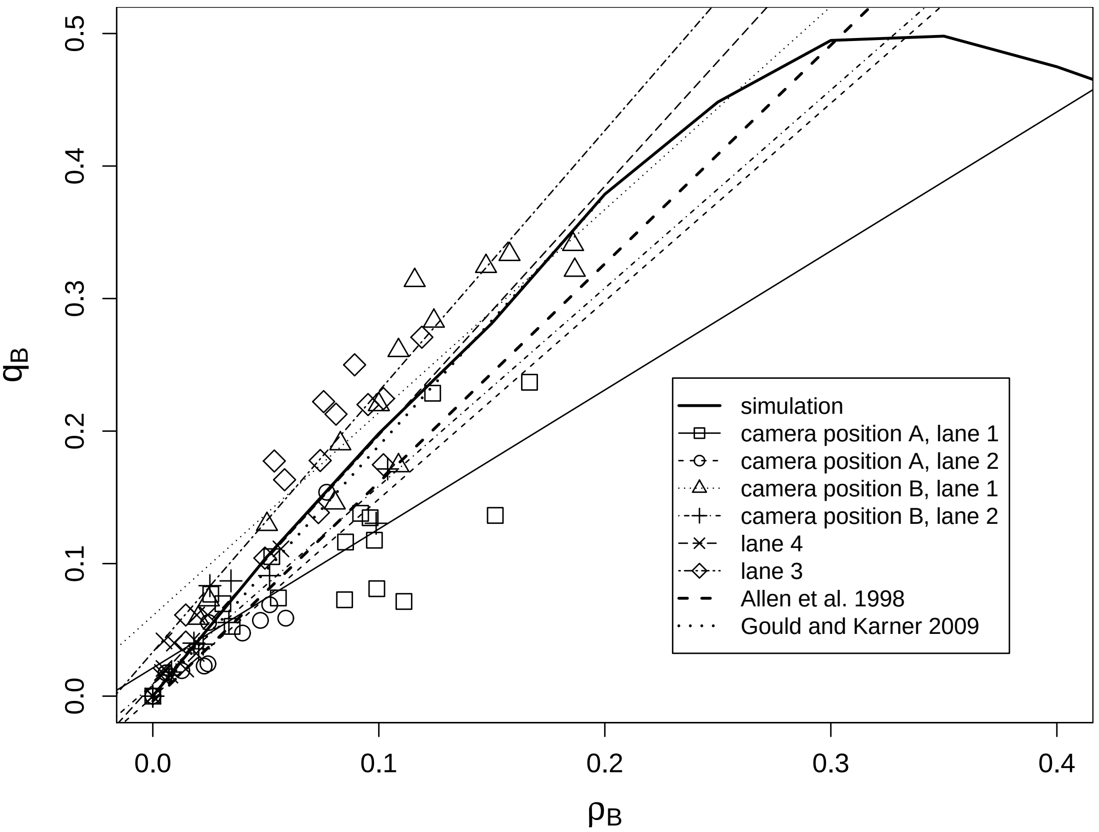
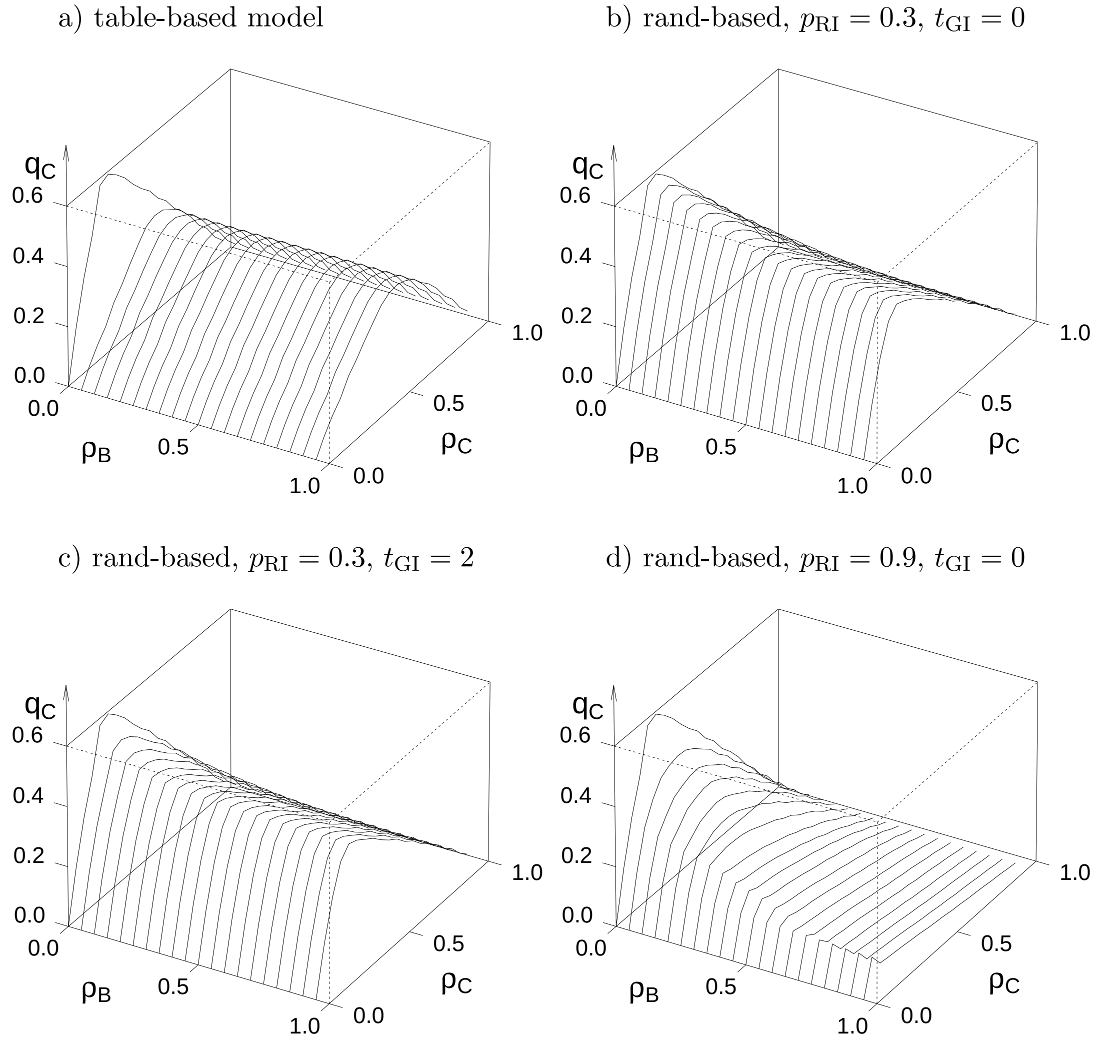


Figure 5.4: Fundamental diagram for bicycles, comparison with empirical data. Axes are: ρ_{B} - bicycle density, q_{B} - bicycle flow. The FD is shown for simulation, for 6 empirical data sets, each pertaining to a filmed lane, and 2 data sets reported by other researchers.

A comparison of this fundamental diagram with measured values is shown in Figure 5.4. While a measured bicycle fundamental diagrams including the congested region is reported in few publications (Gould and Karner, 2009), its free-flow region has been measured by several authors and features in our data sets. The groups of points shown in Figure 5.4 for each lane at the data collection site (1 and 2 measured from two different points and 3 and 4) are accompained by linear regression lines and the fact that these all have a y-intercept close to 0 shows that the points are approximately in the free-flow region of the fundamental diagram. In addition to these, amd apart from the bicycle fundamental diagram produced by the simulation model, the picture also shows lines representing data reported by two other authors (Allen et al., 1998a; Gould and Karner, 2009). Of our data, the set corresponding to lane 3, i.e. the bicycle lane, shows the highest velocity values, while that corresponding to lane 1, when collected from position A, displays the lowest ones. The average velocity is very similar for the two viewpoints of lane 2. Since the current model concerns itself with unconstricted bicycle flows in shared lanes, in order to validate it we look for correspondence with the free-flow regions for lane 4 and lane 1 from position B (the latter represents traffic at the intersection, where bicycles have some more lateral freedom than upstream in the very narrow lane 1), and this exists in the picture. The lower velocities of lane 2 and lane 1 viewed before the intersection are a feature of the measured traffic probably due to very restricted space (the width of these lanes is only 2.2m each), insufficient for ongoing side-by-side sharing.

The road stretch topological element implies the sharing of a narrow but sufficiently wide lane between cars and bicycles (a relationship between tracks that requires handling as part of the movement rules). In Section 3.3.6, we proposed two ways of handling interactions between cars and bicycles due to lane sharing: the deceleration table-based method and randomisation parameter variation. Figure 5.5 presents car-only fundamental diagrams (see Section 5.1.1) as a function of bicycle density, for different car-bicycle interaction models (as described in Section 3.3.6). In all these models the presence of bicycles reduces average car flow but the mechanisms by which this is effected are different. In Figure 5.5a, the car fundamental diagram takes on the even-sided triangle shape of low-velocity traffic, brought about by the slowing down of cars in the presence of bicycles (prescribed by the deceleration table-based model, presented in Section 3.3.6). The diagrams in Figures 5.5b, c and d are all for the randomisation parameter variation-based model of car-bicycle interaction (Section 3.3.6). With higher bicycle densities these tend towards congestion at low density values, due to the stochastic nature of the car-bicycle interaction. We find that the time headway has little effect while the adjusted randomisation parameter is crucial for the amount of interaction visible. In Cheng et al. (2008) the two approaches are combined as part of the rules, resulting in general slowing for cars but with less predictability.

Examination of these interaction models with respect to measured data require suitable empirical car flows. We determine suitability based on three criteria: the car flow must share a lane with bicycles, the bicycle flow in the shared lane must have



a) table-based model

b) rand-based, $p_{\mathrm{RI}} = 0.3$, $t_{\mathrm{GI}} = 0$

c) rand-based, $p_{\mathrm{RI}} = 0.3$, $t_{\mathrm{GI}} = 2$

d) rand-based, $p_{\mathrm{RI}} = 0.9$, $t_{\mathrm{GI}} = 0$

Figure 5.5: Road stretch with periodic boundary conditions: fundamental diagrams for cars with different bicycle densities, for (a) *deceleration-table-based model* of bicycle effect on cars and for (b) *randomisation parameter variation*, using different values for adjusted randomisation parameter and time headway (b, c and d). p_{RI} denotes the adjusted randomisation parameter in the randomisation-based interaction model, while t_{GI} is the interaction time-gap (the number of seconds that the car would need to keep moving at its current velocity in order to arrive alongside the cell that contains the closest bicycle).

average velocities similar to those predicted by the model and the car flow macroscopic measurements must be in the free-flow region of the fundamental diagram. While the first two conditions are imposed so that the modeled situation be matched, the last one simplifies the study of the effect bicycle flows have on cars sharing the same lane, since its fulfillment allows the effects on car flow to be observed through a single variable

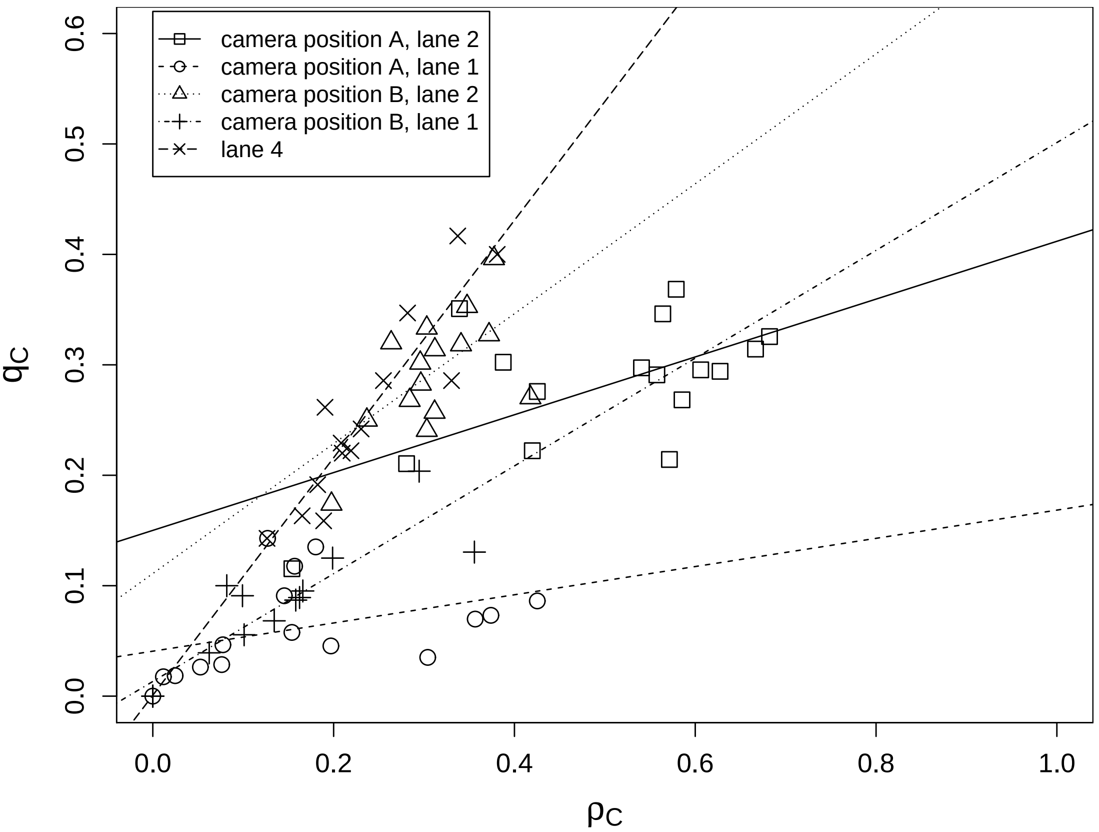


Figure 5.6: Car fundamental diagrams, empirical data. Axes are: ρ_C - motorised vehicle density, q_C - motorised vehicle flow. The FD is shown for 6 empirical data sets, each pertaining to a filmed lane.

- the average velocity. Figure 5.6 shows measured values in the fundamental diagram space for 5 motorised traffic lanes. For two of these, lane 4 and lane 1 viewed from position B, the linear regression lines have y-intercepts at approximately 0, indicating that these data sets can be considered to be in the free-flow region of the fundamental diagram. These are also the two car flows that are adjacent to bicycles with predicted velocities (see Figure 5.4). Figure 5.7 looks at the effect of bicycle flows on average velocities in the lanes filmed from position B. The figure includes simulation output for this dependency. It also includes average car velocities, as influenced by the presence of adjacent bicycle flows, measured by two other authors. The first observation that can be made from the figure is that the motorised vehicle velocities (some buses were recorded in the motorised lanes, but the terms ‘car’ and ‘motorised’ are used here interchangeably when referring to lanes and flows) in lane 2 are higher than those in lane 1. However,

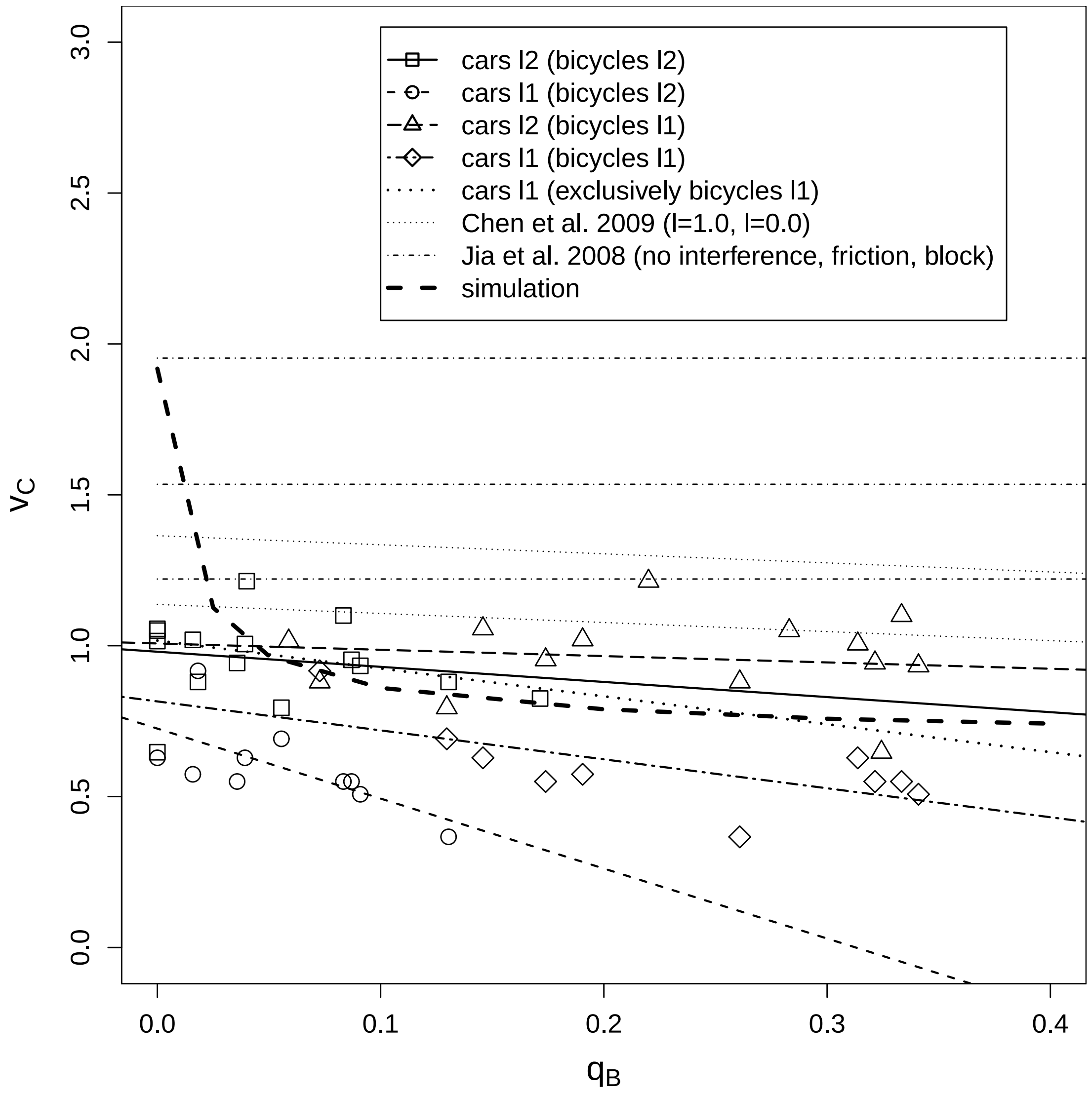


Figure 5.7: Bicycle effect on cars sharing a narrow lane, comparison with empirical data. Axes are: q_{B} - bicycle flow, v_{C} - average velocity of the impacted motorised flow. Car velocity as a function of bicycle flow is shown for four pairs of streams e.g. cars l2 (bicycle l2) pairs the car and bicycle streams in lane 2. Line marked as 'cars l1 (exclusively bicycles l1)' shows an estimate of the impact of lane 1 bicycle flow on lane 1 average motorised velocities in isolation from other factors. Data reported by Jia et al. (2008) and Chen et al. (2009) is, in each case, represented as a series of lines showing average velocities at different bicycle impact levels, with the higher lying lines corresponding to lower impact and vice versa. Jia et al. (2008) use descriptive levels, while Chen et al. (2009) have estimated the impact using a linear equation, with the lateral distance between cars and the bicycle lane, l, as one variable and bicycle flow the other. Finally, bicycle flow values are plotted against corresponding shared lane car velocities for the road stretch scenario simulation.

in both cases, the dependency on bicycle flow in the same lane has a downward trend and values that are in the same order of magnitude as the predicted ones (diamonds and squares for lanes 1 and 2, respectively, in comparison with the short-dashed thick line). It can also be seen that the effect of bicycles in the 'other' lane is very different for the two lanes. In the case of motorised lane 1, there is a pronounced negative effect from bicycles in lane 2, since they are adjacent to the motorised traffic on lane 1, from the right-hand-side, while bicycles in lane 1 have little effect on motorised traffic in lane 2. This partly explains the lower velocities of motorised traffic in lane 1. In order to estimate the effect of lane 1 bicycles on lane 1 cars in isolation, a number of quantities that were measured and that are suspected as possibly of impact on the velocity of lane 1 cars were treated as independent variables in a multiple regression exercise with lane 1 motorised vehicle average velocity as the dependent variable. The independent variables were added one by one and each increased the multiple regression coefficient, with the final value of R=0.91 and the regression formula:

$$v_{\mathrm{C1}} = 1.0171 - 2.4163q_{\mathrm{B2}} - 0.9241q_{\mathrm{B1}} - 0.1338p_{\mathrm{L}} + 0.3104p_{\mathrm{TC}} - 0.5636p_{\mathrm{TB}} \qquad (5.12)$$

where v_{C1} is he velocity of motorised vehicles in lane 1, q_{B1} and q_{B2} are the bicycle flows in lanes 1 and 2, respectively, p_{L} is the proportion of long vehicles in the motorised stream of lane 1, p_{TC} is the proportion of motorised vehicles turning from lane 1 just before the measured section of that lane and p_{TB} is the proportion of bicycles turning from lane 1 just before the filmed section. The dependency of average motorised velocity in lane 1 on the bicycle flow in lane 1, isolated from the other independent variables, is drawn into the diagram with a dotted line and is a much improved match for the simulation output than the measured velocity values. Both our measurements and simulation output differ greatly from empirical values reported by other authors (Figure 5.7). Apart from the free flow average velocity given by Jia et al. (2008), which matches the simulation free flow velocities, we find that the other two authors' reported bicycle 'interference' with motorised flow is much less pronounced, with higher motorised velocities even with 'block' interference i.e. the situation where bicycles are impinging on car lanes. However, if the studies are closely examined, it can be seen that in Jia et al. (2008) the car lanes are as wide as 3.7m, which would suggest that there is enough space for safe passing and that the slowing stems only from car re-positioning on the road when needed. In the case of Chen et al. (2009), the lane width is not reported but the 'bicycle effect' is found to be dependent on the lateral distance of cars from the bicycle lane, again suggesting an exclusive motorised lane of considerable width and a smaller impact of adjacent bicycle flows than in the case we are modelling. The different measurements would suggest that the amount of 'interference' increases with decreasing lane widths as indicated by Wilkinson et al. (1994). As regards the two different models of bicycle-motorised interaction that we have implemented, the deceleration table based model seems to be more suitable, since it involves a negative effect of ajdacent bicycle flow on car velocity in the free flow region of the fundamental diagram, a relation that is found in the field data. With the randomisation based interaction model car velocity in the free flow region does not change with adjacent bicycle flow, hence this model does not fit the collected data. For this reason, in all other scenarios exclusively deceleration table-based interaction is used.

In Figure 5.8 the bicycle interference is shown for the car lane (number 4 in Figure 5.2) running alongside the bicycle line (number 3 in Figure 5.2). The bicycle flow used as the influencing factor consists only of bicycles that are actually travelling in the car lane and not bicycles that stay inside the bicycle lane. Here the car velocities are higher and bicycle flows remain low, however, the downward slope of the linear regression line for the velocities plot against bicycle flows would indicate an impact. The motorised average velocities are a bit higher than those for lane 1 in Figure 5.7, which fits with literature prediction of better sharing performance in wider lanes (Wilkinson et al., 1994).

5.4 Left and right turns

The *left turn* and *right turn* Topological Elements are shown in Figures 3.7 and 3.8. Table 5.3 shows the applicable simulation parameters and their values.

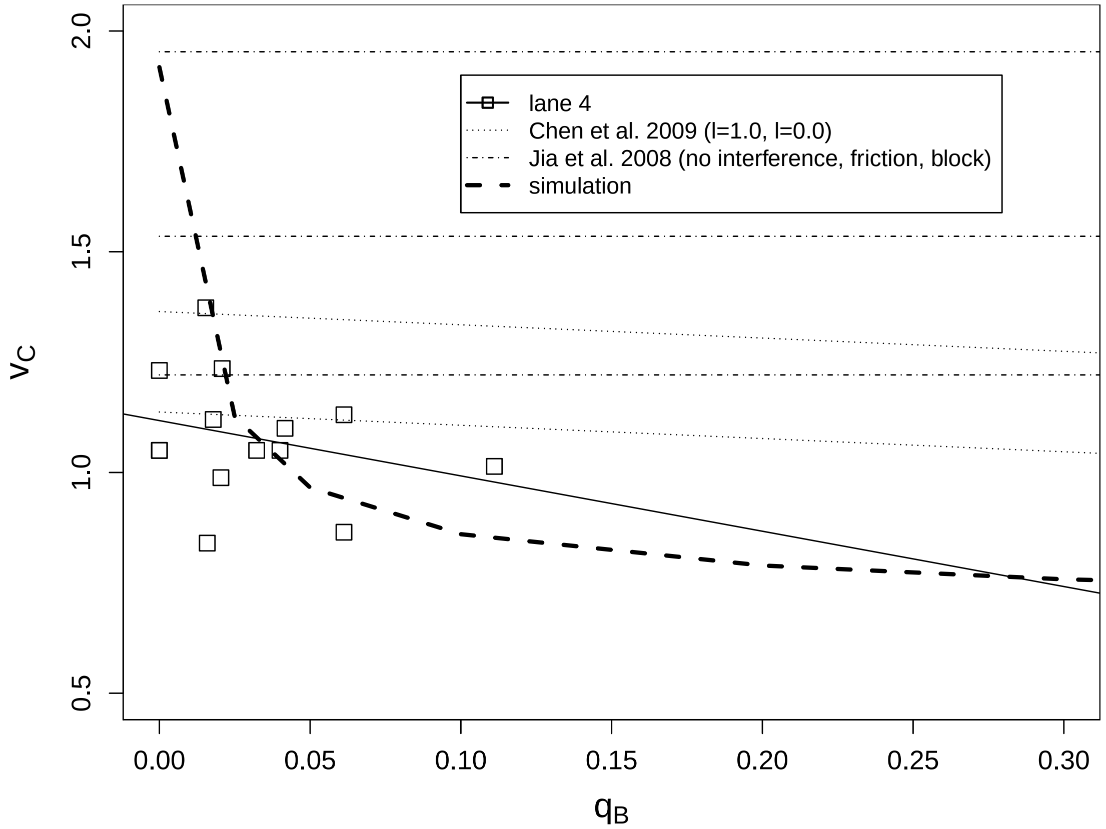


Figure 5.8: Bicycle effect on cars alongside bicycle lane, comparison with empirical data. Axes are: q_B - bicycle flow ocurring in motor vehicle lane, v_C - average velocity of the impacted motorised flow. The car flow in lane 4 as a function of bicycle flow in the same lane is shown, alongside the same data from literature and simulation output as in Figure 5.7

The simulated fundamental diagram for the left turn is shown in Figure 5.9 for values of turning probability for cars. The fundamental diagram lines in this picture are typical of open boundary conditions (Barlovic et al., 2002) in that they do not have the descending portion representing congested traffic. The present sections correspond to free-flow, however, their extension far into the high-density regions indicates queueing by vehicles. Longer queues, accompanied by lower capacities, are the effect of the car-bicycle cross-flow in the scenario. The effect is more pronounced with the higher car turning probability value of 1.0.

Figure 5.10 shows the measured impact of the proportion of left-turning vehicles in the motorised stream of lane 1 (filmed from camera position A) on the average velocity in that stream. The simulation output in this space is shown for two values of bicycle

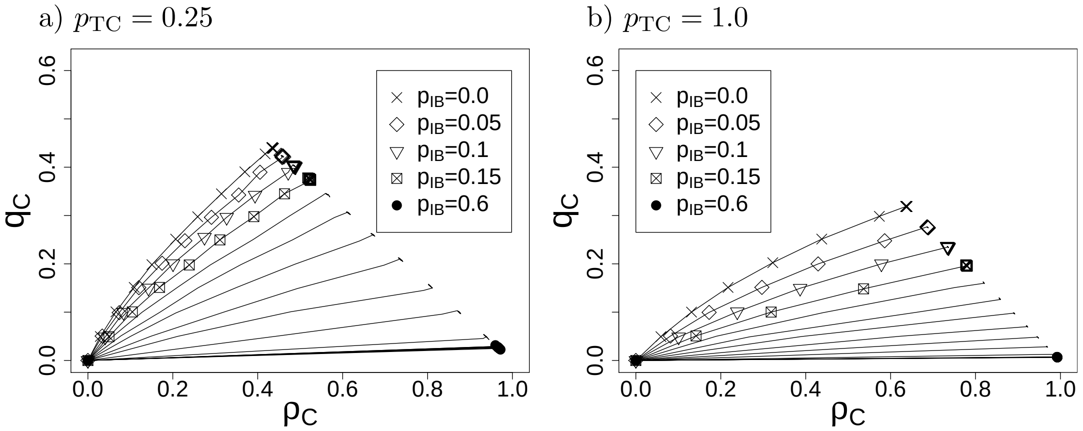


Figure 5.9: Left turn: cars-only fundamental diagram, for different bicycle insertion probabilities, p_{IB}, in the absence of interaction due to lane-sharing. p_{TC} is the turning probability for cars, while the turning probability for bicycles is $p_{\mathrm{TB}} = 0.0$. Points in the graphs are shown for selected values of p_{IB}.

arrival rate. The bicycle arrival rate, p_{IBM} can be considered to be identical in value to bicycle flow, q_{B}, at low values. The measured motorised average velocity values are somewhat lower than predicted by simulation based on the proportion of turning vehicles only, the bicycle arrival rate being about $p_{\mathrm{IB}} = 0.11$ on average. This, however, is in fact favourable for the validity of the model, considering that other car-velocity reducing factors are present in the field data, such as bicycle flow in lane 2 and the presence of buses in the motorised stream of lane 1, which have not been accounted for in the model. Finally, Figure 5.10 includes velocities measured by Chen et al. (2007) for a right-turning car flow in the absence of bicycles sharing the same line, with such bicycles waiting to pass and with bicycles actually moving forward, forming a cross-flow for the cars. While bicycle flow values are not specified in Chen et al. (2007), these points in the picture do give some indication of the bicycle cross-flow effect on turning cars in their data: the velocity they measured for 'careful' turning cars matches that manifested by simulated turning motorised flow in the absence of bicycles and the velocity they measured for cars crossing a moving bicycle stream matches the average velocity in the simulation with $p_{\mathrm{IB}} = 0.1$

Further information on how different model parameters and inputs influence the interaction between flows can be gleaned from a diagram comparing *capacity curves*,

Parameter	**Value for left turn**	**Value for right turn**
Configuration elements	left turn (LT1) and three road stretches (RS1, RS2 and RS3)	right turn (RT1) and three road stretches (RS1, RS2 and RS3)
Boundary conditions	open	open
Connections (see Figure 4.4 for port numbers)	RS1:1 → LT1:0, LT1:1 → RS2:0, LT1:2 → RS3:0	RS1:1 → RT1:0, RT1:1 → RS2:0, RT1:2 → RS3:0
Road length	20 car cells (each road stretch)	20 car cells (each road stretch)
Variable difficulty of turning	1, 2, 3 (cars)	1, 3, 5 (bicycles)
Routing	macroscopic, with turning probabilities: 0.0, 0.5, 1.0 (cars), 0.0 (bicycles)	macroscopic, with turning probabilities: 0.0 (cars), 0.0, 0.5, 1.0 (bicycles)
Simulation length per distinct parameter set	25000 time-steps (corresponds to about 7 hours)	25000 time-steps (corresponds to about 7 hours)
Randomisation parameter	0.1	0.1
Accepted gap	1, 2	1, 2
Car insertion probability	$0.0 \leq p_{\mathrm{IC}} \leq 0.7$, with step 0.05	$0.0 \leq p_{\mathrm{IC}} \leq 0.7$, with step 0.05
Bicycle insertion probability	$0.0 \leq p_{\mathrm{IB}} \leq 0.7$, with step 0.05	$0.0 \leq p_{\mathrm{IB}} \leq 0.7$, with step 0.05

Table 5.3: Left and right turn simulation parameters. Each combination of the listed values represents a distinct simulation instance.

shown for the left turn in Figures 5.11 and 5.12 and for the right turn scenario in Figure 5.13. The capacity curve for a particular traffic stream with respect to another stream is the plot of flow values for the first stream as a function of the second stream's flow values, while the first stream is fed by vehicles at a level of input that would produce capacity flows in that, first stream in isolation. In Figures 5.11a and b the observed vehicle type is the car. Since the *left turn* scenario is shown, the car stream yields to the bicycle stream and is affected by the flow levels in the bicycle stream. Figure 5.11a is for the case of the deceleration table-based car-bicycle interaction and displays reduction in car flows with increase in bicycle inputs (insertion probabilities, p_{IB}) even when no cars are turning ($p_{\mathrm{TC}} = 0$), because of lane-sharing interaction. The other curves in Figure 5.11 show car flow capacity for different combinations of values of the *proportion of turning cars* (p_{TC}), the *difficulty of turning parameter* (d_{TC}) and the *accepted gap for cars with respect to bicycles* (g_{A}). The *difficulty of turning parameter* is

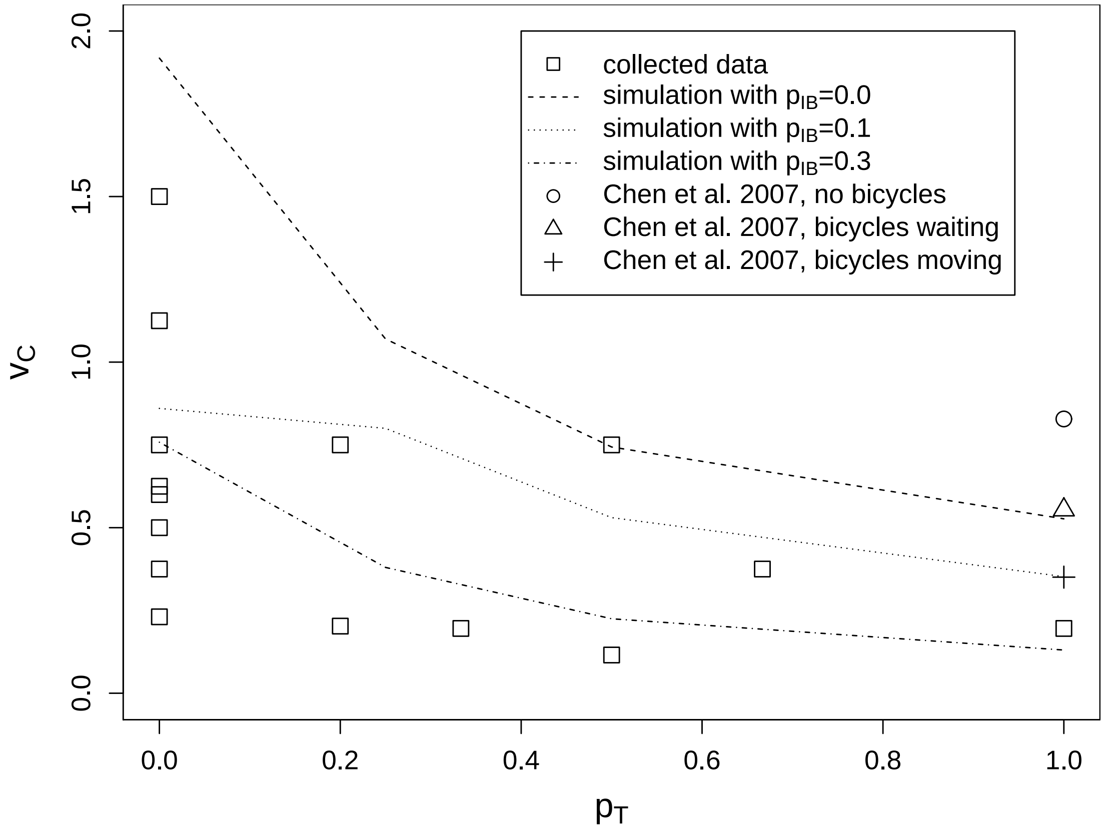
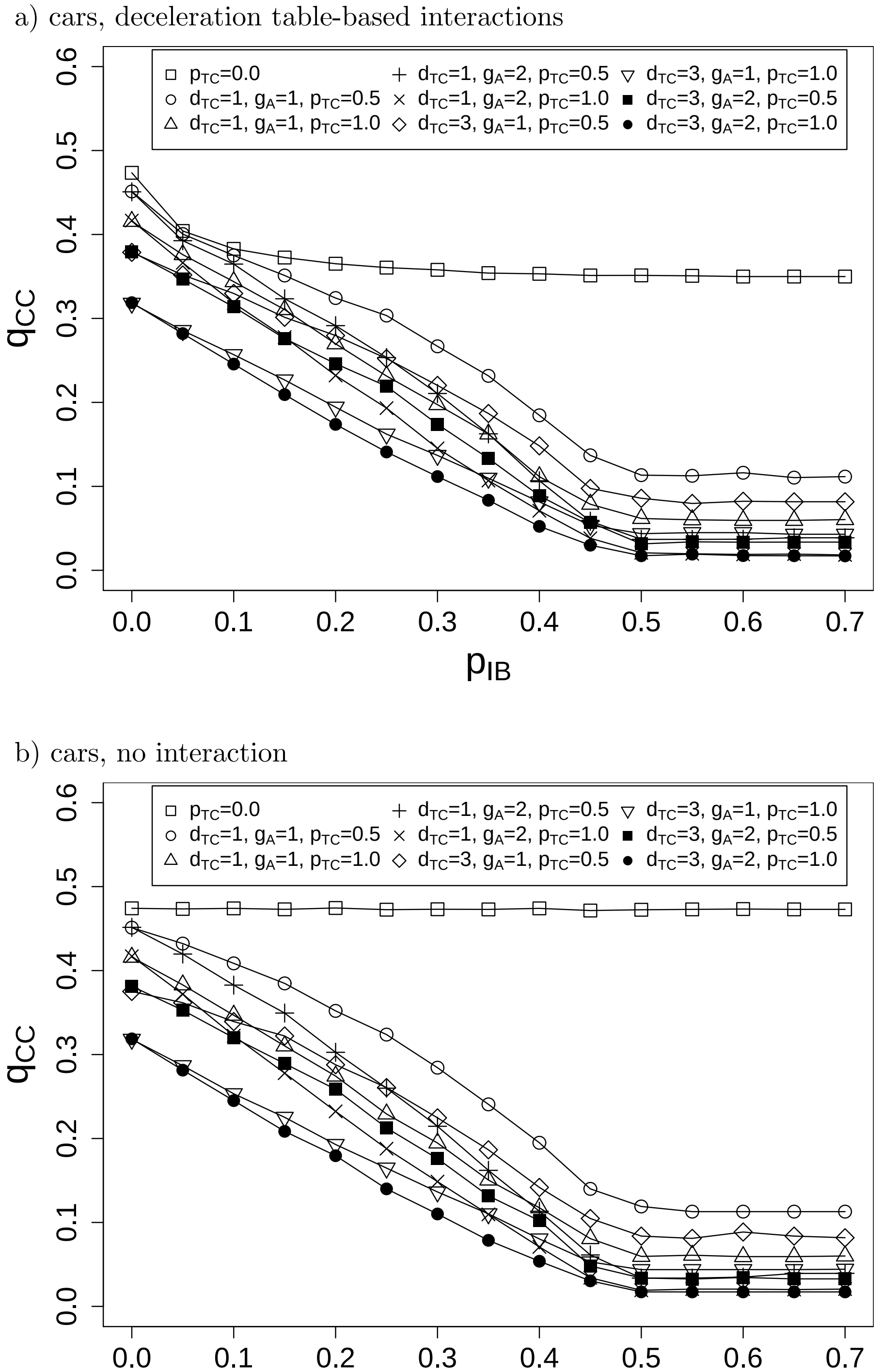


Figure 5.10: Car fundamental diagrams, empirical data. Axes are: p_{T} - proportion of motorised vehicles that are turning, v_{C} - motorised vehicle velocity. The relationship is shown for: collected data (lane 1 with camera position A), simulated flow with $d_{\mathrm{TC}} = 3$ and three different values of p_{IB}; and values observed by Chen et al. (2007) in flows with $p_{\mathrm{T}} = 1.0$.

a construct devised for our model and is effectively the cell count for a turning track in the intersection and is one of the configurable parameters when it comes to intersection interactions in the model. Capacity is reduced for greater proportions of turning cars, for higher difficulty of turning parameter values and for larger accepted gaps. The difficulty of turning for cars, d_{TC}, is, in this context, the number of overlapping cells in the left turning car track. The graphical representation of the left turn model, in Figure 3.7 shows three cells, corresponding to $d_{\mathrm{TC}} = 3$. The capacity car flow, q_{CC}, for all cases where the car turning probability is non-zero is lower than for the case where no cars turn ($p_{\mathrm{TC}} = 0.0$), even if no bicycles are involved ($p_{\mathrm{IB}} = 0$). This is due to cars slowing down before a turn, and is further compounded by higher values for d_{TC}, whereby turning cars take longer for the manoeuvre. The sets of curves in



a) cars, deceleration table-based interactions

b) cars, no interaction

Figure 5.11: Left turn: capacity curves for cars, probability of turning for bicycles is $p_{\mathrm{TB}} = 0.0$. Axes are: p_{IB} - insertion probability for bicycles; q_{CC} - car capacity flow. Parameters are: d_{TC} - difficulty of turning for cars; g_{A} - universal accepted gap; p_{TC} - probability of turning for cars.

c) bicycles, deceleration table interactions

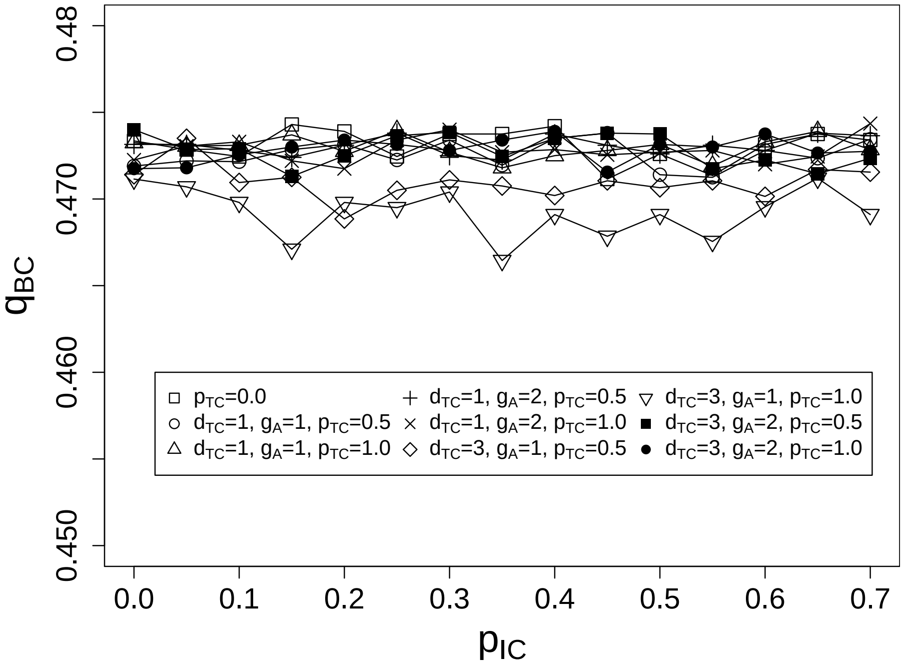
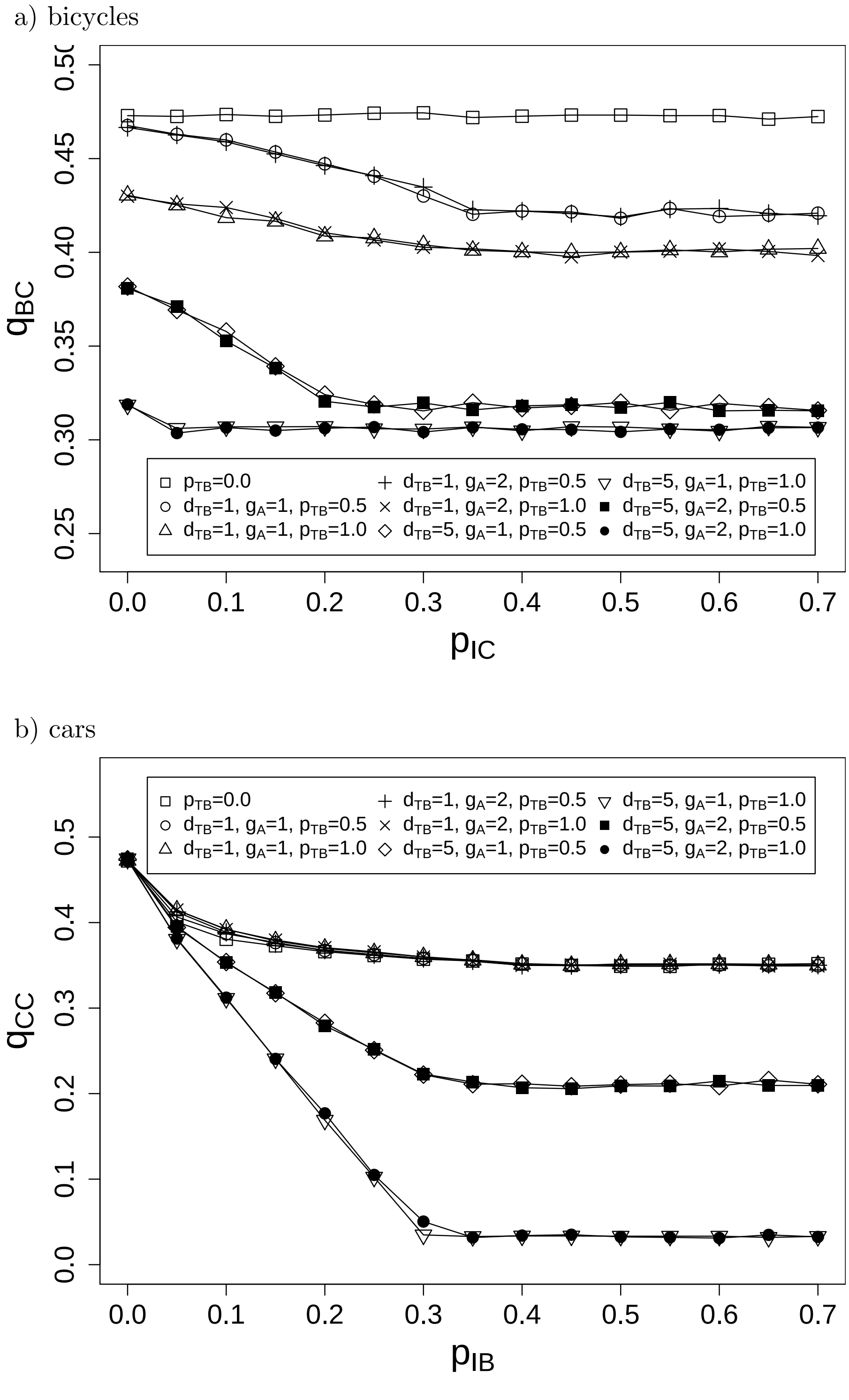


Figure 5.12: Left turn: capacity curves for bicycles, probability of turning for bicycles is $p_{\mathrm{TB}} = 0.0$. Axes are: p_{IC} - insertion probability for cars; q_{CB} - bicycle capacity flow. Parameters are: d_{TC} - difficulty of turning for cars; g_{A} - universal accepted gap; p_{TC} - probability of turning for cars.



a) bicycles

b) cars

Figure 5.13: Right turn: capacity curves for deceleration table-based model of interactions due to lane-sharing, probability of turning for cars is $p_{\mathrm{TC}} = 0.0$. Axes are: p_{IB} - insertion probability for bicycles; q_{CC} - car capacity flow; p_{IC} - insertion probability for cars; q_{BC} - bicycle capacity flow. Parameters are: d_{TB} - difficulty of turning for bicycles; g_{A} - universal accepted gap; p_{TB} - probability of turning for bicycles.

Figures 5.11b are similar to those in 5.11a, the only significant difference being in the curve for $p_{\mathrm{TC}} = 0$, where the lane-sharing interaction types vary in the extent and form of negative impact on capacity car flow. Even with the very optimistic scenario of $d_{\mathrm{TC}} = 1$, $g_{\mathrm{A}} = 1$ and $p_{\mathrm{TC}} = 0.5$ and the rigorous deceleration-table lane-sharing interaction model, the cross flow effect is the dominant impact on car flows. These results loosely fit with the observation of Chopard et al. (1996) that in the context of urban traffic (as opposed to motorway traffic), intersections have greater impact on flows than any other factor.

Figure 5.12 shows the bicycle flows for the same scenario, indicating small reductions in bicycle flow for larger d_{TC} and smaller accepted gap. This is caused by by cars 'edging' their way into the conflict zone and lingering there long enough to reduce the bicycle flow slightly. Much more dramatic but similar impact is made by right-turning bicycles on straight going cars in the right turn scenario. This can be seen in Figure 5.13, which contains capacity curves for that case. Figures 5.13a and b are analogous to the left turn diagrams in Figures 5.11a and 5.12. For the parameters given, the negative effect that right turning bicycles have on straight going cars would seem to be somewhat unrealistic, although it could be envisaged as applying, e.g. in the Beijing of 20 years ago, where bicycle streams had sufficient force to 'push through' in the loosely prioritised traffic mix of bicycles and motorised vehicles. A higher accepted gap by bicycles with respect to cars is required to solve the 'problem'. An investigation with a wider range of more granular *accepted gap values* distinguished by vehicle type would be an interesting topic for further work.

The variation in system output, caused by different values of model parameters *difficulty of turning* and *accepted gap* and illustrated by Figures 5.11 and 5.13, demonstrates the possibility of model calibration, in the event of larger and more numerous data sets being available for this kind of mixed traffic.

5.5 Intersection of two one-way roads

As topological elements become more complex, the number of flows entering and exiting them increases and it becomes harder to produce a comprehensive, unifying view of related measured data. An intersection of two 1-way roads has 2 in-connected ports (0 and 1) and 2 out connected ones (2 and 3 - see Figure 4.4). These compose 8 flows in total and pairs of different flows are numerous.

Parameter	**Value**
Configuration elements	intersection of two one-way roads (I1) and 4 road stretch elements (RS1, RS2, RS3, RS4)
Boundary conditions	open
Connections (see Figure 4.4 for port numbers)	RS1:1 → I1:0 RS2:1 → I1:1 I1:2 → RS3:0 I1:3 → RS4:0
Simulation length per distinct parameter set	10^5 time-steps (corresponds to about 28 hours)
Randomisation parameter	0.1
Routing	macroscopic, with turning probabilities: 0.0, 1.0 at each divergence
Road length	50 car cells per road stretch
Car density	$0.0 \leq \rho_\mathrm{C} \leq 0.7$, with step 0.1
Bicycle density	$0.0 \leq \rho_\mathrm{B} \leq 0.7$, with step 0.1
Conflict resolution	RHS and LHS priority
Accepted gap	1

Table 5.4: Intersection of two one-way roads simulation parameters. Each combination of the listed values represents a distinct simulation instance.

Simulations on this type of intersection were performed with parameters shown in Table 5.4 and the resulting capacity curves are shown in Figures 5.14 and 5.15. The former contains capacity curves for the RHS (right-hand-side) conflict resolution scheme and the latter for the LHS (left-hand-side) conflict resolution scheme. The ‘right’ and

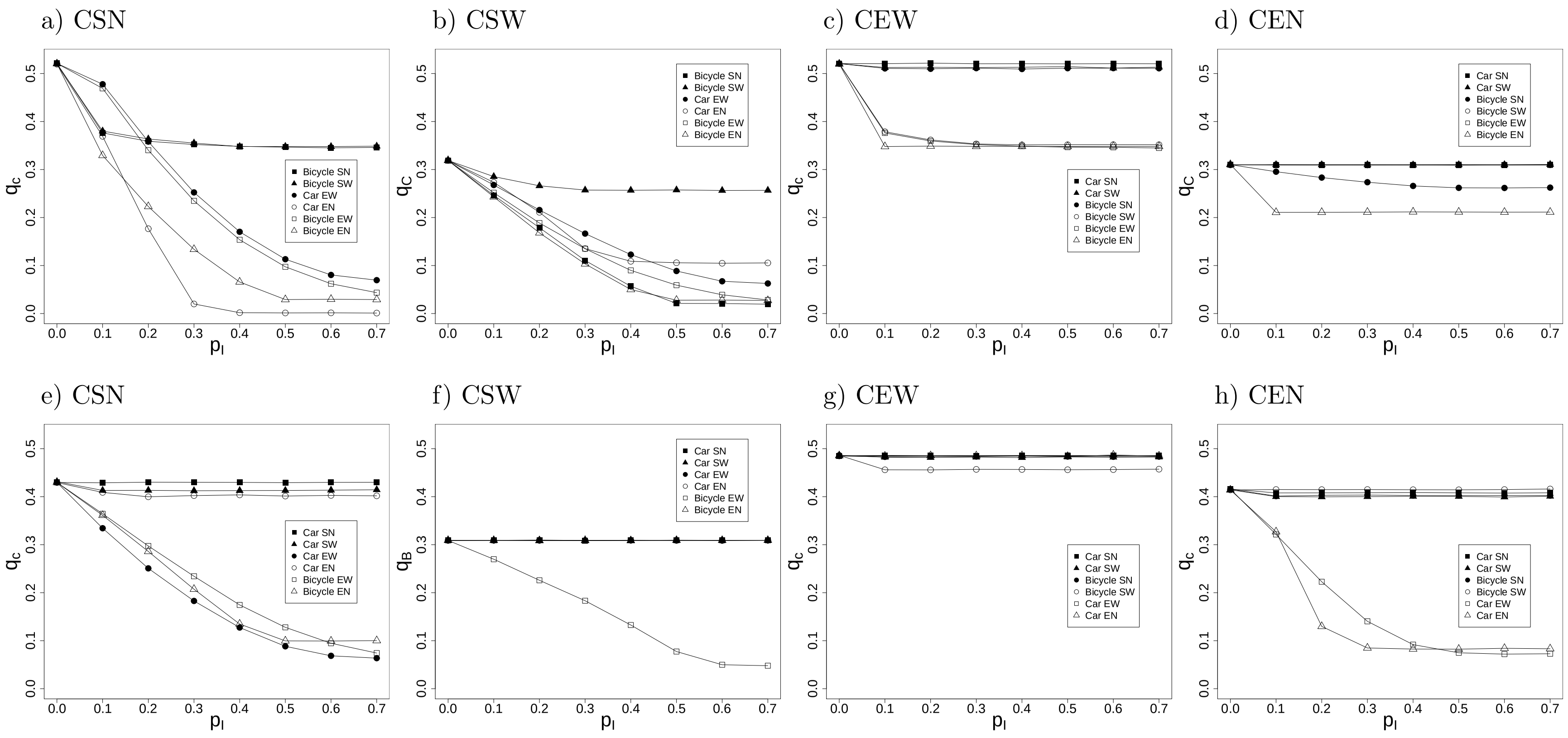
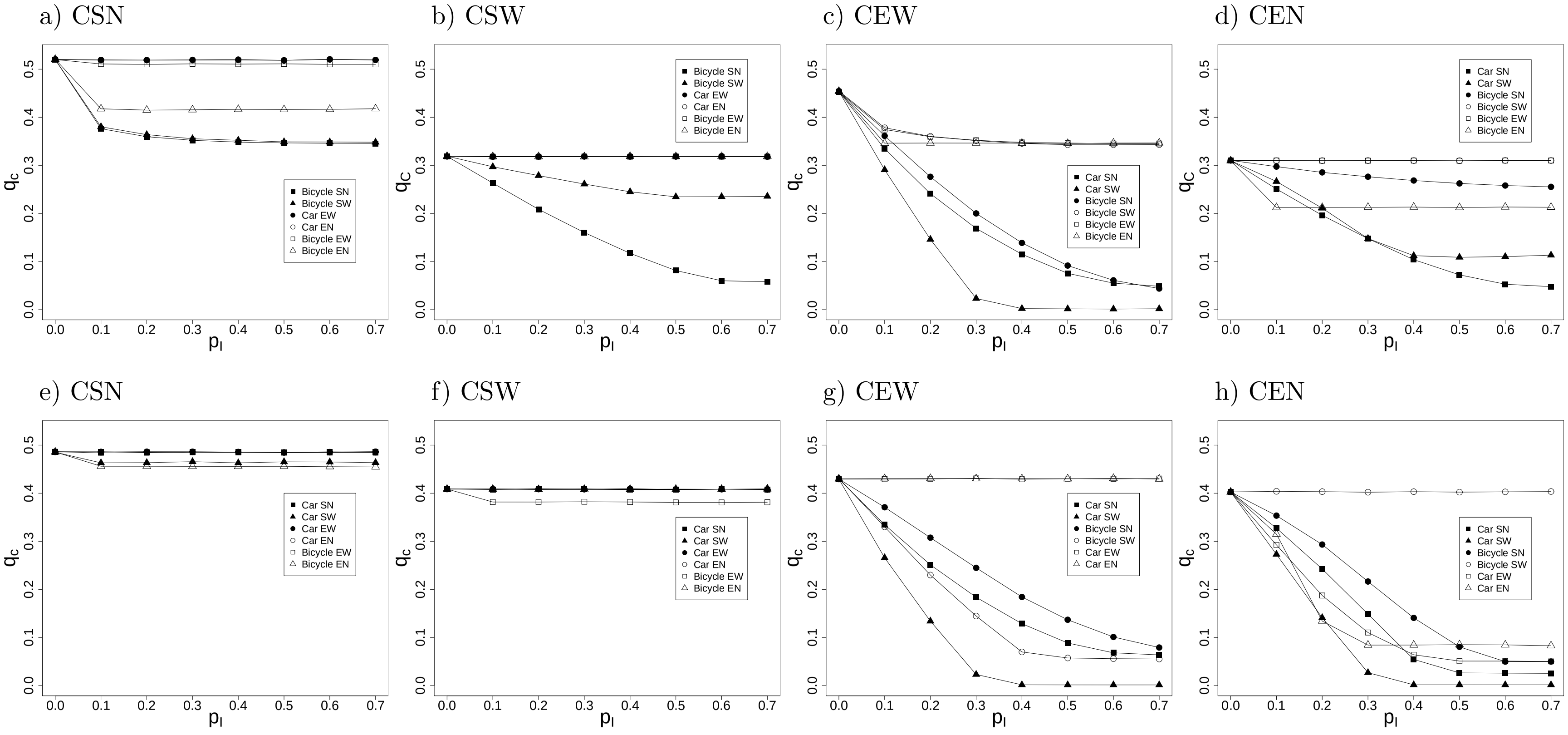


Figure 5.14: Capacity curves for an intersection of 2 1-way single-lane roads, using the RHS conflict resolution scheme: A diagram is shown for each track in the topological element, containing a curve for each of the other track flows as 'interfering'. The 'interfering' track flows are indicated in the legend. The track names follow the pattern ⟨vehicle type letter⟩ ⟨2 track direction letters⟩ (e.g. BSN stands for bicycle south-north track).


a) CSN

b) CSW

c) CEW

d) CEN

e) CSN

f) CSW

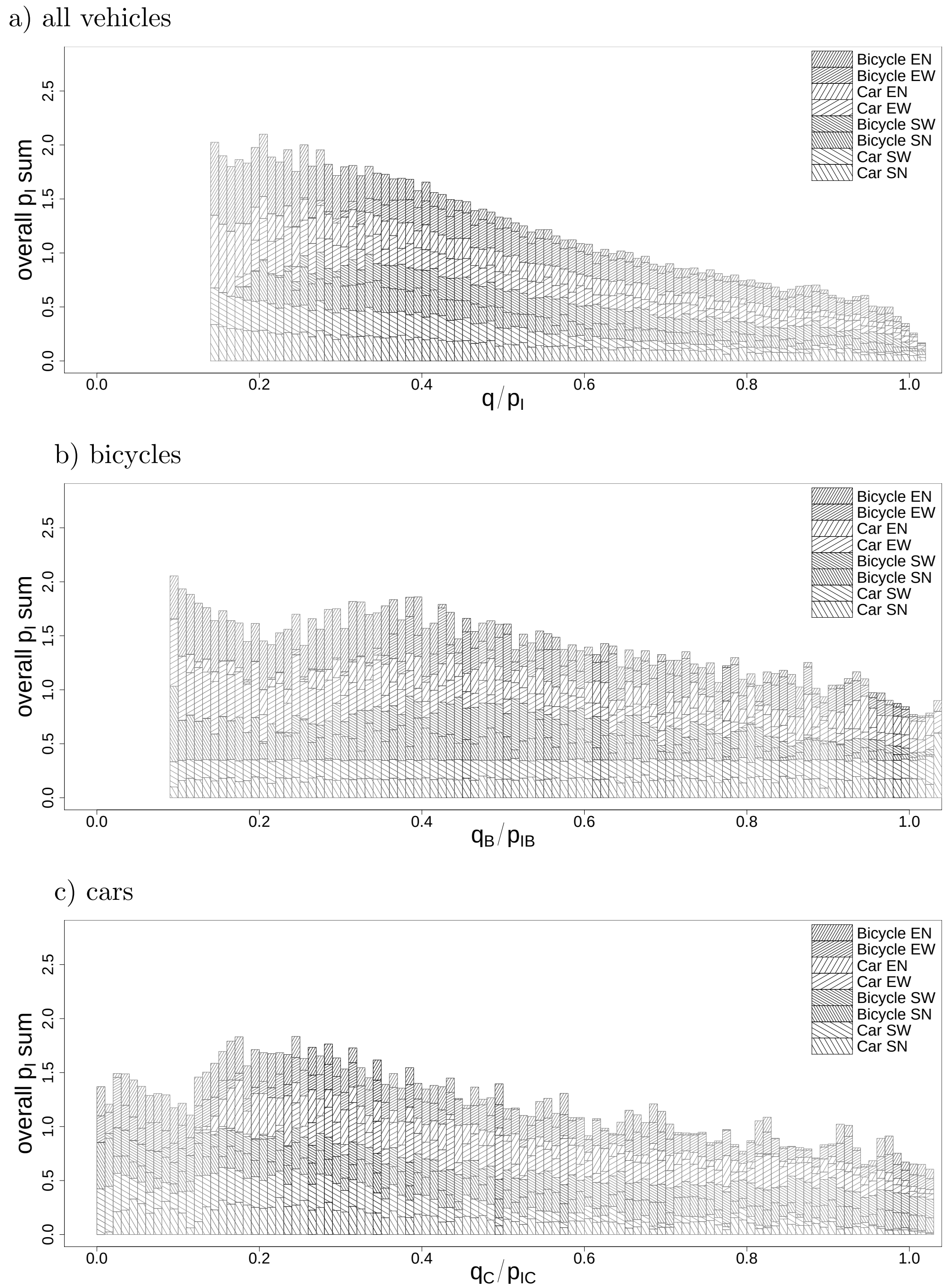
g) CEW

h) CEN

Figure 5.15: Capacity curves for an intersection of 2 1-way single-lane roads, using the LHS conflict resolution scheme: A diagram is shown for each track in the topological element, containing a curve for each of the other track flows as ‘interfering’. The ‘interfering’ track flows are indicated in the legend. The track names follow the pattern ⟨vehicle type letter⟩ ⟨2 track direction letters⟩ (e.g. BSN stands for bicycle south-north track).

‘left’ designation indicates the side at which an observing vehicle finds a higher-priority road. If the observing vehicle arrives on the south port or the intersection (this is port 0 - see Figure 4.4), with the RHS rule any vehicles arriving on the east port of the intersection (port 1) have priority over the observing vehicle. In the case of the LHS rule, the priority is inverted. The parameters used for the simulations are given in Table 5.4. In both Figures 5.14 and 5.15 the dramatic effect of most of the other flows on all the non-priority tracks is immediately noticeable. Also, the CSW and the BEN track flows (following the track naming convention given in the Figures) are negatively affected even when they are in the priority flow, by the flow sharing their road, which has a “lane-local” priority. The turning tracks have lower flows even with no interference from other flows, because of the deceleration before a turn. The same applies to flows on lower-priority tracks, since vehicles ‘know’ they are on a lower priority track and decelerate in order to check for vehicles on the conflicting track.

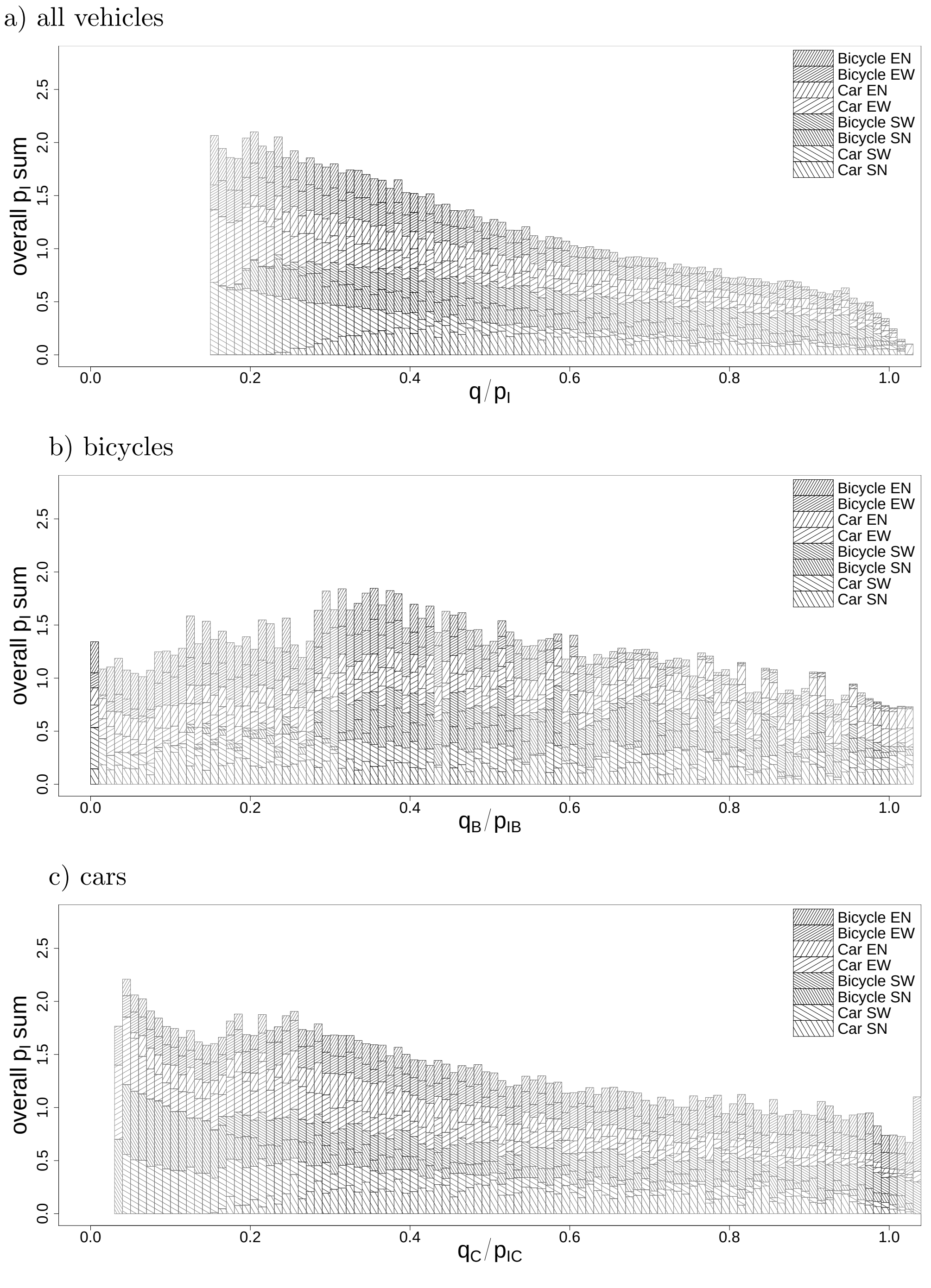
In order to create a single view of a set of inter-dependent flows, we have devised the *realisation diagram*, which provides a visual summary of the flows and their interactions, including useful individual flow information. The derivation of this type of diagram has already been detailed in Section 5.1.4. Here it is applied to the case of an intersection of two one-way roads with single lanes. The realisation diagrams for the two conflict resolution schemes are shown in Figures 5.16 and 5.17. The former is based on simulations with the RHS (right-hand-size) priority rule applied and the latter with the LHS (left-hand-side) rule applied. A general pattern of lower inputs resulting in better *realisations* holds, unsurprisingly, for all the diagrams. Also, when analysed visually, the diagrams show that, on average, bicycles fare better with the RHS rule, in that their flow realisation is higher than that for cars, while with the LHS rule the two types of vehicle have more similar average realisations. This can be concluded from the fact that the darker areas for bicycles in RHS rule simulations (Figure 5.16b) appear at higher realisations than for cars (Figure 5.16c) and at similar realisations (Figure 5.17b and c) in the case of LHS rule simulations. Considering average composition of the traffic for low realisation values, we can identify flows or combinations of flows causing these. For example, the lowest realisation values for cars in the RHS rule case (Figure


a) all vehicles

b) bicycles

c) cars

Figure 5.16: Realisation diagram for intersection of two one-way roads with RHS priority rule: This diagram shows the overall system in-flow values for (a) all vehicles, (b) bicycles and (c) cars, as a function of realisation. The in-flow is displayed in sections pertaining to individual tracks. The RHS rule stipulates that that the east-west in-flowing stream has priority over the south-north one).



a) all vehicles

b) bicycles

c) cars

Figure 5.17: Realisation diagram for intersection of two one-way roads with LHS priority rule: This diagram shows the overall system in-flow values for (a) all vehicles, (b) bicycles and (c) cars, as a function of realisation. The in-flow is displayed in sections pertaining to individual tracks. The LHS rule stipulates that the east-west in-flowing stream has priority over the south-north one).

5.16c) occur at high bicycle inputs and dominant south-to-west car flows. This indicates that south-to-west car flows are impeded by bicycles more than other car flows at the intersection. In the RHS rule case, south-to-west and south-to-north car flows are largely constant across all bicycle realisation values (Figure 5.16b), showing that bicycles are not affected by the low priority cars in this case. In the LHS rule case, SW car flows have a big impact on the overall and car flow realisation (Figure 5.17c) but not on that for bicycles (Figure 5.17b). An interesting piece of information that can be gleaned from the diagrams is that the very low realisations for cars and bicycles do not coincide in any of the simulation instances, as the diagrams in Figures 5.16a and 5.17a do not contain any realisation values below 0.14.

5.6 Intersection of two two-way roads

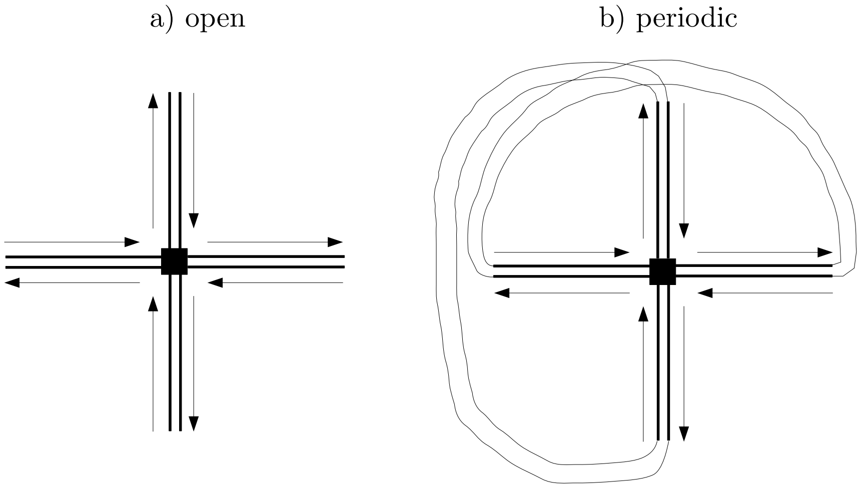


Figure 5.18: Intersection of 2 2-way roads: schematic representation of scenarios with open and periodic boundary conditions. Periodic boundary conditions are achieved by linking the road stretches connected out of the intersection with into road stretches connected into the intersection.

Flow through an intersection of two two-way roads with a single lane in each direction (represented by the spatial model shown in Figure 3.10) has been simulated for each of the four intersection configurations described in Section 3.1.1 and in each case for open and periodic boundary conditions. Schematic representation of open and closed boundary conditions is shown in Figure 5.18 and the parameters used in the

Parameter	Value
Configuration elements	intersection of two two-way roads (I1) and 8 road stretch elements (RS1 to RS8)
Boundary conditions	open, periodic
Connections (see Figure 4.4 for port numbers)	RS1:1 → I1:0 I0:0 → RS2:0 RS3:1 → I1:1 I1:1 → RS4:0 RS5:1 → I1:2 I1:2 → RS6:0 RS7:1 → I1:3 I1:3 → RS8:0 Only with periodic boundaries: RS2:1 → RS5:0 RS4:1 → RS7:0 RS6:1 → RS1:0 RS8:1 → RS3:0
Simulation length per distinct parameter set	10^5 time-steps (corresponds to about 28 hours)
Randomisation parameter	0.1
Routing	macroscopic, with turning probability equal for all directions at all divergences (0.33)
Road length	50 car cells per road stretch
Car density (for periodic boundaries)	$0.0 \leq \rho_{\mathrm{C}} \leq 0.5$, with step 0.05 $0.6 \leq \rho_{\mathrm{C}} \leq 1.0$, with step 0.1
Car insert probability (for open boundaries)	$0.0 \leq p_{\mathrm{IC}} \leq 0.5$, with step 0.05 $0.6 \leq p_{\mathrm{IC}} \leq 1.0$, with step 0.1
Bicycle density (for periodic boundaries)	$0.0 \leq \rho_{\mathrm{B}} \leq 0.5$, with step 0.05 $0.6 \leq \rho_{\mathrm{B}} \leq 1.0$, with step 0.1
Bicycle insert probability (for open boundaries)	$0.0 \leq p_{\mathrm{IB}} \leq 0.5$, with step 0.05 $0.6 \leq p_{\mathrm{IB}} \leq 1.0$, with step 0.1
Accepted gap	1

Table 5.5: Intersection of two two-way roads simulation parameters. Each combination of the listed values represents a distinct simulation instance.

simulations are shown in Table 5.5. The sum of all flows through the intersection for all these scenarios is shown in Figures 5.19 and 5.20, respectively. For open boundary conditions (Figure 5.19) the difference in the flows between schemes is only quantitative and the expected order of growing flow values between schemes, (1) basic (track-based), (2) one-view, (3) cell-less and (4) phantom intersection configuration, is evident. However, the flows for the phantom scheme are dramatically higher than for the other cases, owing to conflicts being disregarded. The graphs for periodic boundary conditions are of two types: in the basic and one-view case, shown in Figures 5.20a and b, there is no flow at all above a certain threshold probability of insertion. In the cell-less and phantom cases, shown in Figures 5.20c and d, curves for bicycles and cars are more

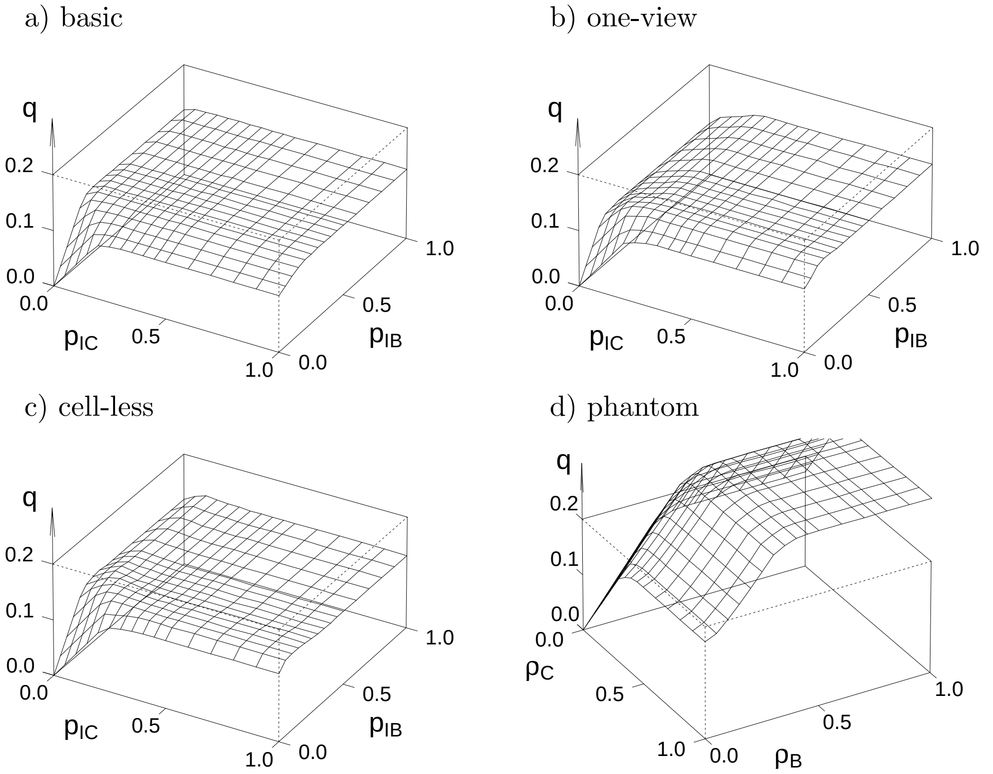


Figure 5.19: Intersection of two two-way roads: flows for different intersection configurations when boundary conditions are open.

or less constant across the range of other vehicle type input values, indicating mutual independence. All cases display the network flow break-down phenomenon described in Daganzo et al. (2011), whereby flows reduce to 0 at a much lower density than the jamming density of the road stretch fundamental diagram (k_J in Figure 2.1), which is equal to a dimensionless 1 for cellular automata traffic models. From this point of view, a single intersection connected back into itself behaves in the same way as a network. However, in the basic and one-view cases the emulated network flows can be thought of as consisting of nodes that are each connected by 4 track flows, a bicycle and car track in each direction. On the other hand, the cell-less and phantom cases operate more like two separate networks, with either two bicycle or two car flows for their edges.

For a more detailed quantification of the impact of intersection flows on each other, we produced the *effects sequence*, described in Section 5.1.5, for the intersection of two two-way roads. The effects sequence diagram for the sum of flows in the intersection of two two-way single-lane roads is shown in Figure 5.21. Here the negative effects on

	Track flow combination	**Negative effect value**	**Track flow combination**	**Negative effect value**
1	CNA, BWR	0.528544	CSA, CER, CNL, CWA	-0.441418
2	CSA, BER	0.515696	CSA, BSR, CER, BEL, BNA, BWR	-0.439236
3	CWL, BWA	0.510743	CSL, CEA, CNA, CWR	-0.427236
4	CEL, BEA	0.504282	BSR, BEA, CNA,	-0.403332
5	BSR, CNA	0.496673	BSR, BER, CNA,	-0.403114
6	CNA, BWA	0.492052	BER, CNA, BWA,	-0.401109
7	CSA, BNR	0.489849	CSL, CER, CNR, CWL	-0.399398
8	CSA, BEA	0.48033	BSA, BEA, BWR,	-0.398384
9	CSR, CNL	0.474967	BER, BNA, BWA,	-0.397722
10	CSL, CNR	0.474635	CSR, CEL, CNL, CWR	-0.397168

Table 5.6: Intersection of 2 2-way single-lane roads: track flow combinations that produce highest absolute effect values.

	Track flow combination	**Negative effect value**
1	CNL CWA	0.473662
2	CSA CER	0.354279
3	CER CWA	0.277073
4	CSA CWA	0.20733
5	CSA CNL CWA	0.161109
6	CER CNL CWA	0.146153
7	CSA CER CNL	0.0114933
8	CNL	0
9	CER	0
10	CSA	0
11	CWA	0
12	CER CNL	-0.00129203
13	CSA CER CWA	-0.0101061
14	CSA CNL	-0.0175178

Table 5.7: Effects values for track flow sub-combinations of tracks CSA, CER, CNL and CWA in the intersection of two two-way single-lane roads. The list is in order of decreasing effect values and the effect values are calculated as explained in Section 5.1.5.

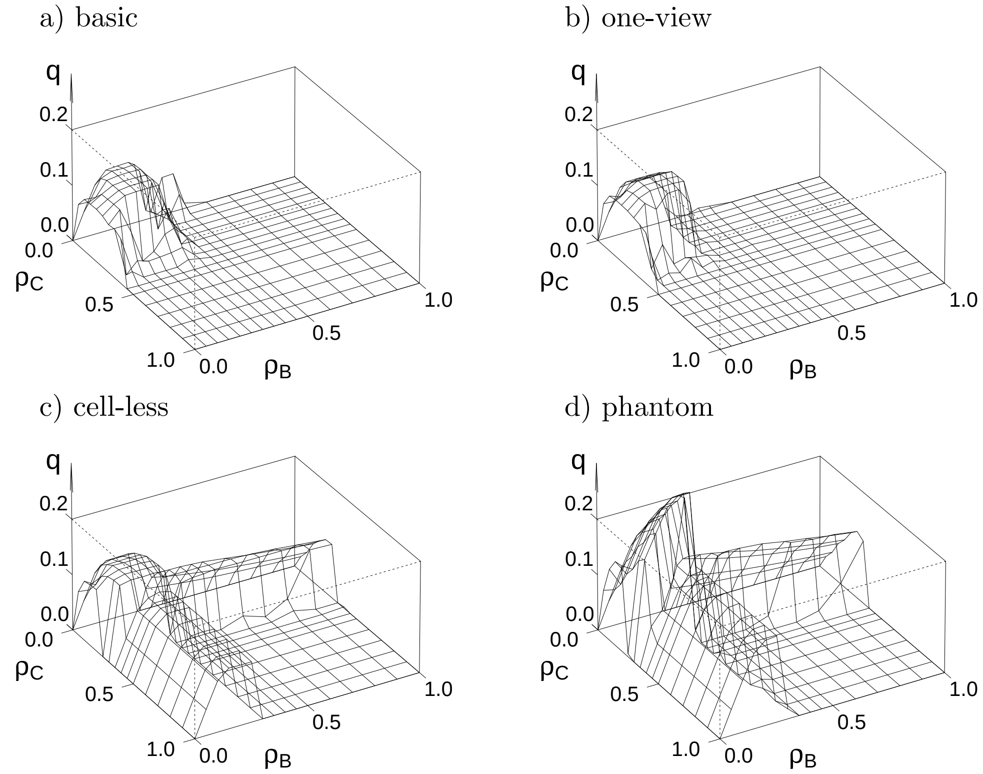


Figure 5.20: Intersection of two two-way roads: flows for different intersection configurations when boundary conditions are periodic.

overall flow in the intersection are shown in sequence, from the largest to the smallest. The x-axis represents the ordinal numbers of the sorted effects. All that can be seen from the diagram is that most of the track flow combinations have no impact on the overall flow while a small number of combinations have a significant impact. The combinations with the 10 highest negative and 10 highest positive effect values are shown in Table 5.6. The greatest negative effects are for conflicts between cars and bicycles, in most cases with bicycles yielding (1, 2, 5, 6, 7 and 8) but also for same-lane conflicts in a low-priority lane (3, 4). The top car-car conflict with regards to negative effect on flow is that between a right turning car track and the left turning car track in the opposite direction, both on the higher-priority road. The positive effects come from the cancelling out of negative effects caused by sub-groups of tracks (Section 5.1.5). For example the highest positive effect comes from the combination of the CSA, CER, CNL and CWA track flows. The effects of the sub-combinations of this combination are shown in Table 5.7. The four tracks of the combination (CSA, CER,

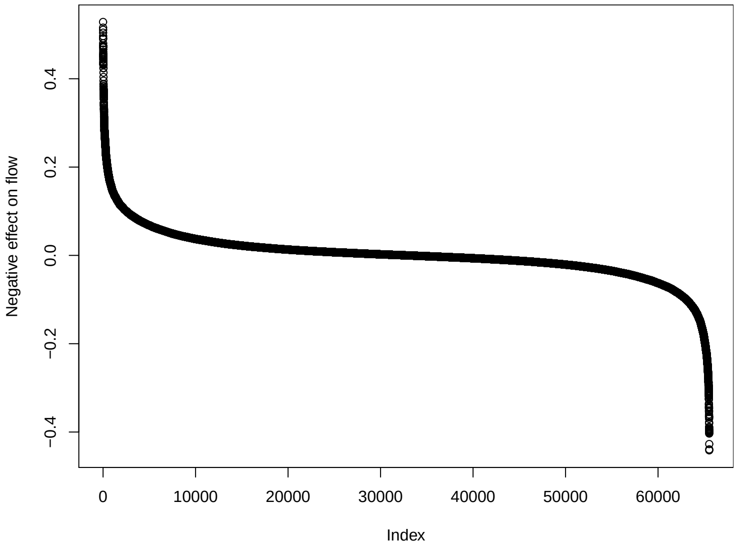


Figure 5.21: Intersection of 2 2-way roads: effects sequence diagram. The x-axis represents the ordinal number in the sequence, sorted by decreasing negative effect on overall intersection flow.

CNL and CWA) form four pairs with high negative effects (positions 1, 2, 3 and 4 in the table) and two groups of three also with high negative effects (positions 5 and 6 in the table). The other effects (apart from the single track-flow ones in positions 8, 9, 10 and 11 in the table, which are 0 by definition) are due to the stochastic simulations as they consist of non-interacting track flows (e.g. the combination CER and CNL consists of two non-overapping car tracks, which do not have an effect on each other). This ranking of influence could be useful in the design of intersections, controls and priority rules.

5.7 Networks

The topological elements investigated in the preceding sections of this chapter can be combined (with numerous variations in type and number) to form a network. The simulations that have been performed with such networks are presented here, starting with a simple grid of one-way single-lane road intersections, in Section 5.7.1. The grid of intersections of one-way single-lane roads is not as realistic with respect to topology but is easier to simulate, given that it does not involve deadlocks, and is a good starting point for studying how bicycle-car interaction effects translate from a single intersection to a grid. Grids with full intersections, i.e. intersections including all directions of movement, require deadlock-related investigation and tuning. This is presented in Section 5.7.2. *Irregular networks* including a variety of different intersections, such as one-way, two way, left and right turn, and with road stretches of different sizes, have not been simulated so far, although the simulation program can be configured to do this.

5.7.1 Grid of 4x4 one-way single-lane road intersections

A grid of 4×4 nodes was configured cor simulation, as shown in Figure 5.22. The simulation parameters are given in Table 5.8. As can be seen from this table, a number of turning schemes have been simulated: one where each vehicle turns exactly once and the turning probabilities are adjusted so that the number of vehicles turning at all nodes is approximately the same (equalised turn-load), one where the 'turning load' is maximised (each vehicle turns only once, at one of the 4 designated, high load, nodes in the grid) and one where the probability of going in any direction at any divergence is equal to any other such probability. These are extreme utilisation patterns which could ocurr in real traffic based on different demand distributions or invoked artificially - they are used here to illustrate a study of network performance under different conditions. The macroscopic densities and flows are presented in the form of fundamental diagrams for RHS and LHS conflict resolution schemes (see Section 5.5), in Figures 5.23 and 5.24, respectively. Because open boundary conditions were used, the plots are all in the free-flow section of the fundamental diagram.

From the diagrams, which are based on density and flow for all vehicles in the system (bicycles and cars), it can be seen that the length of the road stretches in the grid have no bearing on the throughput of the network, since there is no quantitative difference between the diagrams for lengths of 50 car cells and 200 car cells. The diagrams for the *no turning* routing scheme (Figures 5.23a and e and 5.24a and e) all have some outlying curves displaying higher velocities than the rest of the diagram (a steeper line means higher velocity since $v = q/\rho$) - these are the points for the cases where

Parameter	**Value**
Configuration elements	16 intersections of two one-way single-lane roads and 40 road stretch elements
Boundary conditions	open
Connections	each intersection is connected to 4 road stretches, as in Figure 5.22
Simulation length per distinct parameter set	25000 time-steps (corresponds to about 7 hours)
Randomisation parameter	0.1
Routing	four macroscopic schemes: (1) no turning (2) equal probability, (3) maximised turn-load and (4) equalised turn load
Road length	50 and 200 car cells per road stretch
Car insert probability	$0.0 \leq p_{\mathrm{IC}} \leq 0.7$, with step 0.1
Bicycle insert probability	$0.0 \leq p_{\mathrm{IB}} \leq 0.7$, with step 0.1
Accepted gap	1
Netowork size	4×4

Table 5.8: Grid of 4x4 one-way single lane roads: simulation parameters. Each combination of the listed values represents a distinct simulation instance.

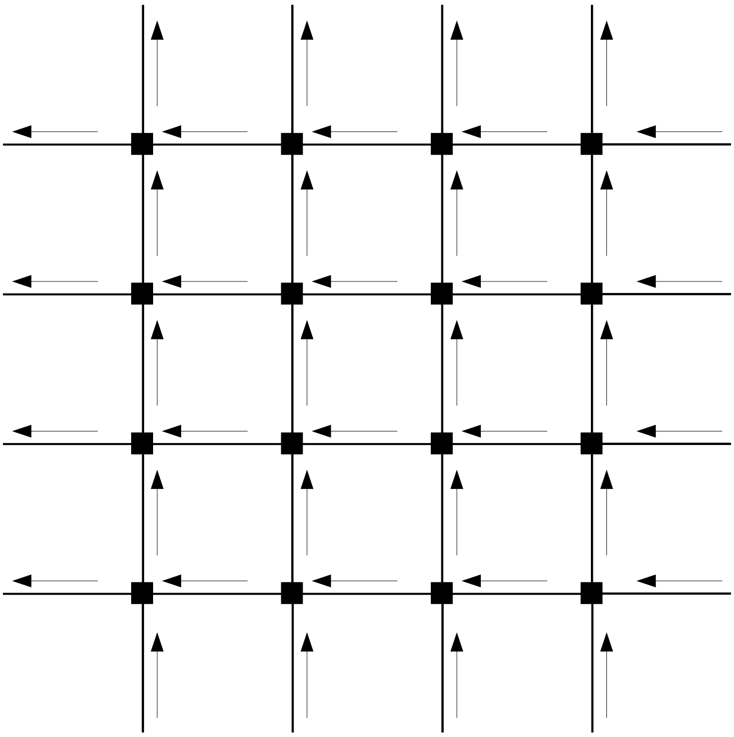
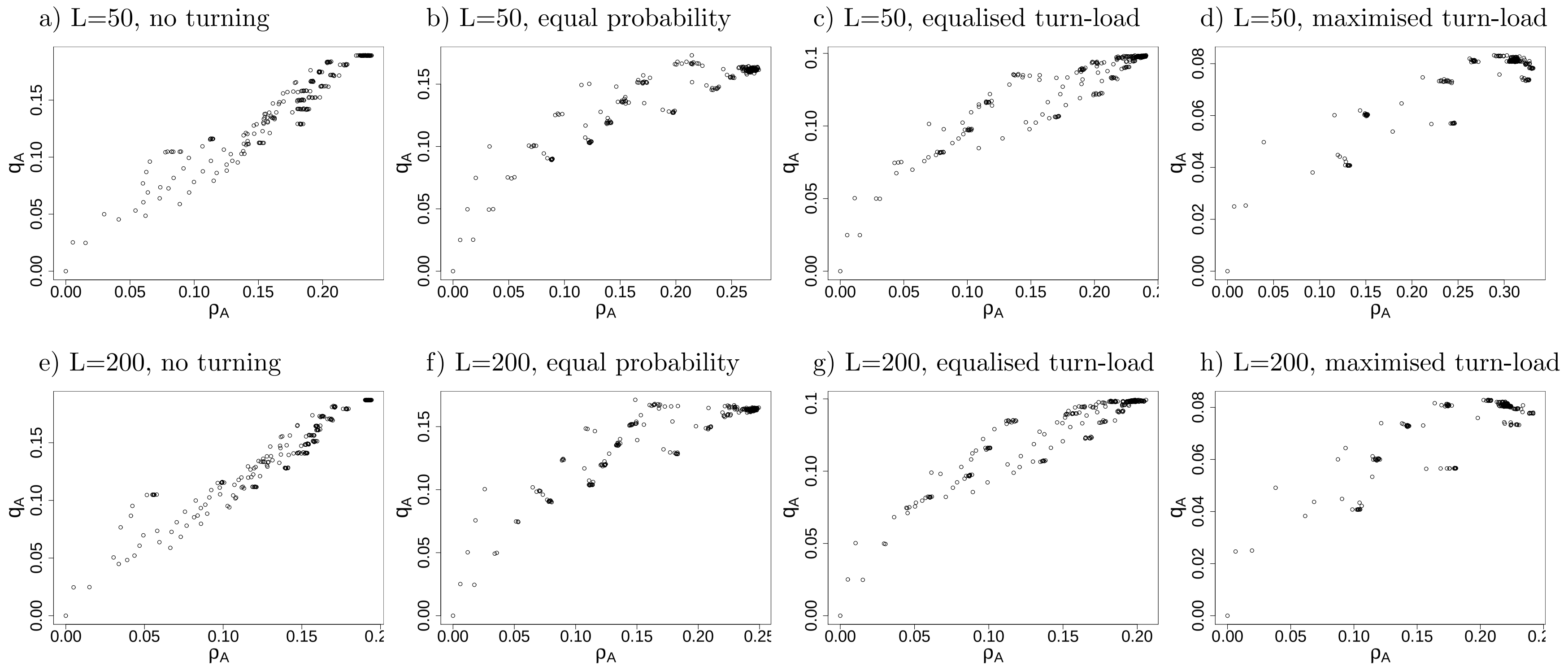


Figure 5.22: Grid of 4x4 one-way single-lane road intersections: scenario schema

the input for one of the vehicle types is zero. The macroscopic routing approaches differ in the maximal flow they produce. The best performance is obtained for the *no*



a) L=50, no turning

b) L=50, equal probability

c) L=50, equalised turn-load

d) L=50, maximised turn-load

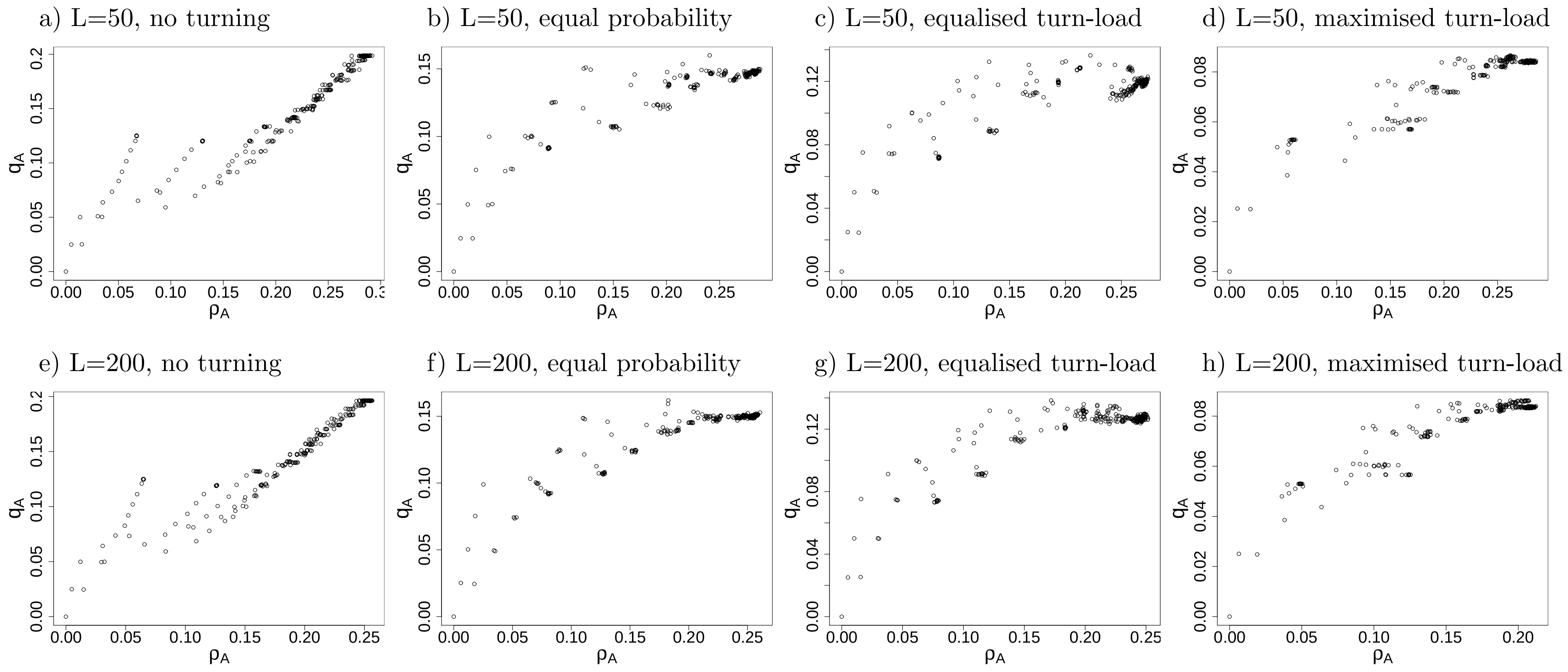
e) L=200, no turning

f) L=200, equal probability

g) L=200, equalised turn-load

h) L=200, maximised turn-load

Figure 5.23: Grid of 4x4 one-way single-lane road intersections: fundamental diagrams for RHS conflict resolution rule. The length, L (in car cells), and the macroscopic routing scheme are indicated for each sub-figure.



a) L=50, no turning

b) L=50, equal probability

c) L=50, equalised turn-load

d) L=50, maximised turn-load

e) L=200, no turning

f) L=200, equal probability

g) L=200, equalised turn-load

h) L=200, maximised turn-load

Figure 5.24: Grid of 4x4 one-way single-lane road intersections: fundamental diagrams for LHS conflict resolution rule. The length, L (in car cells), and the macroscopic routing scheme are indicated for each sub-figure.

turning routing pattern, followed by that for *equal probability* and *equalised turn-load*, with *maximised turn-load*, as might be expected, performing worst. Also, the LHS rule displays better maximal flows than the RHS rule, particularly in the *no turning* and *equalised turn load* cases. This can be explained by the fact that in these cases, which are dominated by intersection interactions without turning, with the RHS rule the low-priority flows have a better chance of getting into and through the intersection, disrupting the high priority flow. The reason for this is that with the RHS rule the conflict that is 'first in line' is that between low priority cars and high priority bicycles, and in which the high priority vehicles are slower, giving better opportunity to the low priority vehicles to enter the conflict. With the LHS rule it is the opposite. In the *equal probability* and *maximised turn load* cases this effect is not pronounced, because of dominating turning interactions.

5.7.2 Grid of 2x2 two-way two-lane road intersections

The grid of two-way two-lane road intersections is the most complex of the configurations studied in this thesis. For the edges in the grid of two-way two-lane roads we have used road stretches with entrances and exits, which allow entry and exit from any point in the network, not only the network boundaries (i.e. the places where the network is 'cut out' of a larger network). Given that the two-way grid carries with it the problem of inter-node deadlocks (see Figure 3.4b), tests were performed with a minimal network for the different configurations for this type of intersection, described in Section 3.1.1. The simulation scenario parameters are shown in Table 5.9 and the schematic representation of its spatial aspect is shown in Figure 5.25.

Different intersection configurations were applied to both the *intersection of 2-way 2-lane roads* and to the entrance-exit areas within the *road stretch with entrances/exits* topological element. Simulation of all the different combinations of intersection configurations has resulted in three patterns in *input-flow* diagrams, shown in Figure 5.26. Each pattern appears for more than one combination of intersection configurations, which are listed to the right of the applicable diagrams (the one-view intersection configuration is not applied to the road stretch with entrances/exits). The diagrams

Parameter	Value
Network size	2×2
Configuration elements	4 intersections of two two-way single-lane roads and 12 two-way road stretch elements with entrances and exits 8 one-way road stretch 'insertion' elements (see Figure 5.25)
Boundary conditions	open
Connections	as in Figure 5.25
Simulation length per distinct parameter set	10^5 time-steps (corresponds to about 28 hours)
Randomisation parameter	0.1
Routing	zone-based, with zones probability: 0 (outside of network): 0.5 1 (entrances in road stretches): 0.5
Road length	10 car cells (5 entrances/exits) per road stretch
Car insert probability	external (through 'plain' road-stretches in Figure 5.25): $0.0 \leq p_{\mathrm{IC}} \leq 0.7$, with step 0.1 internal (through road stretches with entrances and exits in Figure 5.25): $0.0 \leq p_{\mathrm{IC}} \leq 0.02$, with step 0.005
Bicycle insert probability	external (through 'plain' road-stretches in Figure 5.25): $0.0 \leq p_{\mathrm{IB}} \leq 0.7$, with step 0.1 internal (through road stretches with entrances and exits in Figure 5.25): $0.0 \leq p_{\mathrm{IB}} \leq 0.02$, with step 0.005
Accepted gap	1
intersection configuration	track-based, one-view, cell-less and phantom for intersections; track-based, cell-less and phantom for road stretches

Table 5.9: Grid of 2x2 two-way two-lane road intersections: simulation parameters. Each combination of the listed values represents a distinct simulation instance.

show that for the track-based intersection configuration applied to road stretch with entrances/exits flow is zero for all cases except extremely small input values. This is caused by the very high probability of deadlock everywhere in the network, even along the road stretches, owing to the entrances and exits that form mini-intersections. The second group of intersection configuration combinations is that where the road stretch with entrances/exits uses the cell-less or phantom scheme, while the intersection uses the track-based or one-view scheme. The input-flow diagram displayed by these combinations has flow at 'normal' levels for small inputs, but that falls to 0 when inputs increase over 0.2. Finally, any combination of cell-less and phantom intersection configurations results in an input-flow diagram with self-limiting flows for bicycles and cars

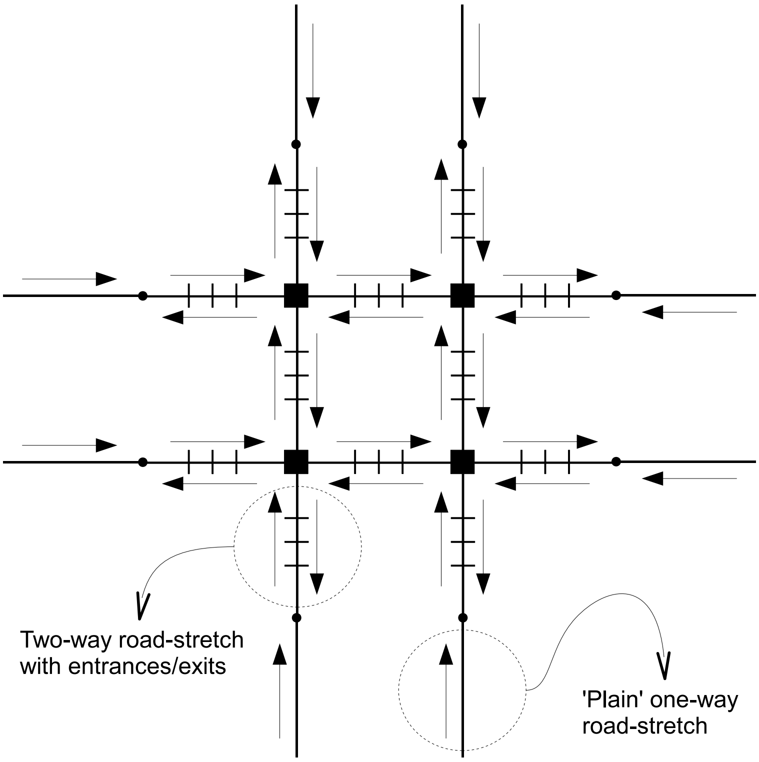


Figure 5.25: Grid of 2x2 two-way two-lane road intersections: scenario schema

that are mostly independent of each-other. The two latter patterns are equivalent to those shown for the same two-way intersection under periodic boundary conditions, in Figure 5.20.

Figure 5.27 contains the fundamental diagrams for all vehicles and for bicycles and cars separately, for the phantom/phantom conflict modelling approach. The input-flow diagram for this case is the most permissive in Figure 5.26 (sub-figure c). Both cars and bicycles (Figure 5.27a and b have fundamental diagrams with a congested slope arriving at flow of 0 at densities lower than 1, which corresponds to the bifurcation shape predicted for macroscopic fundamental diagrams by (Daganzo et al., 2011). The authors also identify 'faster loading' as causing bifurcation at lower densities. Since more bicycles fit into the same length of road than cars, they would take longer to fill up the same amount of space as cars and hence 'load' more slowly. There is also a difference between slower and faster loading bicycles (flows entering through network boundaries vs. flows using entrances alongside roads). The FD for all vehicles suggests

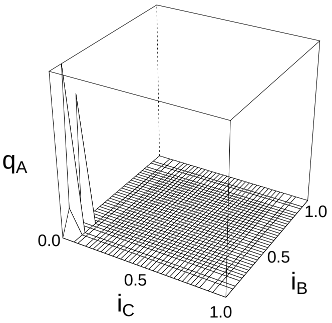
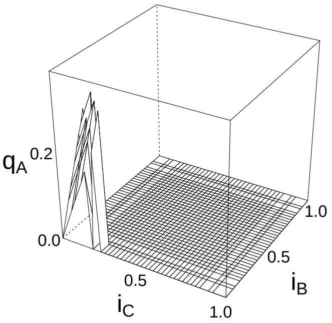
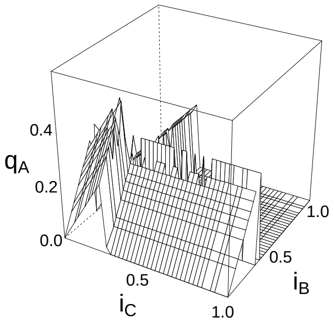
Input-flow diagram	**Conflict modelling schemes for: intersection/ road stretch with entrances and exits**
	phantom/track-based cell-less/track-based one-view/track-based track-based/track-based
	track-based/cell-less one-view/cell-less track-based/phantom one-view/phantom
	cell-less/cell-less cell-less/phantom phantom/cell-less phantom/phantom

Figure 5.26: Grid of 2x2 two-way two-lane road intersections: the three patterns of input-flow diagram. The conflict modelling combinations for intersection and road stretch with entrances/exits that exhibit a pattern are all listed to the right of the corresponding diagram. The flow is the average flow in the system. Axes are: q_{A} - overall average flow per external exit in the network; $i_{\mathrm{B}}, i_{\mathrm{C}}$ - car and bicycle input per external entrance, respectively.

independence between bicycles and car networks (which they are since the phantom intersection configuration is in effect here).

Figure 5.28 contains fundamental diagrams for the same small 'network' as Figure 5.27. Here flows fall to zero at much lower densities than with the *phantom-phantom* intersection configuration. The MFD for all vehicles combined does not display vehicle independence.

A deadlock avoidance approach was implemented for the one-view intersection, whereby vehicles would not enter if there are any other vehicles already in the intersection travelling towards the same road-stretch track beyond the intersection. This prevents vehicles from getting 'stuck' in the intersection. The macroscopic fundamental diagrams for this case are shown in Figure 5.29 and what is seen from the diagrams is that the effect of the measure is more vehicles entering the network and getting sturk, without an improvement in the flows

5.7.3 Grid of 11x11 two-way two-lane road intersections

Some initial work has been done on larger network simulations and application of traffic lights. This section describes simulations on a network of 11 × 11 two-way two-lane road intersections. The cell-less/phantom combination of intersection configurations was used for intersection and road stretch with entrances/exits, as representative of the functioning group in Figure 5.26. Simulation parameters for the 11 × 11 node network are listed in Table 5.10, which complements Table 5.9. The input-flow diagrams for this scenario are shown in Figure 5.30. The input at which deadlocks 'kill' the flow is slightly lower for cars than for bicycles. It is also lower for the shorter road length of 10 than for the road length of L=40.

A test of **physical separation between bicycles and cars** was performed on a very similar network, but with 12 × 12 nodes (an even number in the column and row was required). In this scenario, only road stretches in odd columns and rows could be used by bicycles and only road stretches in even columns and rows could be used by cars. Exceptions were made for access, i.e. a bicycle or car could enter a 'forbidden' road stretch if its destination, determined by the zone routing algorithm, was in that

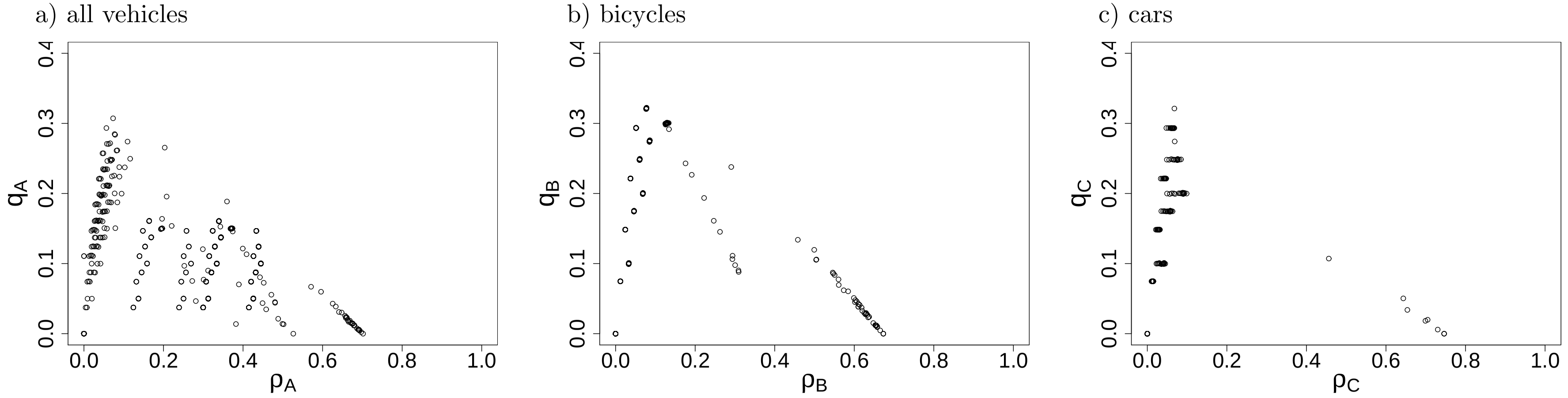


Figure 5.27: Grid 2×2 of two-way two-lane road intersections: macroscopic fundamental diagrams for phantom intersection configuration applied to both intersections and road stretches with entrances/exits. Axes are: $q_{\mathrm{A}}, q_{\mathrm{B}}, q_{\mathrm{C}}$ - overall, bicycle and car average flow per external exit in the network, respectively; $\rho_{\mathrm{A}}, \rho_{\mathrm{B}}, \rho_{\mathrm{C}}$ - system-wide density for all vehicles, bicycles and cars, respectively.

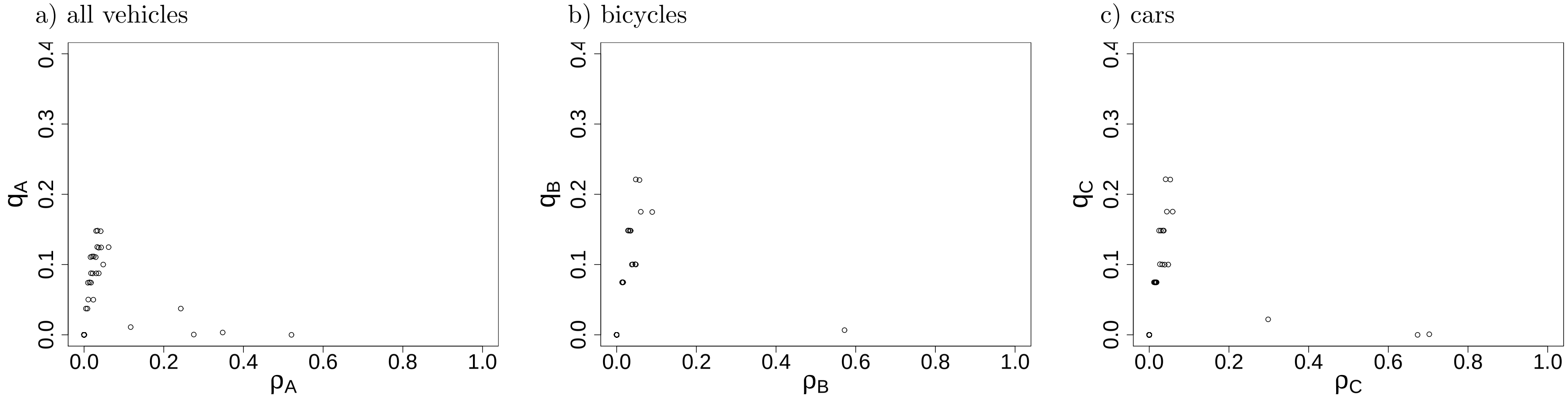
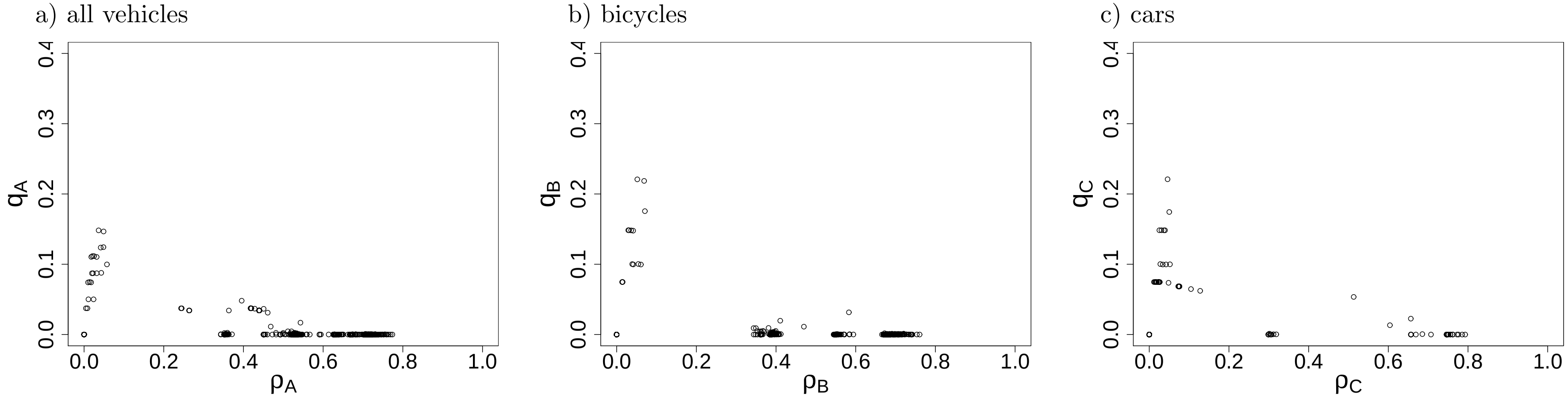


Figure 5.28: Grid 2×2 of two-way two-lane road intersections: macroscopic fundamental diagrams for one-view intersection configuration applied to intersections, phantom applied to road stretches with entrances/exits and with deadlock relief extensions to the simulation. Axes are: $q_{\mathrm{A}}, q_{\mathrm{B}}, q_{\mathrm{C}}$ - overall, bicycle and car average flow per external exit in the network, respectively; $\rho_{\mathrm{A}}, \rho_{\mathrm{B}}, \rho_{\mathrm{C}}$ - system-wide density for all vehicles, bicycles and cars, respectively.


a) all vehicles

b) bicycles

c) cars

Figure 5.29: Grid 2 × 2 of two-way two-lane road intersections: macroscopic fundamental diagrams for one-view intersection configuration applied to intersections and phantom applied to road stretches with entrances/exits. Axes are: $q_{\mathrm{A}}, q_{\mathrm{B}}, q_{\mathrm{C}}$ - overall, bicycle and car average flow per external exit in the network, respectively; $\rho_{\mathrm{A}}, \rho_{\mathrm{B}}, \rho_{\mathrm{C}}$ - system-wide density for all vehicles, bicycles and cars, respectively.

Parameter	**Value**
Network size	11×11
Configuration elements	11×11 intersections etc. (as in Figure 5.25, but scaled)
Car insert probability	external: $0.0 \leq p_{\mathrm{IC}} \leq 0.6$, with step 0.1 internal: $p_{\mathrm{IC}} \in \{0.0, 0.00005, 0.00025, 0.0005, 0.001\}$
Bicycle insert probability	external: $0.0 \leq p_{\mathrm{IB}} \leq 0.6$, with step 0.1 internal: $p_{\mathrm{IB}} \in \{0.0, 0.00005, 0.00025, 0.0005, 0.001\}$
Road length	10 car cells (5 entrances/exits) per road stretch, 40 car cells (80 entrances/exits) per road stretch
Accepted gap	1, 1 except 2 for bicycles w.r.t. cars
intersection configuration	cell-less for intersections phantom for road stretches

Table 5.10: Grid of 11x11 two-way two-lane road intersections: simulation parameters. This table only mentions parameters that are different from Table 5.9. Each combination of the listed values represents a distinct simulation instance.

road stretch. Results of the simulations for the scenario with and without physical separation, where both were based on the exact same inputs, are shown in Figure 5.31. Parameters for these simulations are shown in Figure 5.11. The case with separation shows considerably lower flows than the one without, the effect of having to ‘go around’ being dominant over any speed-up by cars stemming from the separation from bicycles.

Finally, a set of simulations with **zone-based routing** was performed on the 12x12 network used in the previous experiment. Here the system was set-up with three zones for both origin and destination. Zone-routing based simulations described so far have had two zones (external and internal) defined for both origins and destinations and the origin zones were identical to the destination ones in terms of spatial coverage. These final network simulations were configured with different origin and destination zones. Figure 5.32 shows the zone seeds and zones (of width 20 car cells) used in the scenarios. Table 5.12 shows the parameters for this group of simulations. Input-flow diagrams for this case are shown in Figure 5.33. The maximal flows do not seem to differ between scenarios. The results do indicate that even with relatively small zones, flow easily becomes saturated.

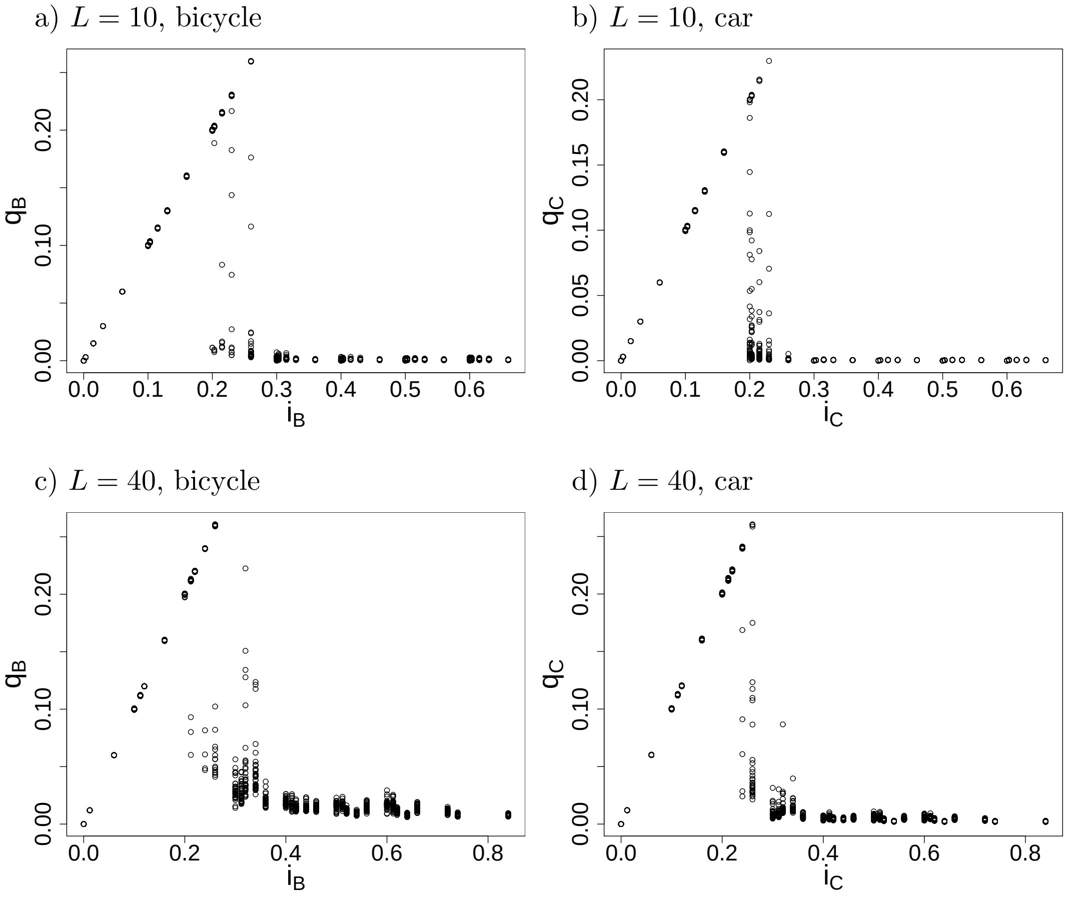


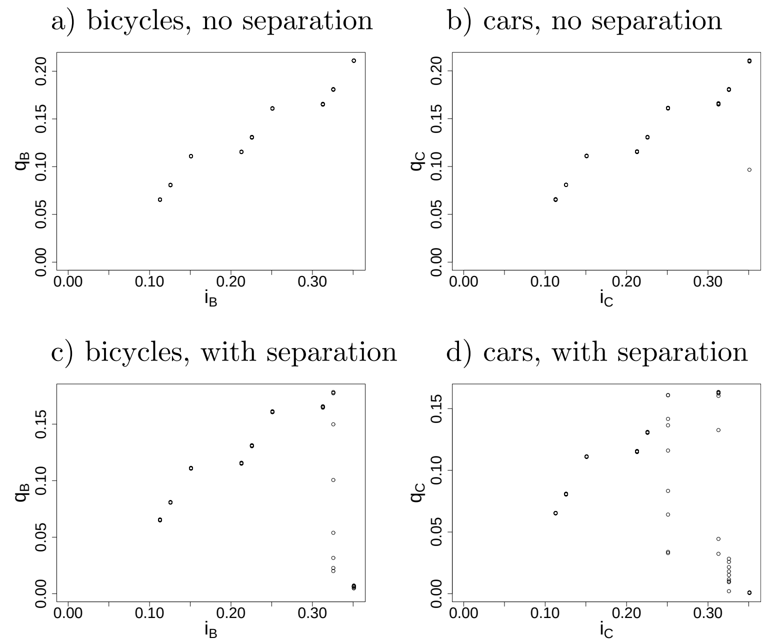
Figure 5.30: Grid of 11x11 two-way two-lane road intersections: input-flow diagrams. L, is the length of the road stretches with entrances/exits, in car cells. All diagrams are for an accepted gap of 1. Axes are: $q_{\mathrm{B}}, q_{\mathrm{C}}$ - bicycle and car average flow per external exit in the network, respectively; $i_{\mathrm{B}}, i_{\mathrm{C}}$ - input per external entrance, for bicycles and cars, respectively.

5.7.4 Traffic lights

Control was not investigated as part of the project to date, except for one case, as a proof of concept for the implementation. The scenario used is the same as that described in Section 5.7.1, the 4 × 4 grid of one-way single lane road intersections. The simulation parameters listed in Table 5.8 were used, except where indicated in the description of the particular scenario. The main modification to that configuration is the use of traffic lights. A number of scenarios were simulated. Their description can be found in Table 5.13 and the input-flow diagrams in Figures 5.34 (for bicycles) and 5.35 (for cars). Both for the case where vehicles turn and that where they do not turn,

Parameter	**Value**
Network size	12 × 12
Configuration elements	12 × 12 intersections etc. (as in Figure 5.25, but scaled)
Car insert probability	external, only on even columns and rows of grid: $0.0 \leq p_{\mathrm{IC}} \leq 0.3$, with step 0.1 internal: $p_{\mathrm{IC}} \in \{0.00111, 0.00056, 0.00028\}$
Bicycle insert probability	external, only on odd columns and rows of grid: $0.0 \leq p_{\mathrm{IB}} \leq 0.3$, with step 0.1 internal: $p_{\mathrm{IB}} \in \{0.00111, 0.00056, 0.00028\}$
Road length	10 car cells (5 entrances/exits) per road stretch,
Accepted gap	1
intersection configuration	cell-less for intersections phantom for road stretches
Scenario specific	cars are only allowed on even columns and rows of the grid and bicycles on odd ones, except for access

Table 5.11: Grid of 12x12 two-way two-lane road intersections: parameters for simulations with physical separation of vehicle types. This table only mentions parameters that are different from Table 5.9. Each combination of the listed values represents a distinct simulation instance.


a) bicycles, no separation

b) cars, no separation

c) bicycles, with separation

d) cars, with separation

Figure 5.31: Grid of 12x12 two-way two-lane road intersections: input-flow diagrams for simulations without and with physical separation of vehicle types. Axes are: $q_{\mathrm{B}}, q_{\mathrm{C}}$ - bicycle and car average flow per external exit in the network, respectively; $i_{\mathrm{B}}, i_{\mathrm{C}}$ - input per external entrance to the network, for bicycles and cars, respectively.

Parameter	**Value**
Network size	12×12
Configuration elements	12×12 intersections etc. (as in Figure 5.25, but scaled)
Car insert probability	external: none internal, zone 1: $p_{\text{IC}} \in \{0.3, 0.2, 0.1, 0.07, 0.05, 0.0266,$ $0.0133, 0.0067, 0.0033, 0.0017, 0.0011, 0.0008\}$ internal, zone 2: none
Bicycle insert probability	external: none internal, zone 1: $p_{\text{IB}} \in \{0.3, 0.2, 0.1, 0.07, 0.05, 0.0266,$ $0.0133, 0.0067, 0.0033, 0.0017, 0.0011, 0.0008\}$ internal, zone 2: none
Road length	10 car cells (5 entrances/exits) per road stretch,
Accepted gap	1,
intersection configuration	cell-less for intersections phantom for road stretches
Routing	all car destinations are in car destination zone 1 all bicycle destinations are in bicycle destination zone 1
Zone definition	5 scenarios are simulated, each with zones 1 and 2 for both cars and bicycles, for both origin and destination; zone 1 is defined for the 5 scenarios as follows: 1) cars-origin: N/A bicycles-origin: zone around seed D in Figure 5.32 cars-destination: N/A bicycles-destination: zone around seed B in same figure 2) cars-origin: zone around seed D in Figure 5.32 bicycles-origin: N/A cars-destination: zone around seed B in same figure bicycles-destination: N/A 3) cars-origin: zone around seed D in Figure 5.32 bicycles-origin: zone around seed D in same figure cars-destination: zone around seed B in same figure bicycles-destination: zone around seed B in same figure 4) cars-origin: zone around seed D in Figure 5.32 bicycles-origin: zone around seed B in same figure cars-destination: zone around seed B in same figure bicycles-destination: zone around seed D in same figure 5) cars-origin: zone around seed D in Figure 5.32 bicycles-origin: zone around seed A in same figure cars-destination: zone around seed B in same figure bicycles-destination: zone around seed C in same figure

Table 5.12: Grid of 12x12 two-way two-lane road intersections: parameters for simulations with small origin and destination zones. This table only mentions parameters that are different from Table 5.9. Each combination of the listed values represents a distinct simulation instance.

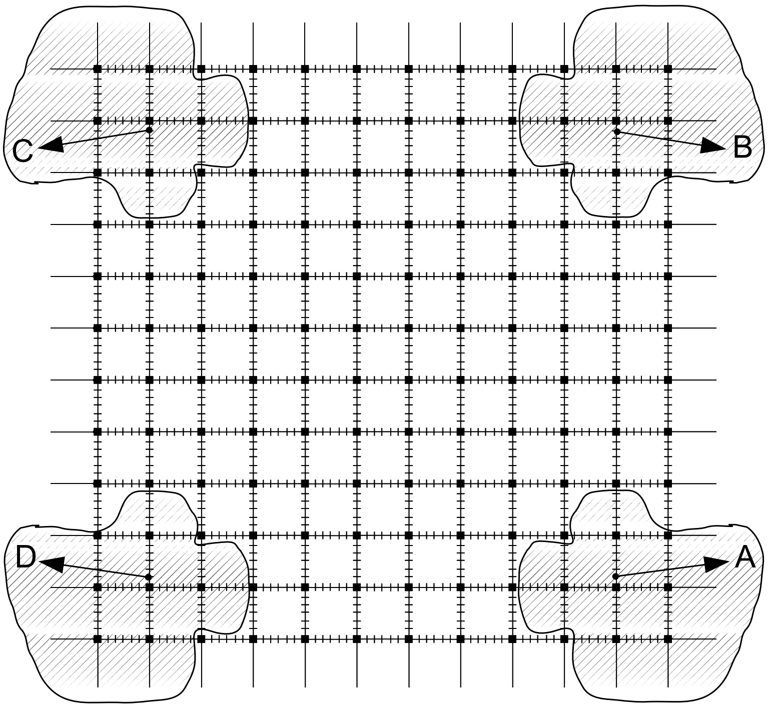
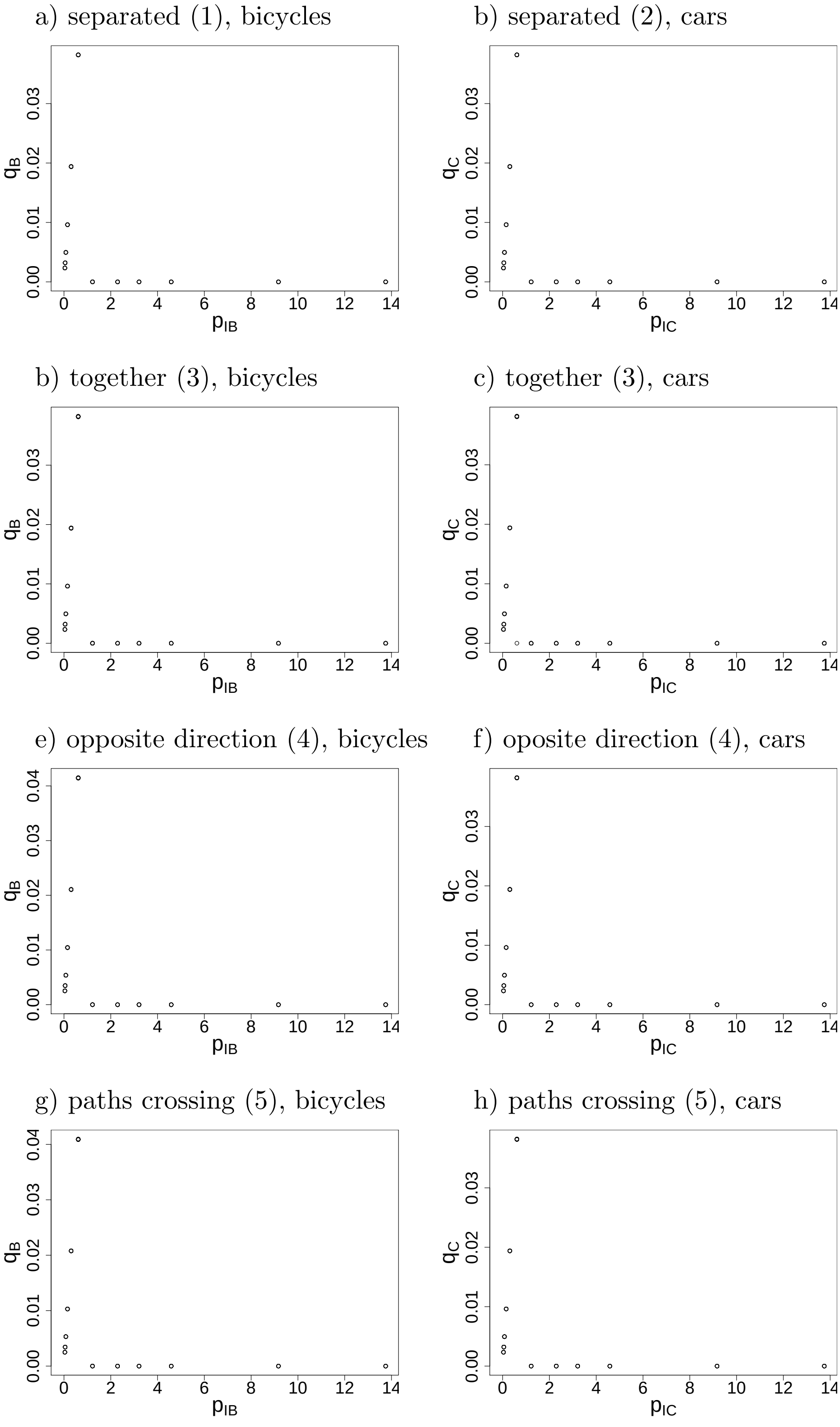


Figure 5.32: Grid of 12x12 two-way two-lane road intersections: scenario for simulations of routing between separated zones. A, B, C and D are zone seeds and the shaded areas around the seeds represent 20 car cell-wide zones.

the even-phase traffic lights improve the flows for both types of vehicles.

Scenario id	Scenario description
1	no turning, no signalisation, RHS priority rule in effect
2	no turning, no signalisation, LHS priority rule in effect
3	no turning, EW and SN in opposite phases: 30s red - 30s green
4	no turning, EW and SN in opposite phases: 60s red - 60s green
5	$p_{\mathrm{T}} = 0.5$, no signalisation, RHS priority rule in effect
6	$p_{\mathrm{T}} = 0.5$, no signalisation, LHS priority rule in effect
7	$p_{\mathrm{T}} = 0.5$, EW and SN in opposite phases: 30s red - 30s green
8	$p_{\mathrm{T}} = 0.5$, EW and SN in opposite phases: 60s red - 60s green
9	$p_{\mathrm{T}} = 0.5$, 4 mutually exclusive green phases: CSN - 15s, BSN - 45s, CEW - 15s, BEW - 45s
10	$p_{\mathrm{T}} = 0.5$, 4 mutually exclusive green phases: CSN - 45s, BSN - 15s, CEW - 45s, BEW - 15s

Table 5.13: Grid of 4x4 1-way 1-lane road intersections: parameter combinations for simulations with traffic lights.


a) separated (1), bicycles

b) separated (2), cars

b) together (3), bicycles

c) together (3), cars

e) opposite direction (4), bicycles

f) oposite direction (4), cars

g) paths crossing (5), bicycles

h) paths crossing (5), cars

Figure 5.33: Grid of 12x12 two-way two-lane road intersections: input-flow diagrams for scenarios with isolated origin and destination zones. Each subheading starts with the scenario name, followed by the number (in brackets) of the same scenario, as given in Table 5.12, and the relevant vehicle type. Axes are: $p_{\mathrm{IB}}, p_{\mathrm{ICu}}$ - overall bicycle and car input, respectively; $q_{\mathrm{B}}, q_{\mathrm{C}}$ - average bicycle and car flow, respectively.

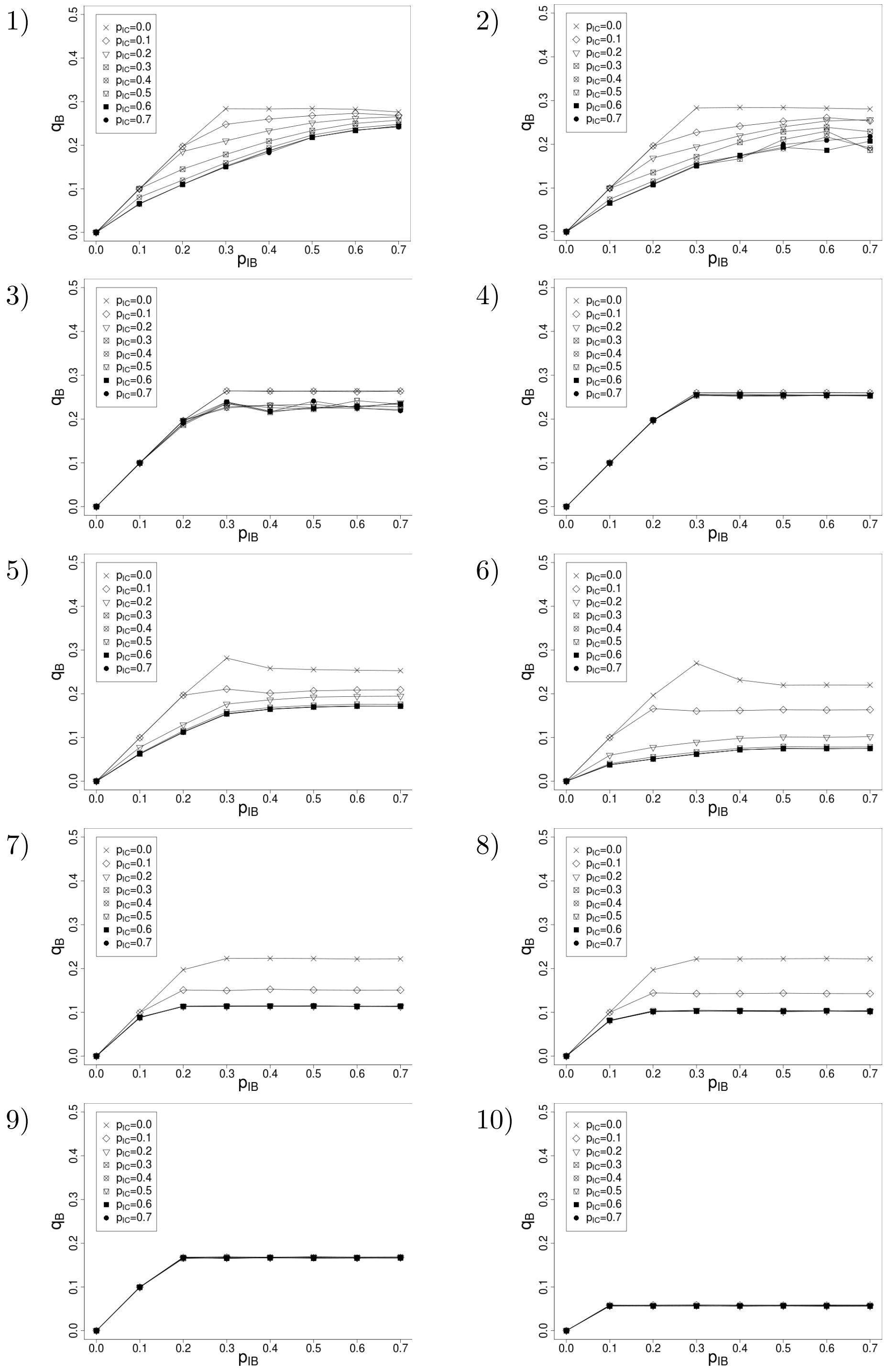


Figure 5.34: Grid of 4x4 one-way one-lane road intersections: input-flow diagrams for bicycles in the ten input scenarios described in Table 5.13. Each sub-figure number refers to the corresponding scenario.

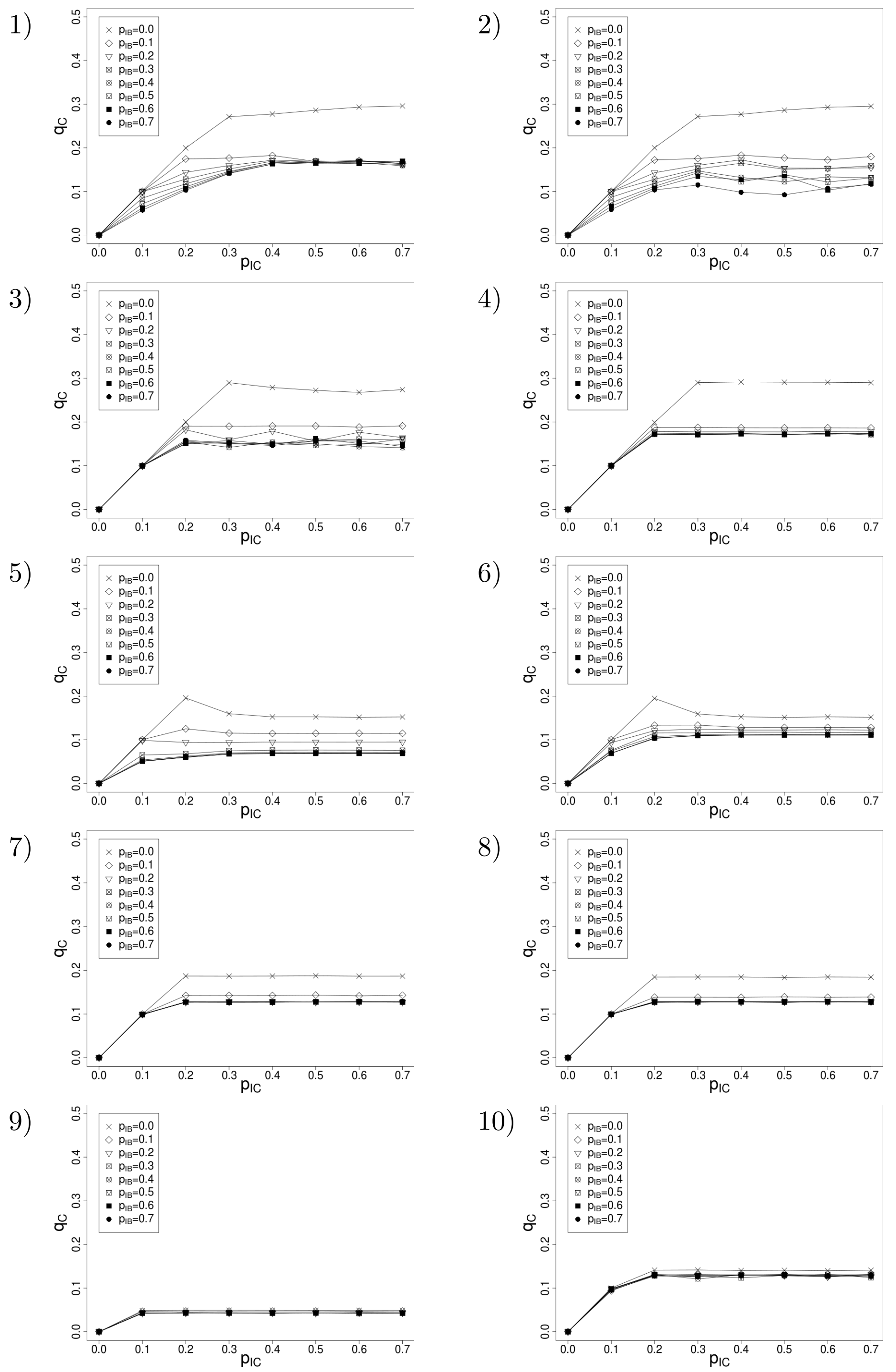


Figure 5.35: Grid of 4x4 one-way one-lane road intersections: input-flow diagrams for cars in the ten input scenarios described in Table 5.13. Each sub-figure number refers to the corresponding scenario.

5.8 Summary

This chapter describes simulations performed for the bi-modal traffic system, based on the model described in Chapter 3 and with implementation features as given in Chapter 4. The simulation scenarios progress from the simplest, which display the modelling of interactions between cars and bicycles on a road stretch, to those for isolated topological elements and, eventually, grid-based scenarios involving either one-directional or two-directional roads. Before dealing with the scenarios, however, the chapter introduces the measurement and presentation approaches that are used for evaluating and understanding the data output and describes the field data collection process that was employed with the purpose of model validation.

As traffic data for an urban network with heterogeneous flows including bicycles is not available for Dublin, or for cities similar with respect to bicycles, a small-scale data collection excercise was carried out for the needs of this project and the simpler scenarios validated using the collected data. The scenarios for which data were collected are: bicycle fundamental diagram, lateral interaction between cars and bicycles (and its variation depending on lane width) and the effect of a left turn on a car flow containing some proportion of vehicles turning left. For all these scenarios simulation output was found to be in good agreement with collected data or to have values and effects with the correct trends and order of magnitude (Sections 5.3 and 5.4). While more validation is needed, both for further investigation of basic model elements and for confirmation of more complex scenario outputs, this initial effort provides some confidence in the premises of the simulation model.

The more complex topological element scenarios were not validated directly with field data but their simulation output was presented, using it to:

- demonstrate scope for calibration, resulting from varying output ranges produced through the variation of parameters, such as the accepted gap, g_{A} or the difficulty of turning, d_{T} (Sections 5.4 and 5.5)
- illustrate the use of output analysis and presentation methods such as the *realisation diagram* and *effects sequence* (Sections 5.5 and 5.6)

Finally, examples of network simulations and corresponding outputs were given, to:

- present the range of intersection dynamics that the model can easily produce, showing isolated intersection output for four different *intersection configurations* of the two two-way road intersection (Figures 5.19 and 5.20) and the effect of using different intersection configurations in a network model (Figure 5.26),
- demonstrate the rough correspondence of the macroscopic fundamental diagram (e.g. Figure 5.27) with the shape expected based on work by other researchers (Geroliminis and Daganzo, 2008; Daganzo et al., 2011) and
- show the results of conducting some simple whole-network assessment: effect of the separation of modes through street allocation (Figure 5.31) and effect of common and separate origin and destination zones for the two modes (Figure 5.33).

The latter also acts as a 'proof of concept' with respect to the ultimate purpose of the simulation model and its implementation. Using it with scenarios of practical importance and with a high level of confidence in the assessment accurracy will be possible with further validation and, for some scenarios, additional implementation.

Chapter 6

Conclusion

The research presented in this thesis was centred around the modelling of network traffic including bicycles and, in particular, traffic of the type found in Dublin and other 'old city centres', where lane sharing is the dominant form of bicycle inclusion, followed by on-street bicycle lanes. This chapter summarises the the work that was carried out and analyses its achievements. The final section outlines possible future work that would improve or extend that already completed, as well as some new study ideas that were inspired by research for the thesis.

6.1 Summary of research and discussion

The content and achievements of the research discussed in this section are grouped by the objectives from Section 1.2 that they meet.

The vision of an ultimate application of the work provided a useful sketch of the bigger picture and a framework of reference for the research. Inputs of interest in relation to bicycles in motorised traffic are the network topology and size, street width, allocation of streets to vehicle types, vehicle type proportions in flows and vehicle properties such as gap acceptance and maximal velocity. Outputs of interest include the macroscopic fundamental diagram, input-flow diagrams but also whole-network profiles with local properties, special diagrams for various contexts etc.

The model of network traffic including bicycles was defined using cellular automata.

The technique was chosen for its compatibility with computer simulation and the fact that even simple CA movement rules are sufficient for successful representation of network traffic (Chopard et al., 1996). The discretised space of CA also facilitated modular implementation, which was an important factor in achieving flexibility with respect to scenario configuration.

Design of a comprehensively formulated bicycle-inclusive network traffic model and its implementation

A CA model was designed for heterogeneous network traffic including bicycles. The problem space is represented in its entirety by the model, which has the following novel features:

- *Heterogeneity of vehicle type* (Section 3.1) is catered for through cells of different sizes and vehicle type-specific parameters, such as maximal velocity. Allowing different cell sizes in the model, which is a novel feature in traffic CA, has the advantage of simple movement rules on a one-dimensional CA lattice for each vehicle type. There is an implicit equivalence between this type of model and multiple-cell occupancy models (Lan and Chang, 2005; Mallikarjuna and Rao, 2009), in terms of cell numbers, in the case of scattered homogeneity (e.g. in Indian streets). However, for lane sharing with positional discipline a large number of positions can be eliminated and our model becomes less redundant than the multiple-cell occupancy ones.

- *Network connections* are realised through tracks in intersections (Section 3.1), rather than specialised cells. This further facilitates the uninterrupted one-dimensional CA lattice as a vehicle's sole view of the network. This is another novel feature, made possible through *cell overlap* being allowed in the model. Not used before with CA models, it simplifies interactions at intersections, reducing them all to conflict constructs of the model.

- Cell overlap is also employed within tracks (one-dimensional CA spaces) in intersections to force deceleration (Figure 3.2d). This is transparent to the vehicle affected and models blockages by slow turning vehicles.

- The connector construct of the spatial meta-model allows definition of re-usable topological elements, which makes for flexibility during scenario building as well as for domain decomposition in preparation for parallel processing (Section 3.1.1).

A vehicle in our model retains the *one-dimensional CA view* of its environment, even though that environment is a network. This means that the usual simple form of cellular automaton rules is possible for a comprehensive formulation of vehicle behaviour. Routing i.e. choice of route at divergences is incorporated as Rule 0 - it does not affect the physics of movement but rather defines the route along which the dynamics unfold. Turns and conflicts encountered along the way affect those dynamics.

Model validation

The model was *validated* with collected field data. Comparisons of simulation output with field data and data published by other authors is discussed in Sections 5.3 and 5.4. The validation was limited to the more basic simulation scenarios but the results are promising, in that simulation and empirical vehicle velocities (Figure 5.4) have similar values for comparable scenarios and that velocities, as affected by other flows (Figures 5.7, 5.8 and 5.10), show values and trends in very similar ranges.

Initially the model was assigned physical values of the original Nagel and Schreckenberg (1992), with 7.5m car-cell lengths. Analysis of the measured data showed that a value of 5m is more accurate with regard to the field measurements, which is not entirely unexpected considering that the average velocities in an urban setting are lower than those on the motorway, which was the case under simulation by Nagel and Schreckenberg (1992).

General applicability and extractability (Chapter 3)

General applicability, while not an early research objective, has emerged as a by-product of the efforts to reconcile differences between bicycles and motorised vehicles in a single model and the model's consequent reduction down to 5 elemental constructs (Section

3.1). The meta-model can be applied to any vehicle type mix. The behaviour rules are also general, apart from parameters, such as the maximal velocity, and inter-type interaction rules. To our knowledge this is the first generally applicable CA model for heterogeneous traffic.

Another property that was not intentionally built into the model but that was found to hold is *automatable extractability*, which means that the metamodel construct instances constituting the model could be extracted from a drawing of the model. This feature, however, is present in many traffic simulation implementations, e.g. Esser and Schreckenberg (1997), who even had the network editor 'guess' priorities at an intersection.

Parallel computing investigations (Section 4.2)

Parallel implementations of the model were carried out and runs performed with a number of different scenario sizes (10 to 10000 cells), both for a ring road and for a grid of two-way two-road intersections. The parallelisation can be considered successful since speed up was achieved with all configurations. An interesting result is that for the grid scenarios, in general the speed-up with message passing is greater than with shared memory. This is a consequence of the special property of traffic simulation whereby the boundary sections are relatively short (only at broken streets/links) in comparison with the problem size.

Study of modelled system from simulation output (Section 5)

Modelled system evaluation with application in the real world, while part of the bigger picture, was not within the scope of the thesis objectives. However, some conclusions could be made about the available simulation results for networks:

- the simulated network of two two-way road intersections, even if it is only of size 2x2, has a critical arrival rate value above which a deadlock and consequently a lower average flow is almost certain; the critical arrival rate for a 2x2 grid is just under 0.5 (Figure 5.26), while for an 11x11 grid it is between 0.2 and 0.3 (Figure 5.30); deadlocks are expected in networks (Daganzo et al., 2011), however, further validation would be needed to confirm that the critical flow value matches reality

- two very rudimentary whole-network-configuration tests, also without validation but were performed: one was dividing the streets into 'car streets' and 'bicycle streets', allowing only access to the 'wrong' type of vehicle, in a 12x12 network and the input-flow diagrams (Figure 5.31) indicate lower critical input flow values for the case with vehicle type separation
- the other whole-network-configuration test performed used zone-based routing to set up origin and destination zones resulting in 4 different scenarios; all the scenarios display a critical arrival rate above which there is no flow; however, there is very litte difference in the maximal flows between the scenarios

Also, two novel forms of presenting simulation results have been described, the *realisation diagram* and *effects sequence*. These are useful in understanding the relationships between different flows through and intersection.

A large proportion of the research reported on in this thesis has been published in journal and conference papers. These are listed in Appendix E.

6.2 Future work

The research objectives outlined in Chapter 1 have been met: a comprehensive model for network traffic including bicycles has been defined, implemented and validated and the possibility of using the model for building and simulating complex scenarios demonstrated. This section outlines a number of possible directions for future work.

Utilisation of the NTIB model

Further research relating to the model for network traffic including bicycles (NTIB) could include the addition of a new track relationship that would represent the gathering of lanes into a multi-lane street. The addition of this element would open up the possibility of modelling an arbitrary real network, including signalling. Then, validation of the model at a network level would be a substantial undertaking from a practical perspective but worth doing for completeness of the NTIB, which, with the multi-lane support and complete validation, would become a useful tool for assessment of bicycle-related properties of a network and for flow optimisation where bicycles are involved.

A graphical interface for simulation building and visualisation

While with the implemented framework so far it is possible to configure and combine complex simulation scenarios, the process is not user-friendly. A front-end complement to the existing code would, if developed for the simulator, add greatly to its usability.

Productisation - viability

A project of a less technical nature and yet just as interesting would be to investigate the viability of a possible productisation of the simulator. This would involve an in-depth study of the existing simulators on the market that account for bicycles in their models, a comparison of their capabilities but also, at a very concrete level, of the differences and similarities of traffic flow they produce, in particular of that involving bicycles. The aim of this would be to gauge the viability of developing the simulation framework into a product.

Dynamic load balancing for parallel processing

The work done in the area of parallelising the simulation is limited in its capacity for optimising loads, since for this it relies on a single application of domain decomposition, albeit performed with the use of a load balancing algorithm. The examples we chose for speed-up measurement were such that they maintained the load balanced for the purpose of measurements, since we were not interested in the loads and, in fact, required them to be equal. Dynamic load balancing (DLB) of a parallel traffic simulation is quite a specific problem: re-balancing during a run (or in the middle of an interrupted run) in this case would seem to require passing of network portions and associated vehicles between processes to avoid ever-increased partitioning of the network. A study to find the best algorithms for DLB and test them through implementation would be an interesting research topic.

Adaptation for other domain applications

The network traffic model could be adapted for simulation of entirely different domains. For example, the model would seem to be suitable for representing water supply and sewerage systems or even the spreading of excess water in the city during rainstorms. The latter could be useful in the prediction of local flooding or as a tool for designing flood-preventive structures such as swales.

References

Akçelik, R. 2007. A Review of Gap-Acceptance Capacity Models *IN: 29th Conference of Australian Institutes of Transport Research (CAITR 2007).*

Allen, D.P., Hummer, J.E., Rouphail, N.M. and Milazzo, J.S. 1998a. Effect of bicycles on capacity of signalized intersections. *Transportation Research Record*, 1646(1646), pp.87–95.

Allen, D.P., Rouphail, N., Hummer, J.E. and Milazzo, J.S. 1998b. Operational analysis of uninterrupted bicycle facilities. *Transportation Research Record*, 1636(1636), pp.29–36.

Ardekani, S. and Herman, R. 1987. Urban Network-Wide Traffic Variables and their Relations. *Transportation Science*, 21(1), pp.1–16.

Aultman-Hall, L. 2004. Chapter 22: Bicycle Transportation *IN:* Kutz, M. (ed.) *Handbook of transportation engineering.* McGraw-Hill Professional, p.22.1.

Aw, A. and Rascle, M. 2000. Resurrection of "Second Order" Models of Traffic Flow. *SIAM Journal on Applied Mathematics*, 60(3), pp.916–938.

Balmer, M., Cetin, N., Nagel, K. and Raney, B. 2004. Towards Truly Agent-Based Traffic and Mobility Simulations *IN: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '04. Washington, DC, USA: IEEE Computer Society, pp.60–67.

Bando, M., Hasebe, K., Nakayama, A., Shibata, A. and Sugiyama, Y. 1995. Dynamical model of traffic congestion and numerical simulation. *Physical Review E*, 51(2), pp.1035–1042.

Barceló, J., Codina, E., Casas, J., Ferrer, J. and Garcá, D. 2005. Microscopic traffic simulation: A tool for the design, analysis and evaluation of intelligent transport systems. *Journal of Intelligent and Robotic Systems*, 41(2-3), pp.173–203.

Barceló, J., Ferrer, J.L., García, D. and Grau, R. 1998. *Microscopic traffic simulation for att systems analysis. a parallel computing version. Contribution to the 25th Aniversary of CRT.*

Barceló, J., J., Casas, J., Ferrer, J. and García, D. 1999. Modelling Advanced Transport Telematic Applications with Microscopic Simulators: The Case of AIMSUN2 *IN:* Brilon, W., Huber, F., Schreckenberg, M. and Wallentowitz, H. (eds.) *Traffic and Mobility.* Springer Berlin Heidelberg, pp.205–221.

Barlovic, R., Huisinga, T., Schadschneider, A. and Schreckenberg, M. 2002. Open boundaries in a cellular automaton model for traffic flow with metastable states. *Physical Review E*, 66(4), p.046113.

Barlovic, R., Santen, L., Schadschneider, A. and Schreckenberg, M. 1998. Metastable states in cellular automata for traffic flow. *European Physical Journal B*, 5(3), pp.793–800.

Beckmann, M., McGuire, C. and Winsten, C. 1955. *Studies in the Economics of Transportation.* Rand Corporation.

Belbasi, S. and Foulaadvand, M. 2008. Simulation of traffic flow at a signalized intersection. *Journal of Statistical Mechanics - Theory and Experiment*, 2008(7), p.P07021.

Biham, O., Middleton, A. and Levine, D. 1992. Self-organization and a dynamical transition in traffic-flow models. *Physical Review A*, 46, pp.R6124–R6127.

Blanco, H., Alberti, M., Forsyth, A., Krizek, K.J., Rodriguez, D.A., Talen, E. and Ellis, C. 2009. Hot, congested, crowded and diverse: Emerging research agendas in planning. *Progress in Planning*, 71(Part 4), pp.153–205.

Botma, H. 1995. Method to Determine Levels of Service for Bicycle Paths and Pedestrian-Bicycle Paths. *Transportation Research Record*, 1502, pp.38–44.

Boyce, D. 2007. Forecasting Travel on Congested Urban Transportation Networks: Review and Prospects for Network Equilibrium Models. *Networks and Spatial Economics*, 7, pp.99–128.

Boyce, D., Mahmassani, H. and Nagurney, A. 2005. A retrospective on Beckmann, McGuire and Winsten's Studies in the Economics of Transportation. *Papers in Regional Science*, 84(1), pp.85–103.

Brackstone, M. and McDonald, M. 1999. Car-following: a historical review. *Transportation Research Part F*, 2(4), pp.181 – 196.

Brilon, W., Koenig, R. and Troutbeck, R.J. 1999. Useful estimation procedures for critical gaps. *Transportation Research Part A*, 33(3-4), pp.161 – 186.

Brilon, W. and Wu, N. 1999. Capacity at unsignalized two-stage priority intersections. *Transportation Research Part A: Policy and Practice*, 33(3-4), pp.275 – 289.

Brockfeld, E., Barlovic, R., Schadschneider, A. and Schreckenberg, M. 2001. Optimizing traffic lights in a cellular automaton model for city traffic. *Physical Review E*, 64(5, Part 2), p.056132.

Brown, J.R., Morris, E.A. and Taylor, B.D. 2009. Planning for Cars in Cities: Planners, Engineers, and Freeways in the 20th Century. *Journal of the American Planning Association*, 75(2), pp.161–177.

Buisson, C. and Ladier, C. 2009. Exploring the Impact of Homogeneity of Traffic Measurements on the Existence of Macroscopic Fundamental Diagrams. *Transportation Research Record*, 2124, pp.127–136.

Cameron, G.D.B. and Duncan, G.I.D. 1996. PARAMICS Parallel microscopic simulation of road traffic. *The Journal of Supercomputing*, 10, pp.25–53.

Cassidy, M.J., Daganzo, C.F. and Jang, K. 2008. Spatiotemporal Effects of Segregating Different Vehicle Classes on Separate Lanes.

Cavill, N., Kahlmeier, S., Rutter, H., Racioppi, F. and Oja, P. 2008. Economic analyses of transport infrastructure and policies including health effects related to cycling and walking: A systematic reviews. *Transport Policy*, 15(5), pp.291–304.

Cervero, R., Sarmiento, O., Jacoby, E., Fernando Gomez, L. and Neiman, A. 2009. Influences of built environments on walking and cycling: Lessons from Bogota. *International Journal of Sustainable Transportation*, 3(4), pp.203–226.

Chandler, R., Herman, R. and Montroll, E. 1958. Traffic dynamics - studies in car following. *Operations Research*, 6(2), pp.165–184.

Chen, C., Varaiya, P. and Kwon, J. 2005. An empirical assessment of traffic operations *IN:* Mahmassani, H.S. (ed.) *Transpn and Traffic Theory, Procs of 16th International Symp on Transpn and Traffic Theory.*

Chen, X.M., Shao, C.F. and Yue, H. 2007. Influence of Bicycle Traffic on Capacity of Typical Signalized Intersection. *Tsinghua Science & Technology*, 12(2), pp.198 – 203.

Chen, Y.H., Wang, D.H. and Tao, Z.X. 2009. Speed Character Study for Motor Vehicle and Bicycle at Non-barrier Section. *Journal of Transportation Systems Engineering and Information Technology*, 9(5), pp.53 – 57.

Cheng, S.H., Yao, D.Y., Zhang, Y., Su, Y.L. and Xu, W.D. 2008. A CA Model for Intrusion Conflicts Simulation in Vehicles-Bicycles Laminar Traffic Flow *IN: Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems Beijing, China, October 12-15, 2008.*

Chopard, B., Luthi, P. and Queloz, P. 1996. Cellular automata model of car traffic in a two-dimensional street network. *Journal of Physics A*, 29(10), pp.2325–2336.

Chowdhury, D. and Schadschneider, A. 1999. Self-organization of traffic jams in cities: Effects of stochastic dynamics and signal periods. *Physical Review E*, 59(2, Part A), pp.R1311–R1314.

Chowdhury, D., Wolf, D. and Schreckenberg, M. 1997. Particle hopping models for two-lane traffic with two kinds of vehicles: Effects of lane-changing rules. *Physica A: Statistical Mechanics and its Applications*, 235(3-4), pp.417–439.

Committee, A.T.N.M. 2010. *A Primer for Dynamic Traffic Assignment.* Technical report, Transportation Research Board.

Cremer, M. and Ludwig, J. 1986. A Fast Simulation-Model for Traffic Flow on the Basis of Boolean Operations. *Mathematics and Computers In Simulation*, 28(4), pp.297–303.

Currie, G., Sarvi, M. and Young, W. 2004. A New Methodology for Allocating Road Space for Public Transport Priority *IN: Urban Transport X: Urban Transport and the Environment in the 21st Century.*

Dabbour, E. and Easa, S.M. 2008. Evaluation of safety and operational impacts of bicycle bypass lanes at modern roundabouts. *Canadian Journal of Civil Engineering*, 35(10), pp.1025–1032.

Daganzo, C. 1995a. The Cell Transmission Model, Part 2: Network Traffic. *Transportation Research Part B*, 29(2), pp.79–93.

Daganzo, C. 1995b. Properties of link travel time functions under dynamic loads. *Transportation Research Part B*, 29(2), pp.95 – 98.

Daganzo, C., Gayah, V. and Gonzales, E. 2011. Macroscopic relations of urban traffic variables: Bifurcations, multivaluedness and instability. *Transportation Research Part B: Methodological*, 45(1), pp.278 – 288.

Daganzo, C.F. 1994. The cell transmission model - a dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B-methodological*, 28(4), pp.269–287.

Daganzo, C.F. 1995c. Requiem for second-order fluid approximations of traffic flow. *Transportation Research Part B*, 29(4), pp.277–286.

Daganzo, C.F. 2006. In traffic flow, cellular automata = kinematic waves. *Transportation Research Part B: Methodological*, 40(5), pp.396 – 403.

Daganzo, C.F. 2007. Urban gridlock: Macroscopic modeling and mitigation approaches. *Transportation Research Part B*, 41(1), pp.49 – 62.

Daganzo, C.F., Cassidy, M.J. and Bertini, R.L. 1999. Possible explanations of phase transitions in highway traffic. *Transportation Research Part A*, 33(5), pp.365 – 379.

Daganzo, C.F. and Geroliminis, N. 2008. An analytical approximation for the macroscopic fundamental diagram of urban traffic. *Transportation Research Part B*, 42(9), pp.771 – 781.

Dai, W., Zhang, J. and Zhang, D. 2010. Parallel simulation of large-scale microscopic traffic networks *IN: Advanced Computer Control (ICACC), 2010 2nd International Conference on*, 3. pp.22–28.

Deo, P. and Ruskin, H.J. 2006a. Comparison of Homogeneous and Heterogeneous Motorised Traffic at Signalised and Two-Way Stop Control Single Lane Intersection *IN:* Gavrilova, M., Gervasi, O., Kumar, V., Tan, C., Taniar, D., Lagana, A., Mun, Y. and Choo, H. (eds.) *Computational Science and Its Applications - ICCSA 2006*, *Lecture Notes in Computer Science* 3980. Heidelberg: Springer, pp.622–632.

Deo, P. and Ruskin, H.J. 2006b. Simulation of heterogeneous motorised traffic at a signalised intersection *IN:* El Yacoubi, S., Chopard, B. and Bandini, S. (eds.) *Cellular Automata*, *Lecture Notes in Computer Science* 4173. Springer Berlin / Heidelberg, pp.522–531.

Dijkstra, E.W. 1959. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1, pp.269–271.

Edie, L.C. 1963. Discussion of traffic stream measurements and definitions *IN:* Almond, J. (ed.) *Proceedings of the 2nd International Symposium on the Theory of Traffic Flow.* OECD, Paris, pp.139–154.

El-Geneidy, A., Krizek, K. and Iacono, M. 2007. Predicting Bicycle Travel Speeds Along Different Facilities Using GPS Data: A Proof-of-Concept Model *IN: TRB 86th Annual Meeting Compendium of Papers.*

Emmerich, H. and Rank, E. 1997. An improved cellular automaton model for traffic flow simulation. *Physica A: Statistical Mechanics and its Applications*, 234(3/4), pp.676 – 686.

Esser, J. and Schreckenberg, M. 1997. Microscopic simulation of urban traffic based on cellular automata. *International Journal of Modern Physics C*, 8(5), pp.1025–1036.

Even, S. 2011. *Graph Algorithms.* New York, NY, USA: Cambridge University Press, 2nd edition.

Faghri, A. and Egyháziová, E. 1999. Development of a Computer Simulation Model of Mixed Motor Vehicle and Bicycle Traffic on an Urban Road Network. *Transportation Research Record*, 1674, pp.86–93.

Fellendorf, M. and Vortisch, P. 2010. Microscopic Traffic Flow Simulator VISSIM *IN:* Barceló, J. (ed.) *Fundamentals of Traffic Simulation.* Springer, pp.63–93.

Feng, Y., Liu, Y., Deo, P. and Ruskin, H.J. 2007. Heterogeneous traffic flow model for a two-lane roundabout and controlled intersection. *International Journal of Modern Physics C*, 18(1), pp.107–117.

Foulaadvand, M. and Belbasi, S. 2007. Vehicular traffic flow at a non-signalized intersection. *Journal of Physics A - Mathematical and Theoretical*, 40(29), p.8289.

Foulaadvand, M., Fukui, M. and Belbasi, S. 2010. Phase structure of a single urban intersection: a simulation study. *Journal of Statistical Mechanics - Theory and Experiment*, 2010, p.P07012.

Foulaadvand, M. and Neek-Amal, M. 2007. Asymmetric simple exclusion process describing conflicting traffic flows. *EPL*, 80(6), p.60002.

Fukui, M. and Ishibashi, Y. 2010. Two-dimensional city traffic model with periodically placed blocks. *Physica A: Statistical Mechanics and its Applications*, 389(17), pp.3613 – 3618.

Garcia, R. and Marin, A. 2005. Network equilibrium with combined modes: models and solution algorithms. *Transportation Research Part B*, 39(3), pp.223–254.

Gayah, V.V. and Daganzo, C.F. 2011. Clockwise hysteresis loops in the Macroscopic Fundamental Diagram: An effect of network instability. *Transportation Research Part B: Methodological*, 45(4), pp.643–655.

Gazis, D., Herman, R. and Potts, R. 1959. Car-Following Theory of Steady-State Traffic Flow. *Operations Research*, 7(4), pp.499–505.

Gazis, D., Herman, R. and Rothery, R. 1961. Nonlinear Follow-The-Leader Models of Traffic Flow. *Operations Research*, 9(4), pp.545–567.

Gazis, D.C. 2002. The Origins of Traffic Theory. *Operations Research*, 50(1), pp.69–77.

Ge, H., Dai, S., Dong, L. and Xue, Y. 2004. Stabilization effect of traffic flow in an extended car-following model based on an intelligent transportation system application. *Physical Review E*, 70(6, Part 2), p.066134.

Geroliminis, N. and Daganzo, C. 2008. Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings. *Transportation Research Part B*, 42(9), pp.759 – 770.

Geroliminis, N. and Sun, J. 2011. Properties of a well-defined macroscopic fundamental diagram for urban traffic. *Transportation Research Part B: Methodological*, 45(3), pp.605 – 617.

Geurs, K.T., Boon, W. and Van Wee, B. 2009. Social Impacts of Transport: Literature

Review and the State of the Practice of Transport Appraisal in the Netherlands and the United Kingdom. *Transport Reviews*, 29(1), pp.69 – 90.

Gonzales, E.J., Geroliminis, N., Cassidy, M.J. and Daganzo, C.F. 2008. *Allocating city space to multiple transportation modes: A new modeling approach consistent with the physics of transport.* UC Berkeley Center for Future Urban Transport. Working Paper UCB-ITS-VWP-2008-1.

Gould, G. and Karner, A. 2009. Modeling Bicycle Facility Operation. *Transportation Research Record*, 2140, pp.157–164.

Gray, R. 2009. *Probability, Random Processes, and Ergodic Properties.* Springer-Verlag US.

Greenshields, B. 1935. A Study of Traffic Capacity. *Proceedings of the Highway Research Board*, 14, pp.448–474.

Gundaliya, P.J., Mathew, T.V. and Dhingra, S.L. 2008. Heterogeneous traffic flow modelling for an arterial using grid based approach. *Journal of Advanced Transportation*, 42(4), pp.467–491.

Hall, R.W., (ed.) 2003. *Handbook of transportation science.* Springer.

Heinen, E., van Wee, B. and Maat, K. 2010. Commuting by Bicycle: An Overview of the Literature. *Transport Reviews*, 30(1), pp.59–96.

Helbing, D. 1995. Improved fluid-dynamic model for vehicular traffic. *Physical Review E*, 51(4, Part A), pp.3164–3169.

Helbing, D. 1996. Gas-kinetic derivation of Navier-Stokes-like traffic equations. *Physical Review E*, 53(3), pp.2366–2381.

Helbing, D. 1997. Modeling multi-lane traffic flow with queuing effects. *Physica A - Statistical Mechanics and Its Applications*, 242(1-2), pp.175 – 194.

Helbing, D. 1998. Structure and instability of high-density equations for traffic flow. *Physical Review E*, 57(5), pp.6176–6179.

Helbing, D. 2003. A section-based queueing-theoretical traffic model for congestion and travel time analysis in networks. *Journal of Physics A - Mathematical and General*, 36(46), p.L593.

Helbing, D. 2009a. Derivation of a fundamental diagram for urban traffic flow. *European Physical Journal B*, 70(2), pp.229–241.

Helbing, D. 2009b. Derivation of non-local macroscopic traffic equations and consistent traffic pressures from microscopic car-following models. *European Physical Journal B*, 69(4), pp.539–548.

Helbing, D. and Huberman, B. 1998. Coherent moving states in highway traffic. *Nature*, 396(6713), pp.738–740.

Helbing, D. and Molnar, P. 1995. Social force model for pedestrian dynamics. *Physical Review E*, 51(5), pp.4282–4286.

Helbing, D., Siegmeier, J. and Lammer, S. 2007. Self-Organized Network Flows. *Networks and Heterogeneous Media*, 2, pp.193–210.

Helbing, D. and Tilch, B. 1998. Generalized force model of traffic dynamics. *Physical Review E*, 58(1), pp.133–138.

Helbing, D. and Treiber, M. 1999. Numerical Simulation of Macroscopic Traffic Equations. *Computing in Science & Engineering*, 1(5), pp.89–99.

Hensher, D.A. and Button, K.J., (eds.) 2000. *Handbook of transport modelling.* Emerald Group Publishing.

Herman, R. and Ardekani, S. 1984. Characterizing Traffic Conditions in Urban Areas. *Transportation Science*, 18(2), p.101.

Herman, R., Montroll, E., Potts, R. and Rothery, R. 1959. Traffic Dynamics: Analysis of Stability in Car Following. *Operations Research*, 7(1), pp.86–106.

Herman, R. and Prigogine, I. 1979. A Two-Fluid Approach to Town Traffic. *Science*, 204(4389), pp.148–151.

Herrey, E.M.J. and Herrey, H. 1945. Principles of Physics Applied to Traffic Movements and Road Conditions. *American Journal of Physics*, 13(1), pp.1–14.

Hoogendoorn, S.P. and Bovy, P.H.L. 2000. Continuum modeling of multiclass traffic flow. *Transportation Research Part B: Methodological*, 34(2), pp.123 – 146.

Hoogendoorn, S.P. and Bovy, P.H.L. 2001. State-of-the-art of vehicular traffic flow modelling. *Proceedings of the Institution of Mechanical Engineers – Part I – Journal of Systems & Control Engineering*, 215(4), pp.283 – 303.

Hossain, M. and McDonald, M. 1998. Modelling of traffic operations in urban networks of developing countries: a computer aided simulation approach. *Computers, Environment and Urban Systems*, 22(5), pp.465 – 483.

Hu, M.B., Jiang, R., Wang, R. and Wu, Q.S. 2009. Urban traffic simulated from the dual representation: Flow, crisis and congestion. *Physics Letters A*, 373(23-24), pp.2007–2011.

Huang, D. and Huang, W. 2003. Traffic signal synchronization. *Physical Review E*, 67(5, Part 2), p.056124.

Jia, B., Li, X.G., Jiang, R. and Gao, Z.Y. 2007. Multi-value cellular automata model for mixed bicycle flow. *European Physical Journal B*, 56(3), pp.247–252.

Jia, S.P., Peng, H.Q., Guo, J.Y. and Chen, H.B. 2008. Quantitative Analysis of impact of Bicycles on Vehicles in Urban Mixed Traffic. *Journal of Transportation Systems Engineering and Information Technology*, 8(2), pp.58 – 63.

Jiang, R., Jia, B. and Wu, Q. 2004. Stochastic multi-value cellular automata models for bicycle flow. *Journal of Physics A - Mathematical and General*, 37(6), pp.2063–2072.

Jiang, R., Wu, Q. and Zhu, Z. 2002. A new continuum model for traffic flow and numerical tests. *Transportation Research Part B: Methodological*, 36(5), pp.405–419.

Jin, X., Zhang, Y., Wang, F., Li, L., Yao, D., Su, Y. and Wei, Z. 2009. Departure headways at signalized intersections: A log-normal distribution model approach. *Transportation Research Part C*, 17(3), pp.318–327.

Kerner, B. and Konhäuser, P. 1993. Cluster effect in initially homogeneous traffic flow. *Physical Review E*, 48(4), pp.R2335–R2338.

Kerner, B.S. and Klenov, S.L. 2002. A microscopic model for phase transitions in traffic flow. *Journal of Physics A - Mathematical and General*, 35(3), pp.L31–L43.

Kerner, B.S., Klenov, S.L. and Wolf, D.E. 2002. Cellular automata approach to three-phase traffic theory. *Journal of Physics A - Mathematical and General*, 35(47), pp.9971–10013.

Kerner, B.S. and Rehborn, H. 1996a. Experimental features and characteristics of traffic jams. *Physical Review E*, 53(2), pp.R1297–R1300.

Kerner, B.S. and Rehborn, H. 1996b. Experimental properties of complexity in traffic flow. *Physical Review E*, 53(5), pp.R4275–R4278.

Kerner, B.S. and Rehborn, H. 1997. Experimental Properties of Phase Transitions in Traffic Flow. *Physical Review E*, 79(20), pp.4030–4033.

Khan, S.I. and Maini, P. 1999. Modeling Heterogeneous Traffic Flow. *Transportation Research Record*, 1678(1), pp.234–241.

Klefstad, R., Zhang, Y., Lai, M.J., Jayakrishnan, R. and Lavanya, R. 2005. A distributed, scalable, and synchronized framework for large-scale microscopic traffic simulation *IN: Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE.* pp.813 – 818.

Knospe, W., Santen, L., Schadschneider, A. and Schreckenberg, M. 2002. Single-vehicle data of highway traffic: Microscopic description of traffic phases. *Physical Review E*, 65(5), p.056133.

Krajzewicz, D., Hertkorn, G., Ŕossel, C. and Wagner, P. 2002. SUMO (simulation of urban mobility); an open-source traffic simulation *IN: Proceedings of the 4th Middle East Symposium on Simulation and Modelling (MESM2002)*. pp.183–187.

Krauss, S., Wagner, P. and Gawron, C. 1996. Continuous limit of the Nagel-Schreckenberg model. *Physical Review E*, 54(4), pp.3707–3712.

Kühne, R.D. 1984. Macroscopic Freeway Model for Dense Traffic - Stop-Start Waves and Incident Detection *IN: Ninth International Symposium on Transportation and Traffic Theory*.

Laemmer, S. and Helbing, D. 2008. Self-control of traffic lights and vehicle flows in urban road networks. *Journal of Statistical Mechanics - Theory and Experiment*, 2008, p.P04019.

Lan, L. and Chang, C. 2005. Inhomogeneous cellular automata modeling for mixed traffic with cars and motorcycles. *Journal of Advanced Transportation*, 39(3), pp.323–349.

Lee, D.H. and Chandrasekar, P. 2002. A Framework for Parallel Traffic Simulation Using Multiple Instancing of a Simulation Program. *Journal of Intelligent Transportation Systems*, 7(3-4), pp.279–294.

Li, S. 2012. A survey of urban traffic coordination controls in intelligent transportation systems *IN: Service Operations and Logistics, and Informatics (SOLI), 2012 IEEE International Conference on*. pp.177–182.

Li, X., Jia, B., Gao, Z. and Jiang, R. 2006. A realistic two-lane cellular automata traffic model considering aggressive lane-changing behavior of fast vehicle. *Physica A: Statistical Mechanics and its Applications*, 367, pp.479–486.

Li, X., Jiang, R. and Wu, Q. 2004. Cellular automaton model simulating traffic flow at an uncontrolled T-shaped intersection. *International Journal of Modern Physics B*, 18(17-19), pp.2703–2707.

Li, X.G., Gao, Z.Y., Jia, B. and Zhao, X.M. 2009a. Cellular automata model for unsignalized T-shaped intersection. *International Journal of Modern Physics C*, 20(4), pp.501–512.

Li, X.G., Gao, Z.Y., Jia, B. and Zhao, X.M. 2009b. Modeling the interaction between motorized vehicle and bicycle by using cellular automata model. *International Journal of Modern Physics C*, 20(2), pp.209 – 222.

Lighthill, M. and Whitham, G. 1955. On kinematic waves 2. A theory of traffic flow on long crowded roads. *Proceedings of the Royal Society of London Series A*, 229(1178), pp.317–345.

Livermore Computing Center (LC) 2013. *MPI Performance Topics* [Online]. Available from: computing.llnl.gov/tutorials/mpi_performance [Accessed July 2013].

Lo, S. and Hsu, C. 2010. Cellular automata simulation for mixed manual and automated control traffic. *Mathematical and Computer Modelling*, 51(7-8), pp.1000–1007.

Maerivoet, S. and De Moor, B. 2005. Cellular automata models of road traffic. *Physics Reports*, 419(1), pp.1–64.

Mallikarjuna, C. and Rao, K.R. 2009. Cellular automata model for heterogeneous traffic. *Journal of Advanced Transportation*, 43(3), pp.321–345.

Martens, K. 2007. Promoting bike-and-ride: The Dutch experience. *Transportation Research Part A*, 41(4), pp.326–338.

Massink, R. 2009. *Estimating the climate value of bicycling in Bogota, Colombia, using a shadow pricing methodology.* Master's thesis, University of Twente.

May, A.D. 1990. *Traffic Flow Fundamentals.* Prentice Hall.

McNally, M.G. 2007. The Four Step Model.

Modesti, P. and Sciomachen, A. 1998. A utility measure for finding multiobjective shortest paths in urban multimodal transportation networks. *European Journal of Operational Research*, 111(3), pp.495–508.

MPI Forum 2009. *MPI: A Message-Passing Interface Standard, Version 2.2* [Online]. Available from: http://www.mpi-forum.org/docs/mpi-2.2/mpi22-report.pdf [Accessed July 2013].

Nagatani, T. 1993a. Jamming transition in the traffic-flow model with two-level crossings. *Physical Review E*, 48(5), pp.3290–3294.

Nagatani, T. 1993b. Self-organization and phase transition in traffic-flow model of a two-lane roadway. *Journal of Physics A - Mathematical and General*, 26(17), pp.L781–L787.

Nagatani, T. 1993c. Shock formation and traffic jam induced by a crossing in the 1d asymmetric exclusion model. *Journal of Physics A - Mathematical and General*, 26(23), pp.6625–6634.

Nagatani, T. 2006. Control of vehicular traffic through a sequence of traffic lights positioned with disordered interval. *Physica A: Statistical Mechanics and its Applications*, 368(2), pp.560–566.

Nagel, K. 1996. Particle hopping models and traffic flow theory. *Physical Review E*, 53(5), pp.4655–4672.

Nagel, K. and Rickert, M. 2001. Parallel implementation of the TRANSIMS microsimulation. *Parallel Computing*, 27(12), pp.1611 – 1639.

Nagel, K. and Schreckenberg, M. 1992. A cellular automaton model for freeway traffic. *Journal de Physique I*, 2(12), pp.2221–2229.

Nagel, K., Wagner, P. and Woesler, R. 2003. Still flowing: Approaches to traffic flow and traffic jam modeling. *Operations Research*, 51(5), pp.681–710.

Nagel, K., Wolf, D., Wagner, P. and Simon, P. 1998. Two-lane traffic rules for cellular automata: A systematic approach. *Physical Review E*, 58(2, Part A), pp.1425–1437.

National Research Council (U.S.). Committee for Determination of the State of the

Practice in Metropolitan Area Travel Forecasting 2007. *Metropolitan travel forecasting: current practice and future direction.* Transportation Research Board.

National Sustainable Transport Office, D.o.T. 2009. *Ireland's first national cycle policy framework 2009-2020.* Technical report, Department of Transport.

Neubert, L., Santen, L., Schadschneider, A. and Schreckenberg, M. 1999. Single-vehicle data of highway traffic: A statistical analysis. *Physical Review E*, 60(6), pp.6480–6490.

Newell, G. 1961. Nonlinear Effects in the Dynamics of Car Following. *Operations Research*, 9(2), pp.209 – 229.

Newell, G.F. 2002. A simplified car-following theory: a lower order model. *Transportation Research Part B*, 36(3), pp.195 – 205.

Nishinari, K. and Takahashi, D. 1998. Analytical properties of ultradiscrete Burgers equation and rule-184 cellular automaton. *Journal of Physics A - Mathematical and General*, 31(24), p.5439.

OpenMP Architecture Review Board 2011. *OpenMP Application Program Interface, Version 3.1* [Online]. Available from: http://openmp.org/wp/ [Accessed July 2013].

Ortuzar, J., Iacobelli, A. and Valeze, C. 2000. Estimating demand for a cycle-way network. *Transportation Research Part A*, 34(5), pp.353–373.

de Palma, A. and Marchal, F. 2002. Real Cases Applications of the Fully Dynamic METROPOLIS Tool-Box: An Advocacy for Large-Scale Mesoscopic Transportation Systems. *Networks and Spatial Economics*, 2, pp.347–369.

Papageorgiou, M., Diakaki, C., Dinopoulou, V., Kotsialos, A. and Wang, Y. 2003. Review of road traffic control strategies. *Proceedings of the IEEE*, 91(12), pp.2043–2067.

Payne, H.J. 1971. *Models of Freeway Traffic and Control.* Mathematical Models of Public Systems. Simulation Councils, Inc.

Pipes, L. 1953. An Operational Analysis of Traffic Dynamics. *Journal of Applied Physics*, 24(3), pp.274–281.

Prigogine, I. and Andrews, F. 1960. A Boltzmann-Like Approach for Traffic Flow. *Operations Research*, 8(6), pp.789–797.

Pucher, J. and Buehler, R. 2008. Cycling for everyone: Lessons from Europe. *Transportation Research Record*, 2074(-1), pp.58–65.

Pucher, J., Dill, J. and Handy, S. 2010. Infrastructure, programs, and policies to increase bicycling: An international review. *Preventive Medicine*, 50(Supplement 1), pp.106–125.

Raksuntorn, W. and Khan, S. 2003. Saturation Flow Rate, Start-Up Lost Time, and Capacity for Bicycles at Signalized Intersections. *Transportation Research Record*, 1852(-1), pp.105–113.

Richards, P. 1956. Shock Waves on the Highway. *Operations Research*, 4(1), pp.42–51.

Rickert, M. and Nagel, K. 2001. Dynamic traffic assignment on parallel computers in TRANSIMS. *Future Generation Computer Systems*, 17(5), pp.637 – 648.

Rickert, M., Nagel, K., Schreckenberg, M. and Latour, A. 1996. Two lane traffic simulations using cellular automata. *Physica A: Statistical Mechanics and its Applications*, 231(4), pp.534–550.

Ruskin, H.J. and Wang, R.L. 2002. Modelling traffic flow at an urban unsignalised intersection *IN:* Sloot, P., Tan, C., Dongarra, J. and Hoekstra, A. (eds.) *Computational Science - ICCS 2002, PT I, Proceedings*, *Lecture Notes in Computer Science* 2329. Springer-Verlag Berlin Heidelberg, pp.381–390.

Sanchez, E., Squillero, G. and Tonda, A. 2010. Evolving Individual Behavior in a Multi-agent Traffic Simulator *IN:* Di Chio, C., Cagnoni, S., Cotta, C., Ebner, M., Ek?rt, A., Esparcia-Alcazar, A., Goh, C., Merelo, J., Neri, F., Preuß, M., Togelius, J. and Yannakakis, G. (eds.) *Applications of Evolutionary Computation*, *Lecture Notes in Computer Science* 6024. Springer Berlin / Heidelberg, pp.11–20.

Sarkar, P. 2000. A brief history of cellular automata. *ACM Comput. Surv.*, 32(1), pp.80–107.

Sasaki, M. and Nagatani, T. 2003. Transition and saturation of traffic flow controlled by traffic lights. *Physica A: Statistical Mechanics and its Applications*, 325(3-4), pp.531–546.

Scellato, S., Fortuna, L., Frasca, M., Gómez-Gardeñes, J. and Latora, V. 2010. Traffic optimization in transport networks based on local routing. *The European Physical Journal B*, 73, pp.303–308.

Schadschneider, A. 1999. The Nagel-Schreckenberg model revisited. *European Physical Journal B*, 10(3), pp.573–582.

Schadschneider, A. 2008. Modelling of Transport and Traffic Problems *IN:* Umeo, H., Morishita, S., Nishinari, K., Komatsuzaki, T. and Bandini, S. (eds.) *Cellular Automata*, *Lecture Notes in Computer Science* 5191. Springer Berlin /Heidelberg, pp.22–31. ISBN 978-3-540-79991-7.

Schadschneider, A. and Schreckenberg, M. 1993. Cellular automation models and traffic flow. *Journal of Physics A - Mathematical and General*, 26(15), pp.L679–L683.

Schoenhof, M. and Helbing, D. 2009. Criticism of three-phase traffic theory. *Transportation Research Part B*, 43(7), pp.784–797.

Schreckenberg, M., Schadschneider, A., Nagel, K. and Ito, N. 1995. Discrete stochastic-models for traffic flow. *Physical Review E*, 51(4, Part A), pp.2939–2949.

Sewall, J., Wilkie, D., Merrell, P. and Lin, M.C. 2010. Continuum Traffic Simulation. *Computer Graphics Forum*, 29(2), pp.439–448.

Shay, E. and Khattak, A.J. 2010. Toward Sustainable Transport: Conventional and Disruptive Approaches in the U.S. Context. *International Journal of Sustainable Transportation*, 4, pp.14 – 40.

Si, B.F., Long, J.C. and Gao, Z.Y. 2008. Optimization model and algorithm for mixed traffic of urban road network with flow interference. *Science in China Series E*, 51, pp.2223–2232.

Simon, P.M. and Gutowitz, H.A. 1998. Cellular automaton model for bidirectional traffic. *Physical Review E*, 57(2), pp.2441–2444.

Simon, P.M. and Nagel, K. 1998. Simplified cellular automaton model for city traffic. *Physical Review E*, 58(2), pp.1286–1295.

Spyropoulou, I. 2007. Modelling a signal controlled traffic stream using cellular automata. *Transportation Research Part C*, 15(3), pp.175–190.

Stevanovic, A. 2009. SCOOT and SCATS: Closer Look into Their Operations *IN: Transportation Research Board 88th Annual Meeting.*

Sugiyama, Y., Fukui, M., Kikuchi, M., Hasebe, K., Nakayama, A., Nishinari, K., Tadaki, S. and Yukawa, S. 2008. Traffic jams without bottlenecks - experimental evidence for the physical mechanism of the formation of a jam. *New Journal of Physics*, 10(3), p.033001.

Takayasu, M. and Takayasu, H. 1993. 1/f noise in traffics model. *Fractals*, 1(4), pp.860–866.

Tanaka, K., Nagatani, T. and Masukura, S. 2008. Fundamental diagram in traffic flow of mixed vehicles on multi-lane highway. *Physica A - Statistical Mechanics and Its Applications*, 387(22), pp.5583–5596.

Tang, T., Huang, H., Zhao, S. and XG 2009. An Extended OV Model with Consideration of Driver's Memory. *International Journal of Modern Physics B*, 23(5), pp.743–752.

Tang, T.Q., Huang, H.J. and Shang, H.Y. 2010. A Dynamic Model for the Heterogeneous Traffic Flow Consisting of Car, Bicycle and Pedestrian. *International Journal of Modern Physics C*, 21(2), pp.159 – 176.

Taylor, S. and Damen, P. 2001. *Traffic Flow Models Allowing for Pedestrians and Cyclists*. Technical report, Austroads.

Tiwari, G., Fazio, J. and Gaurav, S. 2007. Traffic planning for non-homogeneous traffic. *SADHANA - Academy Proceedings In Engineering Sciences*, 32(4), pp.309–328.

Tiwari, G., Fazio, J., Gaurav, S. and Chatteerjee, N. 2008. Continuity equation validation for nonhomogeneous traffic. *Journal of Transportation Engineering - ASCE*, 134(3), pp.118–127.

Toledo, B., Munoz, V., Rogan, J., Tenreiro, C. and Valdivia, J. 2004. Modeling traffic through a sequence of traffic lights. *Physical Review E*, 70(1, Part 2), p.016107.

Tonguz, O.K., Viriyasitavat, W. and Bai, F. 2009. Modeling urban traffic: A cellular automata approach. *IEEE Communications Magazine*, 47(5), pp.142–150.

Treiber, M., Hennecke, A. and Helbing, D. 2000. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, 62(2, Part A), pp.1805–1824.

Treiber, M., Kesting, A. and Helbing, D. 2010. Three-phase traffic theory and two-phase models with a fundamental diagram in the light of empirical stylized facts. *Transportation Research Part B: Methodological*, 44(8-9), pp.983 – 1000.

Van Vliet, D. 1982. SATURN - A Modern Assignment Model. *Traffic Engineering and Control*, 23, pp.578–581.

Vandaele, N., Woensel, T.V. and Verbruggen, A. 2000. A queueing based traffic flow model. *Transportation Research Part D*, 5(2), pp.121 – 135.

von Neumann, J. 1944. *Theory of Games and Economic Behavior*. Princeton University Press.

Wagner, P., Nagel, K. and Wolf, D. 1997. Realistic multi-lane traffic rules for cellular automata. *Physica A: Statistical Mechanics and its Applications*, 234(3-4), pp.687–698.

Wahle, J. and Schreckenberg, M. 2001. A multi-agent system for on-line simulations based on real-world traffic data. *Hawaii International Conference on System Sciences*, 3, pp.3037–3045.

Wang, D., Feng, T. and Liang, C. 2008. Research on bicycle conversion factors. *Transportation Research Part A*, 42(8), pp.1129 – 1139.

Wang, R. and Ruskin, H.J. 2002. Modeling traffic flow at a single-lane urban roundabout. *Computer Physics Communications*, 147(1-2), pp.570–576.

Wang, R. and Ruskin, H.J. 2003. Modelling traffic flow at a multilane intersection *IN:* Kumar, V., Gavrilova, M., Tan, C. and LEcuyer, P. (eds.) *Computational Science and Its Applications - ICCSA 2003, PT 1, Proceedings*, *Lecture Notes in Computer Science* 2667. Springer-Verlag Berlin Heidelberg, pp.577–586.

Wang, R. and Ruskin, H.J. 2006. Modelling traffic flow at multi-lane urban roundabouts. *International Journal of Modern Physics C*, 17(5), pp.693–710.

Wardman, M., Tight, M. and Page, M. 2007. Factors influencing the propensity to cycle to work. *Transportation Research Part A*, 41(4), pp.339–350.

Wardrop, J. 1952. Some Theoretical Aspects of Road Traffic Research. *Proceedings of the Institute of Civil Engineers*, 1, pp.325–362.

Wilkinson, W.C., Clarke, A., Epperson, B. and Knoblauch, R. 1994. *The Effects of Bicycle Accommodations on Bicycle/Motor Vehicle Safety and Traffic Operations.* Technical report, Federal Highway Administration (U.S.), McLean.

Windover, J.R. and Cassidy, M.J. 2001. Some observed details of freeway traffic evolution. *Transportation Research Part A*, 35(10), pp.881 – 894.

Wolfram, S. 1986. *Theory and Applications of Cellular Automata.* World Scientific Publishing.

Woodcock, J., Edwards, P., Tonne, C., Armstrong, B.G., Ashiru, O., Banister, D., Beevers, S., Chalabi, Z., Chowdhury, Z., Cohen, A., Franco, O.H., Haines, A., Hickman, R., Lindsay, G., Mittal, I., Mohan, D., Tiwari, G., Woodward, A. and Roberts, I. 2009. Public health benefits of strategies to reduce greenhouse-gas emissions: urban land transport. *The Lancet*, 374(9705), pp.1930 – 1943.

Wright, L. and Fulton, L. 2005. Climate change mitigation and transport in developing nations. *Transport Reviews*, 25(6), pp.691–717.

Wu, Q., Li, X., Hu, M. and Jiang, R. 2005. Study of traffic flow at an unsignalized T-shaped intersection by cellular automata model. *European Physical Journal B*, 48(2), pp.265–269.

Xie, D., Gao, Z. and Zhao, X. 2010. Combined Cellular Automaton Model for Mixed Traffic Flow With Non-Motorized Vehicles. *International Journal of Modern Physics C*, 21(12), pp.1443 – 1455.

Xie, D.F., Gao, Z.Y., Zhao, X.M. and Li, K.P. 2009. Characteristics of mixed traffic flow with non-motorized vehicles and motorized vehicles at an unsignalized intersection. *Physica A: Statistical Mechanics and its Applications*, 388(10), pp.2041–2050.

Xuan, Y.G., Gayah, V., Daganzo, C.F. and Cassidy, M.J. 2008. *Multimodal Traffic at Isolated Signalized Intersections: New Management Strategies to Increase Capacity.* UC Berkeley Center for Future Urban Transport. Working Paper UCB-ITS-VWP-2009-10.

Yuan, Y.M., Jiang, R., Hu, M.B., Wu, Q.S. and Wang, R.L. 2009. Traffic flow characteristics in a mixed traffic system consisting of ACC vehicles and manual vehicles: A hybrid modelling approach. *Physica A: Statistical Mechanics and its Applications*, 388(12), pp.2483–2491.

Zacharias, J. 2002. Bicycle in Shanghai: Movement patterns, cyclist attitudes and the impact of traffic separation. *Transport Reviews*, 22(3), pp.309–322.

Zhang, X., Guo, Y.M. and Yang, X.K. 2007. A Model of Potential Capacity of Right Turn Movement at Signalized Intersections Under Mixed Traffic Conditions *IN: Proceedings of the 3rd Urban Street Symposium, June 24-27, 2007, Seattle, WA.*

Zhao, B.H., Hu, M.B., Jiang, R. and Wu, Q.S. 2009. A Realistic Cellular Automaton Model for Synchronized Traffic Flow. *Chinese Physics Letters*, 26(11), p.118902.

Appendix A

Topological Element Interface for Agents

The following functions, expressed in C++, constitute the interface exposed by the ***Topological Element*** class for use by the ***Agent*** class:

- **const DORC::P getDORC (const TSPositionCWFC * ppos);**

  This function returns the closest DORC (Divergence or Conflict) object, allowing the calling Agent to 'tap into' the abstract representation of the network at the point where it is located. Parameters:

  – **ppos** : a position in the network

  Returns: the closest DORC in the network

- **int inspectFwdForObstructing (TSPositionCWFC * ppos, int max_distance, TSDirectionList::P pdirs);**

  This function inspects a track for the first impinged cell. Parameters:

  – **ppos** - starting position for the inspection

  – **max_distance** - the distance to cover during inspection

  – **pdirs** - the directions to take if divergences are encountered (the route)

Returns: the number of unimpinged cells

- **int inspectFwdForHighImpactInner (const TSVehicleCWFC * pveh, unsigned char other_vehicle_type, int max_distance, TSDirectionList::P pdirs);**

  This function inspects an adjacent track towards the kerb for the closest vehicle ahead (used by cars to look for bicycles) Parameters:

  - **pveh** - the inspecting vehicle, which includes the starting position for the inspection
  - **other_vehicle_type** - the other vehicle type that is being searched for
  - **max_distance** - the distance to cover during inspection
  - **pdirs** - the directions to take if divergences are encountered (the route)

  Returns: the distance to the closest vehicle in the adjacent track of type other_vehicle_type

- **int inspectFwdForHighImpactOuter (const TSVehicleCWFC * pveh, unsigned char other_vehicle_type, int max_distance, TSDirectionList::P pdirs);**

  This function inspects the adjacent track towards the middle of the road in the same way as the previous function. Not currently used but provided for the case of lane changing.

- **void move (TSVehicleCWFC * pveh, int distance, TSDirectionList::P pdirs);**

  This function moves a vehicle. Parameters:

  - **pveh** - the inspecting vehicle, which includes the starting position for the move
  - **the number of cells that the vehicle should advance** - the other vehicle type that is being searched for
  - **pdirs** - the directions to take if divergences are encountered (the route)

Appendix B

Field Data Summary Tables

The camera positions and lane numbers referred to in the table captions are shown in Figure 5.2.

	Lane 2						Lane 1				
#	ρ_{B}	q_{B}	ρ_{C}	q_{C}	ρ_{L}	q_{L}	ρ_{B}	q_{B}	ρ_{C}	q_{CT}	q_{CL}
1	0.0243902	0.0243902	0.585366	0.268293	0	0	0	0	0.195122	0.0487805	0.0243902
2	0.00584795	0.0175439	0.315789	0.333333	0.0233918	0.0175439	0.0526316	0.105263	0.0116959	0.0175439	0
3	0.0517241	0.0689655	0.425287	0.275862	0	0	0.091954	0.137931	0.252874	0.0689655	0.0172414
4	0.00584795	0.0175439	0.25731	0.192982	0.0233918	0.0175439	0.0350877	0.0526316	0.163743	0.0175439	0.0175439
5	0.0128205	0.0192308	0.102564	0.0961538	0.0512821	0.0192308	0.0961538	0.134615	0.0384615	0.0192308	0.0384615
6	0.0246914	0.0555556	0.296296	0.185185	0.123457	0.037037	0.0555556	0.0740741	0.0246914	0.0185185	0
7	0	0	0.460606	0.272727	0.0969697	0.0181818	0.0848485	0.0727273	0.145455	0.0909091	0
8	0.0769231	0.153846	0.564103	0.346154	0	0	0.025641	0.0769231	0	0	0
9	0	0	0.496124	0.27907	0.186047	0.0465116	0.0852713	0.116279	0	0	0.0697674
10	0	0	0.387597	0.302326	0	0	0.0310078	0.0697674	0.0775194	0.0465116	0
11	0	0	0.578947	0.368421	0	0	0.166667	0.236842	0.0526316	0.0263158	0
12	0.047619	0.0571429	0.666667	0.314286	0	0	0.12381	0.228571	0.0761905	0.0285714	0
13	0	0	0.396396	0.27027	0.144144	0.027027	0.0990991	0.0810811	0.126126	0.108108	0.027027
14	0.0588235	0.0588235	0.627451	0.294118	0	0	0.0980392	0.117647	0.117647	0.0588235	0.0588235
15	0.0227273	0.0227273	0.606061	0.295455	0	0	0.151515	0.136364	0.19697	0.0454545	0
16	0.0396825	0.047619	0.444444	0.190476	0.126984	0.0238095	0.111111	0.0714286	0.126984	0.142857	0

Table B.1: Macroscopic values obtained from field data, camera position A, lanes 1 and 2. ρ_{B}, ρ_{C} and ρ_{L} are bicycle, car and long vehicle densities, respectively, converted to *vehicles per cell.* q_{B}, q_{C}, q_{CT}, qCL and q_{L} are flow values for bicycles, cars, through-moving cars, left-turning cars and long vehicles, respectively, in *vehicles per second.* The values are listed in chronological order of green phases to which they pertain, with corresponding ordinal numbers shown in the first column.

	Lane 2						Lane 1					
#	ρ_{B}	q_{B}	ρ_{C}	q_{C}	ρ_{L}	q_{L}	ρ_{B}	q_{B}	ρ_{C}	q_{C}	ρ_{L}	q_{L}
1	0	0	0.347594	0.352941	0	0	0.0200535	0.0588235	0	0	0	0
2	0	0	0.19656	0.189189	0.22113	0.0810811	0.14742	0.324324	0	0	0	0
3	0.0516529	0.0909091	0.299587	0.295455	0.0413223	0.0227273	0.18595	0.340909	0.0309917	0.0454545	0.103306	0.0227273
4	0.0988142	0.130435	0.197628	0.173913	0	0	0.108696	0.26087	0.118577	0.0869565	0.237154	0.0434783
5	0.0202922	0.0357143	0.284091	0.267857	0	0	0.186688	0.321429	0.162338	0.0892857	0	0
6	0.0181818	0.04	0.263636	0.32	0	0	0.1	0.22	0.0454545	0.08	0.0363636	0.02
7	0	0	0.377358	0.396226	0	0	0.124357	0.283019	0	0	0	0
8	0.103896	0.171429	0.311688	0.257143	0	0	0.38961	0.628571	0	0	0	0
9	0.034585	0.0869565	0.296443	0.282609	0	0	0.108696	0.173913	0.0395257	0.0434783	0.118577	0.0434783
10	0.0222816	0.0392157	0.276292	0.294118	0.0356506	0.0196078	0.115865	0.313725	0.00891266	0.0196078	0.0534759	0.0196078
11	0.00826446	0.0181818	0.322314	0.309091	0.0495868	0.0181818	0.0247934	0.0727273	0.0661157	0.0727273	0.0330579	0.0181818
12	0.0252525	0.0833333	0.30303	0.333333	0	0	0.157828	0.333333	0.0252525	0.0277778	0.0757576	0.0277778
13	0.03367	0.0555556	0.30303	0.240741	0	0	0.0505051	0.12963	0.227273	0.166667	0.0673401	0.037037
14	0	0	0.236742	0.25	0	0	0.0804924	0.145833	0.104167	0.0833333	0.094697	0.0416667
15	0.00721501	0.015873	0.266955	0.285714	0.02886	0.015873	0.0829726	0.190476	0.00721501	0.015873	0.15873	0.0793651

Table B.2: Macroscopic values obtained from field data, camera position B, lanes 1 and 2. ρ_{B}, ρ_{C} and ρ_{L} are bicycle, car and long vehicle densities, respectively, converted to *vehicles per cell*. q_{B}, q_{C} and q_{L} are flow values for bicycles, cars and long vehicles, respectively, in *vehicles per second*. The values are listed in chronological order of green phases to which they pertain, with corresponding ordinal numbers shown in the first column.

	Lane 3		Lane 4					
#	ρ_{B}	q_{B}	ρ_{B}	q_{B}	ρ_{C}	q_{C}	ρ_{L}	q_{L}
1	0.0732601	0.138462	0.003663	0.0153846	0.161172	0.246154	0.029304	0.0153846
2	0.0892857	0.25	0.0042517	0.0178571	0.221088	0.267857	0.0340136	0.0178571
3	0.0755858	0.222222	0.0566893	0.111111	0.15873	0.190476	0.0604686	0.031746
4	0.102041	0.22449	0.0242954	0.0612245	0.126336	0.142857	0	0
5	0.119048	0.270833	0.00496032	0.0416667	0.208333	0.229167	0	0
6	0.0810537	0.212766	0	0	0.182371	0.191489	0	0
7	0.058309	0.163265	0.0145773	0.0204082	0.165209	0.163265	0	0
8	0.102041	0.174603	0.00755858	0.015873	0.188964	0.15873	0	0
9	0.0496032	0.104167	0.00496032	0.0208333	0.337302	0.416667	0	0
10	0.0952381	0.22	0.00952381	0.04	0.171429	0.16	0.0380952	0.06
11	0.0537634	0.177419	0.0192012	0.0322581	0.230415	0.241935	0	0
12	0.0145773	0.0408163	0	0	0.281827	0.346939	0	0
13	0.0145773	0.0612245	0.0194363	0.0612245	0.330418	0.285714	0	0
14	0.0740741	0.177778	0	0	0.31746	0.377778	0.0634921	0.0222222

Table B.3: Macroscopic values obtained from field data, camera position B, lanes 3 and 4. ρ_{B}, ρ_{C} and ρ_{L} are bicycle, car and long vehicle densities, respectively, converted to *vehicles per cell*. q_{B}, q_{C} and q_{L} are flow values for bicycles, cars and long vehicles, respectively, in *vehicles per second*. The values are listed in chronological order of green phases to which they pertain, with corresponding ordinal numbers shown in the first column.

Appendix C

Development and Run Environments

The development and run environments used in the implementation of the simulation model are specified in Tables C.1 and C.2. The implementation code is available for download at https://www.dropbox.com/s/lep7j2xbmtrkvde/jvphdcode.zip

Development	
Programming language	C++
Operating system	Linux (Ubuntu v10.04)
Compiler	g++ v4.5
Integrated Development Environment	Eclipse v3.7

Table C.1: Development environment

Run	
DCU School of Computing cluster Ampato	56 compute nodes, each with: 2 x Quad-Core Intel Xeon E5430 2.66GHz 2x6MB L2 Cache, 8GB RAM
Irish Centre for High-End Computing cluster Stokes	320 compute nodes, each with: 2 x Hex-core Intel Xeon X5650 2.66GHz 2x6MB L2 Cache, 24GB RAM

Table C.2: Run environments

Appendix D

Sample input and output files

Below are listings of a sample configuration file, an example of simulation output and an example of a snapshot of simulation state, used mostly for debugging.

```
:CONFIGURATION
AGENT_TYPE TSAgentCBB
SOURCE 0 TSTEXSCASource TSAgentCBB
SINK 0 TSTEXSCASink
TE 0 TSTEXSCAFullSingle_4_6
TE 1 TSTEXSCASRS_4_6
TE 2 TSTEXSCASRS_4_6
TE 3 TSTEXSCASRS_4_6
TE 4 TSTEXSCASRS_4_6
CONN 0 1 1 0 0
CONN 1 2 1 0 2
CONN 2 0 1 3 0
CONN 3 0 3 4 0
:ARGUMENTS
TSSimulation "" 4000
TSAgentCBB 3 2 0.1 0.1   1 1 1 1
Routing          E
TSVehicleList    1 1 2 1 200 3600 1
SINK0 6 2 B 0 0 C 2 0
TE0 2000 2000 R J 0 0 0
TE1 2000 2000 20 2 C 0 19 200 3600 1 B 0 39 200 3600 1 1 0 2 0 B 0 1 0 0.1 0 C 2 2 0 0.1 0
TE2 2000 2000 20 2 C 0 19 200 3600 1 B 0 39 200 3600 1 1 0 2 0 B 0 1 0 0.1 0 C 2 2 0 0.1 0
TE3 2000 2000 20 2 C 0 19 200 3600 1 B 0 39 200 3600 1 0 1 1 0
TE4 2000 2000 20 2 C 0 19 200 3600 1 B 0 39 200 3600 1 0 1 1 0
```

Figure D.1: A sample configuration file: this file is the configuration for the simulation of an intersection of two one-way single-lane roads.

```
INFO: overallTickCount is 3600
TSVehicleList_0_0_0_0: 3.69056 0.102222 41.7011 36.1223
TSVehicleList_0_0_0_1: 3.96 0.106389 43.7572 37.154
TSVehicleList_0_0_0_A: 7.65056 0.208611 42.7497 36.6485
TSVehicleList_0_0_1_0: 4.19861 0.0908333 84.5199 46.2813
TSVehicleList_0_0_1_1: 4.13278 0.0891667 84.5202 46.3084
TSVehicleList_0_0_1_A: 8.33139 0.18 84.5201 46.2948
TSVehicleList_0_0_A_A: 15.9819 0.388611 62.0972 41.1165
TSVehicleList_A_A_A_A: 15.9819 0.388611 62.0972 41.1165 (15.9822)
TSVehicleList_A_A_0_0: 3.69056 0.102222 41.7011 36.1223
TSVehicleList_A_A_0_1: 3.96 0.106389 43.7572 37.154
TSVehicleList_A_A_0_A: 7.65056 0.208611 42.7497 36.6485 (7.65056)
TSVehicleList_A_A_1_0: 4.19861 0.0908333 84.5199 46.2813
TSVehicleList_A_A_1_1: 4.13278 0.0891667 84.5202 46.3084
TSVehicleList_A_A_1_A: 8.33139 0.18 84.5201 46.2948 (8.33167)
TE1(TSTEXSCASRS_4_6).SensorAtC00019: 0.302222   0.103056
TE1(TSTEXSCASRS_4_6).SensorAtB00039: 0.106944   0.0880556
TE2(TSTEXSCASRS_4_6).SensorAtC00019: 0.167778   0.106111
TE2(TSTEXSCASRS_4_6).SensorAtB00039: 0.0838889  0.0919444
TE3(TSTEXSCASRS_4_6).SensorAtC00019: 0.0480556  0.103889
TE3(TSTEXSCASRS_4_6).SensorAtB00039: 0.0541667  0.0894444
TE4(TSTEXSCASRS_4_6).SensorAtC00019: 0.0488889  0.104722
TE4(TSTEXSCASRS_4_6).SensorAtB00039: 0.0477778  0.0905556
INFO: Simulation ended successfully.
INFO: time (milliseconds): CPU: 142, real: 145
```

Figure D.2: Sample simulation output: this is the output produced for the simulation configured as in Figure D.1.

```
C|2  . .
C<2  . . . .
C-2  . .
C^2  . . 3 .
B|0 ....
B<0 ....
B-0 ....
B^0 ....
[2000] TE1(TSTEXSCASRS_4_6)
C|2  . . . . . 0 . 1 . . . . . . . . . 0 0 0
B|0 ...................2....................
[2000] TE2(TSTEXSCASRS_4_6)
C|2  . . . . . . . . 2 . . . . . . . . . . .
B|0 .......................2................
[2000] TE3(TSTEXSCASRS_4_6)
C|2  . . . . 1 . . . . . . 1 . 2 . . . . . .
B|0 ............2............2.......2......
[2000] TE4(TSTEXSCASRS_4_6)
C|2  . . . . . . . . . . . 1 . 1 . . . . . .
B|0 ...........2.................1..........
```

Figure D.3: Sample simulation snapshot: this is the snapshot at time-step 2000 for the simulation configured as in Figure D.1.

Appendix E

List of Publications

Vasic, J. and Ruskin, H.J. 2013. Interaction of cars and bicycles on a one-way road intersection - a network CA-based model *IN:* Kozlov, V.V., Buslaev, A.P., Bugaev, A.S., Yashina, M.V., Schadschneider, A. and Schreckenberg, M. (eds.) *Traffic and Granular Flow 2011.* Springer.
http://www.springer.com/mathematics/applications/book/978-3-642-39668-7

Vasic, J. and Ruskin, H.J. 2012. Agent-based Space-time Discrete Simulation of Urban Traffic Including Bicycles. *Procedia Computer Science*, 10(0), pp.860 – 865.
http://www.sciencedirect.com/science/article/pii/S1877050912004693

Vasic, J. and Ruskin, H.J. 2012. A Cellular Automata-Based Network Model for Heterogeneous Traffic: Intersections, Turns and Their Connection *IN:* Georgios Ch. Sirakoulis, S.B. (ed.) *ACRI 2012*, *Lecture Notes in Computer Science* 7495. Berlin / Heidelberg: Springer, pp.835–844.
http://link.springer.com/chapter/10.1007/978-3-642-33350-7_87

Vasic, J. and Ruskin, H.J. 2012. Cellular automata simulation of traffic including cars and bicycles. *Physica A: Statistical Mechanics and its Applications*, 391(8), pp.2720–2729.
http://www.sciencedirect.com/science/article/pii/S0378437111009150

Vasic, J. and Ruskin, H.J. 2012. A Discrete Simulation Model for Traffic Including

Bicycles on Urban Networks, Applied to Intersection of Two One-Way Streets *IN:* Wyrzykowski, R., Dongarra, J., Karczewski, K. and Wasniewski, J. (eds.) *PPAM 2011*, *Lecture Notes in Computer Science* 7204. Berlin / Heidelberg: Springer, pp.598–607.
http://www.springerlink.com/content/n608435hj6580530/

Vasic, J. and Ruskin, H.J. 2012. A CA-Based Model for City Traffic Including Bicycles *IN:* Polyzos, S. (ed.) *Urban Development.* Rijeka: InTech, pp.80–92.
http://www.intechopen.com/books/urban-development/a-ca-based-model-for-city-traffic-including-bicycles

Vasic, J. and Ruskin, H.J. 2011. A Discrete Flow Simulation Model for Urban Road Networks, with Application to Combined Car and Single-File Bicycle Traffic *IN:* Murgante, B., Gervasi, O., Iglesias, A., Taniar, D. and Apduhan, B. (eds.) *Computational Science and Its Applications - ICCSA 2011*, *Lecture Notes in Computer Science* 6782. Berlin / Heidelberg: Springer, pp.602–614.
http://dx.doi.org/10.1007/978-3-642-21928-3_44

Vasic, J. and Ruskin, H.J. 2011. Throughput and delay in a discrete simulation model for traffic inlcuding bicycles on urban networks *IN: Proceedings of ITRN 2011, 31 August – 1 September 2011, UCC, Cork, Ireland.*
http://www.itrn.ie/uploads/sesE3_ID129.pdf