Macroscopic Traffic Model Validation of Large Networks and the Introduction of a Gradient Based Solver

ADAM JAMES POOLE

Macroscopic Traffic Model Validation of Large Networks and the Introduction of a Gradient Based Solver

Adam J. Poole

Thesis submitted towards the
degree of Doctor of Philosophy

Computational Mechanics Group
School of Engineering and Computing Sciences
Durham University
United Kingdom

July 2017

Macroscopic Traffic Model Validation of Large Networks and the Introduction of a Gradient Based Solver

Adam J. Poole

Abstract

Traffic models are important for the evaluation of various Intelligent Transport Systems and the development of new traffic infrastructure. In order for this to be done accurately and with confidence the correct parameter values of the model must be identified. The focus of this thesis is the identification and confirmation of these parameters, which is model validation. Validation is performed on two different models; the first-order CTM and the second-order METANET model. The CTM is validated for two UK sites of 7.8 and 21.9 km and METANET for the same two sites using a variety of meta-heuristic algorithms. This is done using a newly developed method to allow for the optimisation method to determine the number of parameters to be used and the spatial extent of their application. This allows for the removal of expert engineering knowledge and ad-hoc decomposition of networks.

This thesis also develops a methodology by use of Automatic Differentiation to allow gradient based optimisation to be used. This approach successfully validated the METANET model for the 21.9 km site and also a large network surrounding the city of Manchester of 186.9 km. This proves that gradient based optimisation can be used for the macroscopic traffic model validation problem. In fact the performance of the developed gradient method is superior to the meta-heuristics tested for the same sites. The methodology defined also allows for more data to be obtained from the model such as its Jacobian and the sensitivity of the objective function being used relative to the individual parameters. Space-Time contour plots of this newly acquired data show structures and shock waves that are not visible in the mean speed contour diagrams.

Declaration

The work in this thesis is based on research carried out in the Computational Mechanics Group, School of Engineering and Computing Sciences, Durham University. No part of this report has been submitted elsewhere for any other degree or qualification and it is all my own work unless referenced to the contrary in the text.

Parts of this work have been published in the following:

Journals

A. Poole and A. Kotsialos, "Swarm intelligence algorithms for macroscopic traffic flow model validation with automatic assignment of fundamental diagrams," *Applied Soft Computing*, vol. 38, pp. 134–150, 2016

A. Poole and A. Kotsialos, "Second order macroscopic traffic flow model validation using automatic differentiation with resilient backpropagation and particle swarm optimisation algorithms," *Transportation Research Part C: Emerging Technologies*, vol. 71, pp. 356–381, 2016

A. Poole and A. Kotsialos, "METANET validation of the large scale Manchester ring-road network using gradient based and particle swarm optimisation," *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, no. 99, pp. 1–11, 2017

Conferences

A. Poole and A. Kotsialos, "METANET model validation using a genetic algorithm," in *Proc. of the 13th IFAC Symp. on Control in Transportation Systems*, vol. 13, pp. 7–12, 2012

A. Poole and A. Kotsialos, "Deterministic model validation with a new approach to fundamental diagrams of the METANET model," in *Transportation Research Board 92nd Annual Meeting*, no. 13-4046 in 92, 2013

A. Kotsialos and A. Poole, "Autonomic systems design for ITS applications," in *Intelligent Transportation Systems (ITSC), 2013 16th International IEEE Conference on*, pp. 178–183, Oct 2013

A. Poole and A. Kotsialos, "Traffic Flow Model Validation Using METANET, ADOL-C and RPROP," in *Proc. of the 14th IFAC Symposium on Control in Transportation Systems*, (Istanbul, Turkey), pp. 291–296, 2016

Book Chapter

A. Kotsialos and A. Poole, "Autonomic systems design for its applications: Modelling and route guidance," in *Autonomic Road Transport Support Systems* (T. L. McCluskey, A. Kotsialos, J. P. Müller, F. Klügl, O. Rana, and R. Schumann, eds.), pp. 131–145, Birkhäuser Basel, 2016

Acknowledgements

This thesis would not be possible without the support and guidance provided by my supervisor Dr. Apostolos Kotsialos. Also vital to this work was the funding provided by the UK Engineering and Physical Sciences Research Council (EPSRC). The author would also like to thank the Highways Agency and Mott MacDonald for the provision of MIDAS data.

Thanks to my family for their patience, Dr. Jonathan Asquith and Dr. Sarah Dempsey for their support and motivation. I would also like to thank my fellow PhD students particularly for some interesting discussions over lunch.

Adam Poole
Durham, July 2017

Contents

List of Figures

List of Tables

List of Algorithms

Acronyms

EA	Evolutionary Algorithm
GA	Genetic Algorithm
PSO	Particle Swarm Optimisation
CS	Cuckoo Search
RPROP	Resilient backPROPagation
LWR	Lighthill-Whitham and Richards
AR	Aw and Rascle
GKT	Gas Kinetic
FD	Fundamental Diagram
AAFD	Automatic Assignment of Fundamental Diagrams
ARZ	Aw, Rascle and Zhang
PW	Payne and Whitham
CTM	Cell Transmission Model
AD	Automatic Differentiation
META	Modèle d'Ecoulement du Traffic Autoroutier
METANET	Modèle d'Ecoulement du Traffic Autoroutier (META) applied to NETworks
ITS	Intelligent Transportation Systems
BC	Boundary Condition
IC	Initial Condition
MIDAS	Motorway Incident Detection and Automatic Signalling
BLAS	Basic Linear Algebra Subprograms

CFL	Courant–Friedrichs–Lewy
PDE	Partial Differential Equation
MAPE	Monitor–Analyse–Plan–Execute
SCT	Source Code Transformation
OO	Operator Overloading
IPC	Inter-Process Communication
CPU	Central Processing Unit
GUI	Graphical User Interface
OS	Operating System
SHM	Shared Memory

Nomenclature

A	Number of dimensions	
B	Number of sections	
$C_1,\ C_2$	Acceleration coefficients bounds	
$E_{\rho,j}$	Density squared error at detector j	
$E_{q,j}$	Flow squared error at detector j	
$E_{v,j}$	Speed squared error at detector j	
G	Gaussian random number	
H	Vector of start links of linear sections	
I_n	Set of in-links to node n	
J'	Scaled objective function	
$J(\mathbf{z})$	Objective function	
J_{norm}	Normalisation fitness	
J_{p}	Penalty component of the objective function J	
J_{s}	Squared error component of the objective function J	
K	Time horizon	
$L_{\text{s},b}$	Length of linear section b	links
M	Number of links	
N_m	Number of segments in link m	
O_n	Set of out-links of node n	
Q_n	Total flow into node n	veh/hr
R	Traffic receive	veh/hr
S	Traffic supply	veh/hr
T	Model time step	s
U	Mapping Algorithm	
V	Equilibrium speed	km/h
W	Boolean vector stating if link has been traversed	
Y	Number of locations measured data available for	
ΔJ	Fitness differential	
Δ	RPROP update step	
$\Delta\lambda$	Number of lanes being dropped	
Γ	Maximum number of fundamental diagrams allowed per section	
Φ	Set of destinations	
Θ	Set of origins	

α	Fundamental diagram parameter	
α_{CE}	Randomness parameter for Chaos Enhanced PSO	
α_{s}	Step size	
$\bar{q}$	Capacity	veh/hr
β	Reduction speed	
β_{CE}	Attraction parameter for Chaos Enhanced PSO	
β_n^m	Turning rate, proportion of flow from node n to link m	
δ	Merging constant	
δ_0	Initial step size	
δ_c	Acceleration rate	
ℓ	Lane index	
η^+	RPROP extension parameter	
η^-	RPROP contraction parameter	
$\gamma_{\widehat{m}}$	Length of fundamental diagram $\widehat{m}$	links
$\hat{K}$	Data time horizon	
$\hat{T}$	Data time interval	s
$\hat{T}$	Data time step	
$\hat{\rho}$	Loop detector density	veh/km/lane
$\hat{k}$	Data time interval	
$\hat{k}_{\mathrm{START}}$	Data time step that corresponds to $k = 0$	
$\hat{q}$	Loop detector flow	veh/hr
$\hat{v}$	Loop detector speed	km/hr
ι	Population index	
κ	Stability constant	
λ	Number of lanes	
λ_1	Eigenvalue 1	
λ_2	Eigenvalue 2	
$\mathbf{J}$	Jacobian of the traffic model	
$\mathbf{U}$	PSO velocity vector	
$\mathbf{Z}$	Population of solution vectors	
$\mathbf{Z}_{\mathrm{mate}}$	Genetic Algorithm mating pool	
$\mathbf{Z}_{\mathrm{new}}$	Interim genetic algorithm population	
$\mathbf{d}$	Disturbance vector	
$\mathbf{s}$	Search direction vector	
$\mathbf{x}$	Model state vector	
$\mathbf{x}_0$	Model initial condition	
$\mathbf{y}$	Measured data vector	
$\mathbf{z}$	Parameter vector	
$\mathbf{z}_c$	Constant model parameter vector	
$\mathbf{z}_{\mathrm{FD},\widehat{m}}$	Parameter vector for road properties of fundamental diagram $\widehat{m}$	
$\mathbf{z}_{\mathrm{L},m}$	Parameter vector for road properties at link m	

$\mathcal{C}$	Maximum number of iterations in optimisation algorithm	
$\mathcal{L}_a$	Length of dimension a	
$\mathcal{P}$	Traffic pressure	
$\mathcal{S}$	Evolutionary state	
$\mathfrak{b}_\iota$	Iteration when best solution found for particle ι	
$\mathfrak{c}$	Optimisation algorithm iteration number	
$\mathfrak{c}_\mathrm{r}$	Restart period	
$\mathfrak{d}^2$	Swarm diversity	
$\mathfrak{f}$	Evolutionary factor	
$\mathfrak{n}_\iota$	Index of best particle in the neighbourhood of ι	
$\mathfrak{r}_\iota$	Remainder of population member ι	
$\mathfrak{s}$	Convergence speed	
$\mathfrak{s}_\iota$	Mean euclidean distance between particle ι and the rest of the swarm	
$\mathfrak{u}$	Membership classification	
μ	Link index	
ν	Anticipation constant	
ω	Inertia weight	
$\overline{J}$	Mean of objective function values	
ϕ	Weaving constant	
ψ	Number of nodes	
ρ	Traffic density	veh/km/lane
ρ_max	Maximum traffic density	veh/km/lane
ρ_b	Boundary density at destination b	veh/km/lane
ρ_{cr}	Critical density	veh/km/lane
$\rho_{m,i}$	Traffic density in segment i of link m	veh/km/lane
σ	Standard deviation	
τ	Relaxation time	s
θ	Iterations since best solution found	
θ_max	Restart threshold	
$\tilde{J}$	Best objective value found to date in search	
$\tilde{\mathbf{z}}$	Best solution vector found to date in search	
$\widehat{M}$	Number of fundamental diagrams	
$\widehat{\mathbf{z}}$	Fundamental diagram based model parameter vector	
$\widehat{m}$	Fundamental diagram index	
ξ	Lower limit on extension parameter η^+	
ζ	Extension parameter η^+ reduction increment	
a	Dimension index	
b	Destination index	
b	Section index	
c	Contraction factor	
c_0	Sound wave speed	km/h

$c_1,\ c_2$	Acceleration coefficients	
c_{mutate}	Non-uniform mutation parameter	
d	Demand	veh/hr
e_ι	Expected value for population member ι	
g	Queue length	veh
i	Segment index	
k	Time step	
m	Link index	
n	Node index	
n_σ	Number of standard deviations to set truncation limit	
n_{elite}	Number of elite individuals	
$n_{\text{d},m}$	Downstream node of link m	
$n_{\text{u},m}$	Upstream node of link m	
o	Origin index	
p_ι	Selection probability of population member ι	
p_{cross}	Crossover probability	
p_{mutate}	Mutation probability	
p_n^m	Priority, proportion of flow into node n from link m	
p_{ABC}	Probability of Artificial Bee Colony operator	
p_{GA}	Probability of Genetic Algorithm operator	
q	Traffic flow	veh/hr
q_e	Equilibrium traffic flow	veh/hr
$q_{m,i}$	Traffic flow leaving link segment i of link m	veh/km
$r_{I,n}$	Major in-link of node n	
$r_{O,n}$	Major out-link of node n	
$s_{I,n}$	Minor in-link of node n	
$s_{O,n}$	Minor out-link of node n	
t	Time	s
u	A component of $\mathbf{U}$	
v	Traffic velocity	km/h
v_f	Free speed	km/h
$v_{m,i}$	Mean speed of traffic in segment i of link m	km/h
w_p	Penalty term weight scalar	
x	Space	m
z	A component of $\mathbf{Z}$	

Chapter 1

Introduction

One of the current issues in the modern world concerns vehicular traffic, in particular, congestion. Congestion leads to large emissions as idling vehicles do not run at peak efficiency. The time spent in traffic and the increased fuel costs also results in a significant economic impact. These issues stem from the roads not running at optimal capacity, or a lack of sufficient control schemes to fully utilise the available facilities. The field of Intelligent Transportation Systems (ITS) aims to combat this by designing suitable control strategies to fully optimise the current infrastructure. ITS use models for tasks like traffic prediction, state and travel time estimation, and real time model-based predictive control. Automatic incident systems also make use of such models. In some cases the infrastructure may not have suitable capacity for the demands placed on them as populations and urban areas grow in size. This then requires the current networks to be upgraded or new roads built. Due to the significant costs involved in these projects, feasible and optimal solutions must be found. There is a requirement for accurate and reliable modelling, either for the development, testing and execution of controllers for ITS systems or to analyse the potential impacts of changes to current infrastructure.

These demands have put a need on the development of accurate simulators that are able to predict and replicate the nature of traffic flow. The models can be used in a variety of ways to help improve the safety, efficiency and capacity of roads. But regardless of the final use of the traffic model, the key aspect should be its robustness and ability to correctly capture the traffic flow dynamics. For the models to be applied in a meaningful way and to provide the desired results it is vital to perform model validation. Every traffic model has some parameters that define physical or non-physical characteristics and these parameters must be selected appropriately so that the model can be used credibly for its intended purpose.

This thesis is concerned with the traffic model validation problem and aims to develop a system that can be used reliably across networks of various sizes and topologies. In this work a selection of real-world sites from the UK are evaluated. Each of the sites is of a different scale and has it own unique properties. These sites are motorway networks and due to the large volumes of traffic involved this thesis considers only the macroscopic traffic models that are computationally efficient. A further aim of this thesis is to develop a methodology such that a gradient based solver can be applied to the calibration problem. Current approaches reported are typically involve the use of meta-heuristics and, to date no consideration has been given to the use of a gradient based approach.

In Chapter 2 an overview of the literature is provided. The current traffic models in use are investigated, as well as the current methods used to perform validation. Chapter 3 provides the numerical description of the simulators being used throughout this thesis: The Cell Transmission Model (CTM) and Modèle d'Ecoulement du Traffic Autoroutier (META) applied to NETworks (METANET), and the chapter provides a background of these models in their current form. A change is included in the CTM to use the same fundamental diagram as in the METANET model. This reduces the difference between the two models and can potentially result in more accurate speed predictions around the transition from free-flowing to congested traffic.

The validation process used in this thesis is defined in Chapter 4. To establish a methodology that can work with any network topology or size a novel method is introduced that allows the optimisation to work with a model that has been decomposed into sub-networks. The method introduced removes the need for ad-hoc sub-division of large networks enabling the whole network to be calibrated as a whole rather than in sections which must then be recombined to make a complete model. The method introduced allows for the optimisation to select the number of parameters required to define a traffic model and determine where homogeneous sections of motorway exist that can be modelled with the same parameters.

Chapter 5 provides the mathematical definitions of the various optimisation algorithms considered. These meta-heuristic algorithms include, Genetic Algorithm (GA), Particle Swarm Optimisation (PSO) and Cuckoo Search (CS). The second part of the chapter concerns gradient based optimisation. This is an approach that has not been previously attempted in the literature. By the novel application of Automatic Differentiation (AD) the Jacobian of the traffic model can be obtained. The chapter provides a discussion on the formulation required to obtain the Jacobian and its properties. Detail is also provided on the gradient based solver Resilient backPROPagation (RPROP), which has been successfully applied to non-linear and non-smooth functions.

The data processing required, including the transformation of data into the correct form, is detailed in Chapter 6. The data source is discussed and explained and the methodologies used to determine the required model inputs, disturbances and boundary conditions are given. The chapter also details how missing data is dealt with. This is important as when large scale networks are being investigated some data is inevitably not available. Also given in Chapter 6 is the location and configuration of the three sites being investigated, Heathrow, Sheffield and Manchester. Manchester is a large network and the results of the novel sectioning method provided in Chapter 5 are shown.

The software developed as part of this thesis are discussed in Chapter 7. The methodology for linking the required systems is described as is a program that was developed to assist in the analysis and inspection of the traffic model relative to measured data. Details are given on how the AD is performed and its methodology. The systems developed are then tested and the run times are compared for each of the different methodologies, gradient based versus the meta-heuristic approach. A variety of different communication processes are evaluated and tested to try and create the most efficient formulation possible. Parallel computing is investigated in order to speed up the optimisation algorithms by using all of the available computing resource on a multi-processor system. Finally, consideration is given to the emerging technology of autonomics and thoughts are given about how the developed systems could be expanded to create a novel system that could be run without user interaction and maintenance.

The first validation effort is reported in Chapter 8, where the meta-heuristic algorithms are compared to each other for the shortest two sites considered in this thesis. The chapter provides the first reported use of some of these meta-heuristics. Also the novel method for determining the number of parameters to apply, and the spatial configuration of parameters is also investigated. The study is done using the second-order model METANET and the first-order CTM. Comparisons and differences between the solutions obtained are noted.

In Chapter 9 consideration is given to the gradient based optimisation. Methods for the application of RPROP to this complex problem, so that it acts as a global search rather than a local one, are introduced and a study into suitable parameter values is performed. The combinatorial method developed for the spatial assignment of parameters developed in Chapter 4 is modified so that the same philosophy can be used with a gradient based solver. Model calibration is performed using the METANET model and the results are reported for the Sheffield site. The gradient based optimisation is compared to the best performing meta-heuristics from Chapter 8. The novel use of AD provides new data that has not previously been obtained from the model, and sensitivity diagrams and partial derivatives from the model are investigated.

The large network of Manchester is validated in Chapter 10. This is the largest network considered in this thesis and in the literature. The validation is performed for the METANET model using the same methods and algorithms as in Chapter 9. The calibration results and spatial assignment of parameters is evaluated and discussed. Further discussions into the sensitivity data available through the novel application of AD is also reported.

Chapter 11 concludes the thesis and provides an overview of the findings of this work. A discussion upon further extensions to the work presented here are also given.

Chapter 2

Literature Review

2.1 Introduction

This thesis is concerned with macroscopic traffic model validation which is a two phase process. Phase one involves calibrating a model to find a set of parameters to replicate a specific set of conditions. The second phase is verification where the obtained parameters are tested for accuracy and generalisation properties. This literature review will therefore commence with a discussion on the macroscopic traffic simulators currently in use. This will in turn be followed by details on the various validation techniques that have been employed. However the first part of model validation, the model calibration or parameter identification process is typically an optimisation problem. A short section on optimisation methods follows. The chapter closes with shortcomings that in the current field that this thesis addresses and some conclusions.

2.2 Traffic Flow models

2.2.1 Categorisation by Modelling Scale

Traffic flow models can be grouped into three distinct categories based upon their level of detail. The models with the highest level of detail are microscopic. In microscopic models every vehicle is an entity within the model and is explicitly traced. Mesoscopic models have a reduced number of entities, with an entity describing a group of vehicles with similar characteristics. The models with the lowest level of detail are macroscopic models. Macroscopic models do not track vehicles but consider the flow of vehicles along a highway as an incompressible fluid with a flow-rate q (veh/hr), speed v (km/h) and density ρ (veh/km). Macroscopic models are typically further defined by the number of coupled Partial Differential Equations (PDEs), referred to as the model order.

The different categories of model are suited to various purposes. Microscopic models are typically used to model interrupted traffic flows, such as urban networks. This is due to the fact that the number of interactions is large so the high level of detail is necessary. Obviously this level of detail has an associated computational cost. In this regard as we move towards uninterrupted traffic flow, typically highways and motorways, mesoscopic and macroscopic models come to the fore. In these networks the number of vehicles is usually large due the high capacity and flow rates of these roads. In these networks the level of detail is not required due to the reduced external interactions. Macroscopic models also confer a few additional benefits, they are computationally inexpensive so are ideal for on-line model based control. In this thesis the focus is on modelling highway networks and only macroscopic models are being considered. An overview of microscopic and mesoscopic models are available in [9]. An overview of the background and development of traffic flow models including hybrid models that merge aspects from each category is given in [10]. Macroscopic models main advantages over microscopic models are their numerical efficiency and reduced parameter set. Furthermore due to their analytical form macroscopic models can also be used for various Intelligent Transportations Systems (ITS) [11] such as controllers and strategy evaluation as well as for simulation.

2.2.2 First order macroscopic models

Lighthill–Whitham and Richards Model

The development of macroscopic traffic flow models started in the 1950's by Lighthill and Whitham [12], and independently Richards [13]. The flow of traffic along sections of highway was compared to fluid flow. The Lighthill-Whitham and Richards (LWR) is a first order model and its main principle is the conservation equation,

$$\rho_t + (\rho v)_x = 0, \tag{2.1}$$

where the conserved quantity is the number of vehicles. This means that the vehicles cannot be created or destroyed; the number of vehicles depends on the vehicles already within the system and the flow into and out of the system. The mean speed was assumed to be a linearly decreasing function of the traffic density,

$$v = V(\rho) = v_f \left(1 - \frac{\rho}{\rho_{\max}}\right), \tag{2.2}$$

where v_f is the free flow traffic speed at zero density and $\rho_{\max}$ the maximum density possible, i.e. vehicles are bumper to bumper. Flow can then be calculated by the hydrodynamic relation,

$$q = v\rho. \tag{2.3}$$

By substituting eq. (2.2) into eq. (2.3) an expression for flow can be found as a function of density this is the Fundamental Diagram (FD) of traffic.

The resulting LWR model is a hyperbolic conservation law, in which density is the conserved variable. The LWR model can create shock-waves and non-smooth solutions from smooth initial conditions. The model has a few limitations however, such as the admissible generalised solutions are not unique [14]; it also fails to reproduce some traffic phenomena. The model cannot reproduce oscillatory stop and go waves, the hysteresis effect [15, 16] of the capacity drop and, phantom jams which arise from amplifications of small disturbances [9]. Furthermore due to the mean speed being defined by the FD there is no speed variation and the speed instantaneously responds to changes density meaning that the acceleration of the flow is infinite.

Cell Transmission Model

The CTM [17, 18] is a first order model that is a special case of the Godunov scheme of the LWR [14]. This model provides a way of using the LWR to model highways. The model splits the highway into uniform sections called cells. The density in each cell is calculated by a finite difference approximation of eq. (2.1) to balance the flow leaving and entering a cell in a single time step. The flow out of a cell is calculated from logical rules to ensure that it does not exceed the cell's capacity and supply, or the downstream cell's capacity and demand. As the CTM is a numerical representation of the LWR model it suffers from the same limitations.

2.2.3 Second order macroscopic models

Basic form

Second-order models explicitly model speed by having a second PDE. For all of the models discussed here can be expressed in the form

$$v_t + vv_x + \frac{1}{\rho}\mathcal{P}(\rho)_x = \frac{1}{\tau}[V(\rho) - v]. \tag{2.4}$$

with $\mathcal{P}$ the traffic pressure. In the models that follow, although derived in different ways they can be expressed in this form with differences coming from the functions used for the equilibrium speed V and the traffic pressure $\mathcal{P}$.

Payne-Whitham type models

To tackle the problems encountered by the LWR model Payne and Whitham (PW) [19] and [20] utilised a second PDE to model the speed explicitly.

$$v_t + vv_x + \frac{c_0^2}{\rho}\rho_x = \frac{1}{\tau}[V(\rho) - v] \tag{2.5}$$

This equation was derived from a microscopic car following model. The equation consists of three terms

- *Convection:* vv_x, stating the evolution in mean speed due to the speed of out and inflowing traffic.
- *Anticipation:* $\frac{c_0^2}{\rho}\rho_x$, which determines the drivers reaction to spatial change in the downstream traffic state. The variable c_0 is the sound speed, c_0^2 is the anticipation constant.
- *Relaxation:* $\frac{1}{\tau}[V(\rho)-v]$, describes the fact that the traffic wishes to relax to an equilibrium state that is defined by the FD. The parameter τ is the relaxation time.

The density is calculated by the conservation of vehicles equation (2.1).

The PW model has several desired characteristics. It can exhibit and model hysteresis, it can produce metastable states, phantom jams as well as stop and go waves [9]. There are several criticisms about the PW model due to it being too similar to fluid flow [21]. Traffic flow by its very nature is anisotropic, as drivers mainly react to frontal stimuli responding to what they see before them. In fluid and gas flow a stream is influenced by the pressure upstream as well as downstream. Slow vehicles in a traffic flow will remain slow, in a fluid particles travelling slowly will be accelerated by the flow. The PW model is known to exhibit waves which travel faster than the speed of the traffic. The wave propagation speeds can be found by the eigenvalues of the system which are,

$$\lambda_1 = v + \mathcal{P}(\rho), \quad \lambda_2 = v - \mathcal{P}(\rho), \tag{2.6}$$

the model exhibits an eigenvalue which is always greater than the traffic flow speed. This breaks the anisotropic property. However, in models where multiple lanes are considered the speed of the flow predicted by the model represents the aggregate value. There will exist vehicles that are travelling faster than this mean speed and some that are slower. If the propagation speed is lower than the fastest vehicles in the stream then the anisotropic property still holds [22, 23]. The LWR model however does exhibit anisotropy as its single eigenvalue, $f'(\rho) = \rho V'(\rho) + V(\rho)$ is always less than the flow's speed since $V(\rho)$ is decreasing and therefore $\rho V'(\rho)$ is always negative.

The PW model is the basis of the METANET simulator [24]. The METANET simulator is a finite difference scheme that provides a discrete model that can cope with networks of any topology. The PW model was shown to suffer in cases where merges and lane drops occur so the METANET model has additional terms included into the speed expression to cope with these situations [25, 26].

Aw-Rascle and Zhang model

To address the issues highlighted in [21], Aw and Rascle [27] analysed the original PW model to try and remedy the isotropic nature. PW models and their variants have a form of momentum conservation, where the fluid pressure term has been replaced with an anticipation factor. They identify that this is not sufficient to remove the fluid properties from the model and is the source of the problem. Including diffusion into the acceleration equation exacerbates the problem, whilst a relaxation term does not cause harm but cannot prevent the paradox. The anticipation term involves the derivative of pressure with respect to space, which the authors prove to be incorrect with the following scenario:

> "Assume for instance that in front of a driver travelling with speed v the density is increasing with respect to x, but decreasing with respect to $(x - vt)$. Then the PW type of models predicts that this driver would slow down, since the density ahead is increasing with respect to x! On the contrary any reasonable driver would accelerate since this denser traffic is travelling faster than him."

Instead of a spatial derivative of the pressure they suggest the correct form is the convective derivative

$$\partial_t + v\partial_x \tag{2.7}$$

of the pressure $\mathcal{P}$. Which they take to be a increasing function of the density,

$$\mathcal{P} = \mathcal{P}(\rho) = \rho^\gamma, \quad \gamma > 0. \tag{2.8}$$

Assuming no diffusion or relaxation they then obtain the following,

$$(v + \mathcal{P}(\rho))_t + v(v + \mathcal{P}(\rho))_x = 0. \tag{2.9}$$

Aw and Rascle also state that a macroscopic traffic model should have the following properties:

- The system should be hyperbolic.
- Density and speed must remain non-negative and be bounded from above.
- The propagation speed (eigenvalue or shock wave speed) must at most be equal to the traffic speed v.
- The solutions should exhibit driver characteristics; braking produces shock waves with negative or non-negative propagation speeds, acceleration produces rarefaction waves whose propagation speed must must be at maximum v.

The model system defined by equations (2.1) and (2.9) is indeed hyperbolic and its eigenvalues are:

$$\lambda_1 = v - \rho\mathcal{P}'(\rho) \quad \leq \quad \lambda_2 = v \tag{2.10}$$

therefore satisfying the anisotropy principle. One problem identified by the authors is that in an empty highway the speed of the vehicles depends only on the initial data so they suggest the inclusion of the relaxation term $\frac{1}{\tau}[V(\rho) - v]$ as a source term of the momentum equation.

Zhang [28] independently came to the same conclusions about the cause of the gas-like behaviour in the PW class of models. From using a car following model Zhang derived the same model but with a different pressure term $\mathcal{P}$,

$$\mathcal{P} = -\rho V'(\rho). \tag{2.11}$$

Zhang notes that this model correctly describes queue end behaviour and of course is anisotropic and devoid of gas-like behaviour. This class of second order models that utilises the convective derivative of pressure is referred to as the Aw, Rascle and Zhang (ARZ) model.

Further developments to the model and a Lagrangian formulation of the model is done by Greenberg [29]. A downwind difference scheme is exploited in the solution due to the fact the speed and density behind a contact are determined by the state ahead of it. In traffic flows various states exist around the equilibrium and the initial data can not be assumed smooth or in this equilibrium state. This means that the Riemann problem formulated by the ARZ model may not have solutions. In [30] extensions to the FD are proposed so that solutions to the Riemann problem can be obtained without violating the bumper to bumper density. The authors argue that this is the best solution to the problem as the other method, domain invariance, involves manipulation of the initial data and boundary conditions to ensure solutions can be obtained. This means the real system is no longer being modelled. Also, there is no guarantee that the data will conform to the restrictions of domain invariance.

2.2.4 Gas Kinetic Models

The Gas Kinetic (GKT) model is described in [31–33]. The model was developed by considering a Gas-Kinetic system where the molecules represent a single vehicle. This behaviour was aggregated to give the overall system dynamics rather than a microscopic model. From its formulation the model introduces variations between vehicles in the system and braking dynamics. If a faster region of traffic catches a slow moving region it has a probability to overtake or it brakes reducing its speed to match the slower traffic. In both scenarios the slower region of traffic is unaffected and the anisotropic nature of traffic flow is replicated in the model.

The model can be expressed as,

$$v_t + vv_x + \frac{1}{\rho}(\theta\rho)_x = \frac{1}{\tau}\left[V(\rho) - v\right] \tag{2.12}$$

where the pressure term θ is a density ratio of the traffic speed that controlling the transition between un-congested and congested states. The equilibrium speed function used in the model includes the braking term $B(\delta v)$. This term adds non-local properties by looking at the traffic state at an interaction point downstream $x_a = x + \gamma 1/\rho_{\max} + T_h \cdot v$, where T_h is the time headway between vehicles.

A function $A(\rho)$ defines a variance in the square of the mean speed between vehicles and is present in the braking and pressure functions. The spatial derivative of $\rho\theta$ describes the dispersion of the mean speed. The mean speed will increase ahead of a congested cluster as the free flowing traffic leaves the group behind. The braking response $B(\delta v)$ is determined by the difference in conditions at the current location and a downstream interaction location. Deceleration is Coulomb-like, with speed reduction increasing as the space headway decreases. When the flow is homogeneous the competing acceleration and braking terms equilibrate when the space-headway reaches a safe distance. The resulting non-local model produces the desired anisotropic properties.

2.2.5 Traffic Simulators

The creation of a robust solvers for the underlying PDEs is important to produce a useful traffic simulator. Several investigations into various numerical-schemes have been analysed, low-order finite volume schemes in [34–37], with a higher order Weighted Essentially Non-Oscillatory (WENO) schemes applied to the LWR model in [38] and a WENO based scheme is applied to the GKT model in [39]. However a numerical solver is not sufficient to create a simulator, as source and sink terms also need to be considered. These are the origins of traffic flow into the model and destinations branching off from it. Also to be considered are the various possible topologies that need to be modelled in a real system with roads merging and diverging. In all the works listed the problem domain is restricted to analytical tests and an uninterrupted section of highway.

The two popular macroscopic models that have been applied and tested on real sites are the CTM and METANET. As previously mentioned the CTM is a Godunov approximation of the LWR model. The METANET model is also a finite difference scheme, but is based on the Payne model with some extensions that are discussed in the next chapter. These two simulators have been used for a variety of real world locations and will be the considered in this thesis. To date a thoroughly tested ARZ model has not been developed and applied to a real network. Although the works [29] and [30] are steps towards the development of a simulator. As for the GKT model a unidirectional stretch has been calibrated with source and sinks considered in [40] but merges and diverges have not yet been considered.

2.3 Model Validation

All of the macroscopic traffic models discussed earlier include a variety of parameters. The selection of suitable values of these parameters is critical to obtain a model that reproduces the traffic characteristics observed within the network to the best possible accuracy. This is done through a the process of model validation. Validation is typically a two phase process; the first involves the calibration (parameter identification) of the model against some known data. The second process is a confirmation of the selected values, the identified parameters are evaluated against additional conditions to determine that they do indeed reproduce the correct traffic dynamics. In [22] it is claimed that the model validation is the key test of a traffic simulator. A valid model proves that the simulator is able to accurately reproduce real traffic dynamics and phenomena. The verification phase of validation involves checking the obtained parameters from calibration. This is usually done in a qualitative manner with comparison of plots and the calibration metrics. Therefore the remainder of this review focuses on the calibration aspect.

The most popular method for model calibration used within the literature is the minimisation of a squared errors between the state predicted by the model and real data collected from various points along the highway. What varies between the calibration attempts reported is the model being used, the site under investigation and the optimisation algorithm employed. An optimisation algorithm is typically used to try and search the solution space in an intelligent way so that a set of parameters are found in a timely manner that results in a minimal error. The minimisation of these squared errors results in a non-linear, non-convex problem which has numerous local minima; an example of the solution space is given in [41].

In [26, 42–45] the models (all variations and developments of METANET) were calibrated using the same technique. The algorithm used was the complex algorithm of Box [46]. This created a generation of solutions and at each iteration the worst was replaced by a new member selected by the algorithm until no further improvement could be achieved. This reliably reproduced the required results by minimising the squared error. The algorithm of Box was also employed in [47] where the LWR model was calibrated to test a merging and diverging modelling. Weightings were applied to each variable to ensure that the sum of the squared error terms were of a similar magnitude to avoid bias in the calibration phase. This method is able to validate models on a variety of scale, from a short section of Boulevard Periphique [26] to network sections of the ring roads around Amsterdam [44]. Apart from Amsterdam all the sites considered are sections of highway with only origins and destinations considered, no merging or diverging occurs with other sections of modelled highways. In Amsterdam [44, 48] to achieve the calibration the model was split into smaller sub sections that were calibrated independently and then combined to make the final model. In a qualitative analysis manual adjustments were made to the flow assignment at junctions to achieve the final validated model.

The Nelder-Mead algorithm [49] has also been applied to the validation problem with it being applied to the METANET model in [50]. This was again used the minimisation of the least squares value as the objective function. The work was extended in [36] to work with various numerical implementations of Payne's model. This shows that the process is robust enough to solve the problem when the process is changed. Both the METANET and CTM model are calibrated using this algorithm in [51]. The Nelder-Mead algorithm has similar methodology to the complex algorithm of Box. The sites investigated using this algorithm range from 7-12 km and are simple linear sections of highway.

Cross-Entropy Method [52] has been used in two calibration attempts [41] and [53]. The method is a Monte-Carlo approach, where a population of solutions are generated according to an initial distribution. These solutions are evaluated and sorted, and the top percentage are used to inform the update of the distribution. In the next iteration the new distribution is used to select the population.

As is mentioned in [41], the optimisation problem related to model calibration has numerous local minima. Hence, efficient optimisation algorithms need to be used for obtaining parameter sets that make models capable of representing traffic dynamics. Most of the proposed, if not all, algorithms used are population based derivative free methods, employing direct or stochastic search. In a recent overview of non-linear programming methods used for macroscopic traffic flow model calibration, [54], gradient based optimisation algorithms are not considered as a viable option due to the non-linear and non-convex nature of the least-squares optimization problem.

A Differential Evolution Algorithm form the basis of the work in [40]. This algorithm is similar to a GA, but the authors use Artificial Neural Networks to evaluate solutions that are trained to the underlying model in this way the costly objective function evaluations are shortened and replaced with an estimates provided from the neural networks. During the course of the search the neural-networks are maintained if necessary retraining is performed.

A number of population based derivative free optimisation algorithms used for calibration are discussed in [53]. The algorithms considered and compared are the Nelder-Mead algorithm, Genetic Algorithm and the Cross-Entropy method. All three algorithms find solutions of a similar quality but the Nelder-Mead method does not require a large population of solutions that was used in the other two algorithms as such it had a far shorter computation time. This study was done on a short site with a low number of parameters and the authors note that these findings may change for problems of different scales.

The key parameters relating to the fundamental diagram of the METANET model were found by using a maximum likelihood estimator in [55], although this does require a good prediction of the remaining parameters. In [56] a model of a 1.8km stretch of the A12 highway in the Netherlands over a time horizon of 3 hours using a linear parameter varying formulation of the Payne model was validated. The same method was also utilised in [57]. In an attempt to develop an independent framework for validation a methodology using statistical tests between various objective values functions and distributions is proposed in [58], instead of the typical least squares method. However, if the model captures the correct dynamics but has a small bias over a large time period the statistical tests outlined will fail. A wider consideration of model calibration using data mining techniques is described by [59], although a large scale trial has not yet been undertaken. In [60] a METANET model parameter identification algorithm is discussed using data from a 4.65 km stretch of a California highway; the original expression used for FD in METANET is replaced with a two-regime model and the resulting optimisation problem is solved using a sequential quadratic programming algorithm.

For model validation to be correct the objective value or data that is used in the validation process must be accurate and analogous to the outputs of the model. This is usually simple for macroscopic models but this is not the case for microscopic models. However, it is not always possible to get the data required for a specific location or of the correct form. To overcome this [61] developed a method in which separate metrics for both the data and the model are calculated independently. These metrics are then used in comparison to create an objective function. In [61] the properties of the congestion waves are found for the data supplied, and are used to validate the model when the same wave-properties are calculated for the model output. The work created a valid 10 km model of a UK highway. Models can also be validated by using data from mobiles phones [62], although the traffic model has to be formulated to coincide with the cell boundaries of the mobile phone network.

The cell transmission model has been validated in [63] and [64]. The method utilised here is split into sections to determine the parameters that govern certain conditions. The initial phase is to calculate the free flow parameters; this done by performing a least squares fit on the flow versus density data early in the morning when traffic is free flowing. This is then used to calculate the free flow speed for each detection point and the other cells are given their free speed by liner interpolation. Bottlenecks are then found by examining contour plots to find locations where the downstream conditions are free flowing and congestion occurs upstream. The capacity of the cells that are not located at a bottleneck is determined by choosing values larger than the maximum observed flow. For the bottleneck capacity the downstream sections capacity is given as for a non bottleneck location but the upstream cells capacity is set to the average flow into the downstream cell. The congestion parameters are found by a constrained least squares optimisation of the the flow versus density data. Firstly the jam density is found by taking the point at which the maximum flow occurs, this is then followed by fitting the graph so the maximum flow allowed at the bottlenecks can be achieved by the model. This was successfully applied to moderately sized sections of U.S. highways. However the process results in a large parameter set that changes over each model cell. Also using fitting procedures of the flow versus density data can provide erroneous results when congestion waves propagate and create artificially low flows that are not due to the properties of the particular section of highway. Although this method is a based on analysis of available data the calibration of the model is rapid and does not require numerous function evaluations like the earlier discussed meta-heuristics.

2.4 Optimisation Algorithms

This literature review mainly focuses on the traffic model validation. There is vast area of literature on available optimisation algorithms that could be potentially applied to this problem. A sample of meta-heuristics applied to real engineering problems. A multi-objective genetic algorithm has been used to optimise electrical drives [65]. A gravitational search is conducted to optimise a fuzzy servo controller in [66]. Particle swarm optimisation has been used for reservoir optimisation [67], and hydrothermal scheduling [68]. Examples of further possible meta-heuristics that could be applied are given in the following [69–83]. This is a popular field with a large number of publications although care is needed as structural bias in the algorithms can make them perform well for specific scenarios or test functions [84]. Further more there is a trend of developing algorithms that have a 'metaphor' that mimics or replicates a dynamic seen in man-made or natural systems, often these algorithms are misunderstood or the difference to an existing method is minor [85]. That is not to say that some of the ideas produced have lead to novel operations that can assist in field of meta-heuristics, but they can make the understanding of the process more difficult by changing the language and terminology.

For gradient based optimisation approaches this thesis will consider the Resilient backPROPagation (RPROP) algorithm [86]. Like with meta-heuristics there are a variety of methods that could be used, this algorithm however is being investigated is that the algorithm has been tested on non-smooth functions in [87] and non-linear problems in [88]. A further reason for this algorithm being selected is that the search method only relies on the sign of the gradient. This means that any methodology used to obtain the gradients from the traffic model can introduce small errors as long as the sign of the gradient is calculated correctly. More details about RPROP algorithm are given in Chapter 5.

To use the RPROP algorithm the gradient from the traffic model must be obtained. One possible method would be divided differences, this however would be impractical, depending on the method used each point on the stencil used would require A evaluations of the traffic model with A being the number parameters to be identified. Numerical approaches are also prone to rounding and truncation errors. A better approach would be to use Automatic Differentiation (AD) [89]. AD takes a function code performs an efficient calculation of derivatives with the same accuracy as the original code. Details about how AD works is given in Section 7.3.

2.5 Thesis Contributions

The aims of this thesis are to try an develop a methodology that is able to validate large networks as a whole. Currently there is only one published work that looks at a large network [44] which was published in 2002. Since then no further consideration have been given to validation on large scale macroscopic models with reported works restricted to shorter sites that consider uni-direction unconnected sections of highway. The FD describes the local properties of a section of highway and using a single set of parameters across all links in a large network will not provide sufficient accuracy. So part of this thesis is the development of a methodology that allows for the optimisation to determine the spatial extent of an FD across links. The other choice is to allow each link to have its own set of parameters but this runs the risk of over-parametrisation.

The other key area considered in this thesis is feasibility and practicality of using a gradient based solver for the optimisation problem. Part of the challenges involved in this is the calculation of an accurate and correct gradient from the traffic model. But also the solution space is complex and considerations need to be given to make sure that a global search is undertaken as gradient methods are by their nature a local search method. It is for these reasons that the use of gradient based solvers has been dismissed in the literature [54].

2.6 Conclusion

The most popular and currently adopted method of calibration is to use a squared error metric between a models output and measured data. This metric is then minimised by use meta-heuristic, Differential Evolution, Genetic Algorithm, Nelder-Mead and Cross Entropy methods being the most recently used approaches. No consideration has been given to the application of a gradient-based algorithm due to the nature of the solution space. The calibration attempts that do not use the squared error metric are typically aimed at using additional data sources, or making up for a lack of data. The exception to this are [63] and [64] where parameters are fitted to the FD directly, this could cause issues if conditions are driven by bottlenecks that were not previously identified.

Apart from the calibration of the Amsterdam network all of the current methodologies have been applied to uni-directional sections of highway typically less than 20 km and don't have to cope with network interactions beyond on- and off-ramps. No recent efforts have been done into a network wide calibration process. The only reported validation of a large network modelled the roads around Amsterdam and to achieve this manual adjustments were performed after calibration. The calibration itself did not consider the entire network but was done independently on sub-networks and not the whole problem geometry.

Chapter 3

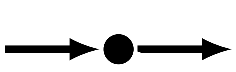
Macroscopic Traffic Models

3.1 Model selection

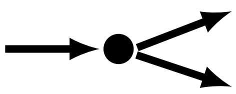
Various traffic flow models were discussed in Chapter 2. For one of them to be used in a real network a numerical form of the relevant PDE must be developed. Furthermore, the PDE do not define how traffic should be treated at merges and bifurcations. As such the models selected must have a proven numerical form. To ensure that the techniques developed for solving the validation problem (Chapter 4) are applicable to many macroscopic simulators, first and second-order models have been selected. One of the most popular first-order traffic models is the Cell Transmission Model (CTM) [17, 18]. The CTM is a Godunov scheme [90] of the LWR model and has been successfully used at numerous sites [51, 63, 64, 91].

A popular second-order model is METANET, which is based on the Payne model [19]. It has been used in many past studies [26, 41, 42, 51, 92] including for a large-scale network [44]. The Payne model has been criticised, as some incorrect model predictions can occur. These include the potential to predict negative speeds and for the speed to be affected by upstream conditions beyond propagation effects [27]. However, the model has still been shown to reproduce real world conditions in an accurate manner. Another interesting second-order model was proposed by Aw, Rascle and Zhang (ARZ) [27, 28] which overcomes the mathematical errors of Payne-type models. This is achieved by altering the speed equation to include a 'traffic pressure' term. This model has not been applied to any real networks. The models considered in this thesis are the CTM and METANET.

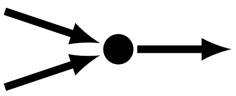
This chapter's focus is on these models' dynamics. As such it serves as a reference to their current status in the literature and does not provide significant original contributions. The material included here mainly follows the choices of the models' developers. However, a different form of sending and receiving functions for the CTM model are defined here.



(a) Simple node.

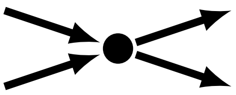


(b) Diverge node.

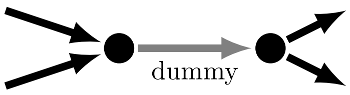


(c) Merge node.

Figure 3.1: Valid node representations.



(a) Invalid.



(b) Valid.

Figure 3.2: Resolving invalid representations, additional dummy arc of zero length shown in grey.

The chapter is arranged as follows; first the representation of networks in a form that can be used by the macroscopic simulators is discussed in 3.2. The macroscopic traffic variables are introduced in 3.3. The governing equations and definitions of the CTM are stated in 3.4, followed by METANET in 3.5. Both models use the same FD and the chapter concludes with their summary.

3.2 Network representation

Any network can be expressed as a directed graph with nodes and arcs. To simplify the expressions and cases that have to be evaluated by the simulator only certain topologies are considered. The total number of arcs entering and leaving a node is restricted to three (in the model versions used by this thesis). Therefore the only node representations allowed in the discrete network are as shown in Figure 3.1. To ensure that any network can be expressed correctly dummy arcs of zero length are introduced to create a valid model as shown in Figure 3.2.

This representation is still not entirely sufficient as the arcs may not express homogeneous sections of highway. To resolve this the arcs are subdivided into homogeneous links. Nodes are added at locations where the geometry of the highway changes. Each link m represents a section of highway where the topology is consistent; no on/off-ramps occur within it and the number of lanes is constant. Further subdivision of each link m into N_m segments of equal length is also performed. The link representation is shown in Figure 3.3.

3.3 Macroscopic Variables

The models are also discrete in time, using a constant time step T. For each time step $k = 0, 1, \ldots, K$, with K the time horizon, the following macroscopic variables are defined for every segment i of each link m.

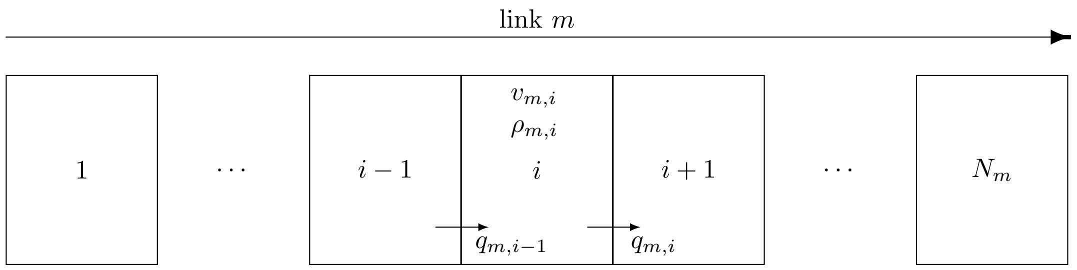


Figure 3.3: A discrete highway link.

- *Traffic flow:* $q_{m,i}(k)$ (veh/h) is the number of vehicles exiting segment i of the link m during the the interval $[k \cdot T, (k+1) \cdot T]$ divided by T.
- *Traffic density:* $\rho_{m,i}(k)$ (veh/km/lane) measures the numbers of vehicles in segment i of link m at time $k \cdot T$ divided by the length of the segment L_m and the number of lanes λ_m.
- *Mean speed:* $v_{m,i}(k)$ (km/h) is the average speed of all the vehicles over the segment i of link m at time $k \cdot T$.

3.4 Cell Transmission Model (CTM)

3.4.1 Overview

In the CTM the traffic network is represented by a set of cells [18]. The methodology in that work however uses cells of a constant size throughout the entire network and no link discretisation existed. The flow was computed in units of (vehicles/time step). Here the expressions are given in a form where links are used so cells within a link are constant size, but not all links has to use the same size, and the flow expressed in (veh/hr). For the discrete form of the network as discussed in Section 3.2 cells of the CTM are analogous to segments.

3.4.2 Link model

Link types

Links of the model belong to one of four types; *highway links* are the main component of the model and represent homogeneous sections of the highway. *Dummy links* which are introduced for simplifying complex topologies and have zero length. *Origin links* that pass flows into the network and apply upstream Boundary Conditions (BCs) and finally *destination links* that allow for the application of the downstream BCs and receive the outflows from the network.

Highway links

The CTM is a Godunov scheme [90] of the LWR model and the density $\rho_{m,i}$ is computed by the upwind finite difference scheme of the conservation equation,

$$\rho_{m,i}(k+1) = \rho_{m,i}(k) + \frac{T}{L_m \lambda_m}[q_{m,i-1}(k) - q_{m,i}(k)]. \tag{3.1}$$

with λ_m the number of lanes in link m. For numerical stability the CFL condition must be satisfied, $L_m \geq T v_{f,m}$. This condition ensures that no vehicle travelling with free speed will pass a segment during one simulation time step. Flow is computed as,

$$q_{m,i}(k) = \min\{S_{m,i}(k), R_{m,i+1}(k)\} \tag{3.2}$$

with $S_{m,i}(k)$ being the supply of the segment i and $R_{m,i+1}(k)$ the possible flow that can be received by the segment $i+1$ of link m at the time step k. The exact nature of the functions S and R depend on the FD being used. The CTM uses a triangular or trapezoidal FD,

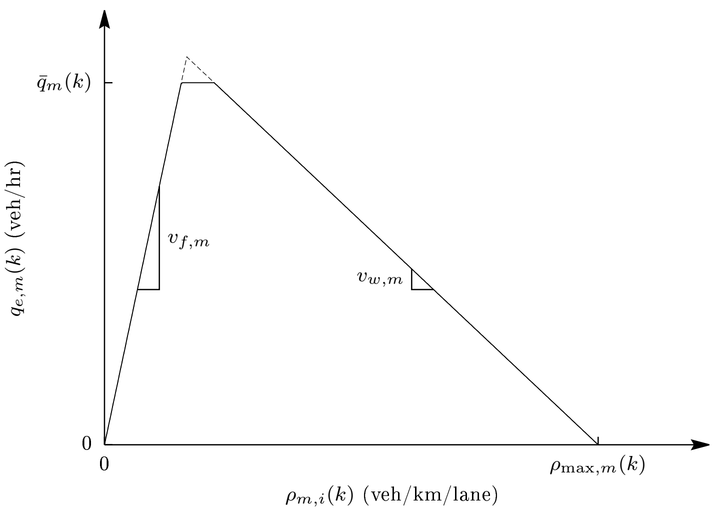
$$q_{e,m}(\rho_{m,i}(k)) = \min\{v_{f,m}\rho_{m,i}(k)\lambda_m, \bar{q}_m, v_{w,m}[\rho_{\max,m} - \rho_{m,i}(k)]\lambda_m\} \tag{3.3}$$

where the parameter $v_{f,m}$ is the free flow speed of link m, $\bar{q}_m$ the capacity of the link and $v_{w,m}$ the backwards propagation wave-speed (Figure 3.4a). The speed (and the equilibrium speed $V(\rho)$) is calculated from the flow via the hydrodynamic relation (2.3),

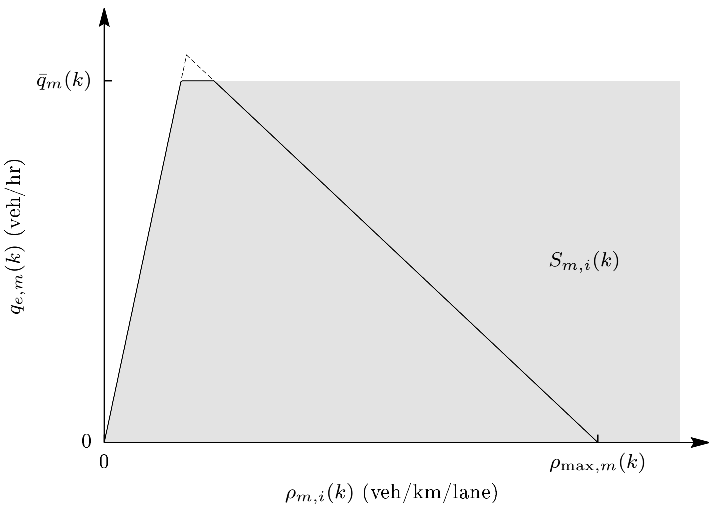
$$v_{m,i}(k) = V(\rho_{m,i}(k)) = \frac{q_{e,m}(\rho_{m,i}(k))}{\rho_{m,i}(k)\lambda_m} \tag{3.4}$$

$$= \min\left\{v_{f,m}, \frac{\bar{q}_m}{\rho_{m,i}(k)\lambda_m}, v_{w,m}\left(\frac{\rho_{\max,m}}{\rho_{m,i}(k)} - 1\right)\right\}. \tag{3.5}$$

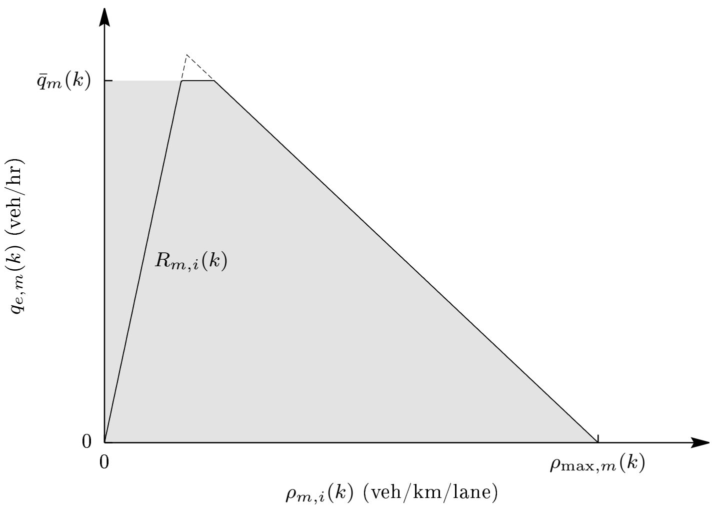
The supply is the maximum possible flow that a segment has available as shown in Figure 3.4b. At low densities (free flow conditions) it is equal to the fundamental diagram flow. If density increases beyond the free flow region then the supply is limited to the capacity of the cell. The receiving flow is the available space in the downstream cell as illustrated in Figure 3.4c. It is equal to the downstream cell's capacity in free flowing conditions and is calculated from the downstream FD. If the link m is an upstream link of a merge then a different expression is required to calculate the flow for $i = N_m$ to account for merging effects, this is part of the node model discussed in 3.4.3.



(a) Piece-wise linear fundamental diagram.



(b) Sending function.



(c) Receiving function.

Figure 3.4: Piece-wise linear fundamental diagram used by the CTM and the Sending and Receiving functions associated with it.

The supply function for a trapezoidal FD is given as,

$$S_{m,i}(k) = \min\{v_{f,m}\rho_{m,i}(k)\lambda_m, \bar{q}_m\} \tag{3.6}$$

and the receiving function from

$$R_{m,i}(k) = \min\{\bar{q}_m, v_{w,m}[\rho_{\max,m} - \rho_{m,i}(k)]\lambda_m\}. \tag{3.7}$$

This formulation means that flow $q_{m,i}(k)$ is calculated from the FD in a manner that preserves the available space and allows for the propagation of queues through the network.

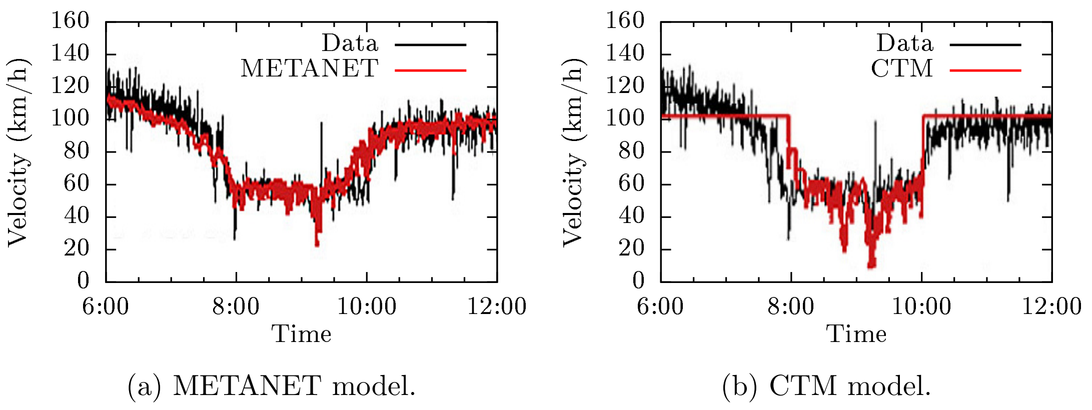
The piece-wise linear fundamental diagram has been used in past validation attempts for the CTM and this has been compared to the METANET model [51]. In METANET the equilibrium speed is defined as,

$$V[\rho_{m,i}(k)] = v_{f,m} \cdot \exp\left[-\frac{1}{\alpha_m}\left(\frac{\rho_{m,i}(k)}{\rho_{cr,m}}\right)^{\alpha_m}\right] \tag{3.8}$$

with $\rho_{cr,m}$ the critical density of link m at which capacity flow occurs and α_m a parameter. This gives a FD of the form

$$q_{e,m}(\rho_{m,i}(k)) = \rho_{m,i}(k)\lambda_m V(\rho_{m,i}(k)) = \rho_{m,i}(k)\lambda \cdot v_{f,m} \cdot \exp\left[-\frac{1}{\alpha_m}\left(\frac{\rho_{m,i}(k)}{\rho_{cr,m}}\right)^{\alpha_m}\right]. \tag{3.9}$$

The comparison of the predicted speeds in the two models in [51] shows that METANET predicts a closer match to real-world conditions than CTM. In the study most of the error between the CTM's predicted speed and the real world data occurs in free-flow conditions as shown in Figure 3.5. The parameters found for the



(a) METANET model. (b) CTM model.

Figure 3.5: Comparison of calibrated METANET and CTM models' speed profiles, result taken from [51] for detector location 29.2km.

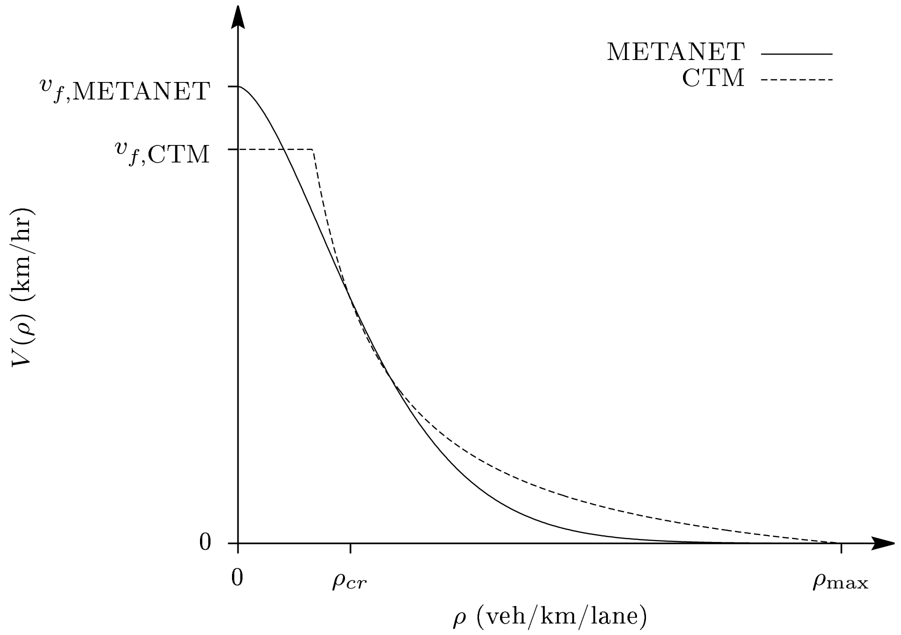


Figure 3.6: Equilibrium speed functions of the METANET and CTM model.

CTM in the study typically under predict the speed at low densities, 6:00–7:00 in Figure 3.5b. As the density approaches the critical density and during the onset of congestion (7:00–8:00) the CTM model over predicts the mean speed. In the congestion region the two models are similar, but during recovery the CTM suffers the same problems as at congestion onset. The equilibrium speed functions used by the two models are compared in Figure 3.6. The parameters of the speed functions shown are equal to those identified in [51]. The figure shows that the trapezoidal FD defines an equilibrium speed function with a constant free speed for free-flowing conditions. Meanwhile METANET's FD has a gradual (and increasing) reduction in speed as density increases. The piece-wise linear FD is also a non-smooth function with $V'(\rho)$ having discontinuities between the three regions (free-speed, capacity flow and congested). In Figure 3.6 the choice of parameters makes the difference in curvature between the capacity flow and congested regions hard to see. Discontinuities are not likely to be exhibited by natural driving behaviour. The equilibrium speed of the METANET model aims to to mimic driver behaviour in this regard as the function is smooth. This function could be used for the CTM, allowing for increased accuracy in the free-flow conditions. In Figure 3.6 the METANET FD is defined with a larger free-speed than for the trapezoidal FD this accounts for the trajectory of the speed profile. The fact that the speed decreases with density means that it is possible to track the speed for all density values, Figure 3.5a. Comparing the FDs of CTM and METANET identified in [51] lead to the two equilibrium speed functions having a similar trajectory around the critical density (capacity flow).

Using the METANET FD (3.9) the sending function becomes

$$S_{m,i}(k) = \begin{cases} q_{e,m}(\rho_{m,i}(k)) & \text{if } \rho_{m,i}(k) < \rho_{cr,m} \\ \bar{q}_m & \text{else.} \end{cases} \tag{3.10}$$

the capacity of the link is no longer a parameter but is calculated from the FD parameters.

$$\bar{q}_m = q_{e,m}(\rho_{cr,m}) = \rho_{cr,m}\lambda_m v_{f,m} \cdot \exp\left[-\frac{1}{\alpha_m}\right] \tag{3.11}$$

The receiving function becomes

$$R_{m,i}(k) = \begin{cases} \bar{q}_m & \text{if } \rho_{m,i}(k) < \rho_{cr,m} \\ q_{e,m}(\rho_{m,i}(k)) & \text{else.} \end{cases} \tag{3.12}$$

The CTM's sending and receiving function using the same FD as the METANET model are shown in Figure 3.7.

The sending and receiving functions defined by the METANET FD have similarities with those for a piece-wise linear FD. By comparing Figure 3.4 to Figure 3.7 it can be seen that the sending function has a slightly lower values as the density approaches $\rho_{cr,m}$ otherwise it is limited to capacity which can be the same for both functions. For the receiving function the main difference is the tail at high densities.

Origin links

Origin links receive the demand at inflows into the network and pass this into the system via a simple queueing model. The outflow $q_o(k)$ at origin o is given by

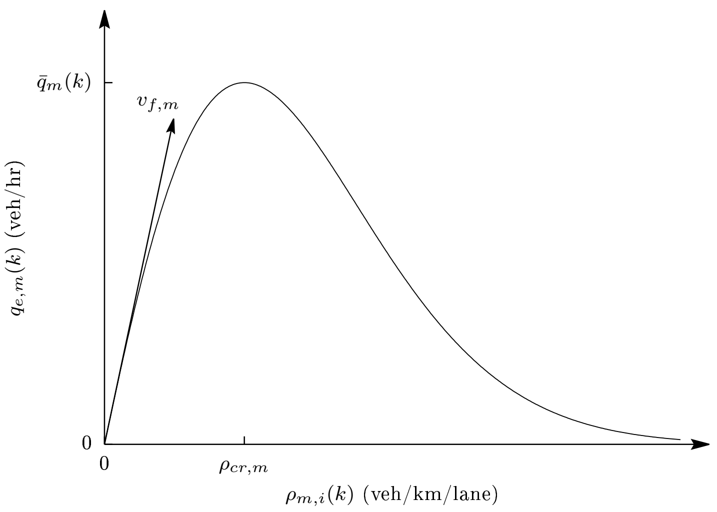
$$q_o(k) = \min\left\{d_o(k) + \frac{g_o(k)}{T}, \bar{q}_o, R_o(k)\right\} \tag{3.13}$$

with $d_o(k)$ the demand at origin o for the time step k, $g_o(k)$ is the queue length in vehicles at time step k and $\bar{q}_o$ the capacity of the origin o. $R_o(k)$ is the receiving function for the downstream segment. A simple conservation scheme is used to update the queue length

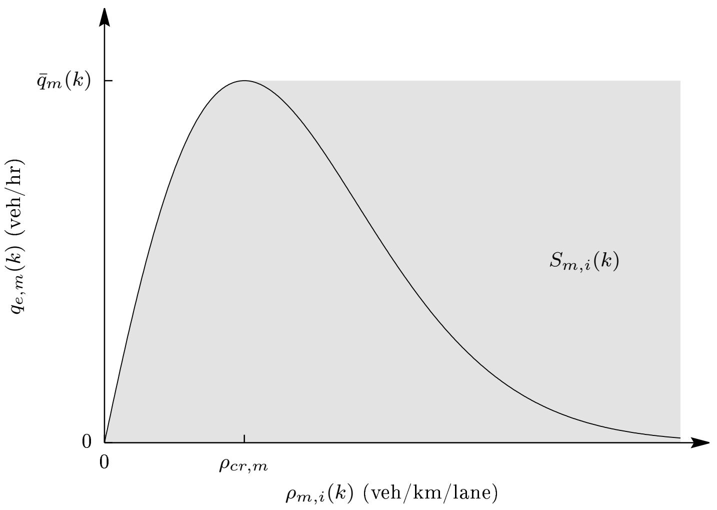
$$g_o(k+1) = g_o(k) + T[d_o(k) - q_o(k)]. \tag{3.14}$$

Destination links

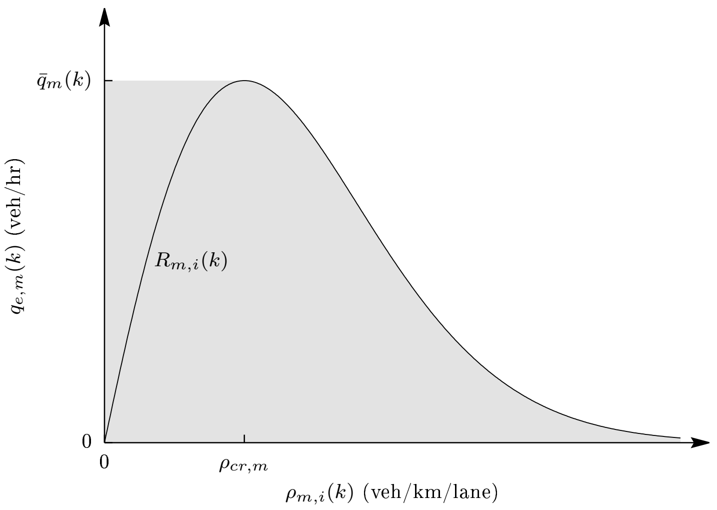
The destination links in the CTM have a FD so that a receiving function can be defined based on the supplied boundary condition. This allows for queues at the destination to be modelled and the available space at off-ramps can be accounted



(a) Smooth fundamental diagram.



(b) Sending function.



(c) Receiving function.

Figure 3.7: Smooth fundamental diagram used by METANET and the CTM Sending and Receiving functions associated with it.

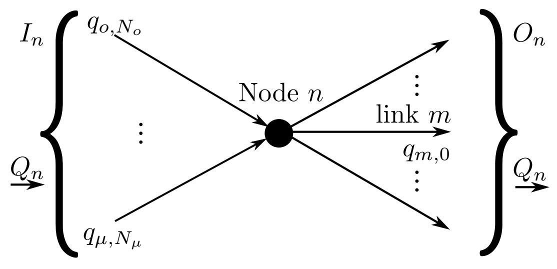


Figure 3.8: Node model.

for. The required boundary condition that has to be supplied is the downstream density ρ_b for each destination. Destinations are therefore treated as single segment links with zero length, the boundary density is applied to the destinations FD to determine the flow that the destination can receive.

3.4.3 Node model

For equation (3.1) a flow must be defined at the node, i.e. $q_{m,0}(k)$, for $i = 1$ of each link m. Let I_n be the in-flowing links and O_n be the out-flowing links of the node n upstream of the link m. Let the total flow entering the node be Q_n which is calculated by

$$Q_n(k) = \sum_{\mu \in I_n} q_{\mu,N_\mu}(k) \tag{3.15}$$

The flow entering link $m \in O_n$ from node n can then be calculated as,

$$q_{m,0}(k) = \beta_n^m(k) \cdot Q_n(k) \tag{3.16}$$

where $\beta_n^m(k)$ is the turning rate that defines the proportion of flow from node n that is passed to link m in time step k. The node model is depicted in Figure 3.8.

When i is equal to N_m a value for $R_{m,N_m+1}(k)$ needs to be defined for eqn. (3.2); similarly the receiving function at origins $R_o(k)$ needs to be expressed. At all nodes this is calculated as

$$R_{m,N_m+1}(k) = \min_{\mu \in O_n} \left\{ \frac{R_{\mu,1}(k)}{\beta_n^\mu(k)} \right\} \tag{3.17}$$

with n the downstream node of link m [17]. This provides the maximum possible flow to the more congested out-link, $R_{m,N_m+1}(K)$ is greater than any of the individual out links receiving flow $R_{\mu,1}(k)$ but it defines the receiving flow of the node as required in (3.2). By application of (3.16) the proportion of traffic split to each out-link cannot exceed its receiving flow. For origin link o,

$$R_o(k) = \min_{\mu \in O_n} \left\{ \frac{R_{\mu,1}(k)}{\beta_n^{\mu}(k)} \right\} \tag{3.18}$$

with n the downstream node of origin o.

For $i = N_m$ a different expression is required to calculate the flow $q_{m,N_m}(k)$ out of the final segment of link m to account for merging effects. Three possible scenarios exist at a merge [17]

Forward, all approaches are free flowing and conditions depend on upstream traffic and waves propagate downstream.

Backward, all approaches are congested, conditions depend on downstream flow and waves propagate upstream.

Mixed, one approach (free flowing) has a higher priority and restricts the flow of the other approach, conditions for the high priority approach depend on upstream traffic and waves propagate downstream. For the other approach conditions depend on the downstream state and waves propagate upstream.

The forward and backwards cases are the most common. The backward case can occur due to congestion downstream of the junction or due to a lack of capacity at the junction itself. Mixed interaction is less common and typically happens at on-ramps that are merging with a high-flow mainline.

Let n be the downstream node of link m. If the flow being sent from all the in-links of the node n is less than the receiving flow of the downstream link then eqn. (3.2) holds and the shocks travel downstream for all approaches. Otherwise the flow out of the segment needs to reduced so that the total in-flow to the node does not exceed the receiving flow of the downstream link. Let each of the in-flowing links have a priority $p_n^m(k)$ that defines the proportion of the flow into the node n that comes from the upstream link m. Due to the allowable network representations there will be only two incoming links at a merge. Let m be the incoming merging link of interest and μ the alternate in-link. The flow out of the incoming merge-link m can be calculated as,

$$q_{m,N_m}(k) = \begin{cases} \text{mid}\{S_{m,N_m}(k), R_{m,N_m+1}(k) - S_{\mu,N_\mu}(k), p_n^m(k) R_{m,N_m+1}(k)\} \\ \qquad\qquad \text{if } R_{m,N_m+1}(k) < S_{m,N_m}(k) + S_{\mu,N_\mu}(k) \\ \min\{S_{m,N_m}(k), R_{m,N_m+1}(k)\} \quad \text{else.} \end{cases} \tag{3.19}$$

This covers the three possible scenarios and allows for the full traffic flow model to be defined.

3.5 METANET

3.5.1 Link model

Link types

As with the CTM links belong to one of four types. *Highway links* which define the main sections of highway. *Dummy links* of zero length to model complex topologies. *Origin links* to apply upstream BCs and pass flows into the network. *Destination links* to apply the downstream BC and output flow from the network.

Highway links

The equations to calculate the macroscopic traffic variables for segment i of link m are:

$$\rho_{m,i}(k+1) = \rho_{m,i}(k) + \frac{T}{L_m \lambda_m}[q_{m,i-1}(k) - q_{m,i}(k)] \tag{3.20}$$

$$q_{m,i}(k) = \rho_{m,i}(k) \cdot v_{m,i}(k) \cdot \lambda_m \tag{3.21}$$

$$\begin{aligned} v_{m,i}(k+1) = v_{m,i}(k) &+ \frac{T}{\tau}\{V[\rho_{m,i}(k)] - v_{m,i}(k)\} && \text{(relaxation)} \\ &+ \frac{T}{L_m} v_{m,i}(k)[v_{m,i-1}(k) - v_{m,i}(k)] && \text{(convection)} \\ &- \frac{\nu \cdot T}{\tau \cdot L_m} \frac{\rho_{m,i+1}(k) - \rho_{m,i}(k)}{\rho_{m,i}(k) + \kappa} && \text{(anticipation)} \end{aligned} \tag{3.22}$$

where ν is an anticipation constant, τ is the relaxation time constant, κ is a constant for numerical stability. All three are constants of the network and are the same for all links. $V[\rho_{m,i}(k)]$ is the equilibrium speed function with,

$$V[\rho_{m,i}(k)] = v_{f,m} \cdot \exp\left[-\frac{1}{\alpha_m}\left(\frac{\rho_{m,i}(k)}{\rho_{cr,m}}\right)^{\alpha_m}\right] \tag{3.23}$$

where $v_{f,m}$ is the free speed of link m, $\rho_{cr,m}$ denotes the critical density of link m and α_m is a parameter. Additional terms are included in the speed equation to account for speed drops due to merging and lane drop phenomena. At merges the term $-\delta T q_{\mu,N_\mu}(k) v_{m,1}(k)/(L_m \lambda_m(\rho_{m,1}(k)+\kappa))$ is included in the speed calculation (3.22) with δ a parameter and μ the merging link and m is the downstream link to which the calculation is applied for $i = 1$. At lane drops the speed is reduced in the final segment of the upstream link m to account for weaving effects by including the term $-\phi T \Delta\lambda \rho_{m,N_m}(k) v_{m,N_m}(k)^2/(L_m \lambda_m \rho_{cr,m})$ in (3.22) where $\Delta\lambda$ is the number of lanes being lost and ϕ is a constant parameter.

The density calculation (3.1) is from the conservation of vehicles, which is part of all macroscopic traffic models, hence eqn. (3.1) and eqn. (3.20) are identical. Flow is calculated via the hydrodynamic relation (3.21), and this is different from the CTM as mean speed dynamics are explicitly modelled in METANET.

The speed calculation (3.22) has three different terms. The relaxation term allows the mean speed to approach the equilibrium state defined by $V(\rho)$. The convection term propagates the upstream condition to the downstream. Finally, the anticipation term describes the drivers reaction to the downstream traffic density.

Origin links

The origins of the METANET model are also treated by a queuing model. The outflow from origin o is calculated as

$$q_o(k) = \min\left\{d_o(k) + \frac{g_o(k)}{T}, q_{\max,o}(k)\right\} \tag{3.24}$$

with $d_o(k)$ the demand at the origin o for time step k, $g_o(k)$ is the queue length in vehicles at time step k and $\bar{q}_o$ the capacity of the origin o. $q_{\max,o}(k)$ is the flow capacity of the origin o and is calculated by

$$q_{\max,o}(k) = \begin{cases} \bar{q}_o & \text{if } \rho_{\mu,1}(k) < \rho_{cr,\mu}(k) \\ \bar{q}_o \frac{\rho_{\max} - \rho_{\mu,1}(k)}{\rho_{\max} - \rho_{cr,\mu}} & \text{else} \end{cases} \tag{3.25}$$

where $\bar{q}_o$ is the capacity of the origin o, $\rho_{\max}$ is a network wide constant defining the maximum possible link density, and μ the index of the primary downstream link. This limits the outflow of the origin o to its capacity or the available space on the downstream link μ. This is similar to how origins are handled in CTM (3.13), the flow from the origin is the minimum of what will fit in the downstream, the origins capacity and the demand. The queue length is updated via a conservation equation,

$$g_o(k+1) = g_o(k) + T \cdot [d_o(k) - q_o(k)]. \tag{3.26}$$

Destination and dummy links

The downstream density for the destination ρ_b is supplied for every boundary b. This is slightly different to the CTM as the speed equation (3.22) requires a downstream density in the anticipation term. The CTM does not explicitly model the speed and instead requires a receiving function to be defined for destinations, to which ρ_b is applied.

3.5.2 Node Model

The node model is used to assign flows at highway junctions. Let $Q_n(k)$ be the sum of all flow entering the node n during time period k.

$$Q_n(k) = \sum_{\mu \in I_n} q_{\mu,N_\mu}(k) \quad \forall n \tag{3.27}$$

where I_n is the set of links entering node n and $q_{\mu,N_\mu}(k)$ is the flow at the end of the last segment N_μ of link μ during period k. The turning rate $\beta_n^m(k)$ defines the proportion of flow entering the node $Q_n(k)$ that during the time interval k leave through the out link $m \in O_n$, where O_n is the set of links leaving node n. This means that $q_{m,0}(k)$ required by eqn. (3.20) when $i = 1$ can be found by

$$q_{m,0}(k) = \beta_n^m(k) \cdot Q_n(k) \quad \forall m \in O_n. \tag{3.28}$$

In the case of a divergence node the density of the leaving links influence on the upstream incoming link's density needs to be accounted for. This is achieved by

$$\rho_{m,N_{m+1}}(k) = \frac{\sum_{\mu \in O_n} \rho_{\mu,1}^2(k)}{\sum_{\mu \in O_n} \rho_{\mu,1}(k)} \tag{3.29}$$

where $\rho_{m,N_{m+1}}(k)$ is the virtual density downstream of link m as required by (3.22) when $i = N_m$, and $\rho_{\mu,1}(k)$ is the density of the first segment of the out link μ. The reason for the quadratic term in (3.29) is that even though multiple out links could be clear (low density) if one is blocked (high density) it may still cause congestion to propagate upstream.

For merging nodes the speed of the upstream links has to be accounted for in the outgoing link. The speed upstream of the out link m, $v_{m,0}$ as required for $i = 1$ in (3.22), is calculated by

$$v_{m,0} = \frac{\sum_{\mu \in I_n} v_{\mu,N_\mu}(k) \cdot q_{\mu,N_\mu}(k)}{\sum_{\mu \in O_n} q_{\mu,N_\mu}(k)}. \tag{3.30}$$

More details can be found in the METANET manual [93].

3.6 Summary of Models

Both models described in the previous sections are non-linear dynamic systems in state space

$$\mathbf{x}(k+1) = f[\mathbf{x}(k), \mathbf{d}(k), \mathbf{z}] \qquad k = 0, 1, \ldots, K-1 \tag{3.31}$$

$$\mathbf{x}(0) = \mathbf{x}_0 \tag{3.32}$$

where $\mathbf{x}$ is the state, $\mathbf{d}$ the disturbance and $\mathbf{z}$ the parameter vector. The system is obtained for the CTM by substituting (3.10), (3.12), (3.17), (3.2), (3.19), (3.16) into (3.1); and (3.13), (3.18) into (3.14). For METANET (3.21), (3.27), (3.28) are substituted into (3.20); (3.29), (3.30) into (3.22); and (3.24),(3.25) into (3.26). The components of the vectors for each of the models is detailed in Table 3.1. Speed is not a state component in the CTM as it is a first order model, but values can be obtained using density from the fundamental diagram. The next key difference comes in the disturbance vector, the CTM needs a priority ratio to be defined at every merge. This defines the proportion of flows to come from the upstream links when the downstream is congested and cannot handle the demand. METANET with the speed equation slows down the upstream flows and therefore does not use a priority rate. In the parameter vector the CTM has more parameters due to its handling of boundary conditions. In CTM the boundary b has an FD and this is used to calculate a destinations receiving function. In METANET the boundary is handled through the speed equation and the downstream density at the destination is sufficient to handle the boundary. Also included in the METANET system is a limit on the speed to always be greater than the parameter $v_{\min}$. This is to ensure that the traffic flow does not change direction and to mitigate the error induced by the formulation of the original PDE. For both models the correct and valid $\mathbf{z}$ vector needs to be obtained for a specific site based on real data. Data must be supplied for the disturbance vector $\mathbf{d}$ as well as for comparison with the model outputs. The validation problem is discussed in Chapter 4 and the required data processing for the model inputs in Chapter 6. The two models were described here for reference and a change made to the CTM to use the same fundamental diagram as the METANET model.

Table 3.1: Components of the state-space model vectors.

	Quantity	Extent	CTM	METANET
x				
	$\rho_{m,i}$	$\forall\ m, i$	Y	Y
	$v_{m,i}$	$\forall\ m, i$	N	Y
	g_o	$\forall\ o$	Y	Y
d				
	d_o	$\forall\ o$	Y	Y
	ρ_b	$\forall\ b$	Y	Y
	β_m^n	for every out-link m of each bifurcation node n	Y	Y
	p_m^n	for every in-link m of each merge node node n	Y	N
z				
	α_m	$\forall\ m$	Y	Y
	$\rho_{cr,m}$	$\forall\ m$	Y	Y
	$v_{f,m}$	$\forall\ m$	Y	Y
	α_b	$\forall\ b$	Y	N
	$\rho_{cr,b}$	$\forall\ b$	Y	N
	$v_{f,b}$	$\forall\ b$	Y	N
	τ	Network	N	Y
	κ	Network	N	Y
	ν	Network	N	Y
	$\rho_{\max}$	Network	N	Y
	$v_{\min}$	Network	N	Y
	δ	Network	N	Y
	ϕ	Network	N	Y

Chapter 4

Model Validation

4.1 Introduction

All the traffic models discussed in Chapter 3 have parameters determining their response to various inputs and traffic states. They represent physical properties, such as a drivers' reaction time and the characteristics of the highway. Some however do not have an explicit physical meaning, such as κ in the METANET model (c.f. 3.5) which exists to ensure numerical stability. The characteristics and properties that they represent are specific to the model used, region and traffic flow. This means that a single set of parameters can not be used for all traffic models and the correct values must be identified for each. Model validation is the process of finding the correct parameter values for a particular model and it is a two phase process. The process of estimating a suitable set of parameters is the calibration (or parameter estimation) phase. The second is verification, which tests the calibrated values for a variety of traffic conditions to provide assurance that the model remains valid. The model validation problem is therefore a data driven activity. The calibration process is an optimisation problem. Verification uses the calibrated parameters to corroborate the models' accuracy by applying different BCs and ICs. This chapter aims at outlining a novel validation method that is reliable and independent of macroscopic simulator. Reliable in the sense that the method should be able to locate a suitable set of parameters for any dataset that it is supplied with. Independent from the model means that the developed method should not contain bespoke routines aimed at specific characteristics of a particular model.

Calibration is a quantitative procedure. The most common approach is to use a fitness measure to define the performance of the predicted model output relative to measured reference data as in [4, 4, 26, 36, 41, 42, 44, 50, 51]. Depending on the optimisation algorithm used the objective function may lead to a maximisation or minimisation problem. Another method is to calculate the properties of the shock waves captured in the available data and then use an algorithm to match the shock-speed, spatial growth rate and wavelength of any oscillations with those found by the model [61]. This approach was used to calibrate a microscopic model and allows for multiple data sources to be combined and used in the calibration process. In [63, 64] the parameter values are identified directly from the data. A combined calibration method is outlined in [92] where data analysis is used to find an initial point for a global optimisation. This method tries to ensure that the optimisation finds a global minimum and does not get stuck in a sub-optimal local minima.

The calibration methods used in [63, 64, 92] use the data directly to set FD parameters. These are set by minimising the squared error between the reference data and the FD. This approach is limited due to the complex interactions of traffic. Data measured at a location may not give the full FD or be affected by shock-waves. If a section of highway always exhibits free flow then it is not possible to determine the capacity and critical density directly from the measured data. Furthermore, the traffic at a specific location can be influenced by upstream or downstream conditions. Therefore the properties set using these methods are based on the combination of the highway characteristics at the current location and all points between the shock origin. An example of this can be seen in data from the M4 motorway with two measurement locations, one upstream and the other downstream of the off ramp at Junction 4B (to the M25, see Chapter 6 for details of the site). This junction has a congested off ramp and the congestion propagates back onto the main carriageway, as illustrated in Fig 4.1. Figure 4.2 shows the raw data from the upstream loop detector (red) and downstream (green). The upstream data measures a congested highway and it is possible to determine the highway's capacity. It is difficult to do so for the downstream detector. For the data shown a simple least-squares curve fitting algorithm minimising the residuals is used to predict the FD parameters in a similar

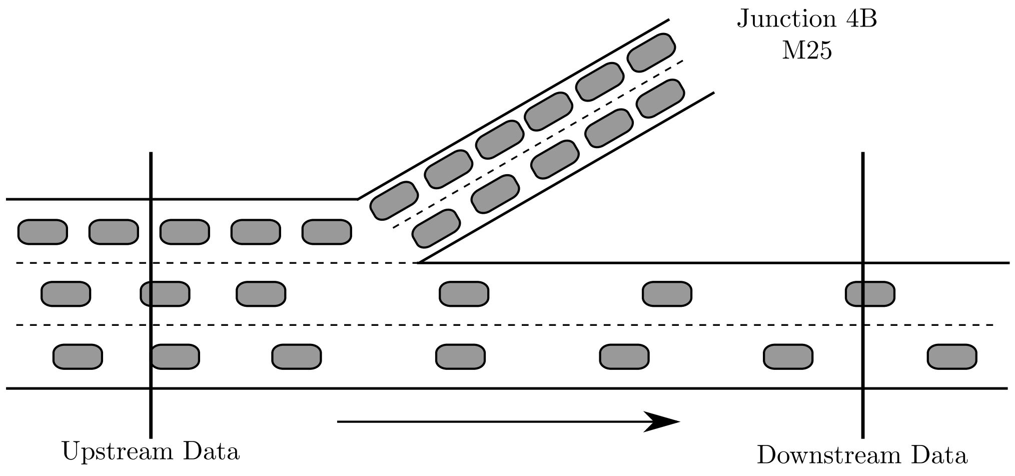


Figure 4.1: Congested off-ramp at Junction 4B of the eastbound M4.

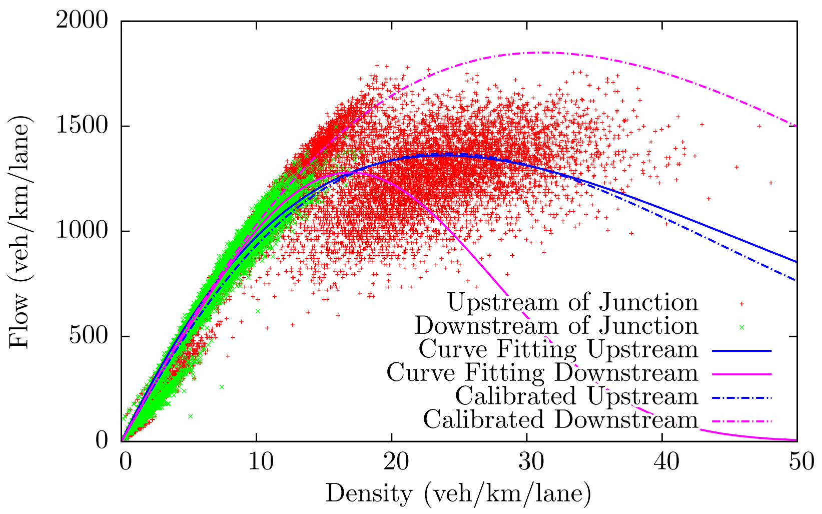


Figure 4.2: Sample data from M4 eastbound at Junction 4B, 5 randomly selected 24h periods from Feb 2010 with various FDs.

manner to the methods used in [63, 64, 92]; this is compared to the calibrated values from [4]. For the non-congested section of highway downstream of the off-ramp the FD parameters found by curve fitting are considerably lower than those found in [4]. This is because the full FD profile is not available to the curve-fitting algorithm. Upstream of the off-ramp both methods find similar parameters as congestion occurs at this location and a fuller profile of the FD is measured.

A drawback of the methods used in [63, 64, 92] is that each and every link within the model has a unique set of parameters. This could potentially result in an over parametrised model. An over-parametrised model will work well in calibration, but be finely tuned to small disturbances which are specific to the particular data and will not correctly express the aggregate driving characteristics of the network, hence the model is more likely to perform poorly in verification. In other works such as [4, 26, 36, 41, 42, 44, 50, 51] the model is restricted to a single set of parameters for the FD that is network-wide. However, these models do not cover large areas. In the calibration of the Amsterdam network[44], the model was split (prior to calibration) into four regions and each was given its own FD, preventing over parametrisation. This decision to split the network was subjective and based on expert opinion. Ideally any decision to split a model and define a new set of parameters due to a bottleneck or change in characteristics of a highway section should be included within the calibration problem.

Verification usually employs the same metrics as calibration. In [63, 64] a performance metric is not defined for calibration and verification uses the total travel time predicted by the model. In all cases the model is deemed to be valid or invalid based on the metric values found. A degradation in the metric value is expected as

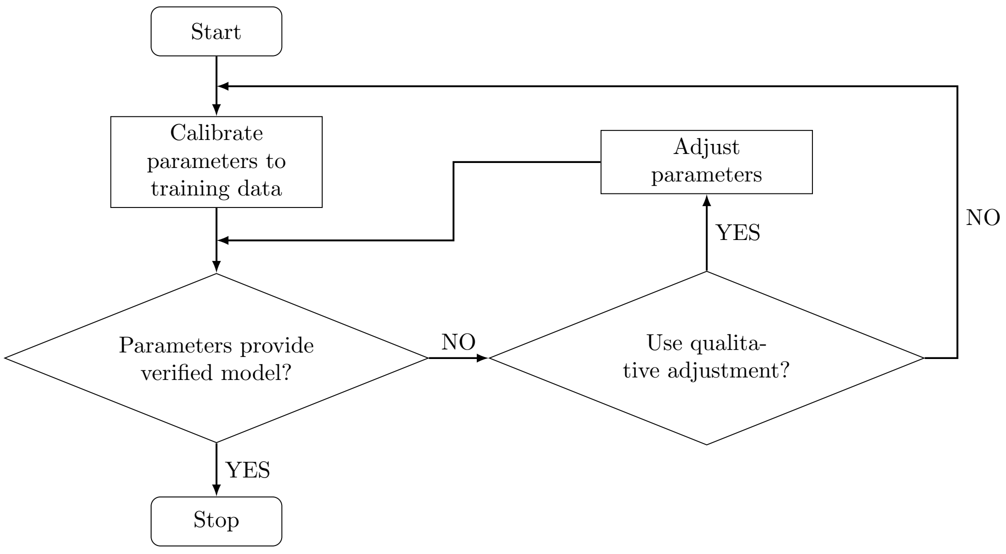

Start
Calibrate parameters to training data
Adjust parameters
YES
NO
Parameters provide verified model?
NO
Use qualitative adjustment?
YES
Stop

Figure 4.3: Validation procedure.

the training data will express different conditions than the verification data. A suitable threshold can be selected to decide if a model should be accepted as verified or not. In most cases however the determination of the model's validity is a qualitative evaluation by an expert. If the model is deemed invalid a qualitative procedure such as that used in [44] can be employed to improve it and the verification process is then repeated. In most cases if a set of calibrated parameters does not pass the verification test the entire procedure is restarted and a new set of calibrated parameters found. The validation problem is conceptually described in Figure 4.3.

In previous work the models being validated were typically short (approx. 5km) to mid scale (approx. 20km) unidirectional stretches. Ideally a networked site as considered in [44] is preferred. A road network site poses the additional challenge for the model to propagate congestion correctly from one highway to another. This is particularly useful in traffic control situations as a model of a network will allow for a traffic engineer to see the effects of any spill-back and ensure that the control strategy effectively relieves congestion rather than moving it to another region. Although [44] validates a network model, consisting of the Amsterdam ring road and the highways that supply it, a qualitative verification process was required to obtain a valid model. This adds an element of subjectivity and expert knowledge is required to adjust the model inputs to get the desired performance.

An ideal validation method should be applicable to large scale sites, which includes networks and not just linear topologies. As networks are considered, the model's size increases in both length and number of links. Using a different set of parameters for each link results in a large number of parameters and increases the probability of over-parametrisation. FDs are a spatial characteristic of traffic and should be applied over large regions. Bottlenecks or the network topology give good indicators to where a particular FD should end and another start. In a network model each highway will exhibit different characteristics. Also highway connections and merges are also likely to exhibit different characteristics as linking junctions usually have greater elevation and curvature changes. One of the contributions of this thesis is to select the number of FDs and their spatial assignment. The calibration problem with Automatic Assignment of Fundamental Diagrams (AAFD) can remove subjectivity and the requirement of expert analysis. For a robust and versatile solution, the validation method should, as far as possible, be independent of the macroscopic traffic flow model used.

This chapter outlines such a validation method addressing problems associated with large networks. Networks recieve little attention in the literature and most studies are undertaken on uni-directional stretches. Here the aim is to develop methods to not only cope with network topologies but to do so in a way that scales so entire regions of highways such as ring road systems around large metropolitan areas can be analysed. The chapter describes the calibration problem in detail. The optimisation problem formulation is presented and modifications allowing for AAFD on networks of any topology are incorporated. This is followed by a discussion of the verification method, metrics used and their calculation.

4.2 The Calibration Problem

4.2.1 General Problem Formulation

Recall from the previous chapter that the macroscopic models under investigation can be expressed as discrete time state space systems of the form,

$$\mathbf{x}(k+1) = \mathbf{f}[\mathbf{x}(k), \mathbf{d}(k), \mathbf{z}] \qquad k = 0, 1, \ldots, K-1 \tag{4.1}$$

$$\mathbf{x}(0) = \mathbf{x}_0 \tag{4.2}$$

with $\mathbf{x}(k)$ the state vector at discrete time step k, $\mathbf{d}$ is the disturbance vector, that also provides inputs and boundary conditions to the model, and $\mathbf{z}$ the parameter vector and K the time horizon. For any macroscopic model the parameter vector $\mathbf{z}$ can be generalised as,

$$\mathbf{z} = [\mathbf{z}_c^\top, \mathbf{z}_{\mathrm{L},1}^\top, \ \ldots \ , \mathbf{z}_{\mathrm{L},M}^\top]^\top \tag{4.3}$$

$\mathbf{z}_{\mathrm{L},m}$ are the FD parameters for link m, and M being the number of links within the model; for the CTM $\mathbf{z}_c$ contains a list of FDs for describing the properties of the destinations; in METANET $\mathbf{z_c}$ contains the set of network-wide parameters used in the speed equation.

The components of $\mathbf{x}(k)$ depend on the model order. In first-order models the only dynamically modelled macroscopic variable is the density, therefore the state vector solely consists of densities. In second-order models the speed is explicitly modelled as a dynamic variable and is therefore also included in the state vector.

Calibration and verification require reference data and in this thesis they come from the Highways Agency's MIDAS system (see Chapter 6). A measurement vector $\mathbf{y}$ is defined for each time step k,

$$\begin{aligned}\mathbf{y}(k) = [&y_{\rho,1}(k), y_{v,1}(k), y_{q,1}(k),\\ &y_{\rho,2}(k), y_{v,2}(k), y_{q,2}(k), \ \dots \ , y_{\rho,Y}(k), y_{v,Y}(k), y_{q,Y}(k)]^\top\end{aligned} \tag{4.4}$$

which gives the traffic state at various detector locations $j = 1, 2, \dots Y$, with Y the total number of detector locations. A measure of error can be defined based on $\mathbf{y}$.

The squared error between the predicted model density at time step k and the measured value at detector j can be expressed as,

$$E_{\rho,j}\left[\mathbf{x}, \mathbf{y}, k\right] = \left[y_{j,\rho}(k) - \rho_{m_j,i_j}(k)\right]^2 \tag{4.5}$$

where m_j, i_j give the link and segment index of detector j. The mean speed error,

$$E_{v,j}\left[\mathbf{x}, \mathbf{y}, k\right] = \left[y_{j,v}(k) - v_{m_j,i_j}(k)\right]^2. \tag{4.6}$$

The flow error is expressed as

$$E_{q,j}\left[\mathbf{x}, \mathbf{y}, k\right] = \left[y_{j,q}(k) - q_{m_j,i_j}(k)\right]^2. \tag{4.7}$$

From this, a weighted non-linear least-squares minimisation of the difference between the model $\mathbf{x}$ and measurements $\mathbf{y}$ can be formulated with an objective function given by

$$J(\mathbf{z}) = \sum_{k=1}^{K} \sum_{j=1}^{Y} \left(w_\rho E_{\rho,j}\left[\mathbf{x}, \mathbf{y}, k\right] + w_v E_{v,j}\left[\mathbf{x}, \mathbf{y}, k\right] + w_q E_{q,j}\left[\mathbf{x}, \mathbf{y}, k\right]\right) \tag{4.8}$$

with w_ρ, w_v and w_q being weights for the density, speed and flow error components respectively subject to (4.1), (4.2). These weights are included to account for the variance in the different macroscopic variables' magnitude [26, 42]. However, all three of the variables need not be considered by the optimisation, as they are linked via the hydrodynamic relation (2.3). The variables usually chosen are mean speed and flow as these are the most readily available from data sources, this involves setting $w_\rho = 0$ in (4.8) to give

$$J(\mathbf{z}) = \sum_{k=1}^{K} \sum_{j=1}^{Y} \left(w_v E_{v,j} \left[\mathbf{x}, \mathbf{y}, k \right] + w_q E_{q,j} \left[\mathbf{x}, \mathbf{y}, k \right] \right). \tag{4.9}$$

The minimisation of this function is a non-linear, non-convex least squares problem with numerous local minima. The complexity of this problem is discussed in [41]. The form of $J(\mathbf{z})$ given in (4.9) can be reduced to consider speed errors only. This still is sufficient to generate accurate models since the conservation of vehicles equation is used by the model and it is fed with the correct BCs, the flow and density can be implicitly obtained by the correct behaviour of the speed dynamics. Therefore only the errors the in mean speed have to be considered [51]. Furthermore the form of (4.9) will change with the size and scale of the model which can make comparisons of the performance difficult as the baseline value for a good solution will vary this issue can be reduced by scaling by the number of time steps K and measurements locations Y,

$$J(\mathbf{z}) = \frac{1}{KY} \sum_{k=1}^{K} \sum_{j=1}^{Y} E_{v,j} \left[\mathbf{x}, \mathbf{y}, k \right]. \tag{4.10}$$

Note that (4.10) is a version of (4.9) with $w_v = \frac{1}{KY}$ and $w_q = 0$.

Each measurement location has an equal weighting in this objective function. Longer links within the model will on average contain more measurement locations and therefore have a greater influence on the final objective function and shorter links will have a smaller impact.

The full calibration problem involves finding parameters $\mathbf{z}$ to minimise eqn. (4.9) subject to the traffic model (4.1), (4.2) as shown in Figure 4.4.

To make the optimisation independent of the simulator used, during calibration the required simulation specific input files are generated and $\mathbf{x}$ is read from the simulator output. From this the objective function is evaluated for the optimization algorithm.

4.2.2 Linear Networks

Consider the linear site as shown in Figure 4.5. It consists of a series of links in sequential order without any highway-highway junctions, i.e the only connections are to on/off ramps attached to the sides. The network topology is fully defined by the link indices sequence,

$$p_1, p_2, \ldots, p_M \tag{4.11}$$

where p_i is the index of the link in position i. An assignment of FDs (that cover several links) consists of an array of integers

$$\gamma_1, \gamma_2, \dots, \gamma_{\widehat{m}}, \dots, \gamma_{\widehat{M}} \tag{4.12}$$

where $\gamma_{\widehat{m}}$ denotes the number of links FD $\widehat{m}$ is applied to and $\widehat{M}$ the maximum number of FDs used for this site. The list of links where FD 1 is applied is initialised at p_1; the final link where FD 1 is applied is p_{γ_1}. The second FD starts at p_{γ_1+1} and ends at $p_{\gamma_1+\gamma_2}$ and so forth. Hence, the vector $\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \dots, \gamma_{\widehat{m}}]^\top$ defines the assignments of FDs to a linear site. Obviously

$$\sum_{\widehat{m}=1}^{\widehat{M}} \gamma_m = M \tag{4.13}$$

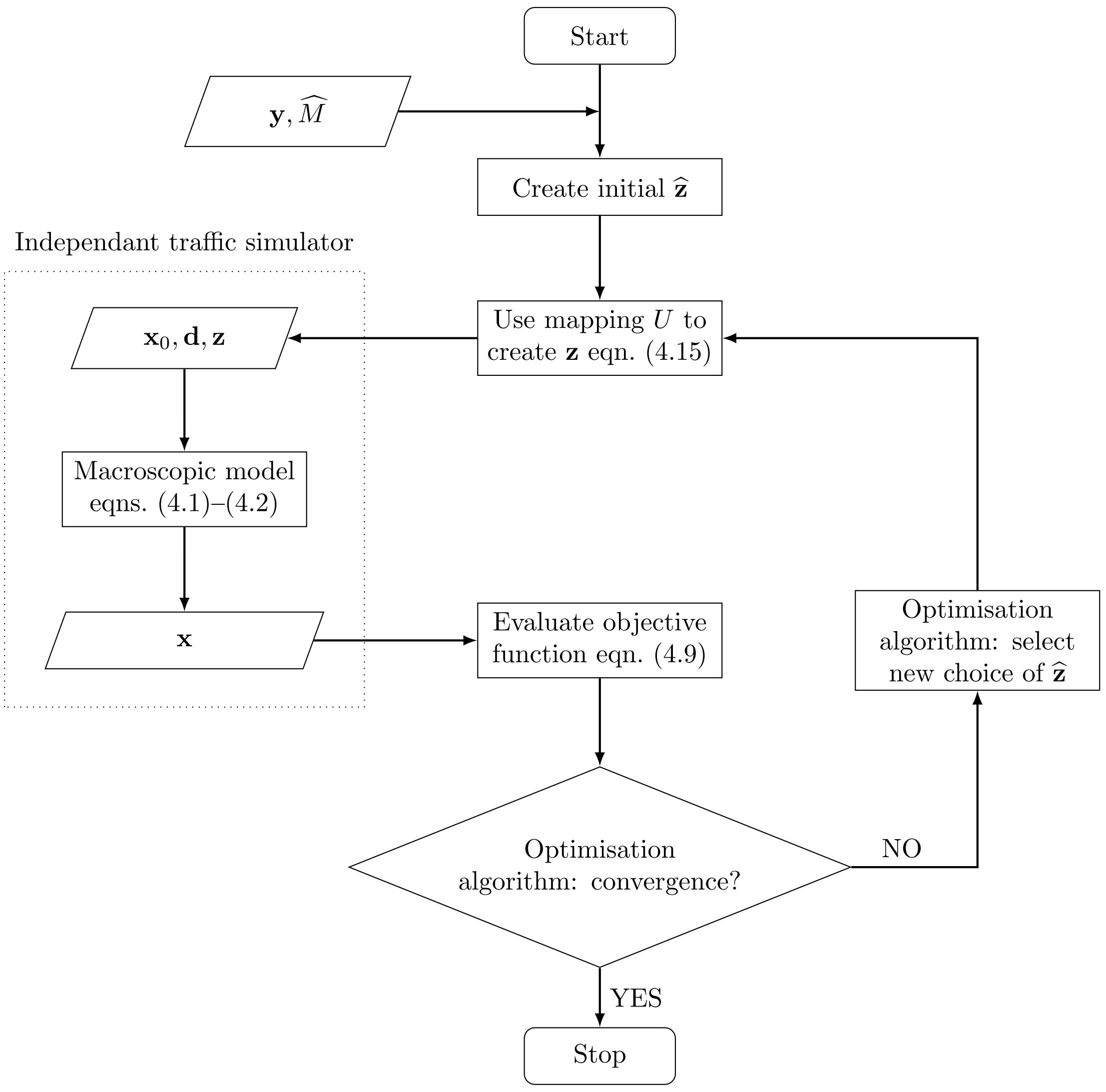

Start
$\mathbf{y}, \widehat{M}$
Create initial $\widehat{\mathbf{z}}$
Independant traffic simulator
$\mathbf{x}_0, \mathbf{d}, \mathbf{z}$
Use mapping U to create $\mathbf{z}$ eqn. (4.15)
Macroscopic model eqns. (4.1)–(4.2)
$\mathbf{x}$
Evaluate objective function eqn. (4.9)
Optimisation algorithm: select new choice of $\widehat{\mathbf{z}}$
Optimisation algorithm: convergence?
NO
YES
Stop

Figure 4.4: Calibration process with a previously determined mapping.

Algorithm 4.1 Mapping for a linear network.

```
Input: $\widehat{\mathbf{z}} = [\widehat{\mathbf{z}}_c^\top, \mathbf{z}_{\text{FD},1}^\top, \ \dots \ , \mathbf{z}_{\text{FD},\widehat{M}}^\top, \gamma_1, \ \dots \ , \gamma_{\widehat{M}}]^\top$
Output: $\mathbf{z} = [\mathbf{z}_c^\top, \mathbf{z}_{\text{L},1}^\top, \ \dots \ , \mathbf{z}_{\text{L},M}^\top]^\top$
 1: $\mathbf{z}_c := \widehat{\mathbf{z}}_c$
 2: $m := 1$
 3: for $\widehat{m} = 1, 2, \dots, \widehat{M}$ do
 4:     $j := 0$
 5:     if $\widehat{m} = \widehat{M}$ then
 6:         $\gamma_{\widehat{m}} := M$
 7:     end if
 8:     while $j < \gamma_{\widehat{m}}$ do
 9:         $\mathbf{z}_{\text{L},m} := \mathbf{z}_{\text{FD},\widehat{m}}$
10:         $m := m + 1$
11:         if $m > M$ then
12:             return $\mathbf{z}$
13:         end if
14:         $j := j + 1$
15:     end while
16: end for
```

to ensure all links are assigned an FD. If an FD of the possible $\widehat{M}$ is not used $\gamma_{\widehat{m}} = 0$. A parameter vector $\widehat{\mathbf{z}}$ is defined as

$$\widehat{\mathbf{z}} = [\mathbf{z}_c^\top, \mathbf{z}_{\text{FD},1}^\top, \ \dots \ , \mathbf{z}_{\text{FD},\widehat{M}}^\top, \gamma_1, \ \dots \ , \gamma_{\widehat{M}}]^\top \tag{4.14}$$

where $\mathbf{z}_{\text{FD},\widehat{m}}$ are the parameters for FD $\widehat{m}$. With a suitable algorithm $\mathbf{z}$ of eqn. (4.3) can be obtained from $\widehat{\mathbf{z}}$ eqn. (4.14) as,

$$\mathbf{z} = U[\widehat{\mathbf{z}}] \tag{4.15}$$

where U is a mapping of $\widehat{\mathbf{z}}$; sequential links in $\mathbf{z}$ that are part of the same FD have the same parameters. For a linear site, U is implemented by Algorithm 4.1. The optimisation problem now involves finding $\mathbf{z}$ via $\widehat{\mathbf{z}}$.

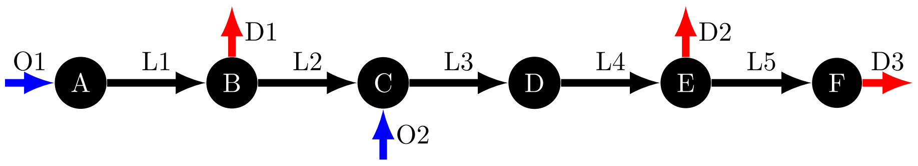
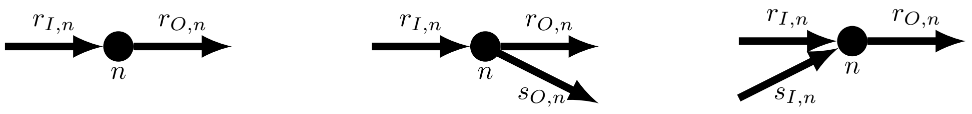


Figure 4.5: Example of a linear site.



(a) Simple node. (b) Diverge node. (c) Merge node.

Figure 4.6: Node model, with major and minor links defined.

4.2.3 Complex Networks

Algorithm 4.1 only works for linear sites with ordered link indices. To apply it to on arbitrary topology networks they need to be decomposed into a series of linear sections. Algorithm 4.1 can be applied to each section and networks of any topology can be calibrated. It is possible to perform such a split based on topology information. A linear section has the following characteristics:

- It starts at an origin of a highway or a diverge from a mainline.
- It consists of a set of links that represents a mainline.
- It ends at a destination of a highway or a merge into another mainline.

Each section is treated as a separate linear site for mapping its links' parameters from the vector of decision variables. Adjoining sections are allowed to have different FD parameters.

To determine the linear sections of a specific network, the network geometry needs to be known. The definition of a section means that at each node the mainline needs to be identified. Figure 4.6 shows how this is achieved, via the definition of major incoming $r_{I,n}$ and minor incoming $s_{I,n}$ links at node n. The same is applied to the outgoing links O of node n; one is set as major $r_{O,n}$ and at diverge nodes the other out-link is set as minor $s_{O,n}$.

To apply Algorithm 4.1 the section start location needs to be defined and the total number of links within the section to be known. Furthermore, links may no longer be in index order so a traversal of the mainline is required for the application of parameters. Simple nodes that receive flow from an origin and pass it to a normal link will always be a start of a section. Other section start points, like those that are diverges from a mainline still need to be identified.

By traversing the network from a known start location, Algorithm 4.2 calculates the length of the section. Algorithm 4.2 also calculates start points from diverges. The algorithm does this by travelling along the graph in the direction of flow choosing the path along the major out links at each node. During this traversal the section starting locations at diverges are identified as any minor out link coming off the mainline. This process is repeated until all known start points have been visited.

Algorithm 4.2 Walk algorithm to traverse section.

```
Input: link index $m$ of section start, index of section $b$, set of destinations $\Phi$, the index of major links $r$, index of minor links $s$, list of downstream nodes $n_d$
Output: Section length $L_{s,b}$, list of section start locations $H_w$, number of sections $B_w$, boolean list stating if link has been traversed $W_w$
 1: $B_w := 0$
 2: $n := n_{\mathrm{d},m}$
 3: $W_{w,m} :=$ True
 4: $L_{\mathrm{s},b} := 1$
 5: if $s_{O,n}$ exists then
 6:     if $s_{O,n} \notin \Phi$ then
 7:         $B_w := B_w + 1$
 8:         $H_{w,B_w} := s_{O,n}$
 9:     end if
10: end if
11: while $m = r_{I,n}$ and $r_{O,n} \notin \Phi$ do
12:     $m := r_{O,n}$
13:     $n := n_{\mathrm{d},m}$
14:     $W_{w,m} :=$ True
15:     $L_{\mathrm{s},b} := L_{\mathrm{s},b} + 1$
16:     if $s_{O,n}$ exists then
17:         if $s_{O,n} \notin \Phi$ then
18:             $B_w := B_w + 1$
19:             $H_{w,B_w} := s_{O,n}$
20:         end if
21:     end if
22: end while
23: return $L_{s,b}, B_w, H_w, W_w$
```

The algorithm takes as input the starting link index m of the section, the section index b, network information such as the set of destinations Φ, the list of major links r and minor links s, and nodes n_d. The algorithm commences by initialising the count of identified starting locations B_w, where w is the section being walked. Link traversal state $W_{w,m}$ is set to true and the length of the current section s, $L_{\mathrm{s},b}$ is set to 1. Lines 5–10 determine if a new section starts at the node n. This happens when the node has a diverge and the divergent minor link $s_{O,n}$ is not a member of the set of destinations Φ. In this case the counter B_w is incremented and the link index of the newly found section is appended to H_w. H_w is one of the outputs of the algorithm and contains a list of section start indices diverging from the walked section.

The main loop of Algorithm 4.2 occurs in lines 11–22. Here the algorithm moves along the major out link of node n (line 12) and finds the downstream node (line 13). $W_{w,m}$ is updated and the length of the section is incremented. Any additional starting locations are identified (lines 15–20). The while loop ends when the whole section has been traversed. This happens when either the downstream major link of the node is a destination $r_{O,n} \in \Phi$ or the section merges into another mainline. A merge into another mainline can be evaluated by comparing the current link index m with the major in-link of the downstream node $r_{I,n}$; if it is different then a merge has occurred.

The walk algorithm (Alg. 4.2) evaluates a section and identifies the sections that start as diverges from it, storing them in H_w. To find all the sections within a network Algorithm 4.3 is used. The splitting algorithm (Alg. 4.3) starts by setting the indexed walked flag to false for all links. Section start points that occur from highway origins are identified in lines 5–10. The walk algorithm is then applied to these known start locations (lines 11–18). Any start locations identified by walks are appended to H and B is incremented and the walked flag updated. Therefore after completion of the **for** loop all of the sections found initially from major origins and those identified during the walks themselves have been traversed.

However, closed loops can exist in networks (ring roads, roundabouts). These links have not been assigned to a section. By evaluating W, the algorithm confirms that all links have been traversed; while this is false the untravelled link with the smallest index is selected as a section start and a new series of walks are performed. This ensures that the algorithm traverses all links and closed loops are captured by the sectioning algorithm.

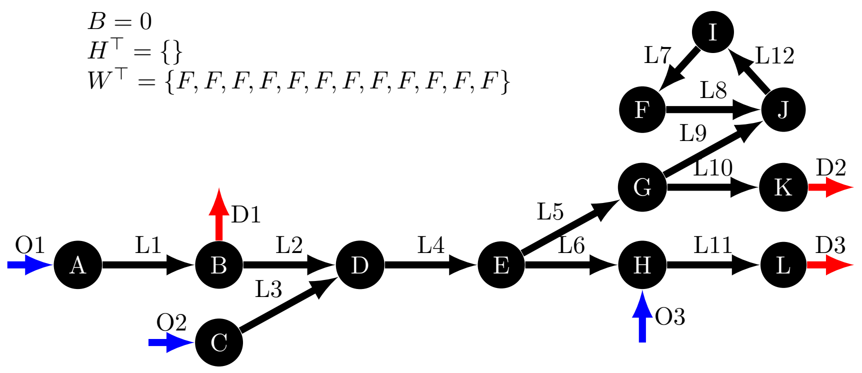
Algorithm 4.3 is applied to an example network in Figure 4.7, where multiple out and in-links exist. The horizontal oriented links are the mainlines, i.e. high capacity highways. The first stage of the algorithm is to set the walked parameter to false as in Figure 4.7a. Afterwards the set of origins is used to initialise the lists B and H; in the example origins $O1$ and $O2$ are the major in-links of nodes A and H respectively so they are added to H and B is set to 2, Figure 4.7b. Figure 4.7c shows the first walk; as each link is walked its parameter W is set to true; the divergence at node B is ignored as it is to a destination and is the minor out-link. At node E the out-links are both ordinary links, the minor out-link $L5$ is therefore added to H and B is incremented. The walk stops at node L as the only out-link is a destination. For the next walk shown in Figure 4.7d the section is only one link long as a merge occurs at node D and the link $L3$ is not the major in-link so the walk stops and the algorithm moves on to the next start location $L5$, which was found on the first walk. This walk (Figure 4.7e) identifies another diverge, saves the minor out-link $L9$ as a start point and continues along the major out-links until it stops at the destination $D2$. The divergence found in the previous walk is then used as a start point in Figure 4.7e, which immediately ends as it runs into a merge and is

Algorithm 4.3 Splitting networks into linear sections.

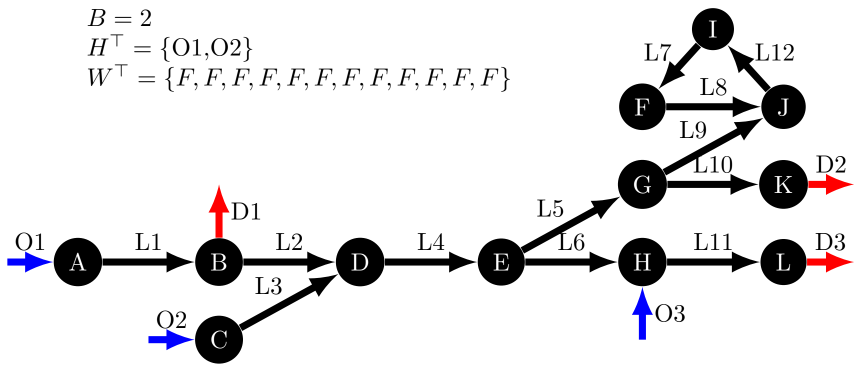
Input: Set of origins Θ, Set of Destinations Φ, Major links r, minor links s, Number of nodes ψ, Downstream node n_d

Output: Set of section start locations H, Number of sections B, Length of sections L_s

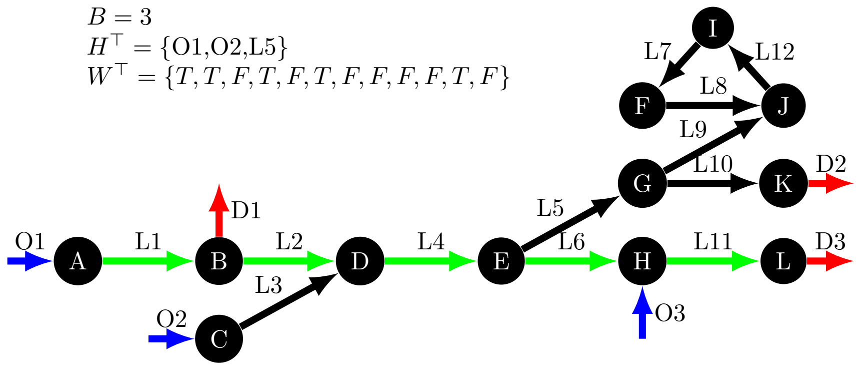
```
1: $B := 0$
2: for $m = 1, 2, \dots, M$ do
3:     $W_m = \text{False}$
4: end for
5: for $n = 0, 1, \dots, \psi$ do
6:     if $r_{I,n} \in \Theta$ then
7:         $B = B + 1$
8:         $H_B = r_{O,n}$
9:     end if
10: end for
11: for $b = 1, 2, \dots, B$ do
12:     $(L_{s,b}, H_w, B_w, W_w)$=$\textsc{Walk}(H_b,b,\Phi,r,s,n_d)$
13:     $B := B + B_w$
14:     $H := [H^\top, H_w^\top]^\top$
15:     for $m = 1, 2, \dots, M$ do
16:         $W_m := W_m$ or $W_{w,m}$
17:     end for
18: end for
19: for $m = 1, 2, \dots, M$ do
20:     if $W_m = \text{False}$ then
21:         $B := B + 1$
22:         $H_B := m$
23:         $b := b + 1$
24:         while $b \leq B$ do
25:             $(L_{s,b}, H_w, B_w, W_w)$=$\textsc{Walk}(H_b,b,\Phi,r,s,n_d)$
26:             $B := B + B_w$
27:             $H := [H^\top, H_w^\top]^\top$
28:             for $m = 1, 2, \dots, M$ do
29:                 $W_m := W_m$ or $W_{w,m}$
30:             end for
31:             $b := b + 1$
32:         end while
33:     end if
34: end for
35: return $H, B, L_s$
```


$B = 0$
$H^\top = \{\}$
$W^\top = \{F, F, F, F, F, F, F, F, F, F, F, F\}$
A B C D E F G H I J K L
L1 L2 L3 L4 L5 L6 L7 L8 L9 L10 L11 L12
O1 O2 O3
D1 D2 D3

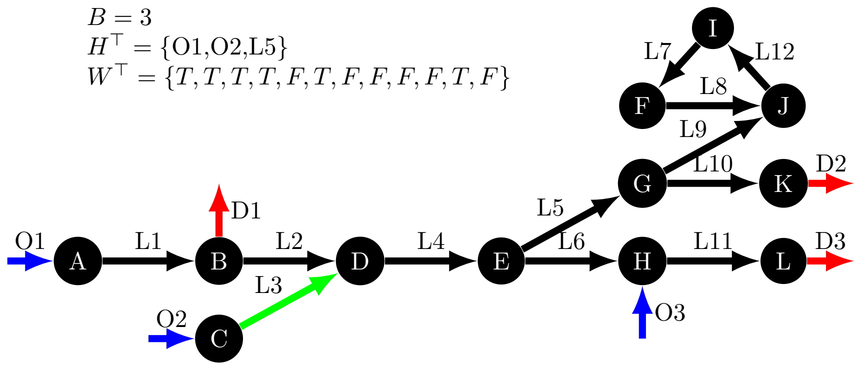
(a) State of Algorithm 4.3 after line 4.


$B = 2$
$H^\top = \{$O1,O2$\}$
$W^\top = \{F, F, F, F, F, F, F, F, F, F, F, F\}$
A B C D E F G H I J K L
L1 L2 L3 L4 L5 L6 L7 L8 L9 L10 L11 L12
O1 O2 O3
D1 D2 D3

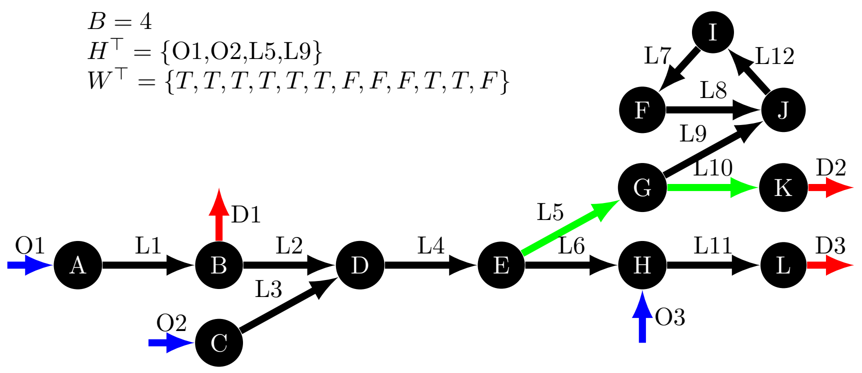
(b) State of Algorithm 4.3 after line 10.


$B = 3$
$H^\top = \{$O1,O2,L5$\}$
$W^\top = \{T, T, F, T, F, T, F, F, F, F, T, F\}$
A B C D E F G H I J K L
L1 L2 L3 L4 L5 L6 L7 L8 L9 L10 L11 L12
O1 O2 O3
D1 D2 D3

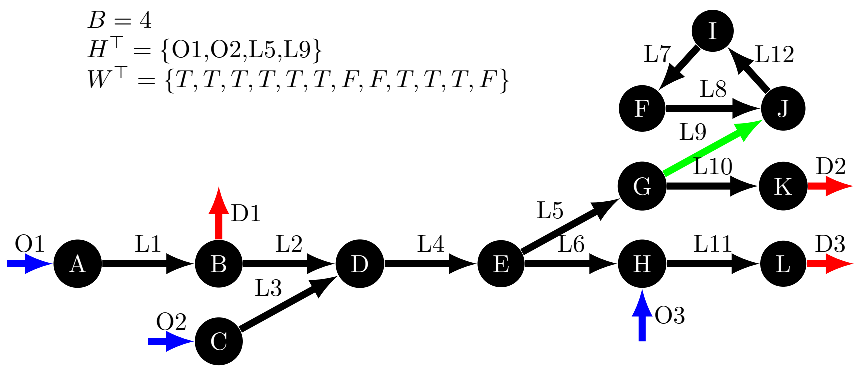
(c) State of Algorithm 4.3 after line 16 for $b = 1$, links walked since Figure 4.7b coloured in green.


$B = 3$
$H^\top = \{$O1,O2,L5$\}$
$W^\top = \{T, T, T, T, F, T, F, F, F, F, T, F\}$
A B C D E F G H I J K L
L1 L2 L3 L4 L5 L6 L7 L8 L9 L10 L11 L12
O1 O2 O3
D1 D2 D3

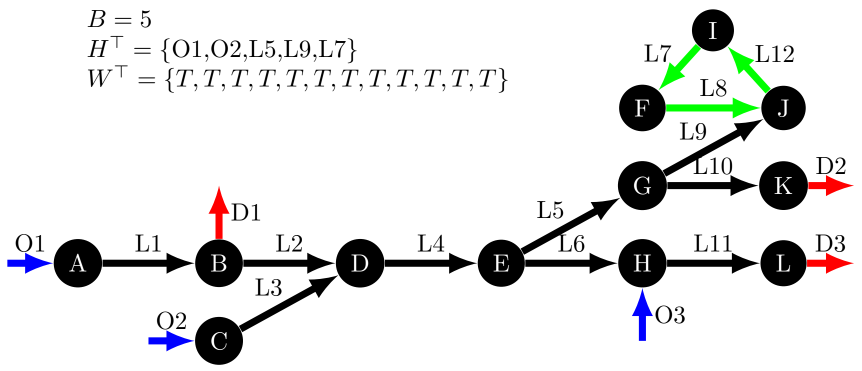
(d) State of Algorithm 4.3 after line 16 for $b = 2$, links walked since Figure 4.7c coloured in green.


$B = 4$
$H^{\top} = \{O1,O2,L5,L9\}$
$W^{\top} = \{T,T,T,T,T,T,F,F,F,T,T,F\}$
I
L7
L12
F
L8
J
L9
G
L10
K
D2
D1
L5
O1
A
L1
B
L2
D
L4
E
L6
H
L11
L
D3
L3
O3
O2
C

(e) State of Algorithm 4.3 after line 16 for $b = 3$, links walked since Figure 4.7d coloured in green.


$B = 4$
$H^{\top} = \{O1,O2,L5,L9\}$
$W^{\top} = \{T,T,T,T,T,T,F,F,T,T,T,F\}$
I
L7
L12
F
L8
J
L9
G
L10
K
D2
D1
L5
O1
A
L1
B
L2
D
L4
E
L6
H
L11
L
D3
L3
O3
O2
C

(f) State of Algorithm 4.3 after line 16 for $b = 4$, links walked since Figure 4.7e coloured in green.


$B = 5$
$H^{\top} = \{O1,O2,L5,L9,L7\}$
$W^{\top} = \{T,T,T,T,T,T,T,T,T,T,T,T\}$
I
L7
L12
F
L8
J
L9
G
L10
K
D2
D1
L5
O1
A
L1
B
L2
D
L4
E
L6
H
L11
L
D3
L3
O3
O2
C

(g) End of Algorithm 4.3, links walked since Figure 4.7f coloured in green.

Figure 4.7: Splitting algorithm on an example network.

not the major in-link of node J. At this point all of the identified start points have been used and the sections from them traversed so the boolean value W for each link is evaluated. This identifies that $L7$ has not been used so it is added as a start location and another walk is performed as in Figure 4.7g. W is then updated and as all links have been walked the algorithm ends. We now know the network contains 5 separate sections and the start points are available through the vector H. The sections are defined as the following routes; $O1 \rightarrow L1 \rightarrow L2 \rightarrow L4 \rightarrow L6 \rightarrow L11$, $O2 \rightarrow L3$, $L5 \rightarrow L10$, $L9$ and $L7 \rightarrow L8 \rightarrow L12$.

With the vector of section starting locations H identified along with lengths L_s and number B, it becomes possible to map the FD parameter vector $\widehat{\mathbf{z}}$ to the link parameter vector $\mathbf{z}$ for any network. A section must have at least one FD assigned to it, therefore the minimum number of FDs is B. To further minimise the number of parameters a maximum number Γ of FDs per section is set. Each section b is assigned a set of FDs it can apply which is the minimum of Γ, or one FD per link. This means that the total number of possible FDs $\widehat{M}$ the model can use is,

$$\widehat{M} = \sum_{b=1}^{B} \min(\Gamma, L_{\mathrm{s},b}). \tag{4.16}$$

Algorithm 4.1 needs adjusting so that it travels in a similar manner as the walk algorithm 4.2. It needs to be updated to account for multiple linear sections where the links may not be index ordered as in Figure 4.7. For this more general case the algorithm for implementing the mapping U for networks is given in Algorithm 4.4. The first FD starts at link H_1 and extends for γ_1 links following the same walk as before. The FDs are assigned concurrently in this manner until one of two conditions occurs, the section reaches its end or the final available FD is being used. When the walk reaches the section end, due to a merge or destination, the index of the last FD available $\gamma_{\widehat{m}}$ is ignored. At this point the FD stops being applied to further links and the process begins at the start of the next section H_{b+1} with the next sections starting FD $\widehat{m} = 1 + \sum_{i=0}^{b} \min(\Gamma, L_{\mathrm{s},i})$, skipping over any unused FDs available for the current section. The second case occurs when $\widehat{m} = \sum_{i=0}^{b} \min(\Gamma, L_{\mathrm{s},i})$, meaning that the maximum number of FDs are being applied to the section. In this scenario the value of $\gamma_{\widehat{m}}$ is treated as M and the FD is applied until the end of the section to ensure all links are assigned parameters.

4.2.4 Automatic Assignment of FDs

A method for mapping any network allows for AAFD but there is a potential issue. The optimisation problem has no terms to minimise the number of parameters used. Also the splitting of networks into linear sections imposes the model to use a new

FD for each section. This is acceptable in most cases as changes in the road characteristics occur at merges and diverges, but this does not apply in all situations. The optimisation should attempt to reduce the numbers of FDs within each section and if possible make the FD parameters the same where two sections meet at a bifurcation or merge. This can be achieved by including a penalty function in the objective function (4.9),

$$J(\mathbf{z}) = \sum_{k=1}^{K}\sum_{j=1}^{Y}\left(w_v E_{v,j}\left[\mathbf{x},\mathbf{y},k\right] + w_q E_{q,j}\left[\mathbf{x},\mathbf{y},k\right]\right) \\ + w_p \sum_{m=1}^{M}\sum_{\substack{\mu\in O_{a_m}\\ \mu\notin\Phi}} (\mathbf{z}_{\mathrm{L},m}-\mathbf{z}_{\mathrm{L},\mu})^\top w_{\mathrm{L}}(\mathbf{z}_{\mathrm{L},m}-\mathbf{z}_{\mathrm{L},\mu}) \quad (4.17)$$

Algorithm 4.4 Mapping for an arbitrary topology network.

Input: $\widehat{\mathbf{z}} = [\widehat{\mathbf{z}}_c, \mathbf{z}_{\mathrm{FD},1}^\top, \ \dots\ , \mathbf{z}_{\mathrm{FD},\widehat{M}}^\top, \gamma_1, \ \dots\ , \gamma_{\widehat{M}}]^\top$, number of sections B, section start locations H, section lengths L_s.
Output: $\mathbf{z} = [\mathbf{z}_c^\top, \mathbf{z}_{\mathrm{L},1}^\top, \ \dots\ , \mathbf{z}_{\mathrm{L},M}^\top]^\top$

```
1:  $\mathbf{z}_c := \widehat{\mathbf{z}}_c$
2:  $\widehat{m} := 1$
3:  for $b = 1, 2, \dots, B$ do
4:      $m := H_b$
5:      while SectionEnd = false do
6:          if $\widehat{m} = \sum_{i=1}^{b}\min(\Gamma, L_{\mathrm{s},i})$ then
7:              $\gamma_{\widehat{m}} := M$
8:          end if
9:          $j := 0$
10:         while $j < \gamma_b$ do
11:             $n := n_{\mathrm{d},m}$
12:             $\mathbf{z}_{\mathrm{L},m} := \mathbf{z}_{\mathrm{FD},\widehat{m}}$
13:             if $m = r_{I,n}$ and $r_{O,n} \notin \Phi$ then
14:                 $m := r_{O,n}$
15:             else
16:                 SectionEnd = true
17:                 $\widehat{m} := 1 + \sum_{i=1}^{b}\min(\Gamma, L_{\mathrm{s},i})$
18:                 break
19:             end if
20:             $j := j + 1$
21:         end while
22:         $\widehat{m} := \widehat{m} + 1$
23:     end while
24: end for
25: return $\mathbf{z}$
```

with w_{L} a diagonal matrix to account for variances between the different FD parameters and w_p a scaling factor to control the effect of the penalty term. The index μ provides the index of the out-links of the node a_m downstream of link m that do not belong to destinations Φ. This formulation aims to simultaneously minimise the changes between a link's parameters and those downstream as well as matching the calibration data. The optimisation will aim at selecting solutions where the parameters at a merge and a diverge are similar for all links despite the fact that they will belong to two different sections. In this thesis the smooth FD (3.23) is considered, repeated here for convenience

$$V[\rho_{m,i}(k)] = v_{f,m} \cdot \exp\left[-\frac{1}{\alpha_m}\left(\frac{\rho_{m,i}(k)}{\rho_{cr,m}}\right)^{\alpha_m}\right], \tag{4.18}$$

this has three parameters so for link m,

$$\mathbf{z}_{\mathrm{L},m} = [v_{f,m}, \rho_{cr,m}, \alpha_m]^\top \tag{4.19}$$

which means w_{L} has the form

$$w_{\mathrm{L}} = \begin{bmatrix} w_{\mathrm{L},v} & 0 & 0 \\ 0 & w_{\mathrm{L},\rho} & 0 \\ 0 & 0 & w_{\mathrm{L},\alpha} \end{bmatrix}. \tag{4.20}$$

Where $w_{\mathrm{L},v}$ is the free speed weighting, $w_{\mathrm{L},\rho}$ the critical density weight and $w_{\mathrm{L},\alpha}$ the weight for the α parameter.

With the weightings $w_q = 0$ and $w_v = \frac{1}{KY}$ (4.17) becomes,

$$J(\mathbf{z}) = \frac{1}{KY}\sum_{k=1}^{K}\sum_{j=1}^{Y} E_{v,j}\left[\mathbf{x},\mathbf{y},k\right] + w_p \sum_{m=1}^{M} \sum_{\substack{\mu\in O_{a_m} \\ \mu\notin\Phi}} (\mathbf{z}_{\mathrm{L},m} - \mathbf{z}_{\mathrm{L},\mu})^\top w_{\mathrm{L}} (\mathbf{z}_{\mathrm{L},m} - \mathbf{z}_{\mathrm{L},\mu}). \tag{4.21}$$

4.2.5 Model Calibration

The complete calibration process is conceptually shown in Figure 4.8. The objective is to optimise the FD parameter vector (4.14) to minimise the function (4.17). The link parameter vector $\mathbf{z}$ (4.3), is obtained from $\widehat{\mathbf{z}}$ via the mapping U, eqn. (4.15). To create U the network is split into linear sections by Algorithm 4.3. U is then implemented by Algorithm 4.4. The objective function (4.17) consists of two components. The first component is the squared error between the model predicted output and measured data $\mathbf{y}$. The second is the term penalising the changes in FD parameter values over the network.

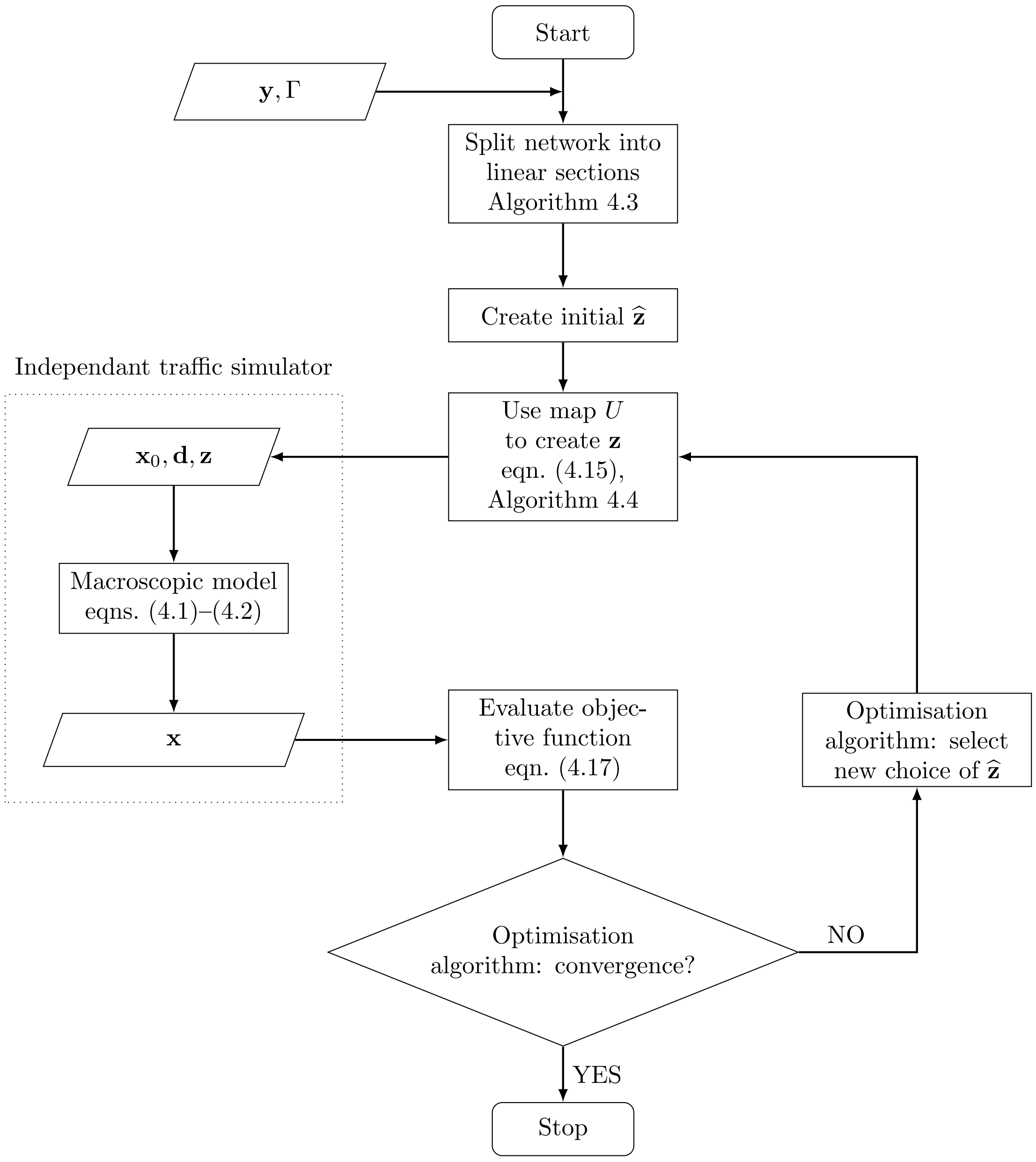

Start
$\mathbf{y}, \Gamma$
Split network into linear sections Algorithm 4.3
Create initial $\widehat{\mathbf{z}}$
Independant traffic simulator
Use map U to create $\mathbf{z}$ eqn. (4.15), Algorithm 4.4
$\mathbf{x}_0, \mathbf{d}, \mathbf{z}$
Macroscopic model eqns. (4.1)–(4.2)
$\mathbf{x}$
Evaluate objective function eqn. (4.17)
Optimisation algorithm: select new choice of $\widehat{\mathbf{z}}$
Optimisation algorithm: convergence?
NO
YES
Stop

Figure 4.8: Calibration process for arbitrary networks of any topology.

The proposed calibration process (Figure 4.8) is independent of both the underlying traffic model equation and the topology of the model site. If a first-order model was to be calibrated by this technique then the parameters of the model are those that define the FD of each link. If a second-order model was to be calibrated the parameter vector would be increased by the introduction of parameters governing the speed equation. All second-order macroscopic models still require the use of the FD parameters as they include a relaxation term in the speed equation. Clearly, the proposed scheme depends on the FD used. If the final output is a solution that uses a minimal number of FDs then the value of w_p can be decreased. Furthermore as

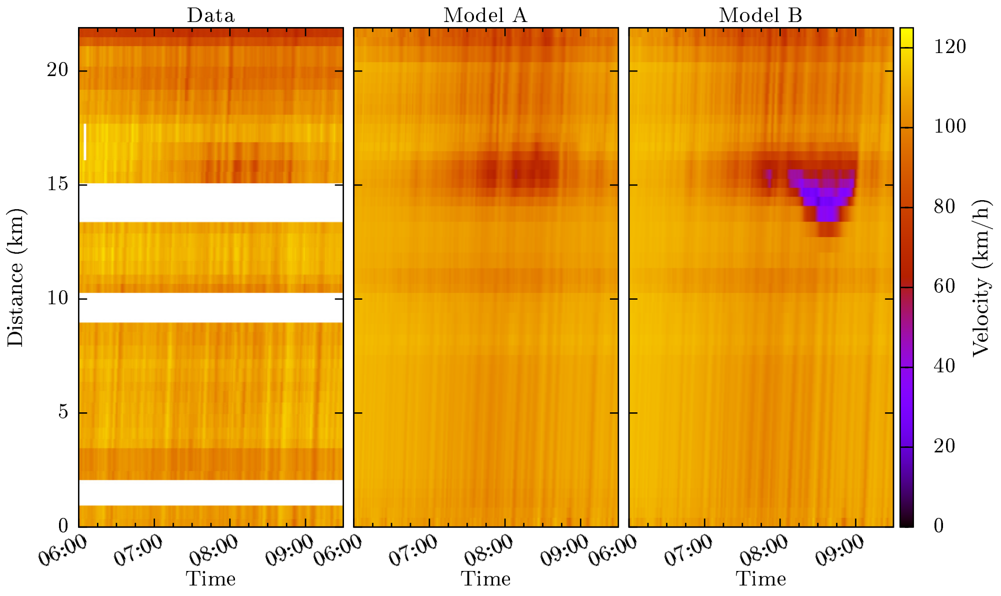


Figure 4.9: Verification test for the Sheffield METANET model using data of the 29th for parameter set A and B calibrated on the 8th. Result an excerpt from Chapter 9.

the penalty function is applied on a link by link basis the scalar w_p will need to be scaled relative to the size of the model being calibrated. With calibration complete the parameter vector $\mathbf{z}$ will need to be verified.

4.3 Verification

The verification process provides assurance that the calibrated parameters work for a range of conditions. A calibration may determine that congestion within a model is caused due to a certain effect (such as a bottleneck) and select parameters appropriately. This, however, may not be the actual cause of the congestion and an alternate could exist (such as speed drops due to merging effects). This may mean that for different sets of input data the model may not correctly reproduce the congestion pattern. This example can be seen in Figure 4.9 where a model of Sheffield (see 6.4.3) calibrated using data from the 8th is applied to data of the 29th in the verification test. The data of the 29th has lower demand inputs and does not exhibit the recurrent congestion pattern. As seen in the figure, model A correctly predicts the free flow conditions and whilst B incorrectly predicts congestion.

To determine if a model should be considered verified the metric from calibration without the penalty term (4.9) will be used for the additional data sets, which is repeated here:

$$J(\mathbf{z}) = \sum_{k=1}^{K} \sum_{j=1}^{Y} \left(w_v E_{v,j} \left[\mathbf{x}, \mathbf{y}, k \right] + w_q E_{q,j} \left[\mathbf{x}, \mathbf{y}, k \right] \right) \tag{4.22}$$

The penalty term is not included for the verification metric as its value represents the difference in the link parameters and will not change. This can then be used to see how the model performs for data other than that used in calibration. The calibration data is evaluated also to provide a benchmark. The solution is expected to degrade when applied to different data, but the trends should be the same. If the metrics are similar then speed-time plots at the cross sections where data are available, or speed-space-time maps (Figure 4.9) can be used to confirm that the model should be considered verified. The squared error metric (4.9) for the models shown in Figure 4.9 are given in Table 4.1 where $w_v = \frac{1}{KY}$ and $w_q = 0$. The table also includes the value of the metric for the calibration data. This provides a baseline value against which comparisons can be judged. A reduction in the accuracy is usually expected in verification this is not the case for parameter set A due to the fact it was calibrated against a data set exhibiting congestion, with shock waves being generated by the model. As the congestion is no longer present the conditions remain stable and it is easier for the model (with correct parameters) to replicate the data. Table 4.1 also illustrates the effect of the speed drop due to the congestion predicted by parameter set B has on the error function. This can be used to deduce that the model should not be considered verified and this can confirmed by Figure 4.9. The white horizontal bars in the data plot show gaps in the data source. As step changes can occur at junctions and topology changes interpolation is not used and the region of the plot is left blank at these locations.

Table 4.1: Squared error for parameter sets A and B of the Sheffield METANET model

Parameter set	8^{th}	29^{th}
A	64.20	63.32
B	59.77	112.05

Verification is performed using data from the same time-frame on the same day of week, year and month as the calibration data. This data should exhibit similar characteristics to the calibration data and therefore the degradation of the models accuracy to the measured data should be small. If the model passes these tests then the verification problem is made harder. The model is then applied to data sets with less resolution to confirm that accuracy is retained. Also the model is verified against datasets from other days of the week and time-frames. The point of this is to see the versatility of the obtained parameter set. Traffic flow has different demands and profiles based on many variables, such as weather, season, time of day, national holidays, school holidays and many others. Data used in validation is restricted to within a calendar month of the training data as demand profiles can change over time, which may not have any kind of cyclic pattern, due to the new business hubs forming or declining and therefore changing the demand profiles.

In the work of [44] a large network is verified by a qualitative process. To achieve a verified model the inputs were perturbed using expert opinion to improve the models accuracy for the extra data sets. This is not ideal as these inputs should be reflect the real data as closely as possible. By changing the inputs in this manner the model is no longer representing the real system as accurately as it could. The process itself also requires a detailed knowledge of the underlying model and network to adjust the inputs correctly. This approach is not repeated here as it involves changing the model away from the real system and requires intensive user involvement. Instead if the model fails the verification tests the calibration process is restarted to find a different parameter set that may be a more accurate representation of the real system.

Due to the specific requirements of the final model, verification decision is left in the hands of the user. A metric based on the objective function can be used to assist in quantifying the accuracy of the calibrated model for various data sets. This in conjunction with examination of heat maps and speed-time profiles at various cross-sections will allow for an expert to determine if the model should be considered verified. It is possible to automate this process by defining a value for the objective function that declares that the model has reached sufficient accuracy. The parameter vector returned from the AAFD process should also be examined at this point to ensure that the locations for changes in the spatial parameters appear logical.

4.4 Validation method

This chapter has developed a novel validation method that allows for autonomous selection of spatial changes to the parameters of a traffic model. The entire validation procedure is illustrated in Figure 4.10. The proposed formulation is independent of simulator and can be easily applied to any macroscopic model with any expression for the FD. The calibration method only uses the outputs and inputs of the traffic model. This means that there are no extra requirements upon a user beyond the initial creation of the traffic model. This may mean that other external studies before performing a validation can be skipped such as network analysis and bottleneck identification. The work can be left to the optimisation algorithm. Verification has remained in the hands of the user at this point to determine proof of concept. This could be automated by either selecting a maximum percentage decay in the objective function (eqn. 4.9), or by using other advances in computation techniques such as pattern recognition on contour plots of the network velocities. The next chapter will

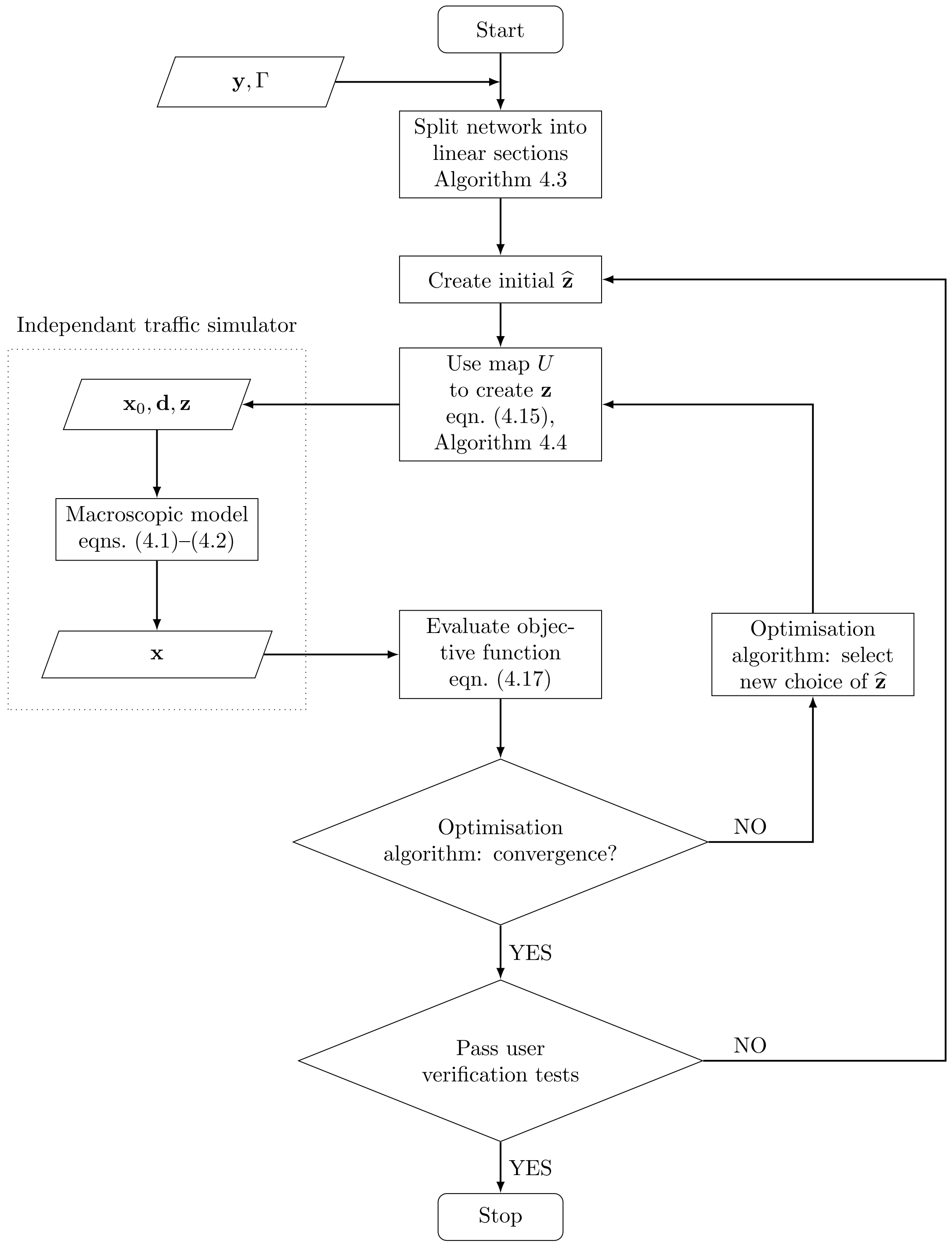

Start
$\mathbf{y}, \Gamma$
Split network into linear sections Algorithm 4.3
Create initial $\widehat{\mathbf{z}}$
Independant traffic simulator
Use map U to create $\mathbf{z}$ eqn. (4.15), Algorithm 4.4
$\mathbf{x}_0, \mathbf{d}, \mathbf{z}$
Macroscopic model eqns. (4.1)–(4.2)
$\mathbf{x}$
Evaluate objective function eqn. (4.17)
Optimisation algorithm: select new choice of $\widehat{\mathbf{z}}$
Optimisation algorithm: convergence?
NO
YES
Pass user verification tests
NO
YES
Stop

Figure 4.10: Validation with automatic spatial assignment of Fundamental Diagrams.

outline a series of possible optimisers to be used in this validation method.

The methods developed for calibration can deconstruct complex networks into a series of uni-directional sections. This method makes it possible for any network regardless of complexity or scale to be analysed. The proposed methods use the topology information provided in the model definitions therefore it is easy to create modular routines. The framework means that it is simple to change the optimisation scheme or underlying traffic model being analysed.

Chapter 5

Optimisation Algorithms

5.1 Introduction

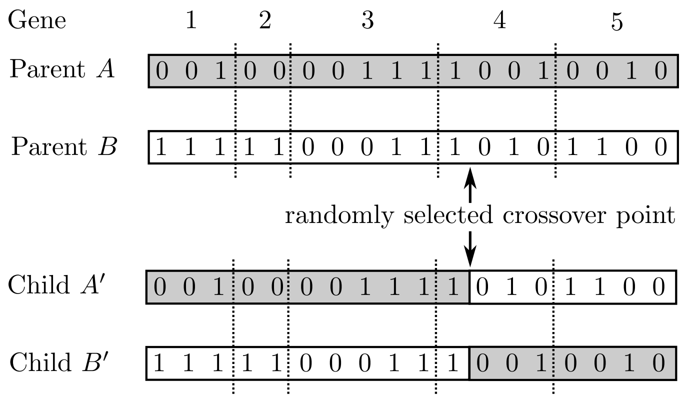
In the previous chapter the validation problem was discussed in detail. To solve the problem an optimisation solver is required to select an appropriate $\hat{\mathbf{z}}$ to minimise the objective function (4.17). As far as the optimisation is concerned the function evaluation and AAFD as discussed previously can be treated as a black box. The solver passes the parameter vector $\hat{\mathbf{z}}$ into a routine and the required information such as $J(\mathbf{z})$ is returned. Many different optimisation algorithms exist and could be applied to the validation problem for example [69, 70, 76, 94, 95]. Past works have used the optimisation method of Box [46] as in [26, 42, 44], Nelder-Mead [49] as in [50, 50, 51] and a cross-entropy method [52] was used in [41]. Here the focus is on nature inspired algorithms such as GAs and PSO. These algorithms use populations of solutions to determine how the solution space is explored. Due to the complex, non-linear solution space none of the prior validation attempts reported in literature used gradient methods. Partly due to the fact that an analytical gradient cannot be determined and the complexity of the problem regarding local minima. In a review about potential methods [54] dismisses grdaient based approaches due to numerous local minima in the solution space. The nature of the problem formulation dictates the AAFD algortihm set-up but through Automatic Differentiation (AD) a gradient for the calibration problem can be obtained. In this chapter the details of the algorithms used, GA, PSO, Cuckoo Search (CS) and Resilient back PROPagation (RPROP) are given. The calculation of the gradient required by RPROP is also discussed as are the required modifications to AAFD.

5.2 Genetic Algorithm (GA)

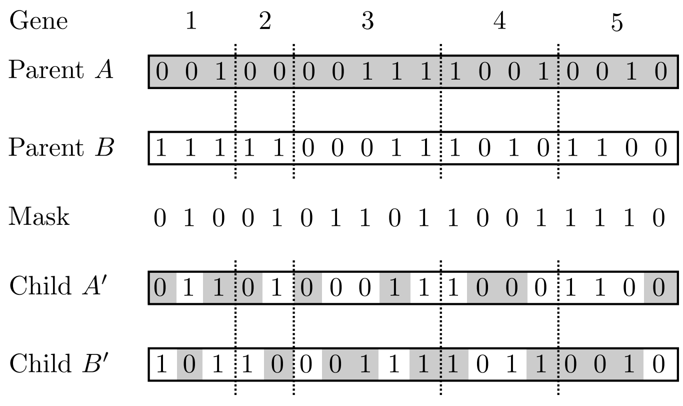
5.2.1 Philosophy

The original GA [94] was inspired by genetics and mimics the process of evolution. The algorithm has a population of solutions with each solution being a chromosome. The chromosome is composed of genes that represent the variables in the optimisation problem. The algorithm has three processes;

Selection a random sample biased towards fitter solutions is selected,
Crossover the selected chromosomes are combined to make offspring,
Mutation a random change is applied to a gene.



(a) Single-point Crossover.



(b) Mask Crossover.

Figure 5.1: Binary crossover operations.

In the original algorithm the genes were expressed in a binary format. Suppose two binary chromosomes A and B were selected as parents for crossover. Figure 5.1 depicts two possible crossover operators for these chromosomes and their resultant children A' and B'. In single point crossover (Figure 5.1a) an arbitrary point is selected, two child chromosomes are then generated as shown.

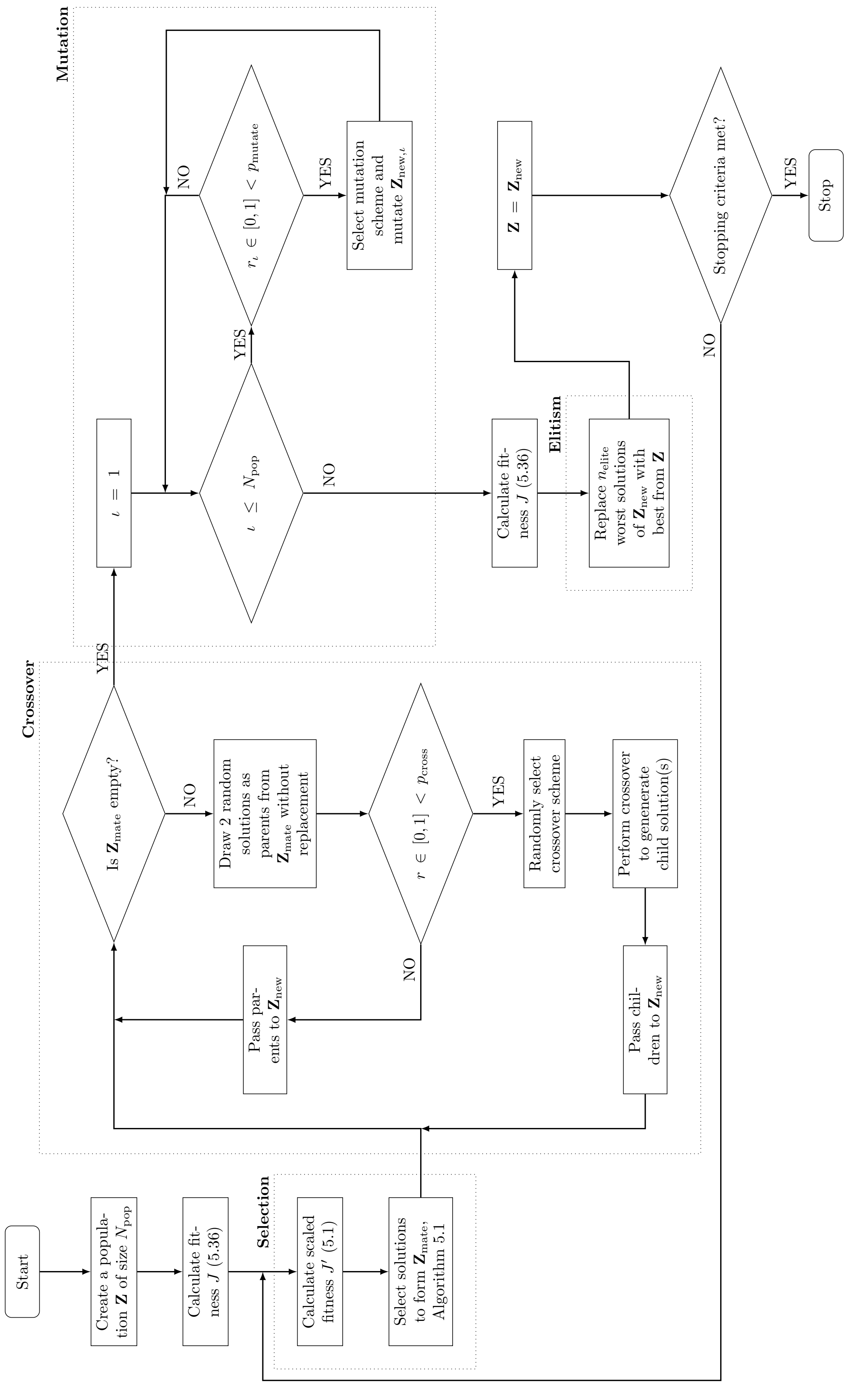

Start
Create a population $\mathbf{Z}$ of size N_{pop}
Calculate fitness J (5.36)
Selection
Calculate scaled fitness J' (5.1)
Select solutions to form $\mathbf{Z}_{\mathrm{mate}}$, Algorithm 5.1
Crossover
Is $\mathbf{Z}_{\mathrm{mate}}$ empty?
NO
Draw 2 random solutions as parents from $\mathbf{Z}_{\mathrm{mate}}$ without replacement
$r \in [0,1] < p_{\mathrm{cross}}$
YES
Randomly select crossover scheme
Perform crossover to genenerate child solution(s)
Pass children to $\mathbf{Z}_{\mathrm{new}}$
NO
Pass parents to $\mathbf{Z}_{\mathrm{new}}$
YES
Mutation
$\iota = 1$
$\iota \leq N_{\mathrm{pop}}$
YES
$r_\iota \in [0,1] < p_{\mathrm{mutate}}$
NO
YES
Select mutation scheme and mutate $\mathbf{Z}_{\mathrm{new},\iota}$
NO
Calculate fitness J (5.36)
Elitism
Replace n_{elite} worst solutions of $\mathbf{Z}_{\mathrm{new}}$ with best from $\mathbf{Z}$
$\mathbf{Z} = \mathbf{Z}_{\mathrm{new}}$
Stopping criteria met?
YES
Stop
NO

Figure 5.2: Genetic Algorithm.

In this thesis a floating point GA was implemented as it allows for the algorithm to be more computationally efficient [96] with no conversions required between an encoding scheme and the real variable values [97, 98]. Floating point GA's have also been shown to provide a faster convergence rate for problems with a large number of dimensions [96] and share convergence properties with its binary predecessor [99, 100]. The GA used is based on that described in [96] and the overview of the specific algrithm used is shown in Figure 5.2.

5.2.2 Selection

The selection process needs to stochastically select individuals with bias on their fitness to create a mating pool of equal size to the population. The mating pool is used to create the next generation and is passed to the crossover operator. The algorithm employed here uses the stochastic remainder selection without replacement method [96], using a fitness function scaled by sigma truncation. Fitness scaling is common practice in GAs to prevent rapid take over of the population by extremely fit chromosomes [94, 96].

Sigma truncation is a linear scaling method that removes solutions that are n_σ standard deviations worse than the average from the search. The scaling sets the fitness of this truncation point as zero and the other solutions are then scaled appropriately. A modified fitness $J'_\iota(\mathbf{z})$ for population member ι, that gives positive fitness values and is a maximisation problem, is achieved using sigma truncation by

$$J'_\iota(\mathbf{z}) = \begin{cases} -J_\iota(\mathbf{z}) + (\overline{J} + n_\sigma\sigma) & \text{if } J_\iota(\mathbf{z}) \leq (\overline{J} + n_\sigma\sigma) \\ 0 & \text{else} \end{cases} \tag{5.1}$$

where $\overline{J}$ is the mean of the objective function values $J_\iota(\mathbf{z})$ of the current population.

With a scaled objective function the selection process is implemented following Algorithm 5.1. Let the current population be $\mathbf{Z}$ and the mating pool be $\mathbf{Z}_{\text{mate}}$. For each population member ι the expected value e_ι is calculated and integer copies are passed to the mating pool (lines 1-5). The remainder $\mathfrak{r}_\iota$ is then used to calculate a selection probability p_ι (lines 7-9). The pool is then completed by random trails using p_ι, once selected the probability is set to zero to prevent more than one copy being added (lines 10-22). This creates a selection pool with a range of solutions close to the expected values but with a bias to allow some weaker solutions.

Algorithm 5.1 Stochastic remainder selection without replacement.

```
Input: $J'_\iota(\mathbf{z})\ \forall\ \iota$, population $\mathbf{Z}$, population size $N_{\text{pop}}$
Output: Mating pool $\mathbf{Z}_{\text{mate}}$
 1: for $\iota = 1, 2, \ldots, N_{\text{pop}}$ do
 2:     $e_\iota = N_{\text{pop}} \left( J'_\iota(\mathbf{z}) / \sum_{j=0}^{N_{\text{pop}}} J'_j(\mathbf{z}) \right)$
 3:     Pass $\lfloor e_\iota \rfloor$ copies of $\mathbf{Z}_\iota$ to $\mathbf{Z}_{\text{mate}}$
 4:     $\mathfrak{r}_\iota = e_\iota - \lfloor e_\iota \rfloor$
 5: end for
 6: $r_{\max} = 1.0$
 7: for $\iota = 1, 2, \ldots, N_{\text{pop}}$ do
 8:     $p_\iota = \mathfrak{r}_\iota / \sum_{j=0}^{N_{\text{pop}}} \mathfrak{r}_j$
 9: end for
10: repeat
11:     $r_{\text{select}}$ = random number $\in [0, r_{\max}]$
12:     $r = 0$
13:     for $\iota = 1, 2, \ldots, N_{\text{pop}}$ do
14:         $r = r + p_\iota$
15:         if $r > r_{\text{select}}$ then
16:             Pass a copy of $\mathbf{Z}_\iota$ to $\mathbf{Z}_{\text{mate}}$
17:             $r_{\max} = r_{\max} - p_\iota$
18:             $p_\iota = 0$
19:             break for
20:         end if
21:     end for
22: until $\mathbf{Z}_{\text{mate}}$ is full
```

5.2.3 Crossover

This is the main operator of the algorithm and combines existing parts of the genetic code from the mating pool to create the new population. Many operators can be used to increase the algorithms ability to search the space effectively [77]. The four different schemes utilised here are, single point, mask, whole arithmetical and heuristic crossover. These are defined as:

1. Single point crossover acts in a manner similar to that of a traditional binary crossover operator. If population members $\mathbf{Z}_1 = \hat{\mathbf{z}} = (z_1, ..., z_A)$ and $\mathbf{Z}_2 = \hat{\mathbf{z}} = (\mathfrak{z}_1, ..., \mathfrak{z}_A)$ are to be crossed after the a^{th} position then the new population members are:

$$\begin{aligned} \mathbf{Z}'_1 &= (z_1, ..., z_a, \mathfrak{z}_{a+1}, ..., \mathfrak{z}_A) \\ \mathbf{Z}'_2 &= (\mathfrak{z}_1, ..., \mathfrak{z}_a, z_{a+1}, ..., z_A) \end{aligned} \tag{5.2}$$

where A is the number of dimensions, the size of vector $\hat{\mathbf{z}}$.

2. Mask crossover is similar to single point crossover however it generates a mask to establish if crossover occurs, hence it has up to $A-1$ crossover points. The mask $\mathcal{M} = (m_1, ..., m_A)$ contains binary bits which are randomly generated. If an index of the mask is equal to 1 then the parents are crossed at that point. If a denotes the index of the dimension, the new members are given by:

$$\mathbf{Z}'_{1,a} = \begin{cases} \mathbf{Z}_{1,a} & \text{if } \mathcal{M}_a = 0 \\ \mathbf{Z}_{2,a} & \text{if } \mathcal{M}_a = 1 \end{cases} \tag{5.3}$$

$$\mathbf{Z}'_{2,a} = \begin{cases} \mathbf{Z}_{2,a} & \text{if } \mathcal{M}_a = 0 \\ \mathbf{Z}_{1,a} & \text{if } \mathcal{M}_a = 1 \end{cases} \tag{5.4}$$

for $a = 0, ..., A$.

3. Whole arithmetical crossover combines two population members as follows:

$$\begin{aligned} \mathbf{Z}'_1 &= r \cdot \mathbf{Z}_1 + (1-r) \cdot \mathbf{Z}_2 \\ \mathbf{Z}'_2 &= r \cdot \mathbf{Z}_2 + (1-r) \cdot \mathbf{Z}_1 \end{aligned} \tag{5.5}$$

where r is a random number $\in [0, 1]$.

4. Heuristic crossover creates a single offspring $\mathbf{Z}_3$ where parent $\mathbf{Z}_2$ is fitter than $\mathbf{Z}_1$, $\mathbf{Z}_3$ is given by:

$$\mathbf{Z}_3 = r \cdot (\mathbf{Z}_1 - \mathbf{Z}_2) + \mathbf{Z}_2 \tag{5.6}$$

The first two operators are modifications of the original binary operators. However, these operators are combinatorial and will only find the best configuration of the initial parameters. This is where the operators such as arithmetical and heuristic crossover come in. They combine the solutions by creating a child solutions between the parents values.

The new population is generated by drawing two random solutions from the pool without replacement. A random number is generated and if this is less than the probability of crossover occurring, p_{cross}, then one of the crossover operations is selected randomly and child solutions generated. In the case of heuristic crossover only a single is child is created so one of the parents is selected at random to survive into the next generation. If crossover does not occur then the two selected individuals from the mating pool are passed to the next generation with no alteration.

5.2.4 Mutation

Mutation inputs new genetic material to move the search to new regions of the solution space. Two mutation operations are utilised in this algorithm; uniform and non-uniform.

1. Uniform mutation takes a single population member at random, and selects a random dimension $a \in (1, ..., A)$ and replaces this with a random number within the search domain [96]. This allows for a detailed search of the space and is important in the early stages of the algorithm to move around the search area.
2. Non-uniform Mutation allows for fine tuning of the parameters and moves the solution locally. The amount of movement from the current parameter value is reduced over time. A random dimension a is selected from $\mathbf{Z} = (z_1, ..., z_a, ..., z_A)$, this value is then mutated to:

$$\mathbf{Z}'_a = \begin{cases} z_a + \triangle(\mathfrak{c}, UB - z_a) & \text{if a random digit is 0} \\ z_a + \triangle(\mathfrak{c}, z_a - LB) & \text{if a random digit is 1} \end{cases} \tag{5.7}$$

   LB and UB are the lower and upper bounds of variable z_a respectively, and $\triangle(\mathfrak{c}, y)$ returns values in the range $[0, y]$ in a way that the probability of the value being close to 0 rising as $\mathfrak{c}$ increases, where $\mathfrak{c}$ is the iteration number and r a random number $\in [0, 1]$ [96]. This ensures that during the beginning of the algorithm a more uniform search is carried out that over time becomes more refined.

$$\triangle(\mathfrak{c}, y) = y \cdot \left(1 - r^{\left(1-\frac{\mathfrak{c}}{\mathfrak{C}}\right)^{c_{\text{mutate}}}}\right) \tag{5.8}$$

   in which $\mathfrak{C}$ is the maximum number of iterations and c_{mutate} is a system parameter describing the dependency on the iteration number.

As with crossover there is a probability of a solution mutating p_{mutate}, at each iteration every member of the new population is checked to see if a mutation occurs. If so, then uniform or non-uniform is selected randomly without bias and the solution altered as described.

5.2.5 Elitism

The elitism operator is the final operation in an iteration of the GA. This operation exists to ensure that the best solutions are not lost and remain in the search [94]. Elitism is very simple, the number of elite solutions to be maintained between generations is defined as n_{elite}. At the end of each iteration the best n_{elite} solutions from the previous generation replace the worst solutions of the new generation.

5.2.6 Genetic Algorithm Overview

The GA's convergence is determined by it's population size N, crossover rate p_{cross} and mutation rate P_{mutate}. The crossover and mutation rates control how often these operations occur. The crossover rate is usually high as this is the main search operator, it is not 100% as this would not allow for members of the previous generation to pass to the next. The mutation rate is much lower than the crossover rate, if the mutation rate is too high the algorithm may become divergent as the solutions don't have time to converge before a mutation potentially causes a detrimental change. However to search the whole solution space effectively the mutation rate needs to be as high as possible. The population size is the key factor in determining the time the algorithm takes to converge, large populations take a longer time to converge but provide a better initial search.

5.3 Particle Swarm Optimisation (PSO)

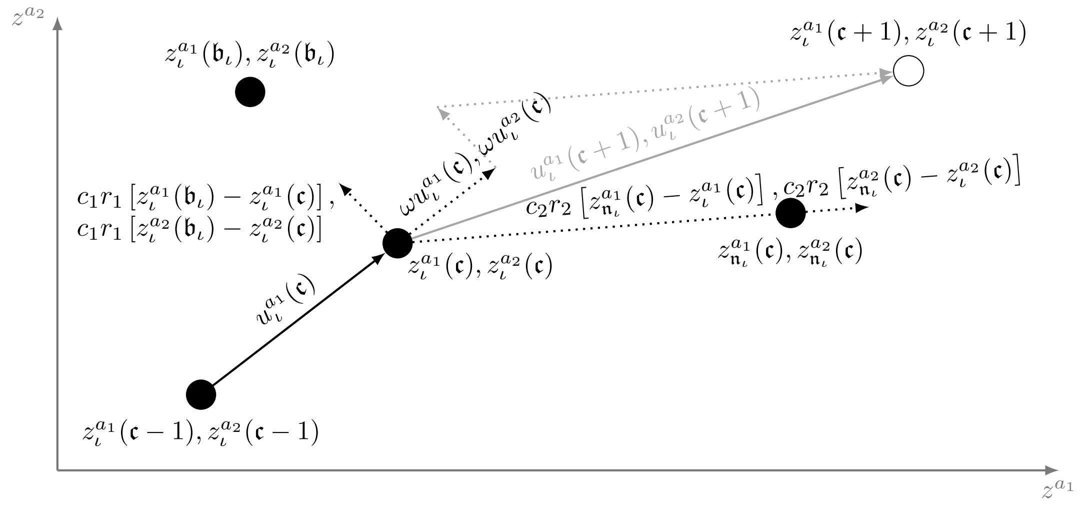
5.3.1 Base PSO

Particle Swarm Optimisation (PSO) was developed by Kennedy and Eberhart [95, 101], it is based on the principle of animal behaviours observed in flocking and schooling. The PSO is a population based algorithm and each population member is referred to as a particle. Let the population be $\mathbf{Z}$, with a particle ι having the parameter vector to be optimised that is the particles position $\mathbf{Z}_\iota = [z_\iota^1, z_\iota^2, \dots, z_\iota^A] = \mathbf{z}$, with A the number of dimensions of $\mathbf{z}$. Each particle has a velocity vector $\mathbf{U}_\iota = [u_\iota^1, u_\iota^2, \dots, u_\iota^A]$ that is updated through three terms. The cognitive term is the difference of the particles current position and its historical best position. The second term defines a social effect, and is the difference between the particles current position and the best position found by any particle within the neighbourhood. Finally an inertia weight term was included in [102], that allows for the particles to have momentum. Particles can therefore 'overshoot' the current optimum found and explore more of the solution space. The evolutionary process is defined as

$$u_\iota^a(\mathfrak{c}+1) = \omega u_\iota^a(\mathfrak{c}) + c_1 r_1 \left[z_\iota^a(\mathfrak{b}_\iota) - z_i^a(\mathfrak{c})\right] + c_2 r_2 \left[z_{\mathfrak{n}_\iota}^a(\mathfrak{c}) - z_\iota^a(\mathfrak{c})\right] \tag{5.9}$$

$$z_\iota^a(\mathfrak{c}) = z_\iota^a(\mathfrak{c}-1) + u_\iota^a(\mathfrak{c}) \tag{5.10}$$

with $\mathfrak{c}$ the iteration of the algorithm, ω the inertia weight, c_1 the cognitive acceleration coefficient and c_2 the social. r_1 and r_2 are uniform random numbers $\in [0,1]$. $\mathfrak{b}_\iota$ is the iteration of the algorithm for which particle ι found its best position and $\mathfrak{n}_\iota$ is the best particle in the neighbourhood of particle ι. The update process is illustrated in Figure 5.3. The parameters $c_1,\ c_2$ can be greater than 1 and therefore it is possible to have a velocity vector that goes beyond a particle as shown in the figure. This results in a simple algorithm that has a fast convergence rate. A selection of different topologies to find $\mathfrak{n}_\iota$ allows for different convergence properties and the choice of the parameters $\omega,\ c_1,\ c_2$ are the main differences between different PSO variants.



z^{a_2}
$z_\iota^{a_1}(\mathfrak{b}_\iota), z_\iota^{a_2}(\mathfrak{b}_\iota)$
$z_\iota^{a_1}(\mathfrak{c}+1), z_\iota^{a_2}(\mathfrak{c}+1)$
$\omega u_\iota^{a_1}(\mathfrak{c}), \omega u_\iota^{a_2}(\mathfrak{c})$
$u_\iota^{a_1}(\mathfrak{c}+1), u_\iota^{a_2}(\mathfrak{c}+1)$
$c_1 r_1 \left[z_\iota^{a_1}(\mathfrak{b}_\iota) - z_\iota^{a_1}(\mathfrak{c})\right],$
$c_1 r_1 \left[z_\iota^{a_2}(\mathfrak{b}_\iota) - z_\iota^{a_2}(\mathfrak{c})\right]$
$c_2 r_2 \left[z_{\mathfrak{n}_\iota}^{a_1}(\mathfrak{c}) - z_\iota^{a_1}(\mathfrak{c})\right], c_2 r_2 \left[z_{\mathfrak{n}_\iota}^{a_2}(\mathfrak{c}) - z_\iota^{a_2}(\mathfrak{c})\right]$
$z_\iota^{a_1}(\mathfrak{c}), z_\iota^{a_2}(\mathfrak{c})$
$z_{\mathfrak{n}_\iota}^{a_1}(\mathfrak{c}), z_{\mathfrak{n}_\iota}^{a_2}(\mathfrak{c})$
$u_\iota^{a_1}(\mathfrak{c})$
$z_\iota^{a_1}(\mathfrak{c}-1), z_\iota^{a_2}(\mathfrak{c}-1)$
z^{a_1}

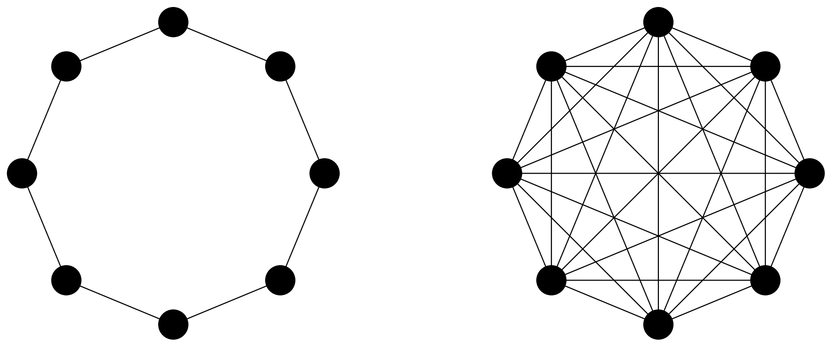
Figure 5.3: Update process for dimensions a_1, a_2 of particle ι of the PSO.

5.3.2 Local PSO

The local PSO (LPSO) algorithm uses the main PSO equations (5.9 – 5.10) with constant values for the parameters $c_1 = c_2 = \frac{1}{2} + \log 2$ and $\omega = 0.721$ as in [103]. A simple linear reduction on the inertia weight over the course of the algorithm has been used in the past. However, that means that in complex functions the algorithm may get stuck in a local minima and cannot improve in the latter stages as the inertia weight reduces forcing a more local search [72]. The topology used in the LPSO is a ring as shown in Figure 5.4. Each particle is connected to the two adjacent to it in memory. This topology has slows convergence as information about the best location takes time to propagate through the swarm but allows for a detailed search of the solution space.

5.3.3 Global PSO

The Global PSO (GPSO) uses the same fundamental expressions as the LPSO but with a different topology defined for the neighbourhood. All particles within the swarm are connected to every other particle. Therefore information of the best location at the current iteration $(\mathfrak{c})$ is known throughout the network as shown in Figure 5.4. This algorithm therefore tends to have a faster initial convergence than the LSPO, this does mean it can skip over some areas of the solution space. Other topologies than the two shown in Figure 5.4 can be used.



(a) Ring-lbest. (b) Fully connected-gbest.

Figure 5.4: Swarm topologies.

5.3.4 Adaptive PSOs

Summary

The parameters of the LPSO and GPSO are time-invariant, by changing these parameters over the search the convergence properties of the algorithm can be changed. Many Adaptive PSOs (APSO) have been developed [67, 68, 81, 83, 104–106] where the parameters of the search vary. Simple APSOs focus on the inertia weight and the acceleration coefficients. The inertia weight controls the domain of the search, a large inertia weight results in global search. The search then becomes more localised with the reduction of the inertia weight. In complex problems a static inertia weight can be preferable as it is unknown when the optimisation will find the correct region. A premature reduction in the inertia weight could lead the algorithm to get stuck at a local minima [72]. As many APSOs have been developed a few have been selected for trials and will be referred to as APSO-XX with XX the year of publication.

APSO-09

To adapt the inertia weight in an intelligent way [81] proposes an APSO with Evolutionary State Estimation (ESE). At each iteration the swarm is profiled and a state assigned based on the swarm composition. Then according to the sate of search the PSO parameters can be adapted appropriately. Four different states are defined:

Exploration: Particles scattered throughout search space.

Exploitation: Particles becoming more clustered, moving towards an optimum.

Convergence: Particles clustered around an optimum.

Jumping out: Best particle away from the swarming cluster, i.e. leaving local optimum.

As each of these states are ill defined fuzzy logic is applied in the ESE. State estimation is done by utilising a metric based on the mean distance between particles. The mean Euclidean distance for particle ι is measured as

$$\mathfrak{s}_\iota(\mathfrak{c}) = \frac{1}{N_{\text{pop}} - 1} \sum_{j=1, j\neq\iota}^{N\text{pop}} \sqrt{\sum_{a=1}^{A} \left[z_\iota^a(\mathfrak{c}) - z_j^a(\mathfrak{c})\right]^2} \tag{5.11}$$

with N_{pop} the number of particles in the swarm (population). The neighbourhood used in APSO-09 is global so $\mathfrak{n}_\iota$ gives the index of the particle with the best solution at the current iteration $\mathfrak{c}$. It is the same for all ι so can be denoted as $\mathfrak{n}$. Using this an 'evolutionary factor' $\mathfrak{f}$ is calculated as

$$\mathfrak{f} = \frac{\mathfrak{s}_\mathfrak{n} - \min\{\mathfrak{s}_0, \mathfrak{s}_1, ..., \mathfrak{s}_{N_{\text{pop}}}\}}{\max\{\mathfrak{s}_0, \mathfrak{s}_1, ..., \mathfrak{s}_{N_{\text{pop}}}\} - \min\{\mathfrak{s}_0, \mathfrak{s}_1, ..., \mathfrak{s}_{N_{\text{pop}}}\}} \in [0, 1]. \tag{5.12}$$

This factor is used to classify the search state to one of the four states $\mathcal{S}_1$, $\mathcal{S}_2$, $\mathcal{S}_3$, $\mathcal{S}_4$, which represent exploration, exploitation, convergence and jumping out respectively. These sets are defined with fuzzy boundaries as there is no definite cut off. The membership classifications (Figure 5.5) are:

$$\mathfrak{u}_{\mathcal{S}_1}(\mathfrak{f}) = \begin{cases} 0, & 0 \leq \mathfrak{f} \leq 0.4 \\ 5 \times \mathfrak{f} - 2, & 0.4 < \mathfrak{f} \leq 0.6 \\ 1, & 0.6 < \mathfrak{f} \leq 0.7 \\ -10 \times \mathfrak{f} + 8, & 0.7 < \mathfrak{f} \leq 0.8 \\ 0, & 0.8 < \mathfrak{f} \leq 1 \end{cases} \tag{5.13a}$$

$$\mathfrak{u}_{\mathcal{S}_2}(\mathfrak{f}) = \begin{cases} 0, & 0 \leq \mathfrak{f} \leq 0.2 \\ 10 \times \mathfrak{f} - 2, & 0.2 < \mathfrak{f} \leq 0.3 \\ 1, & 0.3 < \mathfrak{f} \leq 0.4 \\ -5 \times \mathfrak{f} + 3, & 0.4 < \mathfrak{f} \leq 0.6 \\ 0, & 0.6 < \mathfrak{f} \leq 1 \end{cases} \tag{5.13b}$$

$$\mathfrak{u}_{\mathcal{S}_3}(\mathfrak{f}) = \begin{cases} 1, & 0 \leq \mathfrak{f} \leq 0.1 \\ -5 \times \mathfrak{f} + 1.5, & 0.1 < \mathfrak{f} \leq 0.3 \\ 0, & 0.3 < \mathfrak{f} \leq 1 \end{cases} \tag{5.13c}$$

$$\mathfrak{u}_{\mathcal{S}_4}(\mathfrak{f}) = \begin{cases} 0, & 0 \leq \mathfrak{f} \leq 0.7 \\ 5 \times \mathfrak{f} - 3.5, & 0.7 < \mathfrak{f} \leq 0.9 \\ 1, & 0.9 < \mathfrak{f} \leq 1 \end{cases} \tag{5.13d}$$

This produces overlapping memberships, so a method for determining a unique state is required, this is done by using the singleton scheme with a state transition rule base. The state change process for the PSO is: $\mathcal{S}_1 \rightarrow \mathcal{S}_2 \rightarrow \mathcal{S}_3 \rightarrow \mathcal{S}_4 \rightarrow \mathcal{S}_1...$, so if two states have a degree of membership then this sequence is used to determine the state, then the singleton rule is applied if a classification has not been achieved; select the state with the higher membership value.

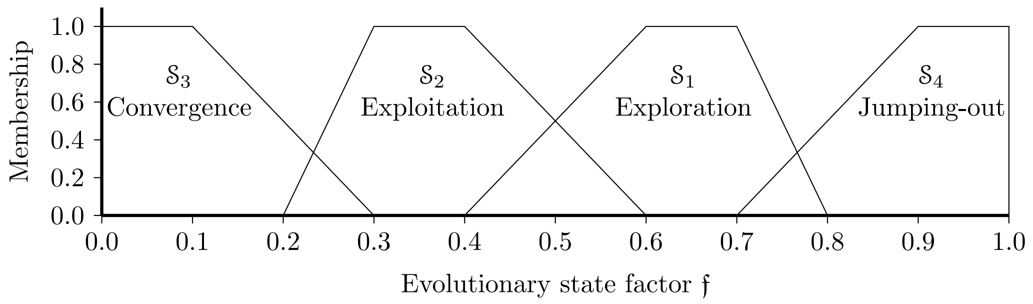


Figure 5.5: Membership functions of the evolutionary states.

With the states and factor $\mathfrak{f}$ defined it it possible to update the PSO parameters; *Inertia Weight:* can be tuned by the evolutionary factor $\mathfrak{f}$. $\mathfrak{f}$ is large when the search is scattered, or jumping-out hence the search should be a global one and the inertia weight should also be large. As the search converges $\mathfrak{f}$ reduces and a more local search should be undertaken so ω should also be reduced. Therefore ω is updated via a sigmoid mapping of the evolutionary factor $\mathfrak{f}$

$$\omega(\mathfrak{f}) = \frac{1}{1 + 1.5 \exp^{-2.6\mathfrak{f}}} \in [0.4, 0.9] \quad \forall \mathfrak{f} \in [0, 1]. \tag{5.14}$$

ω is initialised to 0.9, and will now become large in exploration states and smaller as convergence occurs.

Acceleration Coefficients: are varied in respect to the evolutionary state as described in Table 5.1. Each coefficient is scaled differently to exploit the cognitive knowledge without converging at an excessive rate to the social which could result in convergence at a local optimum in the early stages of the optimisation. In jumping out the social needs to have more influence over the cognitive so the change is reversed. Changes to the scaling parameters are restricted by

$$|c_\iota(\mathfrak{c} + 1) - c_\iota(\mathfrak{c})| \leq \delta_c, \quad \iota = 1, 2 \tag{5.15}$$

to avoid excessive interruptions, where δ_c, the acceleration rate, is a random value in the domain [0.05, 0.1]. Where Table 5.1 indicates slight variations $0.5\delta_c$ is used. c_1 and c_2 are restricted to the interval [0.5, 0.83], and the sum to [1.0, 1.33]. If the sum exceeds this bound, c_1 and c_2 are normalised by

$$c_j = \frac{c_j}{c_1 + c_2} 1.33 \quad j = 1, 2 \tag{5.16}$$

Table 5.1: Adaptive control of c_1 and c_2.

State	c_1	c_2
Exploration	Increase	Decrease
Exploitation	Increase slightly	Decrease slightly
Convergence	Increase slightly	Increase slightly
Jumping out	Decrease	Increase

Even with the adaptive parameters detailed above it is still possible that the algorithm will converge prematurely to a sub-prime local optimum [81]. Therefore a mutation operation is included to encourage jumping-out states. As the parameters will adapt themselves accordingly to the state it is only necessary to modify a single particle. The elitist learning strategy (ELS) randomly selects a single dimension a of the current best particle when the swarm is in a convergence state and applies a random Gaussian perturbation

$$z_{\mathfrak{n}}^{a'} = z_{\mathfrak{n}}^{a} + (z_{\max}^{a} - z_{\min}^{a}) \cdot G(\mu, \sigma^2) \tag{5.17}$$

in which $[x_{\min}^d, x_{\max}^d]$ is the domain of the dimension, $G(\mu, \sigma^2)$ is a random number from a Gaussian distribution with mean μ and standard deviation σ. The mean μ is set to zero and remains constant. To ensure a gradual reduction in the search over time σ is linearly decreased

$$\sigma(k) = \sigma_{\max} - (\sigma_{\max} - \sigma_{\min}) \frac{\mathfrak{c}}{\mathfrak{C}} \tag{5.18}$$

where $\sigma_{\max}$ and $\sigma_{\min}$ are the bounds to σ, $\mathfrak{c}$ is the iteration number and $\mathfrak{C}$ the maximum number of iterations. From testing convergence was found to be best with $\sigma_{\max} = 0.5$ and $\sigma_{\min} = 0.1$. The objective function for a clone of the best particle with this perturbed dimension is evaluated, if an improvement is found the best particle is replaced, if not the worst particle is replaced by the ELS particle.

APSO-12

A simple adaptive scheme is used in [68] to update the PSO parameters based on the iteration number of the search.

$$\omega(\mathfrak{c}) = (\omega_{\max} - \omega_{\min})\exp(-\beta \cdot \mathfrak{c}) + \omega_{\min} \tag{5.19}$$

$$c_1(\mathfrak{c}) = (C_2 - C_1)\frac{\mathfrak{c}}{\mathfrak{C}} + C_1 \tag{5.20}$$

$$c_2(\mathfrak{c}) = (C_1 - C_2)\frac{\mathfrak{c}}{\mathfrak{C}} + C_2 \tag{5.21}$$

where β is a constant to control the reduction speed of ω. $\omega_{\max}$ and $\omega_{\min}$ are the maximum and minimum limits on the inertia weight. This is reduced exponentially from $\omega_{\max}$ at $\mathfrak{c} = 0$ to $\omega_{\min}$ at $\mathfrak{c} = \mathfrak{C}$. C_1 and C_2 define the limits on the acceleration weights. C_1 is set greater than C_2 so that the search is predominantly cognitive in the early iterations as the search progress the cognitive influence is reduced and the social component of velocity becomes dominant in a linear fashion. The parameters are set as $\beta = 0.001$, $C_1 = 2.8$ and $C_2 = 1.2$. $C_1 + C_2 = 4$ to match the findings of [101] about the scaling of the acceleration coefficients.

APSO-14

The final APSO being considered uses constant acceleration weights $c_1 = c_2 = 2$, but uses a calculation based on the diversity of the swarm to update the inertia weight [67]. They critique (5.19) as having the same problem of a linearly decreasing inertia weight as discussed by [72]. The search quickly becomes local as the inertia weight decreases and the search will not be able to leave a local minima as the velocity will rapidly approach 0. The algorithm proposed by [67] again uses a global topology, and they define a fitness differential of the best result

$$\Delta J(\mathfrak{c}) = |J(\mathbf{Z}_{\mathfrak{n}}(\mathfrak{c})) - J(\mathbf{Z}_{\mathfrak{n}}(\mathfrak{c} - 1))| \tag{5.22}$$

which is used to calculate a convergence speed

$$\mathfrak{s}(\mathfrak{c}) = \frac{\Delta J(\mathfrak{c})}{\max\{\Delta J(0), \Delta J(1), ..., \Delta J(\mathfrak{c})\}}. \tag{5.23}$$

Population diversity is calculated by

$$\overline{J(\mathfrak{c})} = \frac{1}{N_{\text{pop}}} \sum_{\iota=1}^{N_{\text{pop}}} [J(\mathbf{Z}_{\iota}(\mathfrak{c}))] \tag{5.24}$$

$$J_{\text{norm}}(\mathfrak{c}) = \max\{|J(\mathbf{Z}_{\iota}(\mathfrak{c})) - \overline{J(\mathfrak{c})}|\} \tag{5.25}$$

$$\mathfrak{d}^2(\mathfrak{c}) = \frac{1}{N}\sum_{\iota=1}^{N_{\text{pop}}}\left(\frac{J(\mathbf{Z}_\iota(\mathfrak{c})) - \overline{J(\mathfrak{c})}}{J_{\text{norm}}(\mathfrak{c})}\right) \tag{5.26}$$

where $\overline{J(\mathfrak{c})}$ is the average fitness of the swarm, $J_{\text{norm}}(\mathfrak{c})$ a normalisation value and $\mathfrak{d}^2$ the population diversity at the iteration $\mathfrak{c}$. The inertia weight is then calculated from the convergence speed $\mathfrak{s}$ and diversity $\mathfrak{d}^2$ by

$$\omega(\mathfrak{c}) = \mathfrak{e}_1 \cdot \mathfrak{s}(\mathfrak{c}) + \mathfrak{e}_2 \cdot \mathfrak{d}^2(\mathfrak{c}) + \omega(0) \tag{5.27}$$

with $\mathfrak{e}_1$ and $\mathfrak{e}_2$ parameters. $\mathfrak{e}_1 = 0.3$, $\mathfrak{e}_2 = 0.3$ and $\omega(0) = 0.4$ are the values set for the parameters. Like APSO-09 this allows for the inertia weight to react to the composition of the swarm. During fast convergence or high diversity the inertia weight is large to conduct a global search. As the convergence speed decreases and the diversity reduces so does the inertia weight to allow for a more local search.

High Exploration PSO

Another variant of the PSO tries to increase the exploration of the search space by incorporating aspects from other evolutionary algorithms. The High Exploration PSO (HEPSO) [75] includes two additional update rules, one based on the Artificial Bee Colony (ABC) algorithm and the other a from a multi-crossover genetic algorithm. The algorithm uses a global search space and adapts the inertia weight as in APSO-09 (5.12,5.14). The acceleration coefficients are calculated according to the iteration of the search and follows the same method as used in APSO-12, eqns. (5.20, 5.21)

Two additional update operators are included. One is based on a multi-crossover genetic algorithm [107]. This uses three parent chromosomes $\mathbf{Z}_1(\mathfrak{c})$, $\mathbf{Z}_2(\mathfrak{c})$ and $\mathbf{Z}_3(\mathfrak{c})$, with $\mathbf{Z}_1(\mathfrak{c})$ being the fittest parent. Then a new chromosome is generated as

$$\mathbf{Z}_1(\mathfrak{c}+1) = \mathbf{Z}_1(\mathfrak{c}) + r(2\mathbf{Z}_1(\mathfrak{c}) - \mathbf{Z}_2(\mathfrak{c}) - \mathbf{Z}_3(\mathfrak{c})) \tag{5.28}$$

with r a random number $\in [0,1]$. This is applied to a random member ι by selecting the global best solution $\mathbf{Z}_\mathfrak{n}(\mathfrak{c})$ as the fittest chromosome, and the personal best solution $\mathbf{Z}_\iota(\mathfrak{b}_\iota)$ as the other required chromosome. This is used to set the velocity as

$$u_\iota^a(\mathfrak{c}+1) = r\left(\frac{c_2(\mathfrak{c})}{2} z_\mathfrak{n}^a(\mathfrak{c}) - z_\iota^a(\mathfrak{b}_\iota) - z_\iota^a(\mathfrak{c})\right) \tag{5.29}$$

The second update operator takes the form of a search according to the ABC algorithm [69] and the particle ι position is set as,

$$z_\iota^a(\mathfrak{c}+1) = z_\iota^a(\mathfrak{c}) + (2r-1)\left(z_\iota^a(\mathfrak{c}) - z_j^a(\mathfrak{c})\right) \tag{5.30}$$

with j a randomly selected member of the population. If the move does not give an improvement then the position is reverted to its previous value $z_\iota^a(\mathfrak{c}+1) = z_\iota^a(\mathfrak{c})$. For control of the algorithm at each iteration two random numbers are generated for every particle $r_{3,\iota}, r_{4,\iota} \in [0,1]$. If $r_{3,\iota} > \sigma$ and $r_{4,\iota} < p_{\text{ABC}}\frac{\mathfrak{c}}{\mathfrak{C}}$ then the particles position is updated by the ABC operator (5.30). With σ the standard deviation of the fitness values of the swarm, and p_{ABC} the probability of using the ABC operator. If a particle is not updated via ABC then a further random number $r_5 \in [0,1]$ is generated. If this random number is less than the probably of using crossover p_{GA} then the algorithm uses (5.29) to update the velocity. Otherwise the normal PSO operator is used to update the velocity (5.9). Both then use the velocity to then update the particle position (5.10).

Chaos Enhanced Accelerated PSO

The formulation of the PSO requires the algorithm to be performed in two steps, velocity calculation and position calculation. Accelerated PSOs were developed to reduce this to a single step to increase the algorithms speed. To simplify the cognitive effect (use of personal best position) of the algorithm is removed as it only serves to increase the diversity of swarm and is replaced with randomness. The removal of cognitive term can have negative effects on some problems, especially those that are highly non-linear and multi-modal [74]. The particles position is calculated as

$$z_\iota^a(\mathfrak{c}+1) = (1-\beta_{\text{CE}})z_\iota^a(\mathfrak{c}) + \beta_{\text{CE}} z_{\mathfrak{n}}^a(\mathfrak{c}) + \alpha_{\text{CE}} r \tag{5.31}$$

with a global topology again being used $z_{\mathfrak{n}}^a(\mathfrak{c})$ defines value of the dimension a of the best particle $\mathfrak{n}$ in the swarm for the iteration $\mathfrak{c}$. The algorithm has two parameters $\alpha_{\text{CE}}, \beta_{\text{CE}}$; α_{CE} provides the randomness and replaces the cognitive term. β_{CE} is the attraction parameter and replaces the acceleration coefficient and inertia weight. r is a random number $\in [0,1]$. The main control on the algorithms convergence is the parameter β_{CE}, $\beta_{\text{CE}} = 1$ will cause all particles to move to the current global best. A value close to zero will have a very slow convergence. As the solution space may vary and the value of beta is important it can be calculated due to from a chaotic map to form a Chaos Enhanced accelerated PSO (CEPSO) [74]. Various maps were trialled in [74] with a sinusoidal mapping providing the best algorithm performance for a range of benchmark functions and engineering problems.

$$\beta_{\text{CE}}(\mathfrak{c}+1) = \sin(\pi\beta_{\text{CE}}(\mathfrak{c})) \tag{5.32}$$

The parameter α_{CE} also influences the diversity of the swarm and is reduced over time to make the search more local and is calculated by

$$\alpha^{a}_{\text{CE}}(\mathfrak{c}) = \delta^{k}_{\text{CE}}\mathcal{L}_a \tag{5.33}$$

with $\delta_{\text{CE}} = 0.8$ a simulated annealing style parameter, and $\mathcal{L}_a$ the length of dimension a.

5.4 Cuckoo Search

5.4.1 Basic Cuckoo Search

Cuckoo Search (CS) was developed by Yang and Deb [108], it mimics the brood parasitism utilised by cuckoos. The algorithm maintains a population of solutions $\mathbf{Z}$ these are referred to as eggs. Eggs are grouped into nests, although most implementations have nests that only contain a single egg. The algorithm then uses three idealised rules to mimic cuckoo behaviour:

- Each cuckoo lays an egg and it is placed in a random nest;
- The best nests are maintained between iterations;
- The number of nests is fixed, but the host can detect cuckoo eggs and decide to create a new nest, abandoning the old one.

The key to the CS algorithm however is its method of generating solutions. Solutions are generated by random walks from drawn from a stable distribution with a heavy-tail (Lévy flight). This mimics the search and foraging pattern often seen in nature. The tails of the distribution follow a power law, meaning that the distribution has an infinite variance. Therefore the majority of walks will have a small step length but large steps are possible this allows for the search to move to entirely new region of the search space whilst also exploring local regions. An example of a series of Lévy flights is shown in Figure 5.6. The figure shows the characteristic nature of the flight, the search direction follows a normal distribution. The step size however does not, this gives a path that searches locally before a large step occurs that relocates the search to another region.

The basic CS algorithm is shown in algorithm 5.2. The three idealised rules are used, at each iteration the $\mathfrak{N}_{\text{cuckoo}}$ cuckoos create new solutions and replace a random egg if they are a fitter solution. The solutions are then ranked and the worst $\mathfrak{N}_{\text{abandon}}$ are replaced with new solutions created by Lévy flights, this takes care of (and is a simplified version) of the last two rules. For the Lévy flights a scaling factor $\mathfrak{a}$ is applied to the step length. This aids in keeping the flight within the domain.

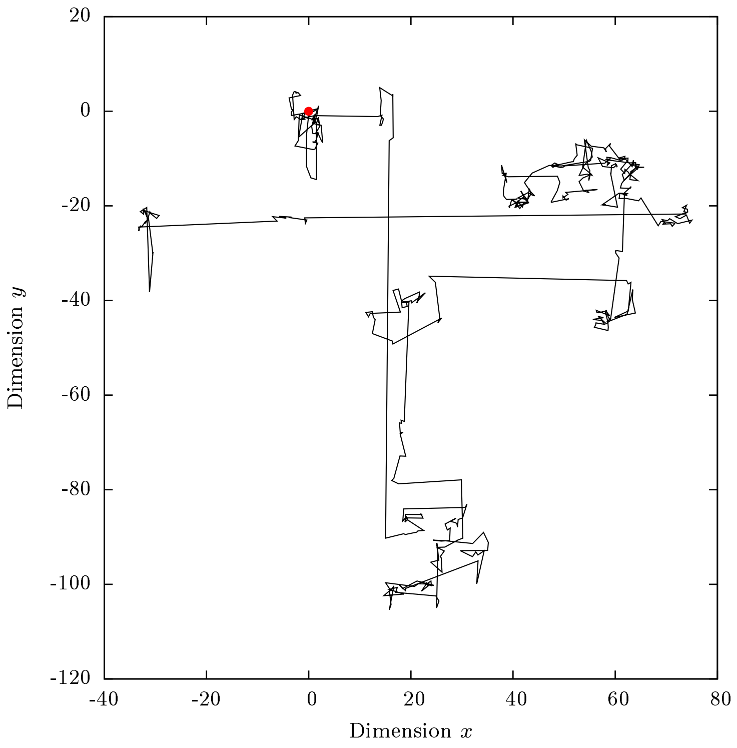


Figure 5.6: Lévy flight in 2 dimensions with 500 steps staring from [0,0] (red circle).

In [78] a recommendation of the weight for a dimension a is $\mathfrak{a}_a = \mathcal{L}_a/100$; with $\mathcal{L}_a$ is the length of the dimension a.

Algorithm 5.2 Cuckoo Search.

```
Input: Objective function: $J(\mathbf{z})$
Output: Optimal solution: $\mathbf{z}$
 1: Create an initial population $\mathbf{Z}$ of $N_{\text{pop}}$ nests
 2: for $\iota = 1, 2, \dots, N_{\text{pop}}$ do
 3:     Evaluate fitness $F_\iota = J(\mathbf{Z}_\iota)$
 4: end for
 5: while Convergence not achieved do
 6:     for $\iota = 1, 2, \dots, \mathfrak{N}_{\text{cuckoo}}$ do
 7:         Generate a cuckoo $\mathbf{Z}_{c,\iota}$ by a Lévy flight from random nest
 8:         Select a random nest $j$
 9:         if $F_\iota = J(\mathbf{Z}_{c,\iota}) < F_j$ then
10:             $\mathbf{Z}_j \leftarrow \mathbf{Z}_{c,\iota}$
11:             $F_j \leftarrow F_\iota$
12:         end if
13:     end for
14:     Rank solutions
15:     Abandon the $\mathfrak{N}_{\text{abandon}}$ worst nests and replace with new nests by Lévy flights
16: end while
```

5.4.2 Modified Cuckoo Search

The original CS algorithm relies on the random walk due to this the convergence of the algorithm is random and a fast convergence cannot be guaranteed. Walton et al. [79] propose a Modified Cuckoo Search (MCS) with two alterations to the original algorithm. The aim of the modifications to improve the convergence rate whilst maintaining the simplicity of the CS algorithm and its search properties. The MCS is detailed in algorithm 5.3.

Algorithm 5.3 Modified Cuckoo Search.

```
Input: Objective function: $J(\mathbf{z})$
Output: Optimal solution: $\mathbf{z}$
 1: Create an initial population $\mathbf{Z}$ of $N_{\text{pop}}$ nests
 2: for $\iota = 1, 2, \ldots, N_{\text{pop}}$ do
 3:     Evaluate fitness $F_\iota = J(\mathbf{Z}_\iota)$
 4: end for
 5: $\mathfrak{c} = 0$
 6: while Convergence not achieved do
 7:     $\mathfrak{c} = \mathfrak{c} + 1$
 8:     Rank solutions
 9:     $\mathfrak{a} = \mathfrak{a}_0/\sqrt{\mathfrak{c}}$
10:     Abandon the $\mathfrak{N}_{\text{abandon}}$ worst nests and replace with new nests by Lévy flights
11:     for the $\mathfrak{N}_{\text{top}}$ best nests do
12:         Current position $\mathbf{Z}_i$
13:         Randomly select another nest from the group of top nests $\mathbf{Z}_j$
14:         if $\mathbf{Z}_i = \mathbf{Z}_j$ then
15:             $\mathfrak{a} = \mathfrak{a}_0/\mathfrak{c}^2$
16:             Generate a cuckoo $\mathbf{Z}_c$ by a Lévy flight from $\mathbf{Z}_i$
17:             Select a random nest $k$
18:             if $J(\mathbf{Z}_c) < F_k$ then
19:                 $\mathbf{Z}_k \leftarrow \mathbf{Z}_c$
20:                 $F_k = J(\mathbf{Z}_c)$
21:             end if
22:         else
23:             $dz = |\mathbf{Z}_i - \mathbf{Z}_j|/\phi_g$
24:             Move distance $dz$ from the worst nest along the line connecting $\mathbf{Z}_i$,
    $\mathbf{Z}_j$ to find $\mathbf{Z}_c$
25:             Select a random nest $k$
26:             if $J(\mathbf{Z}_c) < F_k$ then
27:                 $\mathbf{Z}_k \leftarrow \mathbf{Z}_c$
28:                 $F_k = J(\mathbf{Z}_c)$
29:             end if
30:         end if
31:     end for
32: end while
```

The first modification is to the scaling parameter $\mathfrak{a}$. In CS $\mathfrak{a}$ is a constant, in MCS the parameter decreases every iteration, line 9 of the algorithm. This modification draws parallels with the scaling of the inertia weight in PSO. At the start of the search the aim is to identify the best regions to exploit so a large step size is required. As the search progresses and these regions are being exploited a smaller scaling weight is preferable to perform a local search. This parameter is scaled as;

$$\mathfrak{a}_a(\mathfrak{c}) = \frac{\mathfrak{a}_{a,0}}{\sqrt{\mathfrak{c}}} \tag{5.34}$$

with the parameter $\mathfrak{a}_{a,0}$ being the initial value for the scaling parameter for the dimension a.

The second modification in MCS introduces information exchange between good solutions (lines 11–31 of Algorithm 5.3) in an attempt to speed up the algorithms convergence. This is done in a similar manner to the crossover operator of the GA and differential evolution algorithms. A number of the best nests $\mathfrak{N}_{\text{best}}$ are selected for information exchange. For every solution in $\mathfrak{N}_{\text{best}}$ a second nest from the set is selected at random. A new solution is generated along the line connecting the solutions. The new solution is biased towards the fitter solution by the golden ratio and is generated as,

$$\mathbf{Z}_{c,\iota} = \mathbf{Z}_{g,2} + \frac{\mathbf{Z}_{g,2} - \mathbf{Z}_{g,1}}{\phi_g}. \tag{5.35}$$

With $\mathbf{Z}_{g,1}$, $\mathbf{Z}_{g,2}$ the selected solutions from the best nests, with $\mathbf{Z}_{g,1}$ being the fitter solution; ϕ_g is the golden ratio $(1+\sqrt{5})/2$. This replaces a randomly selected solution if it is a fitter solution.

5.5 Gradient Based Optimisation

5.5.1 Gradient Calculation

In past validation attempts the searches have been limited to solvers that use heuristics that sample the objective value at various points. This is due to the fact that as the underlying traffic model is a non-linear system an analytical expression for the gradient cannot be obtained. Through Automatic Differentiation (AD) it is possible to obtain a gradient of the traffic model for a specific set of parameters. AD calculates a derivative for a computer code by exploiting the fact that a complex program is composed of a series of lines with elementary arithmetic operations (addition, subtraction, division, multiplication, ...) and basic functions (sin, cos, exp and ln, ...). Each line of code can therefore be easily evaluated and derivative calculated. By recursive application of the chain rule the derivative of complex functions can be obtained. This means the calculation of the derivative is limited to machine precision and errors can be compounded.

For a gradient based optimisation algorithms the partial derivatives of the objective function for each parameter in $\mathbf{z}$ are required $\frac{\partial J(\mathbf{z})}{\partial \mathbf{z}}$. With the objective function J as defined in 4.2.4 and repeated here

$$\begin{aligned} J(\mathbf{z}) = &\sum_{k=1}^{K}\sum_{j=1}^{Y}\left(w_v E_{v,j}\left[\mathbf{x},\mathbf{y},\mathbf{z},k\right] + w_q E_{q,j}\left[\mathbf{x},\mathbf{y},\mathbf{z},k\right]\right) \\ &+ w_p \sum_{m=1}^{M}\sum_{\substack{\mu\in O_{a_m}\\ \mu\notin\Phi}} (\mathbf{z}_{\mathrm{L},m}-\mathbf{z}_{\mathrm{L},\mu})^{\top} w_{\mathrm{L}}(\mathbf{z}_{\mathrm{L},m}-\mathbf{z}_{\mathrm{L},\mu}). \end{aligned} \tag{5.36}$$

The model is a black box to which AD will be applied. Therefore AD will give the Jacobian $\mathbf{J}$ of the model outputs in relation to the model parameters

$$\mathbf{J} = \frac{\partial \mathbf{x}}{\partial \mathbf{z}} = \begin{bmatrix} \frac{\partial \mathbf{x}(1)}{\partial \mathbf{z}_1} & \frac{\partial \mathbf{x}(1)}{\partial \mathbf{z}_2} & \cdots & \frac{\partial \mathbf{x}(1)}{\partial \mathbf{z}_A} \\ \frac{\partial \mathbf{x}(2)}{\partial \mathbf{z}_1} & \frac{\partial \mathbf{x}(2)}{\partial \mathbf{z}_2} & \cdots & \frac{\partial \mathbf{x}(2)}{\partial \mathbf{z}_A} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{x}(K)}{\partial \mathbf{z}_1} & \frac{\partial \mathbf{x}(K)}{\partial \mathbf{z}_2} & \cdots & \frac{\partial \mathbf{x}(K)}{\partial \mathbf{z}_A} \end{bmatrix} \tag{5.37}$$

where A is the size of the vector $\mathbf{z}$. Recall that $\mathbf{x}(k)$ is a vector so $\mathbf{J}$ is composed of vectors each containing the partial derivatives for the density, flow and speed at every segment of each link at each time step k. This is a large matrix $(3KA \cdot \sum_{m=1}^{M} N_m)$ that is costly to compute. The matrix is very dense, however, it is sparse for the first few time steps. This is due to the fact that it takes a few iterations for states to propagate downstream. As the time steps increase and a state propagates further through the network the amount of zero entries will reduce as the interdependence increases. The zeros reduce as a link is fed traffic from an upstream cell and then its state depends on the properties of that link and all links which that portion of traffic has passed through. The exact nature and steps for the matrix to become nearly full will depend on the network configuration and time step T. Due to these properties it is not possible to use any techniques to reduce the memory requirements or reduce the computation time for any calculations involving $\mathbf{J}$.

The Jacobian $\mathbf{J}$ that is obtained from AD version of the model code however is not the required derivative for the optimisation problem. To obtain $\frac{\partial J(\mathbf{z})}{\partial \mathbf{z}}$ the chain rule can be applied to Equation (5.36). However the objective function consists of two terms the squared error term J_s and the penalty term J_p with

$$J_\mathrm{s}(\mathbf{z}) = \sum_{k=1}^{K}\sum_{j=1}^{Y}\left(w_v E_{v,j}\left[\mathbf{x},\mathbf{y},\mathbf{z},k\right] + w_q E_{q,j}\left[\mathbf{x},\mathbf{y},\mathbf{z},k\right]\right), \tag{5.38}$$

$$J_\mathrm{p}(\mathbf{z}) = w_p \sum_{m=1}^{M}\sum_{\substack{\mu\in O_{a_m}\\ \mu\notin\Phi}} (\mathbf{z}_{\mathrm{L},m}-\mathbf{z}_{\mathrm{L},\mu})^\top w_\mathrm{L}(\mathbf{z}_{\mathrm{L},m}-\mathbf{z}_{\mathrm{L},\mu}). \tag{5.39}$$

J_s requires the state vector from the model while J_p is a direct function of the parameters and a derivative can easily be calculated. Therefore the required derivative can be obtained by,

$$\frac{\partial J}{\partial \mathbf{z}} = \left(\frac{\partial J_\mathrm{s}}{\partial \mathbf{x}}^\top \frac{\partial \mathbf{x}}{\partial \mathbf{z}}\right)^\top + \frac{\partial J_\mathrm{p}}{\partial \mathbf{z}} \tag{5.40}$$

which can be expressed compactly as,

$$\frac{\partial J}{\partial \mathbf{z}} = \mathbf{J}\frac{\partial J_\mathrm{s}}{\partial \mathbf{x}}^\top + \frac{\partial J_\mathrm{p}}{\partial \mathbf{z}}. \tag{5.41}$$

The derivatives $\frac{\partial J_\mathrm{s}}{\partial \mathbf{x}}$ and $\frac{\partial J_\mathrm{p}}{\partial \mathbf{z}}$ are simple and are computed analytically to remove the overheads included by AD. Utilising the analytical derivative

$$\begin{aligned}\frac{\partial J_\mathrm{s}}{\partial \mathbf{x}_{m_j,i_j}(k)} = 2w_\rho\left[y_{\rho,j}(k)-\rho_{m_j,i_j}(k)\right] + 2w_v\left[y_{v,j}(k)-v_{m_j,i_j}(k)\right]\\ + 2w_q\left[y_{q,j}(k)-q_{m_j,i_j}(k)\right].\end{aligned} \tag{5.42}$$

For locations m, i which do not correspond to any measurement location m_j, i_j $j = 1, 2, \ldots, Y$ the derivative is zero. The other required derivative

$$\left\{\frac{\partial J_\mathrm{p}}{\partial \mathbf{z}_{\mathrm{L},m}}\right\} = 2w_p\left\{\sum_{\substack{\mu\in O_{n_{\mathrm{d},m}}\\ \mu\notin\Phi}} (w_\mathrm{L}\mathbf{1})^\top(\mathbf{z}_{\mathrm{L},m}-\mathbf{z}_{\mathrm{L},\mu}) - \sum_{\substack{\mu\in I_{n_{\mathrm{u},m}}\\ \mu\notin\Theta}} (w_\mathrm{L}\mathbf{1})^\top(\mathbf{z}_{\mathrm{L},m}-\mathbf{z}_{\mathrm{L},\mu})\right\} \tag{5.43}$$

with $n_{\mathrm{d},m}$ the downstream and $n_{\mathrm{u},m}$ the upstream node of link m; Θ the set of origins and Φ the set of destinations of the model. $\mathbf{z}_{L,m}$ is the parameter vector for link m. $\mathbf{1}$ is a vector of 1's so that the term $w_\mathrm{L}\mathbf{1}$ gives the leading diagonal of the link scaling factors w_L as a vector. For all other parameters of $\mathbf{z}$, $\frac{\partial J_\mathrm{p}}{\partial \mathbf{z}_a} = 0$ as the penalty function does not influence these parameters.

The gradient estimation by use of AD does not however allow for AAFD to function in the same manner. There is no gradient given for the fundamental diagram length γ. This is partly due to the process working with the output from a trans-formed set of variables and not the original therefore the parameter γ does not appear in $\mathbf{z}$. This works for optimisers that do not require knowledge of the solution space. However, even if the implementation was adjusted a gradient for the γ parameter would not be possible whilst keeping the model as a black box; it is a constraint upon the system and not a model parameter. Therefore, to keep the applicability of the AAFD scheme the values of γ will be set as 1 and removed from the optimisation. This means that a single unique FD will be used for every link. The penalty term however remains so the optimisation will still aim to reduce the variance between the FD parameters for the adjoining links.

5.5.2 Model Jacobian

The model Jacobian has interesting properties, for the first few states it is sparsely populated. However, as time progresses and waves propagate through the network the spatial influence of parameters increases and the matrix becomes more dense. Table 5.2 shows three regions of a Jacobian for the METANET model of Heathrow (see 6.4.2). The first block is for $k = 1$. It can be seen that the matrix is sparse. Non-zero entries exist at most locations for the network-wide parameters τ, κ and ν. As states depend on initial conditions, rather than a previous state, some values are zero. The limits $v_{\min}$ and $\rho_{\max}$ have zero entries. This is expected as these parameters will only affect the output in extreme scenarios and exist to prevent unrealistic traffic states. δ has zero effect at this early stage as it depends on inflow from origins that are low at early hours in the morning; it is applied downstream of merges only. The final network-wide parameter ϕ has an influence on the destinations and the upstream link as this is where the adjustment is made to the speed equation. At this initial phase the region of the matrix relating to the link parameters is sparsely populated. Note that the first line for each link corresponds to the node and hence the upstream link(s) affects the output. Furthermore non-zero entries exist for the link parameters at destinations directly downstream.

Forwards in time at $k = 5$ the propagation of states can be observed. The influence of the network parameter ϕ has propagated to links further away from the junctions. δ still provides no input but this is due to the model inputs. The parameter follows a similar pattern as ϕ but starting from locations downstream of a merge. The parameters for each link now also affect the output of other links. The table shows not only a propagation in the downstream direction but also upstream, however, the upstream propagation is slower. It does not take long for the matrix to become dense. By $k = 30$ the matrix is nearly entirely composed of non-zero entries. As we get away further away from a link we can see the magnitude of the partial derivative's decreases. This shows that the influence of the parameter reduces as

Table 5.2: Extract of the Jacobian produced for the Heathrow METANET model.

k	Link	τ	κ	ν	$v_{\min}$	$\rho_{\max}$	δ	ϕ	α_1	$v_{f,1}$	$\rho_{cr,1}$	α_2	$v_{f,2}$	$\rho_{cr,2}$	α_3	$v_{f,3}$	$\rho_{cr,3}$	α_4	$v_{f,4}$	$\rho_{cr,4}$	α_5	$v_{f,5}$	$\rho_{cr,5}$
1	O1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	O2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	O3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	L1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		6.95E+02	9.05E-03	5.48E-03	0	0	0	0	2.60E+00	3.71E-01	5.67E-02	0	0	0	0	0	0	0	0	0	0	0	0
		4.68E+02	4.76E-03	2.90E-03	0	0	0	0	2.67E+00	3.70E-01	5.86E-02	0	0	0	0	0	0	0	0	0	0	0	0
		3.54E+02	2.70E-02	1.65E-02	0	0	0	0	2.71E+00	3.70E-01	5.96E-02	0	0	0	0	0	0	0	0	0	0	0	0
		4.72E+02	1.51E-02	9.39E-03	0	0	0	0	2.90E+00	3.69E-01	6.57E-02	0	0	0	0	0	0	0	0	0	0	0	0
		8.52E+02	1.10E-01	6.74E-02	0	0	0	3.01E+00	2.79E+00	3.70E-01	2.68E-01	0	0	0	0	0	0	0	0	0	0	0	0
	L2	8.52E+02	1.10E-01	6.74E-02	0	0	0	3.01E+00	2.79E+00	3.70E-01	2.68E-01	0	0	0	0	0	0	0	0	0	0	0	0
		1.58E+02	9.27E-03	4.86E-03	0	0	0	0	0	0	0	1.15E+00	3.77E-01	3.77E-01	1.98E-02	0	0	0	0	0	0	0	0
		6.73E+01	0	0	0	0	0	0	0	0	0	1.21E+00	3.77E-01	3.77E-01	2.10E-02	0	0	0	0	0	0	0	0
		6.73E+01	0	0	0	0	0	0	0	0	0	1.21E+00	3.77E-01	3.77E-01	2.10E-02	0	0	0	0	0	0	0	0
		7.30E+02	2.38E-01	1.25E-01	0	0	0	0	0	0	0	1.21E+00	3.77E-01	3.77E-01	2.10E-02	0	0	0	0	0	0	0	0
	L3	7.30E+02	2.38E-01	1.25E-01	0	0	0	0	0	0	0	1.21E+00	3.77E-01	3.77E-01	2.10E-02	0	0	0	0	0	0	0	0
		4.19E+02	1.07E-02	6.61E-03	0	0	0	0	0	0	0	0	0	0	0	2.78E+00	3.70E-01	6.19E-02	0	0	0	0	0
		4.46E+02	0	0	0	0	0	0	0	0	0	0	0	0	0	2.86E+00	3.69E-01	6.44E-02	0	0	0	0	0
		2.71E+02	4.45E-02	2.76E-02	0	0	0	3.05E+00	0	0	0	0	0	0	0	2.86E+00	3.69E-01	2.73E-01	0	0	0	0	0
	L4	2.71E+02	4.45E-02	2.76E-02	0	0	0	3.05E+00	0	0	0	0	0	0	0	2.86E+00	3.69E-01	2.73E-01	0	0	0	0	0
		6.23E+02	1.69E-02	1.04E-02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.73E+00	3.70E-01	6.04E-02	0	0
		5.33E+02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.60E+00	3.71E-01	5.67E-02	0	0
		3.96E+02	3.56E-02	2.15E-02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.60E+00	3.71E-01	5.67E-02	0	0
	L5	3.96E+02	3.56E-02	2.15E-02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.60E+00	3.71E-01	5.67E-02	0	0
		2.93E+03	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8.43E+00	3.30E-01
		2.93E+03	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8.43E+00	3.30E-01
		3.78E+03	2.16E-01	1.34E-01	0	0	0	2.57E+00	0	0	0	0	0	0	0	0	0	0	0	0	0	8.43E+00	3.30E-01
	D1	3.78E+03	2.16E-01	1.34E-01	0	0	0	2.57E+00	0	0	0	0	0	0	0	0	0	0	0	0	0	8.43E+00	3.30E-01
	D2	8.52E+02	1.10E-01	6.74E-02	0	0	0	3.01E+00	2.79E+00	3.70E-01	2.68E-01	0	0	0	0	0	0	0	0	0	0	0	0
	D3	2.71E+02	4.45E-02	2.76E-02	0	0	0	3.05E+00	0	0	0	0	0	0	0	2.86E+00	3.69E-01	2.73E-01	0	0	0	0	0
5	O1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	O2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	O3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	L1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		3.70E+02	1.53E-02	9.58E-03	0	0	0	0	2.88E+00	3.91E-01	6.37E-02	0	0	0	0	0	0	0	0	0	0	0	0
		5.93E+02	1.19E-02	6.94E-03	0	0	0	0	4.49E+00	6.33E-01	9.81E-02	0	0	0	0	0	0	0	0	0	0	0	0
		1.23E+02	1.13E-01	6.99E-02	0	0	0	1.87E-02	4.78E+00	6.78E-01	1.05E-01	0	0	0	0	0	0	0	0	0	0	0	0
		1.50E+02	1.16E-01	7.11E-02	0	0	0	2.55E-01	6.29E+00	7.47E-01	1.62E-01	2.28E-03	6.20E-04	6.20E-04	4.13E-05	0	0	0	0	0	0	0	0
		1.20E+03	1.14E-01	7.10E-02	0	0	0	2.91E+00	6.14E+00	8.28E-01	3.35E-01	5.58E-02	1.15E-02	1.15E-02	1.09E-03	0	0	0	0	0	0	0	0
	L2	1.20E+03	1.14E-01	7.10E-02	0	0	0	2.91E+00	6.14E+00	8.28E-01	3.35E-01	5.58E-02	1.15E-02	1.15E-02	1.09E-03	0	0	0	0	0	0	0	0
		8.31E+02	1.65E-02	9.91E-03	0	0	0	1.42E+00	2.80E+00	4.17E-01	1.57E-01	2.24E+00	4.30E-01	4.30E-01	4.44E-02	0	0	0	0	0	0	0	0
		6.83E+02	7.63E-02	4.43E-02	0	0	0	7.61E-01	1.66E+00	2.22E-01	8.89E-02	3.22E+00	6.41E-01	6.41E-01	6.34E-02	0	0	0	0	0	0	0	0
		6.14E+02	8.05E-02	4.66E-02	0	0	0	5.35E-01	9.88E-01	1.28E-01	5.87E-02	3.45E+00	7.78E-01	7.78E-01	6.55E-02	9.82E-03	1.21E-03	2.25E-04	0	0	0	0	0
		1.33E+02	2.86E-01	1.52E-01	0	0	0	3.54E-01	4.34E-01	5.64E-02	3.39E-02	3.45E+00	9.18E-01	9.18E-01	6.26E-02	3.60E-01	4.19E-02	8.41E-03	0	0	0	0	0
	L3	1.33E+02	2.86E-01	1.52E-01	0	0	0	3.54E-01	4.34E-01	5.64E-02	3.39E-02	3.45E+00	9.18E-01	9.18E-01	6.26E-02	3.60E-01	4.19E-02	8.41E-03	0	0	0	0	0
		2.00E+01	1.74E-01	9.08E-02	0	0	0	1.26E-01	9.87E-02	1.31E-02	9.48E-03	1.78E+00	5.22E-01	5.22E-01	3.14E-02	3.65E+00	4.09E-01	8.77E-02	0	0	0	0	0
		2.38E+02	8.53E-02	4.40E-02	0	0	0	3.17E-01	0	0	0	9.13E-01	2.81E-01	2.81E-01	1.60E-02	5.45E+00	6.29E-01	1.49E-01	8.37E-03	1.19E-03	1.82E-04	0	0
		9.51E+02	2.88E-02	2.42E-02	0	0	0	3.23E+00	0	0	0	4.43E-01	1.36E-01	1.36E-01	7.75E-03	5.61E+00	6.49E-01	3.52E-01	1.04E-01	1.63E-02	2.19E-03	0	0
	L4	9.51E+02	2.88E-02	2.42E-02	0	0	0	3.23E+00	0	0	0	4.43E-01	1.36E-01	1.36E-01	7.75E-03	5.61E+00	6.49E-01	3.52E-01	1.04E-01	1.63E-02	2.19E-03	0	0
		4.31E+02	3.28E-03	8.42E-03	0	0	0	1.87E+00	0	0	0	3.11E-01	9.72E-02	9.72E-02	5.42E-03	2.97E+00	3.22E-01	1.99E-01	2.66E+00	3.88E-01	5.75E-02	0	0
		5.19E+02	7.42E-02	4.81E-02	0	0	0	9.84E-01	0	0	0	1.05E-01	3.29E-02	3.29E-02	1.83E-03	2.02E+00	2.25E-01	1.15E-01	4.25E+00	6.18E-01	9.18E-02	6.12E-02	2.39E-03
		2.23E+02	1.69E-01	9.86E-02	0	0	0	6.63E-01	0	0	0	0	0	0	0	1.49E+00	1.90E-01	7.91E-02	4.58E+00	7.42E-01	9.55E-02	1.69E+00	6.19E-02
	L5	2.23E+02	1.69E-01	9.86E-02	0	0	0	6.63E-01	0	0	0	0	0	0	0	1.49E+00	1.90E-01	7.91E-02	4.58E+00	7.42E-01	9.55E-02	1.69E+00	6.19E-02
		2.76E+03	1.35E-01	8.11E-02	0	0	0	5.16E-01	0	0	0	0	0	0	0	6.26E-01	8.08E-02	4.83E-02	2.28E+00	3.62E-01	4.80E-02	1.46E+01	5.03E-01
		2.87E+03	1.28E-01	8.04E-02	0	0	0	3.69E-01	0	0	0	0	0	0	0	8.99E-02	1.16E-02	8.56E-03	1.42E+00	2.00E-01	3.11E-02	2.09E+01	7.09E-01
		2.09E+03	4.42E-01	2.96E-01	0	0	0	1.15E+00	0	0	0	0	0	0	0	0	0	0	5.22E-01	7.37E-02	1.14E-02	2.34E+01	8.01E-01
	D1	2.09E+03	4.42E-01	2.96E-01	0	0	0	1.15E+00	0	0	0	0	0	0	0	0	0	0	5.22E-01	7.37E-02	1.14E-02	2.34E+01	8.01E-01
	D2	1.20E+03	1.14E-01	7.10E-02	0	0	0	2.91E+00	6.14E+00	8.28E-01	3.35E-01	5.58E-02	1.15E-02	1.15E-02	1.09E-03	0	0	0	0	0	0	0	0
	D3	9.51E+02	2.88E-02	2.42E-02	0	0	0	3.23E+00	0	0	0	4.43E-01	1.36E-01	1.36E-01	7.75E-03	5.61E+00	6.49E-01	3.52E-01	1.04E-01	1.63E-02	2.19E-03	0	0
30	O1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	O2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	O3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	L1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		2.85E+02	9.73E-03	6.20E-03	0	0	0	3.99E-04	3.66E+00	4.11E-01	8.67E-02	8.13E-06	1.57E-06	1.57E-06	1.61E-07	4.09E-10	4.48E-11	9.91E-12	4.38E-14	6.50E-15	9.41E-16	1.26E-19	4.94E-21
		3.45E+02	1.19E-02	7.60E-03	0	0	0	4.35E-03	5.96E+00	6.52E-01	1.43E-01	8.95E-05	1.69E-05	1.69E-05	1.78E-06	5.55E-09	5.85E-10	1.37E-10	9.18E-13	1.37E-13	1.97E-14	6.33E-17	2.47E-18
		3.20E+02	1.06E-02	6.83E-03	0	0	8.87E-12	4.43E-02	7.40E+00	7.96E-01	1.81E-01	9.20E-04	1.67E-04	1.67E-04	1.86E-05	6.52E-08	6.58E-09	1.64E-09	1.43E-11	2.13E-12	3.08E-13	5.31E-15	2.04E-16
		2.70E+02	2.66E-02	1.73E-02	0	0	2.12E-09	4.36E-01	8.26E+00	8.78E-01	2.29E-01	9.14E-03	1.57E-03	1.57E-03	1.88E-04	7.10E-07	6.86E-08	1.83E-08	1.90E-10	2.78E-11	4.11E-12	2.04E-13	7.65E-15
		9.18E+02	5.99E-02	4.00E-02	0	0	8.89E-08	3.59E+00	7.86E+00	8.01E-01	4.38E-01	7.69E-02	1.25E-02	1.25E-02	1.61E-03	6.43E-06	5.96E-07	1.68E-07	2.04E-09	2.89E-10	4.45E-11	4.56E-12	1.66E-13
	L2	9.18E+02	5.99E-02	4.00E-02	0	0	8.89E-08	3.59E+00	7.86E+00	8.01E-01	4.38E-01	7.69E-02	1.25E-02	1.25E-02	1.61E-03	6.43E-06	5.96E-07	1.68E-07	2.04E-09	2.89E-10	4.45E-11	4.56E-12	1.66E-13
		9.33E+02	6.00E-02	3.85E-02	0	0	9.89E-06	1.90E+00	4.11E+00	4.22E-01	2.30E-01	2.95E+00	4.50E-01	4.50E-01	6.27E-02	2.66E-04	2.39E-05	7.08E-06	1.02E-07	1.38E-08	2.28E-09	4.03E-10	1.42E-11
		7.36E+02	4.71E-02	2.96E-02	0	0	2.96E-04	1.02E+00	2.18E+00	2.28E-01	1.23E-01	4.30E+00	6.90E-01	6.90E-01	9.02E-02	3.99E-03	3.50E-04	1.07E-04	1.95E-06	2.45E-07	4.44E-08	1.13E-08	3.86E-10
		4.27E+02	1.24E-02	5.82E-03	0	0	8.41E-03	5.57E-01	1.16E+00	1.25E-01	6.60E-02	4.99E+00	8.35E-01	8.35E-01	1.03E-01	5.93E-02	5.08E-03	1.61E-03	3.75E-05	4.39E-06	8.75E-07	2.87E-07	9.53E-09
		3.58E+02	3.49E-01	1.99E-01	0	0	1.84E-01	3.38E-01	7.24E-01	7.50E-02	4.00E-02	5.56E+00	9.34E-01	9.34E-01	1.15E-01	7.96E-01	6.57E-02	2.20E-02	6.20E-04	6.83E-05	1.48E-05	5.99E-06	1.92E-07
	L3	3.58E+02	3.49E-01	1.99E-01	0	0	1.84E-01	3.38E-01	7.24E-01	7.50E-02	4.00E-02	5.56E+00	9.34E-01	9.34E-01	1.15E-01	7.96E-01	6.57E-02	2.20E-02	6.20E-04	6.83E-05	1.48E-05	5.99E-06	1.92E-07
		1.04E+02	2.59E-01	1.03E-01	0	0	1.73E+00	2.60E-01	3.97E-01	4.15E-02	2.24E-02	3.16E+00	5.43E-01	5.43E-01	6.49E-02	5.62E+00	4.44E-01	1.58E-01	4.79E-03	5.07E-04	1.16E-04	5.57E-05	1.74E-06
		2.83E+02	1.50E-01	6.24E-02	0	0	8.90E-01	6.98E-01	2.06E-01	2.09E-02	1.24E-02	1.85E+00	3.24E-01	3.24E-01	3.77E-02	8.33E+00	6.72E-01	2.65E-01	4.06E-02	4.19E-03	9.91E-04	5.64E-04	1.71E-05
		1.28E+03	5.99E-02	6.39E-02	0	0	3.81E-01	4.31E+00	4.71E-02	4.68E-03	4.51E-03	9.50E-01	1.71E-01	1.71E-01	1.92E-02	8.73E+00	6.97E-01	5.28E-01	2.98E-01	3.01E-02	7.32E-03	4.88E-03	1.43E-04
	L4	1.28E+03	5.99E-02	6.39E-02	0	0	3.81E-01	4.31E+00	4.71E-02	4.68E-03	4.51E-03	9.50E-01	1.71E-01	1.71E-01	1.92E-02	8.73E+00	6.97E-01	5.28E-01	2.98E-01	3.01E-02	7.32E-03	4.88E-03	1.43E-04
		1.15E+03	4.63E-02	4.33E-02	0	0	1.88E-01	2.48E+00	8.42E-03	1.92E-03	2.24E-03	5.47E-01	9.89E-02	9.89E-02	1.11E-02	4.99E+00	4.05E-01	3.02E-01	4.30E+00	4.25E-01	1.07E-01	8.14E-02	2.32E-03
		6.81E+02	2.62E-02	1.03E-02	0	0	1.17E-01	1.46E+00	3.46E-02	7.94E-03	1.00E-04	3.06E-01	5.58E-02	5.58E-02	6.16E-03	2.92E+00	2.40E-01	1.77E-01	6.63E+00	6.70E-01	1.63E-01	8.37E-01	2.32E-02
		1.03E+03	3.33E-01	2.16E-01	0	0	2.24E-01	9.29E-01	9.63E-02	1.56E-02	2.91E-03	1.13E-01	2.29E-02	2.29E-02	2.23E-03	1.70E+00	1.44E-01	1.05E-01	8.22E+00	8.26E-01	2.03E-01	7.51E+00	2.03E-01
	L5	1.03E+03	3.33E-01	2.16E-01	0	0	2.24E-01	9.29E-01	9.63E-02	1.56E-02	2.91E-03	1.13E-01	2.29E-02	2.29E-02	2.23E-03	1.70E+00	1.44E-01	1.05E-01	8.22E+00	8.26E-01	2.03E-01	7.51E+00	2.03E-01
		3.92E+03	3.59E-01	2.38E-01	0	0	6.52E-01	6.95E-01	1.80E-01	2.54E-02	6.85E-03	7.89E-02	1.04E-02	1.04E-02	1.68E-03	7.03E-01	6.48E-02	4.67E-02	4.36E+00	4.56E-01	1.06E-01	3.12E+01	8.19E-01
		2.65E+03	1.50E-01	9.97E-02	0	0	2.56E-01	9.82E-01	1.45E-01	1.97E-02	5.95E-03	1.27E-01	2.04E-02	2.04E-02	2.63E-03	1.26E-01	1.51E-02	1.14E-02	1.66E+00	1.82E-01	3.97E-02	3.62E+01	9.01E-01
		4.55E+02	7.23E-02	7.48E-02	0	0	8.69E-02	2.17E+00	1.95E-02	2.44E-03	9.29E-04	1.97E-02	3.12E-03	3.12E-03	4.08E-04	8.23E-02	7.42E-03	5.73E-03	5.73E-01	6.10E-02	1.39E-02	2.63E+01	6.22E-01
	D1	4.55E+02	7.23E-02	7.48E-02	0	0	8.69E-02	2.17E+00	1.95E-02	2.44E-03	9.29E-04	1.97E-02	3.12E-03	3.12E-03	4.08E-04	8.23E-02	7.42E-03	5.73E-03	5.73E-01	6.10E-02	1.39E-02	2.63E+01	6.22E-01
	D2	9.18E+02	5.99E-02	4.00E-02	0	0	8.89E-08	3.59E+00	7.86E+00	8.01E-01	4.38E-01	7.69E-02	1.25E-02	1.25E-02	1.61E-03	6.43E-06	5.96E-07	1.68E-07	2.04E-09	2.89E-10	4.45E-11	4.56E-12	1.66E-13
	D3	1.28E+03	5.99E-02	6.39E-02	0	0	3.81E-01	4.31E+00	4.71E-02	4.68E-03	4.51E-03	9.50E-01	1.71E-01	1.71E-01	1.92E-02	8.73E+00	6.97E-01	5.28E-01	2.98E-01	3.01E-02	7.32E-03	4.88E-03	1.43E-04

the distance from its application increases. For all of the entries shown it can be seen that the model parameters have no influence on the origins. This is due to the set-up invoked where by the origins are set so that the input data are directly applied and no queues are formed. With a more complex network it will take longer for the matrix to fill up and some non-zero entries will exist for a long time if no direct route exists between locations. However, due to the backward and forward propagation of dependencies these values will become non-zero given enough time.

5.5.3 General line-search algorithm

Line-search based gradient optimisers involve the calculation of a search direction. The solution is moved along the calculated trajectory according to a step that is also calculated by the algorithm. This class of gradient based optimisers includes the steepest-descent, Newton's and Quasi-Newton methods. It takes the general form of Algorithm 5.4. With the algorithm supplied an initial starting point $\mathbf{z}_0$, a tolerance ϵ for the convergence criterion and a large number LARGE to use in initialisation.

Algorithm 5.4 Line-Search algorithm.

```
Input: $\mathbf{z}_0$, $\epsilon$, LARGE
Output: $\mathbf{z}^*$
 1: $\mathfrak{c} = 0,\ \tilde{J} = \text{LARGE}$
 2: repeat
 3:     $\mathbf{s}(\mathfrak{c}) \leftarrow$ determine_search_direction
 4:     $\alpha_s(\mathfrak{c}) \leftarrow$ determine_step_size
 5:     $\mathbf{z}(\mathfrak{c}+1) = \mathbf{z}(\mathfrak{c}) + \alpha_s(\mathfrak{c})\mathbf{s}(\mathfrak{c})$
 6:     if $J(\mathbf{z}(\mathfrak{c}+1)) < \tilde{J}$ then
 7:         $\tilde{J} = J(\mathbf{z}(\mathfrak{c}+1))$
 8:         $\tilde{\mathbf{z}} = \mathbf{z}(\mathfrak{c}+1)$
 9:     end if
10:     $\mathfrak{c} = \mathfrak{c} + 1$
11: until converged
12: $\mathbf{z}^* = \tilde{\mathbf{z}}$
```

The sub-algorithms determine_search_direction and determine_step_size define the specific gradient optimiser. Newton's method requires the Hessian matrix and majority of line-search algorithms express the step-size calculation as a function of the gradient. This could cause issues for the model validation problem. The computation of higher-order derivative matrices will significantly increase the computational requirements. Also the problem is non-linear and multi-modal making the choice of step-size a critical part of the algorithm. A step size that is too large could skip large regions of the domain, missing important areas. On the other hand a step size that is too small will limit the algorithm to it's current basis of attraction and only find the local minimum. The step-size also needs to respond correctly to the state of the search.

5.5.4 Resilient back-propagation

The RPROP algorithm is typically applied to the training of neural networks [87]. In [87] the algorithm was used to minimise a series of non-smooth test functions and non-linear functions in [88]. The RPROP solver also bases its update rule only on the sign of the gradient so any errors in the estimated gradient supplied by AD will be minimised as long as the sign is correct (away from zero). These properties make RPROP an ideal optimiser to be evaluated for the non-linear problem of model validation. It's step size calculation is also not connected to the gradient value so the step-size won't be affected by any small steep peaks in the solution space, or discontinuities. RPROP is a modification of the line-search algorithm. RPROP combines steps 3 and 4 of Algorithm 5.4 into a single stage with $\alpha_{\mathrm{s}}(\mathfrak{c})\mathbf{s}_a(\mathfrak{c})$ calculated by,

$$\alpha_{\mathrm{s}}(\mathfrak{c})\mathbf{s}_a(\mathfrak{c}) = \begin{cases} -\Delta_a^{(\mathfrak{c})} & \text{if } \frac{\partial J(\mathfrak{c})}{\partial z_a} > 0 \\ +\Delta_a^{(\mathfrak{c})} & \text{if } \frac{\partial J(\mathfrak{c})}{\partial z_a} < 0 \\ 0 & \text{if } \frac{\partial J(\mathfrak{c})}{\partial z_a} = 0 \end{cases} \tag{5.44}$$

with

$$\Delta_a^{(\mathfrak{c})} = \begin{cases} \eta^+\Delta_a^{(\mathfrak{c}-1)} & \text{if } \frac{\partial J(\mathfrak{c}-1)}{\partial z_a}\cdot\frac{\partial J(\mathfrak{c})}{\partial z_a} > 0 \\ \eta^-\Delta_a^{(\mathfrak{c}-1)} & \text{if } \frac{\partial J(\mathfrak{c}-1)}{\partial z_a}\cdot\frac{\partial J(\mathfrak{c})}{\partial z_a} < 0 \\ \Delta_a^{(\mathfrak{c}-1)} & \text{else.} \end{cases} \tag{5.45}$$

η^+ and η^- are constants with the following restriction $0 < \eta^- < 1 < \eta^+$. To prevent the step size from being too small or too large $\Delta^{(k)}$ is bounded by $0 < \Delta_{\min} < \Delta^{(k)} < \Delta_{\max}$. To satisfy this condition after the calculation of (5.45) a saturation operation is applied

$$\Delta_i^{(k)} := \mathbf{sat}(\Delta_i^{(\mathfrak{c})}) \tag{5.46}$$

where the saturation operation $\mathbf{sat}(\Gamma)$ is defined as,

$$\mathbf{sat}(\Gamma) = \begin{cases} \Gamma_{\min} & \text{if } \Gamma < \Gamma_{\min} \\ \Gamma_{\max} & \text{if } \Gamma > \Gamma_{\max} \\ \Gamma & \text{else.} \end{cases} \tag{5.47}$$

The update process of RPROP is shown in Figure 5.7 for a single dimension a. The search direction is selected based upon the current sign of the gradient only. So as at time-step $\mathfrak{c}$ the gradient is negative, as the previous solution also has a

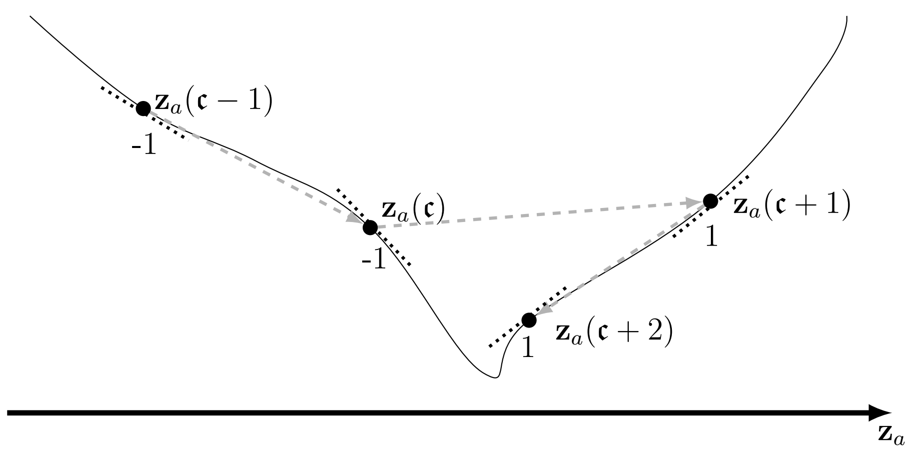

$\mathbf{z}_a(\mathfrak{c}-1)$
-1
$\mathbf{z}_a(\mathfrak{c})$
-1
$\mathbf{z}_a(\mathfrak{c}+1)$
1
$\mathbf{z}_a(\mathfrak{c}+2)$
1
$\mathbf{z}_a$

Figure 5.7: RPROP update rule along a single dimension a.

negative gradient the step size is extended. Assuming now that the step is sufficient to take us over to the other side of the basin and remains within it. The gradient is now therefore positive and the sign has flipped. This causes two things to happen; the step size is reduced and its direction reversed. Once a basis of attraction has been found and a step occurs with a sign change that remains within the basin then converge occurs. No special handling of the box constraints is required, after the position update the saturation operator is applied to ensure the solution remains within the search space

$$\mathbf{z}(\mathfrak{c}+1) := \mathbf{sat}(\mathbf{z}(\mathfrak{c}+1)). \tag{5.48}$$

For a the initial iteration where no previous knowledge exists for the previous gradient a search is conducted along the direction of steepest descent with a step-size δ_0. Once convergence begins within a single basin of attraction the algorithm will find the minimum and become stuck. This may be a local-minima, to make the search global a variety of improvements can be undertaken. A multi-start version of the algorithm can be done utilising starting locations distributed throughout the solution space. Another advance is the use of restarts, a scheme to reset some of the search parameters at a set interval to move the algorithm away from the current solution [88]. In [88] a restart is applied every T iterations according to Algorithm 5.5. The position is updated to the best location found so far by the search, the step length $\Delta_a^{(\mathfrak{c})}$ and the extension parameter η^+ are reduced by an increment ζ. To prevent the extension parameter from becoming too small it is limited to a minimum ξ. The parameter $\theta(\mathfrak{c})$ counts the number of iterations since the the best objective value $\tilde{J}$, for which the vector is $\tilde{\mathbf{z}}$, was found,

$$\theta(\mathfrak{c}) = \begin{cases} \theta(\mathfrak{c}-1)+1 & \text{if} \quad J(\mathfrak{c}) > \tilde{J} \\ 0 & \text{else.} \end{cases} \tag{5.49}$$

If this exceeds a threshold $\theta_{\max}$ then when a restart occurs the contraction factor c that controls the reduction of $\Delta_i^{(\mathfrak{c})}$ is lowered by an order of magnitude, otherwise if it is reset to 1. This algorithm therefore refines the search over time and allows for fine tuning of the solution.

Algorithm 5.5 Restart algorithm.

```
Input: $\mathfrak{c}$, $\mathfrak{c}_r$, $\tilde{\mathbf{z}}$, $c(\mathfrak{c}-1)$, $\Delta_a^{(\mathfrak{c})}$, $\Delta_{\min}$, $\eta^+$, $\zeta$, $\xi$
Output: $\mathbf{z}(\mathfrak{c})$, $c(\mathfrak{c})$, $\Delta_a^{(\mathfrak{c})}$, $\eta^+$
 1: if $\mathfrak{c}_r \mid \mathfrak{c}$ then
 2:     if $\theta(\mathfrak{c}) > \theta_{\max}$ then
 3:         $c(\mathfrak{c}) = 0.1c(\mathfrak{c}-1)$
 4:     else
 5:         $c(\mathfrak{c}) = 1$
 6:     end if
 7:     $\mathbf{z}(\mathfrak{c}) = \tilde{\mathbf{z}}$
 8:     $\Delta_a^{(\mathfrak{c})} = \max\{0.1c(\mathfrak{c})\Delta_a^{(\mathfrak{c})}, \Delta_{\min}\}$
 9:     $\eta^+ = \max\{\eta^+ - \zeta, \xi\}$
10: end if
```

The choice of $\mathfrak{c}_r$ depends on the problem to be solved. A short $\mathfrak{c}_r$ allows for a lot of restarts to allow for the proper exploitation of a complex space. However, the restarts could disrupt the algorithm and if $\mathfrak{c}_r$ is too short the algorithms convergence could be halted. Instead of applying a restart every $\mathfrak{c}_r$ intervals a dynamic $\mathfrak{c}_r$ could be used. This could allow for the algorithm to have a time to converge whilst also having shorter restart periods to allow for the full solution space to be explored. A chaotic restart $\mathfrak{c}_r$ is proposed according to a sinusoidal mapping as inspired by [74]. $\mathfrak{c}_r$ is updated at each restart by

$$\mathfrak{c}_r = \mathfrak{c}_r + r(\mathfrak{c}_{r,\max} - \mathfrak{c}_{r,\min}) + \mathfrak{c}_{r,\min} \tag{5.50}$$

with $r \in [0, 1]$ the seed that is set randomly at the start of the algorithm and updated by the sinusoidal map at each restart using

$$r = \sin(r\pi). \tag{5.51}$$

$\mathfrak{c}_r$ is set at the start of the algorithm to equal $\mathfrak{c}_{r,\max}$ to allow for initial convergence to occur without interruption. Minor adjustments are made to the restart algorithm 5.5 in the case of the chaotic restart. Line 1 the if clause becomes if $\mathfrak{c} = \mathfrak{c}_r$ and the equation (5.51) followed by (5.50) are the first commands executed when the if statement is true. Otherwise the algorithm remains unchanged.

The Algorithm 5.5 as used in [88] reduces the step size and extension parameter

Algorithm 5.6 Updated restart algorithm using chaotic interval.

```
Input: $\mathfrak{c}$, $\mathfrak{c}_{\mathrm{r}}$, $\mathfrak{c}_{\mathrm{r,max}}$, $\mathfrak{c}_{\mathrm{r,min}}$, $r$, $\tilde{\mathbf{z}}$, $c(\mathfrak{c}-1)$, $\Delta_a^{(\mathfrak{c})}$, $\Delta_{\min}$, $\eta^+$, $\zeta$, $\xi$
Output: $\mathfrak{c}_{\mathrm{r}}$, $r$, $\mathbf{z}(\mathfrak{c})$, $c(\mathfrak{c})$, $\Delta_a^{(\mathfrak{c})}$, $\eta^+$
1: if $\mathfrak{c} = \mathfrak{c}_{\mathrm{r}}$ then
2:     $r = \sin(r\pi)$
3:     $\mathfrak{c}_{\mathrm{r}} = r(\mathfrak{c}_{\mathrm{r,max}} - \mathfrak{c}_{\mathrm{r,min}}) + \mathfrak{c}_{\mathrm{r,min}}$
4:     if $\theta(\mathfrak{c}) > \theta_{\max}$ then
5:         $c(\mathfrak{c}) = 0.1c(\mathfrak{c}-1)$
6:     else
7:         $c(\mathfrak{c}) = 1$
8:     end if
9:     if Normal_random_number(0,1)$< 0.02$ then
10:        $c(\mathfrak{c}) = 1$
11:    end if
12:    $\mathbf{z}^{(k)} = \tilde{\mathbf{z}}$
13:    $\Delta_a^{(\mathfrak{c})} = \max\{0.1c(\mathfrak{c})\delta_{0,a}, \Delta_{\min}\}$
14:    $\eta^+ = \max\{\eta^+ - \zeta, \xi\}$
15: end if
```

making the search more refined. This may not be sufficient to move the algorithm away from its current basis of attraction and may remain stuck at a local minima. A more aggressive restart method is to set the step size Δ to a specific value rather than a function of itself. However, the reduction property of the original restart method is also a desired property. This allows for a fine search during the late stages of the algorithm. Therefore a method is proposed whereby the step size Δ is updated as a function of the initial step size δ_0 following the same style as the original procedure. This makes the search more likely move to new regions, rather than performing a refined search at the current found optima. The updated restart interval using a chaotic restart is given in Algorithm 5.6. The value of c reduces over time if no improvements are found to the current best solution making the step away from the current optimum smaller. This algorithm therefore retains the refinement property, but changes to the solution will be greater than the original restart method.

5.6 Overview

5.6.1 Algorithm Parameters

Each of the algorithms discussed have control parameters that influence their search behaviour. Furthermore each algorithm requires a set of initial points from which the search should start. All algorithms were initialised using Latin hypercubes, variables are assumed to have a uniform distribution within its range, as defined in

Table 5.3. The limits in the table are simply handled by adjusting solutions that move outside the domain to the boundary. For the PSO algorithms the velocity component for dimensions that are adjusted in this manner is multiplied by -0.5. This prevents the solution from attempting to leave the space on the next iteration.

Table 5.3: Variable limits.

Variable	τ	κ	ν	$v_{\min}$	$\rho_{\max}$	δ	ϕ	α_m	$v_{f,m}$	$\rho_{cr,m}$
Maximum	60	90	90	8	190	4	3.0	5.00	130	40.0
Minimum	1	5	1	5	160	0.001	0.1	0.40	80	18.0

The GA, PSO and CS are all population based algorithms. The population size for the GA and PSO was set to 30. For CS and MCS a population of 25 was used as this is what is used in [108]. These values were corroborated by simple trials and no significant performance gains were observed for different population sizes. The RPROP algorithm is not population based however a multi-start approach is used to ensure that the minima found is close to optimal, 6 initial starting points are used.

Table 5.4: PSO parameters.

Variant	c_1	c_2	ω	Reference
APSO-09	ESE	ESE	$\frac{1}{1+1.5e^{-2.6\mathfrak{f}}}$	[81]
APSO-12	$(1.2-2.8)\frac{\mathfrak{c}}{\mathfrak{C}}+2.8$	$(2.8-1.2)\frac{\mathfrak{c}}{\mathfrak{C}}+1.2$	$(0.75-0.35)e^{0.001\mathfrak{c}}+0.35$	[68]
APSO-14	2.0	2.0	(5.27)	[67]
HEPSO	$(0.5-2.5)\frac{\mathfrak{c}}{\mathfrak{C}}+2.5$	$(2.5-0.5)\frac{\mathfrak{c}}{\mathfrak{C}}+0.5$	$\frac{1}{1+1.5e^{-2.6\mathfrak{f}}}$	[75]
CEPSO-12	N/A	N/A	N/A	[74]
GPSO	$\frac{1}{2}+\ln(2)$	$\frac{1}{2}+\ln(2)$	$\frac{1}{2\ln(2)}$	[103]
LPSO	$\frac{1}{2}+\ln(2)$	$\frac{1}{2}+\ln(2)$	$\frac{1}{2\ln(2)}$	[103]

The GA parameters are the probability of crossover, set at 0.7, and the mutation probability, set at 0.05. The parameters for the various PSOs is typically inherited from the papers they were proposed in, and shown in Table 5.4. In the table $\mathfrak{c}$ is the search iteration number and $\mathfrak{C}$ is the maximum number of iterations which was set at 5000. $\mathfrak{f}$ defined in [75, 81] is the evolutionary factor that defines the spread of particles relative to the current optimum. For CEPSO the sinusoidal map which performed best in the benchmarks in [74] is used for the chaotic map. The probabilities for the artificial bee colony and GA operators used by HEPSO are the same as those in [75]. For GPSO and LPSO better results were obtained by using the parameter values suggested in [103] rather than the typically used $c_1 = c_2 = 2.0$. In CS and MCS the percentage of nests to be abandoned in each iteration was set equal to 25% and 75% respectively, as suggested in [78] and [79]. The dramatic difference between these values is that fact that MCS makes some significant changes to the algorithm. For CS the characteristic step length was set to 1/100 of the variable search range as shown in Table 5.3 as it is in [108]. For MCS the characteristic length is the same as proposed in [79].

5.6.2 Convergence Properties

This chapter has looked at a variety of optimisation techniques. In general the the convergence properties and computational effort required are similar between the various EAs (GA, PSO and CS). All of these algorithms are population based with the next population being generated based on a heuristic of the current population. There are differences in the complexity involved in the heuristics used. In comparison to the computational complexity of the underlying traffic model this difference becomes insignificant and the runtime of the algorithm is dependant on the number of function evaluations performed. That is to say the number of iterations required and the size of the population ($\mathfrak{c} \cdot N_{\text{pop}}$). Therefore the importance is not on the efficiency of the algorithms update method but the number of function evaluations, hence this metric is used in comparing the algorithms in later chapters. For the gradient based RPROP once a gradient is obtained the update method is very simple and easy to work out, however due to the complexity of obtaining the gradient the number of function evaluations required is again the key factor determining the algorithms run time. But the extra effort required to compute the gradients make a function evaluation much more costly than for the EAs, see Section 7.5.5. However by using the gradient information it is possible to make some efficiencies over the EAs. The gradient provides a search direction to the algorithm so there is no requirement for a population of solutions. This however is not strictly true due to the nature of the solution space.

As for the final solution found by the algorithms the EAs provide no guarantee of convergence to a global minimum. For RPROP the algorithm will find a local minimum based on its starting location. The non-linear and shape of the solution landscape means that there are many minima. By using a multi-start algorithm with different starting locations it is then possible to take the best solution from the minima found. Again this provides no guarantee of a global-minima. The key is to find a suitably acceptable local minima that results in a valid traffic model that is accurate enough for the models end case. For RPROP to determine if the algorithm has found a global minima or maxima the second derivative would have to be evaluated. This would significantly increase the computational requirement. It is easier to keep track of the objective function value between values and ensure that the value is decreasing.

5.6.3 Conclusion

This chapter introduced a different classes algorithm to be applied to the optimisation problem, GA, PSO and CS. For the GA various crossover operators have been outlined. The ideal operator to use and weightings between their choice is an area that will be investigated in Chapter 8. For the PSO and CS class algorithms a few variants are detailed, investigations comparing these variants are also reported in 8. This current chapter also introduced AD and a method for solving the optimisation problem using a gradients. This is more intensive and but provides more information about the solution space than the other algorithms described, so overall the formulation could be more computationally efficient. The use of gradient information however means that the AAFD process requires modification. Tests comparing this simplified AAFD to the formats used by the population based algorithms is done in Chapter 9. With a series of potential algorithms identified to solve the validation problem the next stage is to gather reference data to apply the algorithm on, this is discussed in the next chapter.

Chapter 6

Data Processing and Model Sites

6.1 Loop Detectors

The validation problem shown in Chapter 4 requires a series of reference data for calibration purposes. In this thesis data are sourced from the Highways Agency's (the UK's motorway operator) Motorway Incident Detection and Automatic Signalling (MIDAS) system which is managed by Mott MacDonald. A simple loop detector is shown in Figure 6.1. It consists of two components, a 'loop' of insulated wire embedded in the highway in a shallow saw cut channel and an electronic detector/control module. The module provides power to the loop and energises it at a constant frequency, in the range of 10–50kHz, creating a magnetic field. As a vehicle passes over the loop the inductance of the system is reduced due to the eddy currents induced by the interaction of the vehicles' metallic parts with the generated magnetic field. This inductance drop causes the oscillation frequency to rise. The frequency is monitored by a detector and calculates when a vehicle enters and leaves the loop Figure 6.2. A vehicle is assumed to have entered the detector when the frequency measured surpasses a threshold value and leaves once the frequency drops below this level. The detector is sampled at a high rate to ensure that every vehicle that passes is detected. Different classes of vehicles result in differing frequency responses, typically sports cars result in a higher peak and heavy good vehicles a lower one due to the difference in ground clearance. Depending on the required use the configuration and shape of the loop wire can be changed. The simple single loop detector is able to count the number of vehicles that pass it, as well as the occupancy. Occupancy is the measure of the percentage time that a loop is detecting a vehicle and it can be used as a measure of traffic density. If the vehicles' length was known it would be possible to calculate the speed from the time between the vehicle entering and leaving the detector. This is not the case so another method is used, the double loop detector.

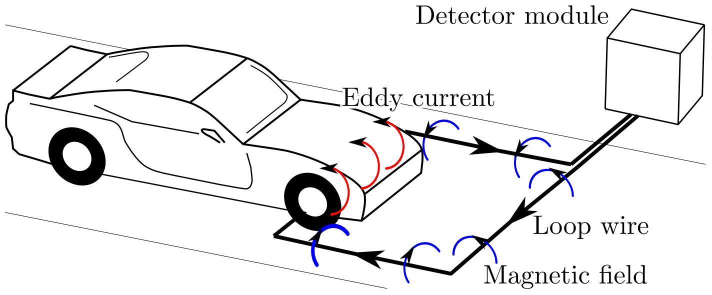


Figure 6.1: Vehicle passing over a simple loop detector.

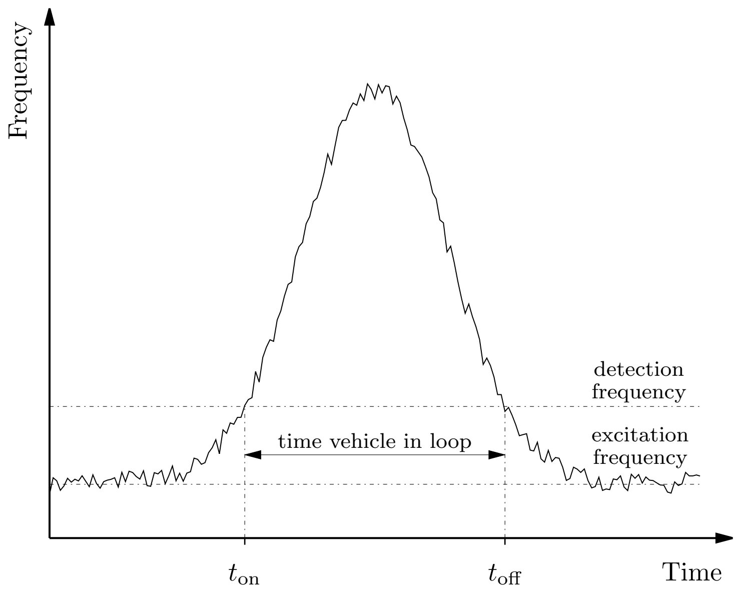


Figure 6.2: Example of a loop detector sensing a vehicle.

A double loop detector measures a vehicles' speed directly. It consists of two inductive loops being embedded in the highway at a set interval. By calculating the time between the first and second detectors becoming occupied and dividing by the loop spacing the vehicles' speed is obtained. With a known speed it also possible to calculate the vehicles' length.

6.2 MIDAS Data

The Highways Agencies MIDAS data comes from double loop detectors with a detector in each lane of the highway, a sample location is shown in Figure 6.3. Detectors are placed throughout the UK's motorway network, typically at a spatial resolution of 500m. Not all of the UK's motorway network is covered by the system, most of England has coverage with preference given to roads with high demand. Gaps can exist in the data coverage due to faulty infrastructure or a lack of infrastructure due to economical considerations.

The MIDAS data provides for each lane of the highway the number of vehicles,

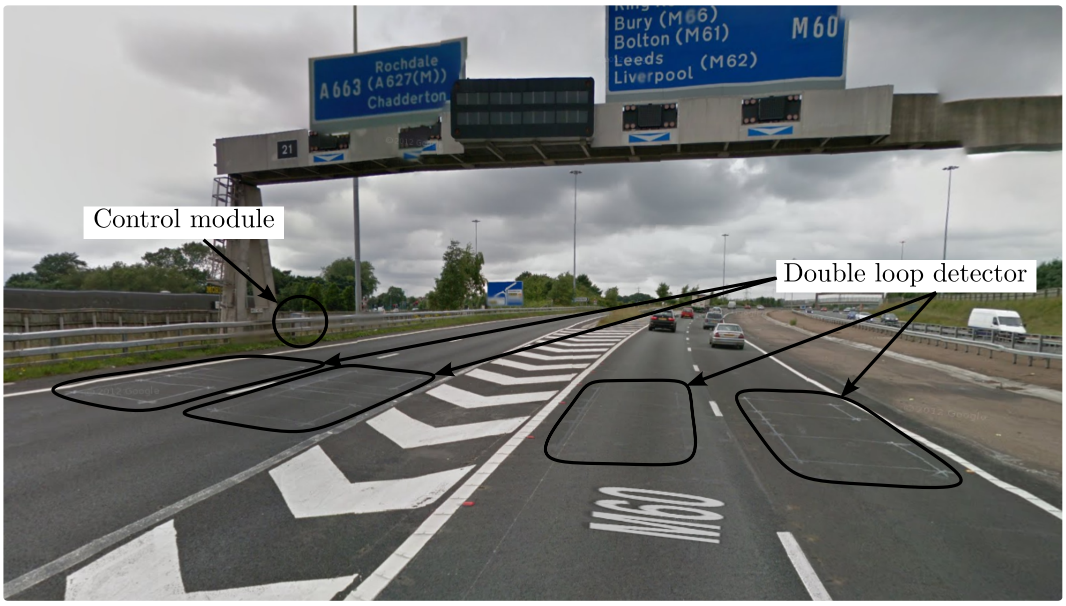


Figure 6.3: MIDAS loop detectors at Junction 21 of the anti-clockwise M60. Image capture: Jul 2012, © 2015 Google.

average speed and occupancy for every minute of every day. The vehicle counts are also available in a broken down set of vehicle classes based on the vehicles length. Each detector within the network has a unique identifier which includes the road number, marker post number (distance from start of highway) and carriageway classification. The classification identifies the direction of travel and also if the detector is located at an on- or off-ramp.

6.3 Data Processing

6.3.1 Data Requirements

A macroscopic model needs to be supplied with the correct boundary data at origins and destinations. The disturbance vector $\mathbf{d}(k)$ also needs to be supplied for each time step and the initial state $\mathbf{x}_0$ given for $k = 0$. The data requirements for an arbitrary site are shown in Figure 6.4. This shows that the origins to the model need to be supplied with a flow rate for each time step k. Furthermore Origin 1 also requires speed data for all time steps. This requirement is only for higher-order models which explicitly model the traffic streams speed. These models propagate speed through the network therefore for major origins (see 4.2.3) a speed has to be supplied. For all destinations density must be given for all time steps to ensure that the effects of congested off-ramps and downstream conditions are accounted for. The disturbance vector contains the splitting rate and information on any incidents. Incident modelling is not part of this work so the disturbance vector consists of the splitting rates β. These have to be supplied for all of the divergence nodes (Nodes C and E in the example). For the CTM model the disturbance vector also contains merging ratios which need to be supplied for the merging nodes (Node D in Figure 6.4) $p(k)$. The initial condition requires that the traffic state is given for every segment of the model at time step $k = 0$. This means that for all of these segments a speed, density and flow must be supplied. Again for first-order models the speed requirement can be relaxed as the speed is not used in the model equations. To calculate the correct inputs for the model the location of the available loop detectors within the model must be found.

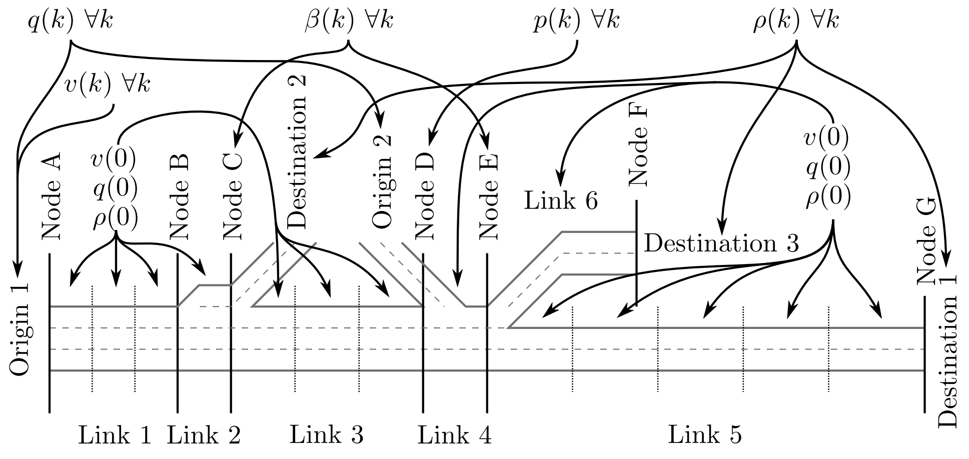
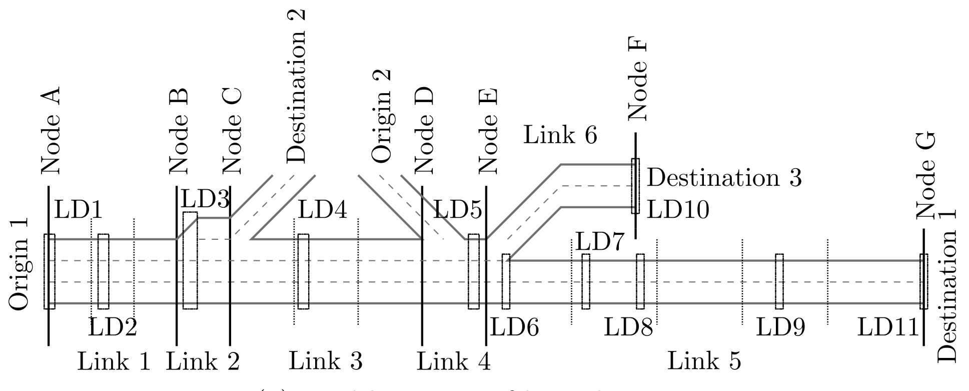
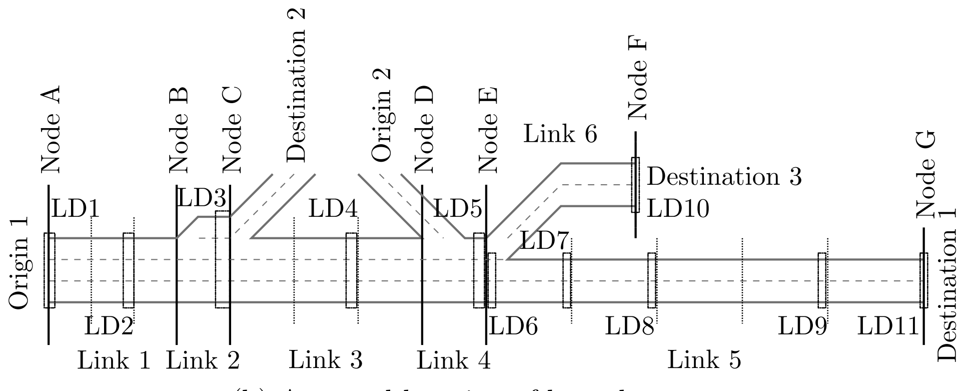


Figure 6.4: Data requirements of an arbitrary model.

Given a particular model of a site with a number of loop detectors, each is assigned to the link m in which it resides. As the model provides outputs at a segment resolution; with the space mean speed over the segment and the flow leaving it, loop detectors will be assumed to be at the end of segments. This means they provide the flow out of the segment and their actual position will not be used. This is indeed true for 2$^{\text{nd}}$ order macroscopic models where it is explicitly assumed that the traffic conditions in a segment are homogeneous. Interpolation of values is therefore not required. If a segment has two available loop detectors within it then the loops are assumed to be located at the closest boundary, one at the upstream and the other downstream. The most upstream loop is therefore moved to segment $i-1$. This can have a knock-on effect as segment $i-1$ could now contain two detectors. If a detector to be moved is in segment 1 then the process stops and the loop being moved is assigned to the node ($i = 0$). The assignment process is shown in Figure 6.5b. As can be seen in the figure LD7 and LD8 are in the same segment so LD7 is moved to the upstream segment. This is the first segment of Link 5 which already contains LD6. Therefore LD6 is assigned to be at the start of Link 5.



(a) Real locations of loop detectors.



(b) Assumed location of loop detectors.

Figure 6.5: Assigning loop detectors to segments within a model.

6.3.2 Loop detector data

With the loop detectors now assigned to a segment of the model the data provided by them must be converted into macroscopic variables. The loop detector data is on a different time-scale to the model and is available for $\hat{k} = 1, \ldots, \hat{K}$ at a time step $\hat{T}$. For MIDAS data $\hat{T} = 60s$, $\hat{k} = 1$ is at 00:00 and $\hat{K} = 1440$ as data is given at minute intervals for an entire day. The loop detector data is given per lane, the flow per lane from the detector is $\hat{q}_{m,i,\ell}$ and speed per lane $\hat{v}_{m,i,\ell}$ but the models require the values over all lanes $\hat{q}_{m,i}$ and $\hat{v}_{m,i}$ respectively. The flow is simply calculated as the sum,

$$\hat{q}_{m,i}(\hat{k}) = \sum_{\ell=1}^{\lambda_m} \hat{q}_{m,i,\ell}(\hat{k}). \tag{6.1}$$

The calculation of the mean speed is not as straightforward. The speed computed by the model is the space-mean speed. The measurement from the loop detectors however is a time-mean speed as the data available is the average value over an interval $\hat{T}$. The time-mean speed is larger than the space mean speed and has been shown to be different by about 1–5% by [109]. The conversion of a time-mean speed to a space-mean speed is not simple and requires the variance of the space-mean speed as shown in [110]. The approximation in this thesis is based on the assumption that the variance in the space mean speed is zero over the segment for each lane and the variance in the speed occurs between lanes. Then the model speed can be estimated as the flow per lane weighted harmonic mean,

$$\hat{v}_{m,i}(\hat{k}) = \frac{\sum_{\ell=0}^{\lambda_m} \hat{q}_{m,i,\ell}(\hat{k})}{\sum_{\ell=0}^{\lambda_m} \frac{\hat{q}_{m,i,\ell}(\hat{k})}{\hat{v}_{m,i,\ell}(\hat{k})}}. \tag{6.2}$$

The comparison between this method and using the arithmetic mean (time-mean speed) is shown in Figure 6.6. The figure shows that by assuming that each lane is homogeneous with zero variance in speed (i.e. the loops provide the space-mean speed per lane) the calculated speed is less than the time-mean speed. This difference between the two values is shown in Figure 6.7. This shows the reduction in the speed to be around 0-2% for free flowing conditions and increases to 2-12% for the congested region. As the estimated speed is less than the time-mean speed then it is likely to be a closer approximation to the unknown true space-mean speed than the time-mean speed. With a value for flow and speed obtained, the density can be estimated from the hydrodynamic relation,

$$\hat{\rho}_{m,i}(\hat{k}) = \frac{\hat{q}_{m,i}(\hat{k})}{\lambda_m \cdot \hat{v}_{m,i}(\hat{k})}. \tag{6.3}$$

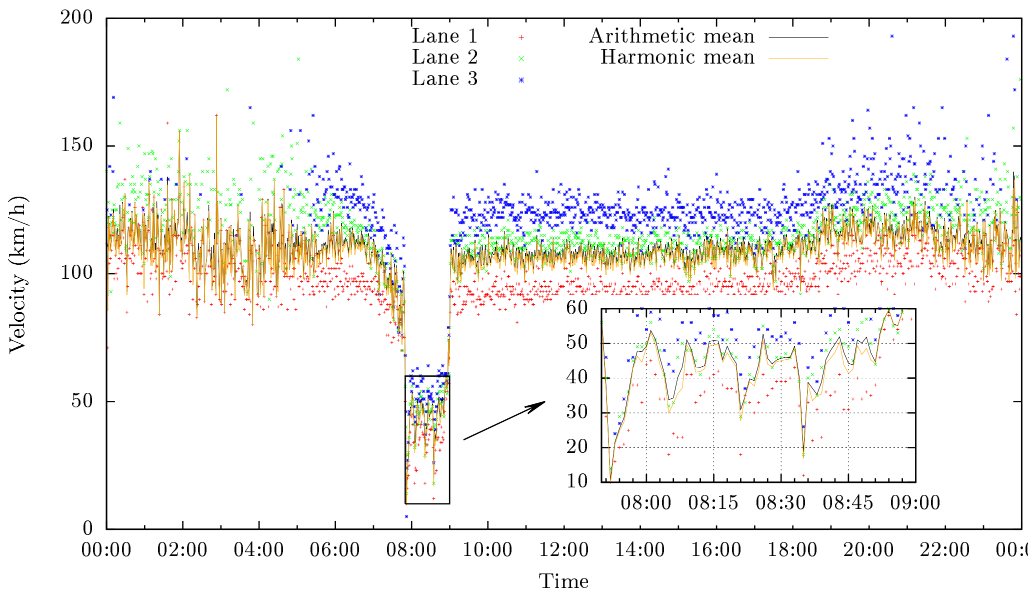


Figure 6.6: Loop detector speed calculation, loop data from M1 Northbound near Sheffield, 1st.

The loop detector data provide the required model inputs; the disturbance vector

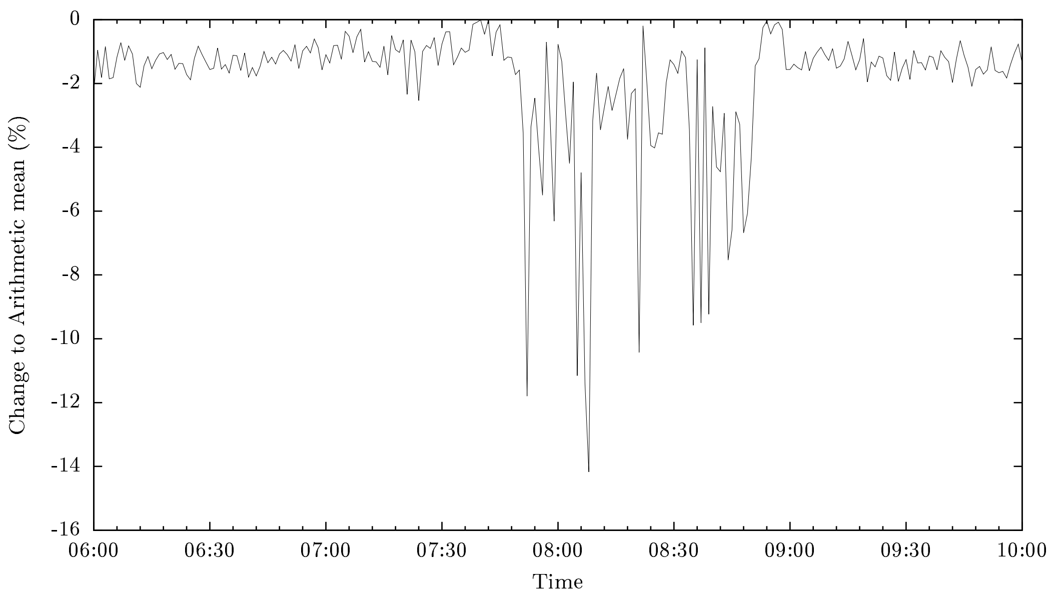


Figure 6.7: Comparison of speed calculated using a harmonic mean to an arithmetic mean, loop data from M1 Northbound near Sheffield, 1st.

d and the initial condition $\mathbf{x}_0$ of the state-space model eqn. (4.1). Before this is done the values obtained from the loop detectors need to be checked to ensure that the detectors are fully operational and provide realistic measurements.

6.3.3 Network flow evaluation

To evaluate the consistency of the available data the flows throughout the network are evaluated. This is done by confirming that along each link and at every node no flow is being created or lost. If flow is not being preserved it could be due to a faulty loop detector providing incorrect data. The flow evaluation has two phases, one on the link level and the other on the nodes. The first phase involves confirming that the flow measurements within each link are consistent with each other. The second then is to confirm that the flow over nodes sums to zero, i.e. vehicle conservation is preserved.

The first phase, evaluating the flow consistency within a link, can only be done for links that contain multiple loop detectors. For these links the flow over the whole model's time horizon is summed for each loop detector. If the difference between any summed detector's flow is greater than 5% then it is possible that a detector is faulty. The detector's locations can then be analysed for any errors and a fix be applied or the faulty loop detector removed from the data. The second phase confirms that flow is preserved over nodes. For each node n the incoming flow over the time horizon is compared to the out-going flow, and again any difference greater than 5% was flagged and evaluated. To ensure that all links were involved in the evaluation process an estimated flow was calculated for links without detectors. Estimate flow was calculated using a 5 minute moving average of the known flows over the node,

$$\hat{q}_{m,0}(\hat{k}) = \frac{1}{5} \cdot \left(\sum_{\mu \in O_n} \sum_{j=-2}^{2} \hat{q}_{\mu,i}(\hat{k}+j) - \sum_{\mu \in I_n} \sum_{j=-2}^{2} \hat{q}_{\mu,s}(\hat{k}+j) \right) \qquad \hat{k} = 0, \dots, \hat{K}. \quad (6.4)$$

A moving average is used to account for travel time between the measurement locations and flow variation. The estimated value can then be used in evaluating the balance at remaining nodes. In Figure 6.8 an example network with loop detectors is shown. No loop data is available for links L3, L4 and L7. When the balance is confirmed at node B an estimate flow for L4 will be calculated. This value is then used in evaluating the flows through node C. An estimate flow can be found for L3 by the evaluation of node G. This then makes it possible to estimate flows for L7 that can be compared with the outflow at node F. This allows for all of the nodes to be checked for balance. Any links that cause for imbalance in flow to occur can then be cross checked to ensure the integrity of the data. If an estimated value at a node appears to be incorrect then the error comes from the one of the links at the node which the estimate was calculated.

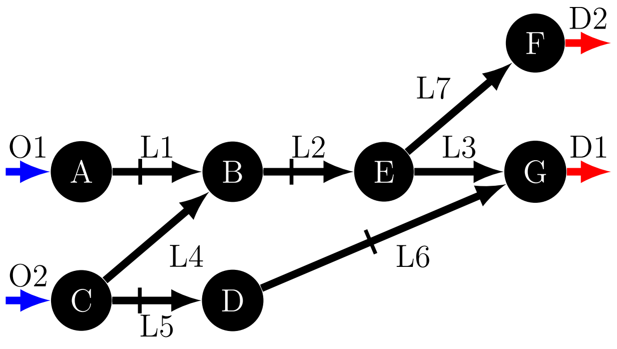


Figure 6.8: Checking node flows. Links with bars contain loop detectors. Origins and destinations also have loop detector data availiable.

6.3.4 Boundary conditions

At origins into the model a flow vector needs to be provided. For origins that are major in-links to nodes speed also needs to be provided for the second order models. This is taken from the data where it is available and if not the estimate generated in eqn. (6.4) is utilised. For the destination links the downstream density is required. If data is not available then the density is estimated by assuming a speed equal to the upstream link if data exists, 80km/hr if it does not, and calculating a density using eqn. (6.3). The estimates where data are not available are calculated for all $\hat{k}$.

6.3.5 Turning and priority rates

Turning rates are calculated for all $\hat{k}$. When each link has a flow (estimated or measured) their calculation is simple. For each diverge node the percentage of the incoming flow to each of the out-links is computed. As there is a travel time between the measurement locations using different link pairs will give slightly different results. This will be minimal as the node balance was confirmed earlier. Preference is given to use real data if it is available, then to data (real or estimated) that are closest to the node.

Priority rates are required for merging nodes of the CTM model. They define how much flow comes into the merge from each of the supplying links when the downstream link is unable to receive all flow. These proportions are calculated from the real data for all time steps $\hat{k}$ using the same method as the turning rates.

The boundary conditions, turning and priority rates are calculated for all $\hat{k}$ and no conversion to the model time step k is required as the models linearly interpolate between supplied values.

6.3.6 Initial condition

At the initial time point $k = 0$ the initial state of the variables q, v, ρ needs to be given for every segment in each link. Where loop data are available they are taken directly from the calculated values. It is not likely that all the segments in a model will contain loops so data will not be available. Once the known points are set linear interpolation is performed between these locations within a link for segments that do not have data. If data is missing from the start or end of a segment then the initial condition is set to be the same as the nearest loop within the link. Interpolation or comparisons are not performed across link boundaries as topology changes can occur that could lead to large changes in flow, speed, and density. If a link does not contain a loop detector then the estimated (eqn. 6.4) flow is utilised. All segments within the link are assumed to have the same flow as estimated at the node. The link is assumed to be free flowing and all segments are assumed to have a mean speed of 110km/hr the density then calculated by eqn. (6.3). This means that for every segment i of each link m,

$$q_{m,i}(0) = \begin{cases} \hat{q}_{m,i}(\hat{k}_{\text{START}}) & \text{if } \hat{q}_{m,i}(\hat{k}_{\text{START}}) \text{ exists} \\ \hat{q}_{m,0}(\hat{k}_{\text{START}}) & \text{else} \end{cases} \tag{6.5}$$

$$v_{m,i}(0) = \begin{cases} \hat{v}_{m,i}(\hat{k}_{\text{START}}) & \text{if } \hat{v}_{m,i}(\hat{k}_{\text{START}}) \text{ exists} \\ 110 & \text{else} \end{cases} \tag{6.6}$$

$$\rho_{m,i}(0) = \begin{cases} \hat{\rho}_{m,i}(\hat{k}_{\text{START}}) & \text{if } \hat{\rho}_{m,i}(\hat{k}_{\text{START}}) \text{ exists} \\ \frac{q_{m,i}(0)}{\lambda \cdot v_{m,i}(0)} & \text{else} \end{cases} \tag{6.7}$$

with $\hat{k}_{\text{START}}$ the time-step of the data that corresponds to start of the model time period.

6.3.7 Calibration data

In the objective function of the calibration problem there is a requirement for mean speed data for every time instant of the model (eqn. (4.17)) at as many points as possible. Therefore, data from all measurement locations are applied to the calibration problem. As the measured data are available at a constant interval of $\hat{T}$ seconds a conversion to the model time step T is required. The data provide the average over the interval $\hat{T}$. The optimisation problem is one of least-squares so the measured data can be applied without manipulation to all time-steps which correspond to that particular measurement. The objective function will then be responsible for making the model respond in such a way so it fits the data without the assumption of a particular trend between time-steps as required for an interpolation scheme. The calibration data $\mathbf{y}$ can therefore be calculated as,

$$\mathbf{y}_{v,j}(k) = \hat{v}_{m,i}\left(\left\lfloor \frac{k \cdot T}{\hat{T}} \right\rfloor + \hat{k}_{\text{START}}\right) \quad \forall k. \tag{6.8}$$

The values of m, i for which data are available are mapped to the lowest possible index j.

6.4 Model Sites

6.4.1 Overview

Three different model sites are considered in this work. Each one has its own congestion pattern, and exhibits periods of recurrent congestion. The sites considered compose of a section of the eastbound M4 around Heathrow, the northbound M1 on the approach to Sheffield and the orbital motorway network around Manchester as shown in Figure 6.9. The M4 and M1 sites were selected due to the fact that they exhibit recurrent congestion. For a large scale network Manchester was selected over other options such as Birmingham and London due to the fact that the motorways that form these other orbitals are part of the Highways Agencies managed motorway network, with various schemes such as variable speed limits, ramp-metering and hard shoulder running.

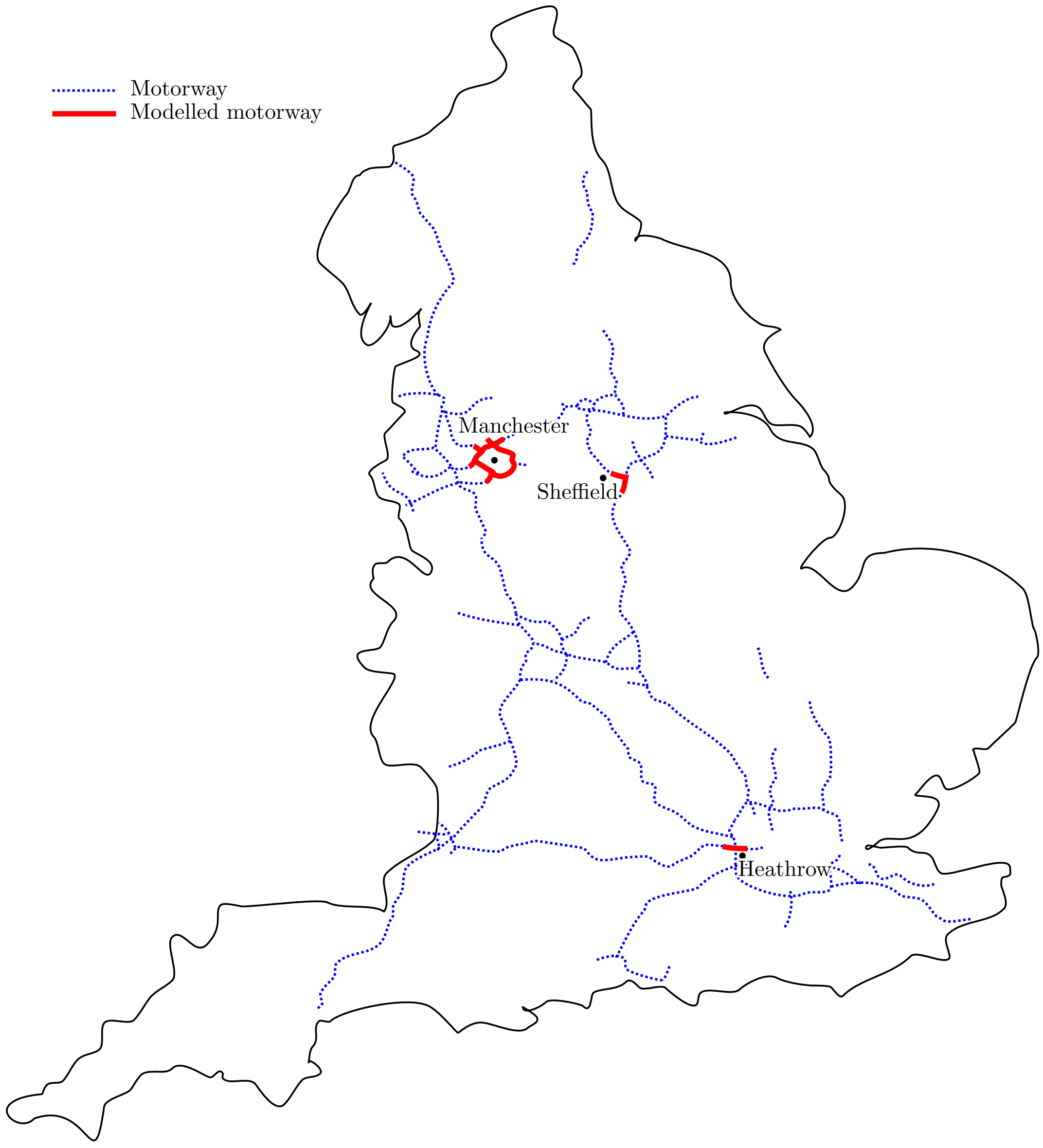


Figure 6.9: Map of model sites within the English motorway network. Drawn using Ordnance Survey data, © Crown Copyright and Database Right 2015, Ordnance Survey (Digimap Licence).

6.4.2 Heathrow-M4 site

The Heathrow model is the shortest site tested in this thesis at 7.8 km. The model comprises the eastbound M4, which links London to Reading and Bristol and a map of the site is shown in Figure 6.10. This site has large inflows and outflows due to the junction with London's orbital, the M25. It also provides access to parts of Heathrow Airport. The model consists of 5 links, is supplied flow from 3 origins and serves 3 destinations as shown in Figure 6.11. The calibration process was carried out using data from Monday 8th, 15th and 22nd of February 2010. This is a linear model with origins and destinations acting directly on the mainline. Therefore, the network splitting algorithm (4.2.3) treats the entire model as a single section. The main challenge of the Heathrow site is data availability, none of the on- or off-ramps contain loop detectors. This provides an ideal test of the data processing done in 6.3.4.

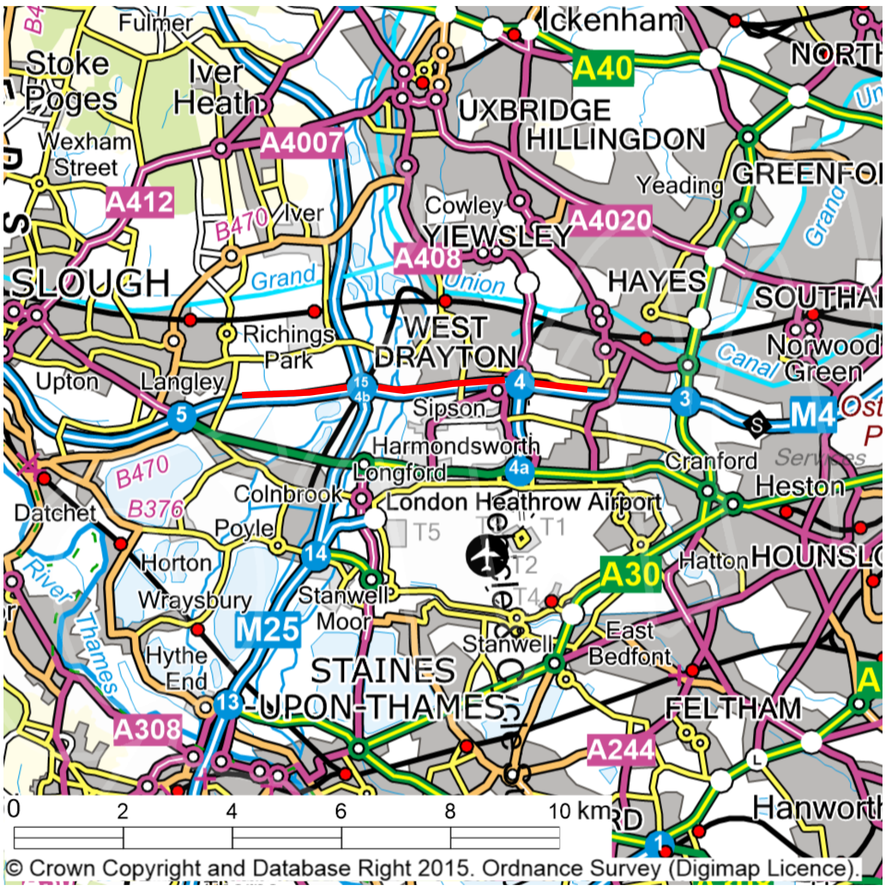


Figure 6.10: Map of the Heathrow site, modelled carriageway highlighted in red. © Crown Copyright and Database Right 2015, Ordnance Survey (Digimap Licence).

Congestion typically occurs between the merge of the traffic from the M25 at Junction 4B and the M4 link road to Heathrow at Junction 4. The off-ramp at Junction 4, is usually congested which spills back onto the main M4, this compounds

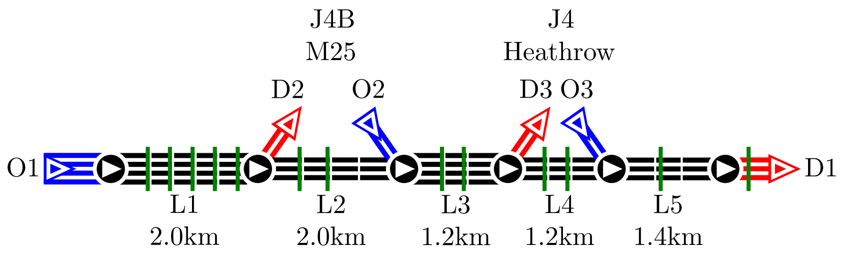


Figure 6.11: Schematic of the Heathrow model. A line represents a lane of traffic, breaks indicate segments. Origins are shown in blue and destinations in red. Locations of available loop detectors are shown by green bars.

with the merge of a large number of vehicles from the M25. Also seen is a small congestion period towards the end of L1 as congestion from the M25 starts to impact on the vehicles trying to access it from the M4.

6.4.3 Sheffield-M1 site

The Sheffield model is a 21.9 km section of the northbound M1 (Figure 6.13. A schematic of the model is shown in Figure 6.12. The model is composed of 10 links, 5 origins and 5 destinations. One of the origins, destination pairs are for the M18 motorway, with the remaining connecting to major roads and urban networks.

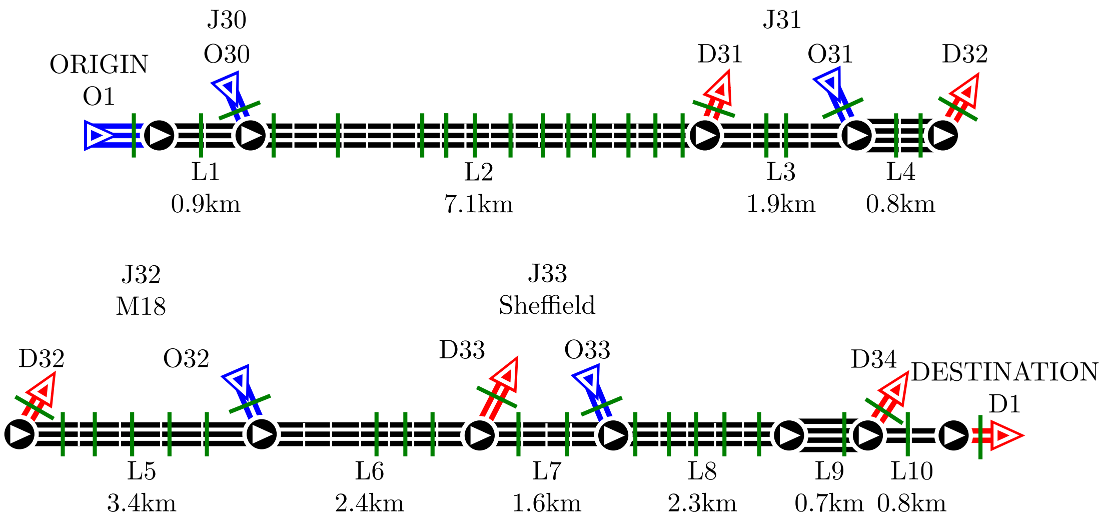


Figure 6.12: Schematic of the Sheffield model, including the location of available loop detectors.

Data are being used used from June 2009, with calibrations being done using the Monday morning congestion on the 1st, 8th and 15th. Congestion occurs at this site due to the queues that form at the off-ramp at Junction 33 spilling back onto the main highway. This creates a backwards propagating wave of congestion throughout links 6 that then disperses within link 5. This is only a short section of the site and

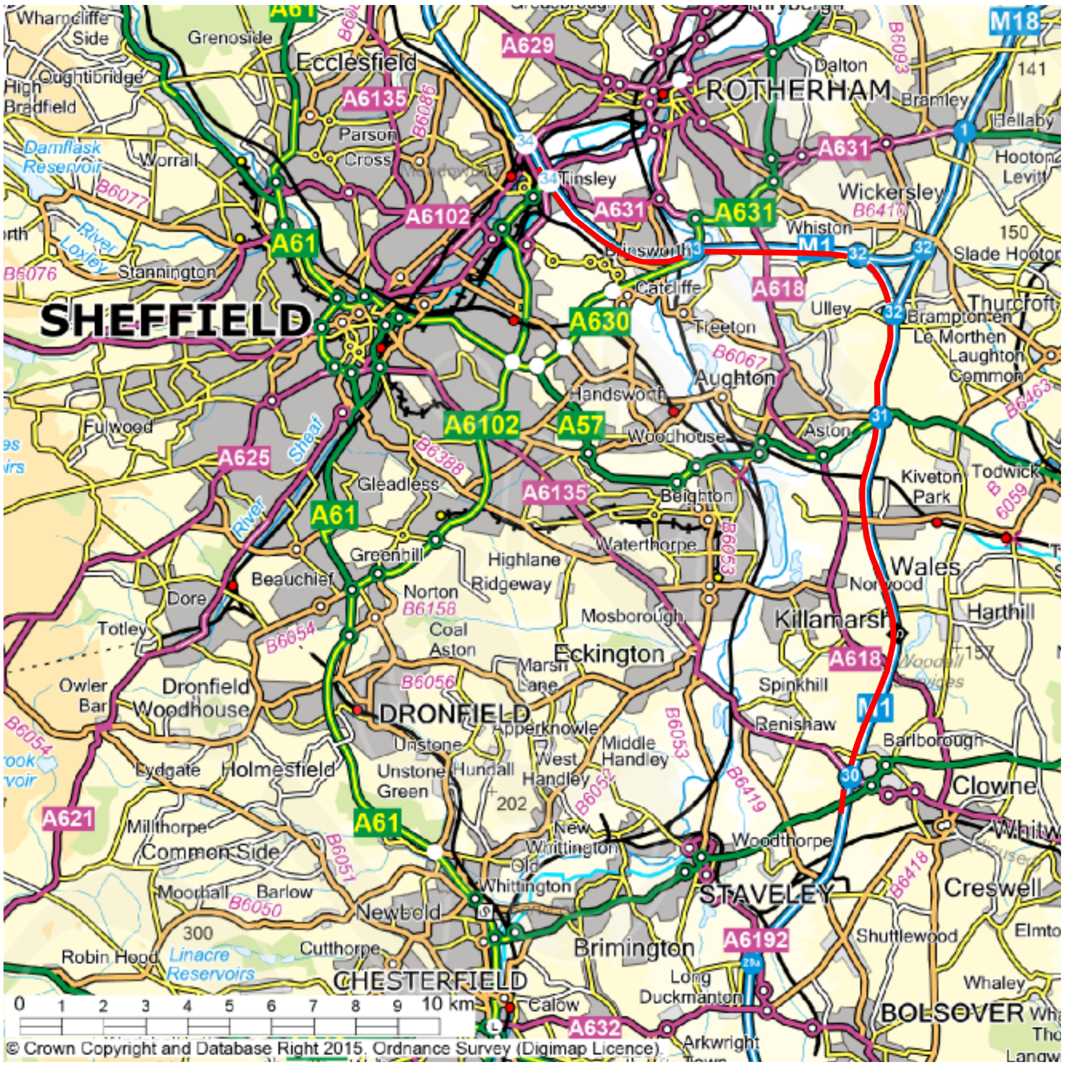


Figure 6.13: Map of the Sheffield site, modelled carriageway highlighted in red. © Crown Copyright and Database Right 2015, Ordnance Survey (Digimap Licence).

detectors are not available throughout the congestion region (see Figure 6.12 for detector locations). The reason for making the model as long as it is, with large regions of free flowing traffic upstream and downstream of the congestion region, is twofold. Firstly, the congestion is not input into the model via a congested origin and the same applies to the destinations. Although the latter cannot be totally avoided as the congestion in this area is caused by a congested off-ramp. This makes the BC applied at Junction 33 critical. Secondly, a long model site provides a test for the proposed schemes at a network level. Only small regions inter-dispersed in free flowing traffic may experience congestion. The optimisation techniques have to be able to find and replicate the congestion at these small resolutions within a large dataset.

6.4.4 Manchester large scale network

The model of Manchester incorporates both directions of the M60 ring road. Also included are the motorways that connect to the orbital, the model site includes the M56, M61, M62 and M602 (see Figure 6.14). The M67 is not included as data leading up to the merge of the motorway are unavailable. This problem is also compounded by the junction of this motorway to the M60 being a roundabout that also incorporates roads to the surrounding urban network. These roads are not covered by the MIDAS data. Each of the motorways connecting the M60 is modelled only for a short area. Except for the M56 which is modelled for a longer stretch to extend it past the access to Manchester Airport. Also included in the model is a section of the A5103, no data is supplied to the model for this road as it

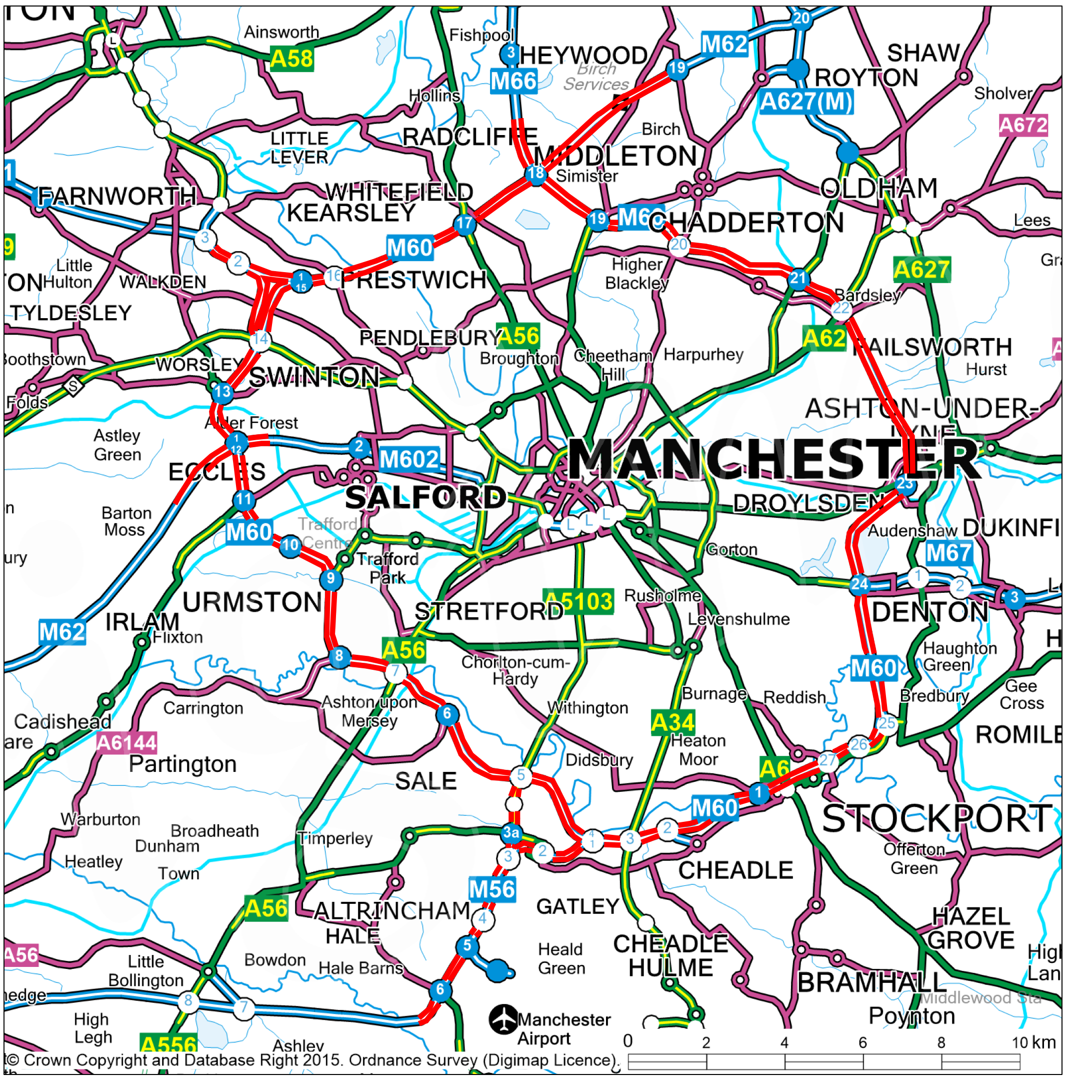


Figure 6.14: Simplified map of the Manchester region with the modelled highways shown in red. © Crown Copyright and Database Right 2015, Ordnance Survey (Digimap Licence).

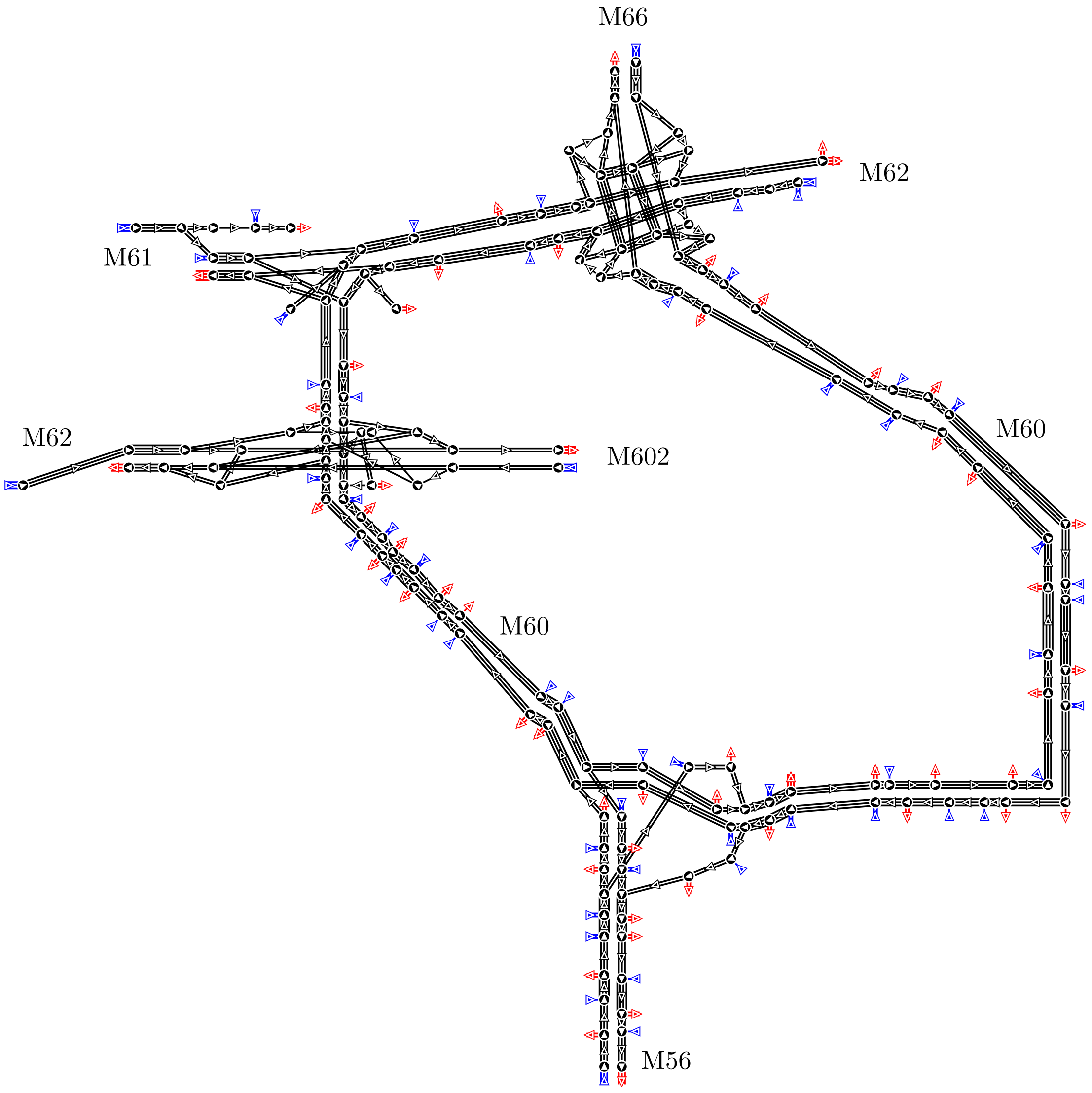


Figure 6.15: Schematic of the Manchester model.

is not part of MIDAS. This A-Road is included as it is the optimal route (shortest and quickest) between Junction 5 of the M60 and Junction 3 of the M56, ideal for anti-clockwise traffic from the M60 to access the M56 and for M56 traffic accessing the clockwise M60. A simplified map of the region with the modelled highways highlighted is shown in Figure 6.14.

In total the site covers 186.92km of highways, and the model uses 192 links with 56 origins and 55 destinations. A schematic of the full model is shown in Figure 6.15 (not to scale). Table A.1 details the origins, destinations and links that compose the Manchester model. This model is split into sections following Algorithm 4.3. This decomposes the network into 39 linear sections as detailed in Table 6.1 and shown in Figures 6.18–6.23. The sections 0–8 are those that are included as starting locations to the algorithm as they are fed by major origins. These sections make up the bulk of the model length. The main part of the model is the M60, the anti-clockwise loop is covered by section 2 (Figure 6.19) and the clockwise loop is included in section 4 (Figure 6.20). The M60 is not a complete ring road in the fact that it is not possible to complete ‘laps’ of the M60 without leaving the main highway at the Simister roundabout (Figure 6.16), Junction 18, and using this roundabout to remain on the M60. This is reflected in the sectioning. Section 2 starts with the major origin O62, that provides flows from the M62. The section represents the M62 up to Junction 18 of the M60 where the road designation changes. The motorway continues uninterrupted (with on- and off-ramps to access the Simister Roundabout) as does the section. It follows the M60 around in the anti-clockwise direction ignoring origins, destinations and connecting motorways (including the M60 which was the designation of the road at the beginning of the section) until it returns to Junction 18. Again the road classification changes, this time to the M66, but the section continues until it reaches its end at D66. The opposite path to this makes up section 4. Sections 9–33 are those that started from saved diverges from previous walks along the network. Section 34 is a closed loop that represents the Simister roundabout and is detected at the final phase of the algorithm, when a check is performed to ensure all links have been traversed. The remaining sections, 35–38, are diverges from the roundabout. Details of the sections and their lengths are given in Table 6.1.

The Simister roundabout at Junction 18 of the M60 is signalled and provides an interesting modelling problem. Here the road is modelled exactly as it exists in the real world. This means that it is possible for traffic in the model to use wrong routes, such as using the roundabout to perform a U-turn or exiting a carriageway from the main motorway and rejoining the same carriageway without changing direction. A simplified model could be used where no interactions between the roundabout’s

Table 6.1: Sections of the Manchester model.

Section	Motorways	Origins	Destinations	Links	Length
0	A5103/M56	5	4	9	6.90
1	M61	2	0	4	3.15
2	M60 (+M62,M66)	19	18	49	62.50
3	M62/M602	1	1	5	6.55
4	M60 (+M62,M66)	20	20	51	62.27
5	M602	1	0	3	1.50
6	A5103/M56	4	5	9	7.00
7	M60	1	0	1	0.90
8	M56	1	1	3	3.45
9	M60/A5103	0	0	1	0.70
10	M61/A580	1	1	3	2.30
11	M61/M60	0	0	1	1.20
12	M60/R'bout	0	0	2	0.90
13	M60/M61	0	0	1	1.00
14	A580	0	1	1	0.80
15	M60/M602	0	0	2	0.90
16	M60	0	0	1	0.45
17	M60/A5103	0	0	1	1.30
18	M66/R'bout	0	0	2	0.90
19	M62/M60	0	0	2	0.90
20	M66/R'bout	0	0	2	0.90
21	M56	1	1	3	3.55
22	M62	0	1	3	1.80
23	M61	0	1	2	1.70
24	M60/R'bout	0	0	2	0.90
25	M602/M60	0	0	2	1.15
26	R'bout/M60	0	0	2	0.90
27	M60/M62	0	0	1	0.80
28	R'bout/M60	0	0	2	0.90
29	M62/M60	0	1	4	2.35
30	R'bout/M62	0	0	2	0.90
31	M60/M602	0	0	1	0.50
32	R'bout/M66	0	0	2	0.90
33	M602/M60	0	0	1	0.50
34	R'bout	0	0	8	1.80
35	R'bout	0	0	1	0.45
36	R'bout	0	0	1	0.45
37	R'bout	0	0	1	0.45
38	R'bout	0	0	1	0.45
	TOTAL	56	55	192	186.92

parts are considered as shown in Figure 6.16. This approach was not used as the attempt of this model is to try and represent the real-world system as closely as possible. Also a simplified representation of the roundabout would not propagate congestion correctly. If an exit of the roundabout is blocked then other flows through the roundabout would be unaffected. Information about the signal control is not available and no modelling of the signalling is done. It is up to the calibration to select appropriate parameters that define an urban-style macroscopic FD that includes the influence of the signals.

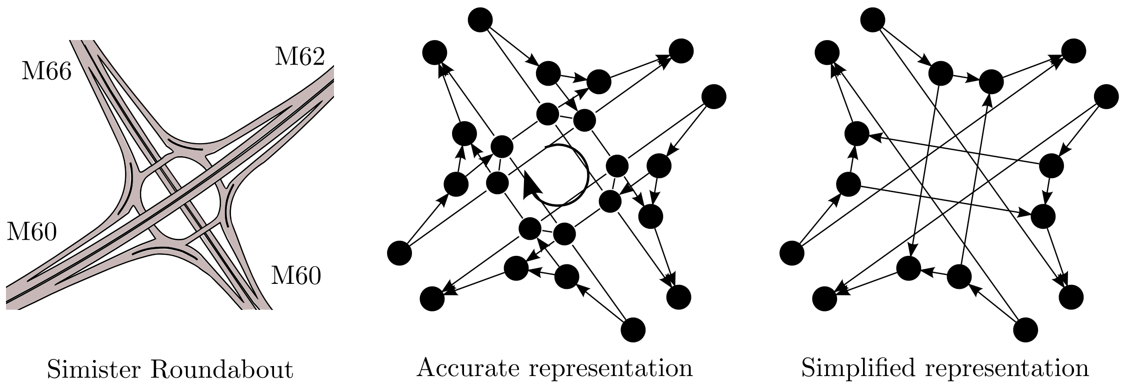


Figure 6.16: Modelling the Simister Junction.

Data are being used from May 2012, with calibration data sets being Monday 14th, 21st and 28th. The morning congestion pattern is going to be analysed and the model run between 06:00 and 09:30. Congestion is typically seen in various parts of the network during these times. A major congestion wave happens between the merge of the M61 and the M62 with shock-waves seen along the joining stretch of the anti-clockwise M60. Further congestion is seen at small localised regions along the M60. Another large congested region occurs on the clockwise M60 on the southern side from Junction 27 leading up to the diverge for the M56, Junction 4. This congestion causes some speed drops along the start of the southbound M56 that quickly recovers.

The available data do not allow for a full description of the traffic flows throughout the network to be determined. The two problem areas are the flows through Simister roundabout and the unavailable data for the short section of the A5103 between the M56 and M60. For the A5103 no information is available to be supplied as the major origin of the southbound carriageway and for the destination of the northbound carriageway as shown in Figure 6.17. The flows are known at junction of the M56 and A5103, this means that there exists a junction on the A5103 where the origin and destination data needs to be assumed. For the northbound carriageway it is assumed that the origin provides 1200veh/hr and the destination receives a constant 800veh/hr. This means the number of vehicles increases through the junction to ensure that flows remain above zero regardless of the prevailing traffic conditions. The flow from the northbound A5103 to the M60 is known. This provides enough information for the destination flow to be estimated by balancing node flows. For the southbound carriageway the destination is set receive a constant flow of 1200veh/hr and the unknown origin provides 800veh/hr. The net result of this is a reduction of vehicles in the carriageway. When balancing is used to calculate flows from the major origin are therefore guaranteed to be positive and non-zero, as again data is available for the flows provided to the A5103 from the M60. The major origin is assumed to experience no congestion and the flow is assumed to enter the network at a constant speed of 110km/hr.

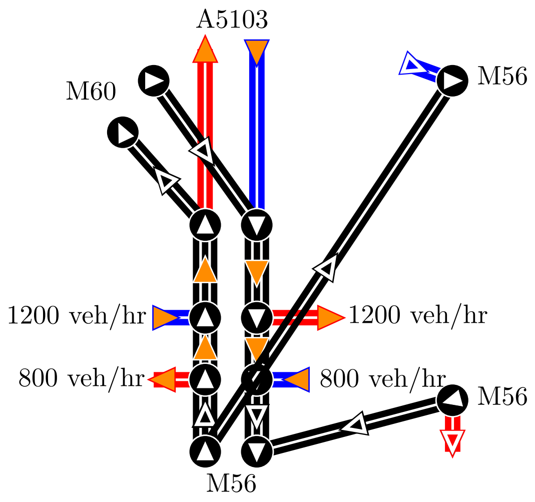


Figure 6.17: Schematic of Manchester network around A5103, links without data shown with orange filled triangles.

At Simister data are available at various points throughout the junction, but not enough locations are available to calculate the flows using (6.4). As there are no origins or destinations at this junction the problem is in determining how to split the known flow as it cannot be created or destroyed. By assuming the turning rates at a couple of the nodes then the flows throughout the rest of the junction can be estimated. The assumed turning rates are then adjusted so that the flow up- and down-stream of the junction on every carriageway matches the known data.

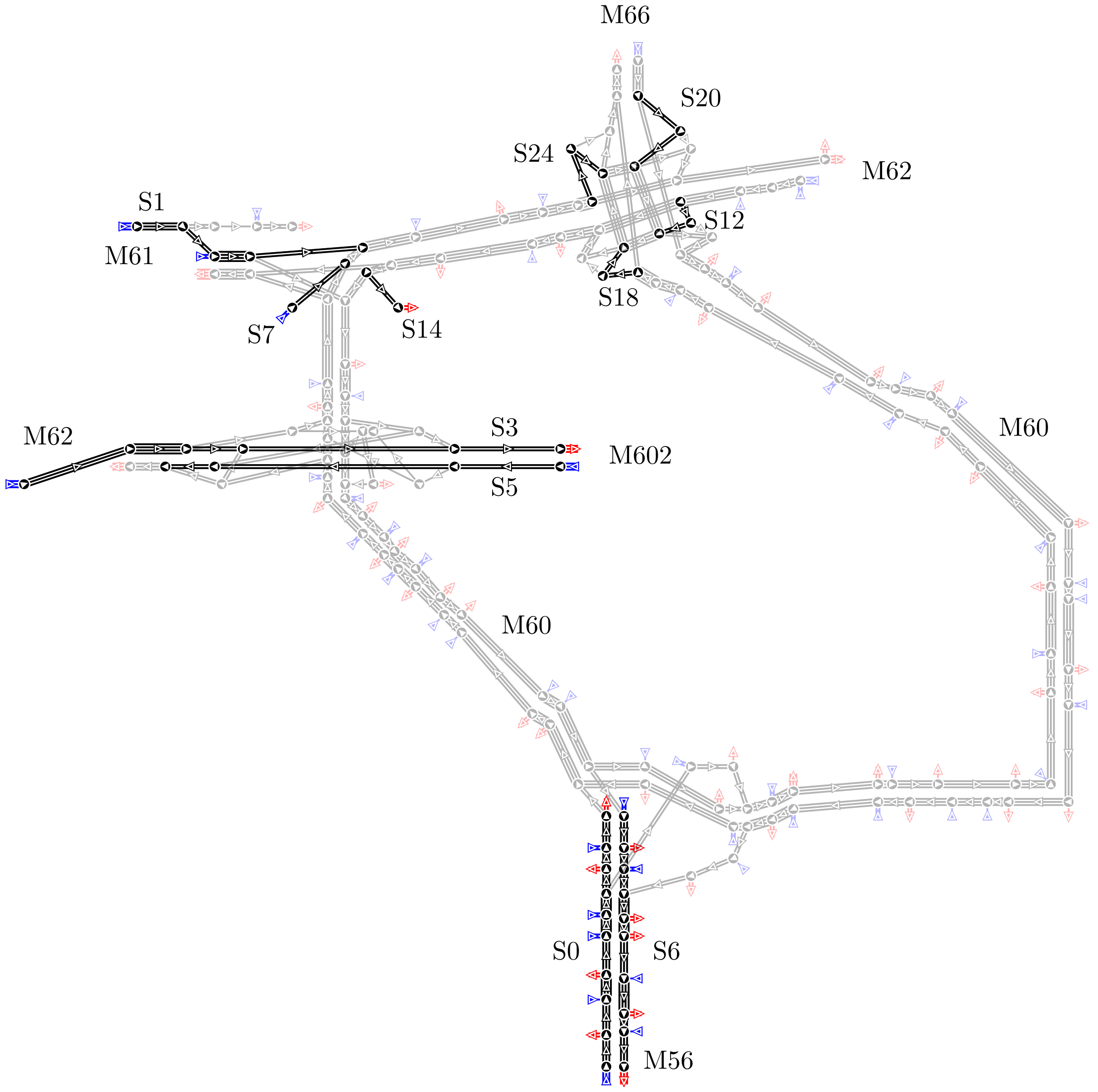


Figure 6.18: Sections 0, 1, 3, 5–7, 14, 18, 20 and 24 of the Manchester network.

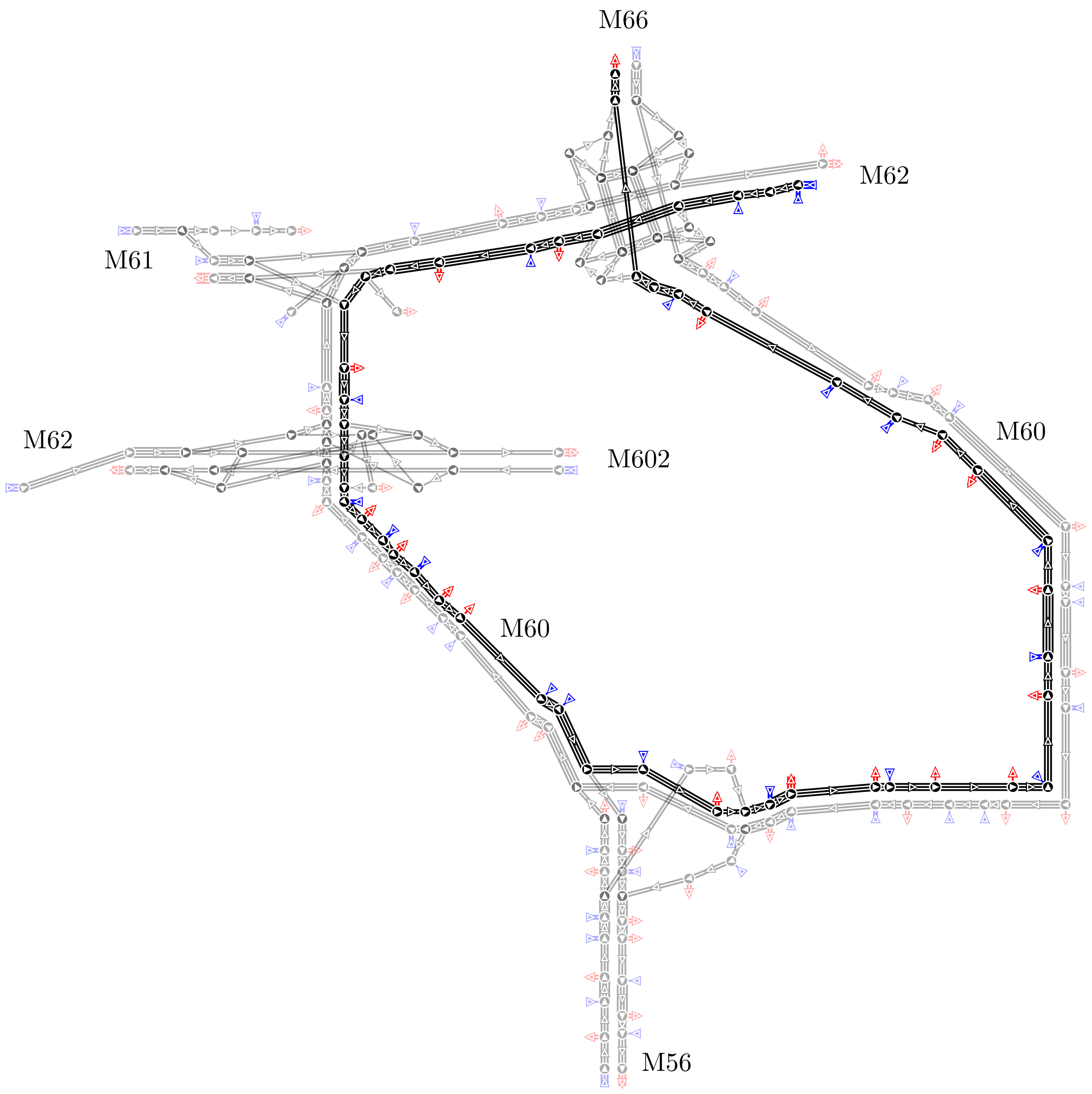


Figure 6.19: Section 2 of the Manchester network.

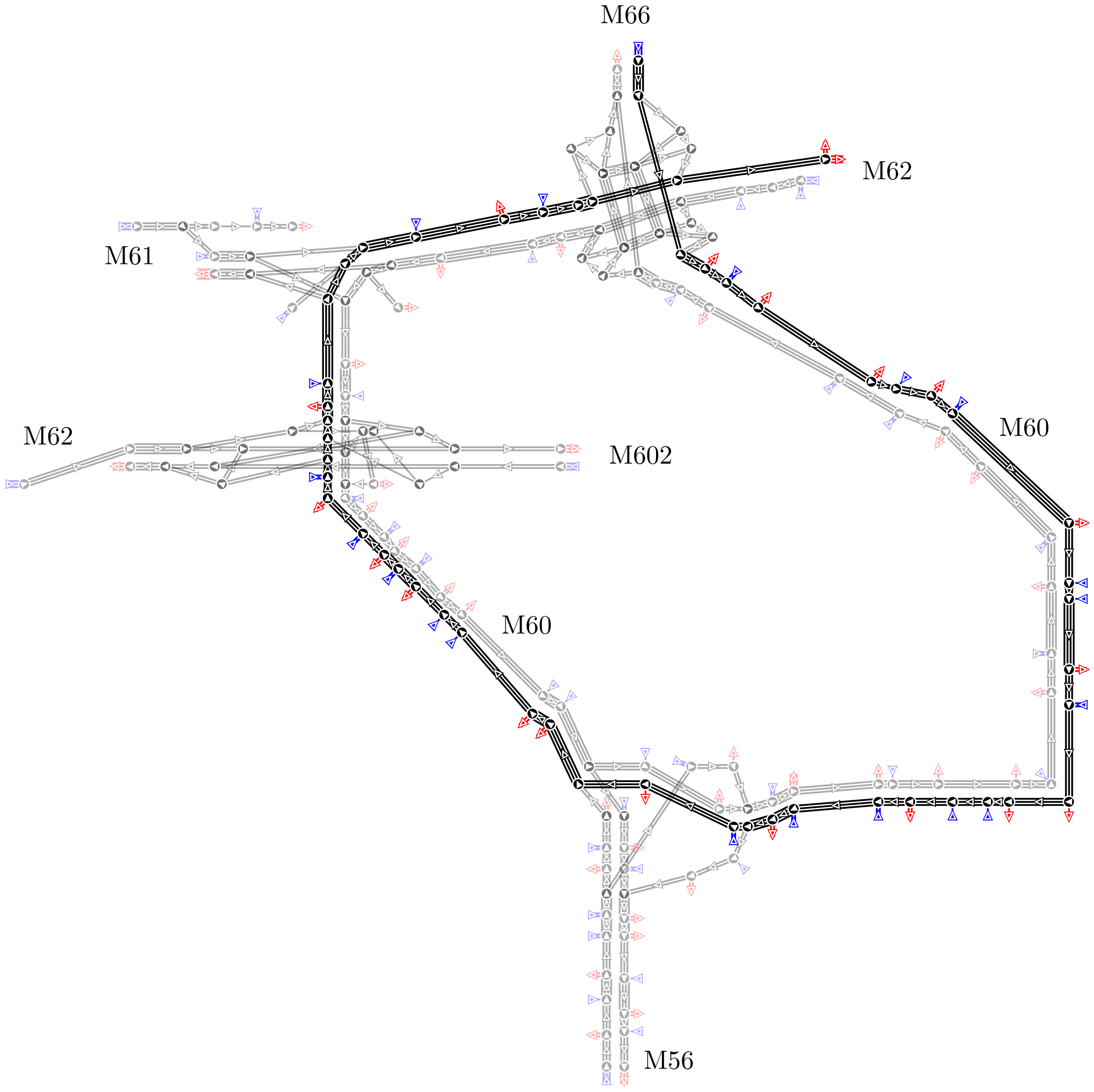


Figure 6.20: Section 4 of the Manchester network.

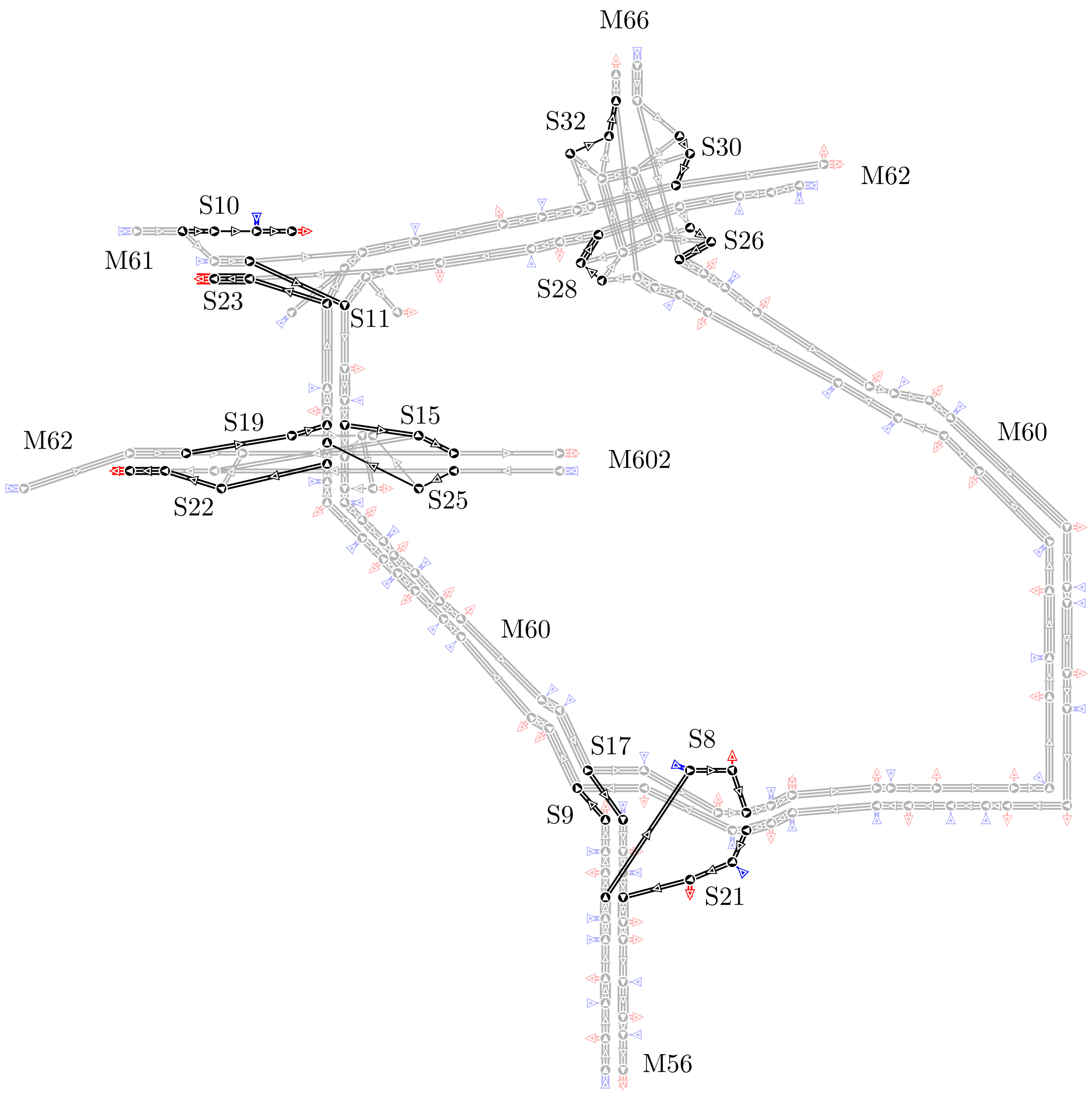


Figure 6.21: Sections 8–11, 15, 17, 19, 21–23, 25, 26, 28, 30 and 32 of the Manchester network.

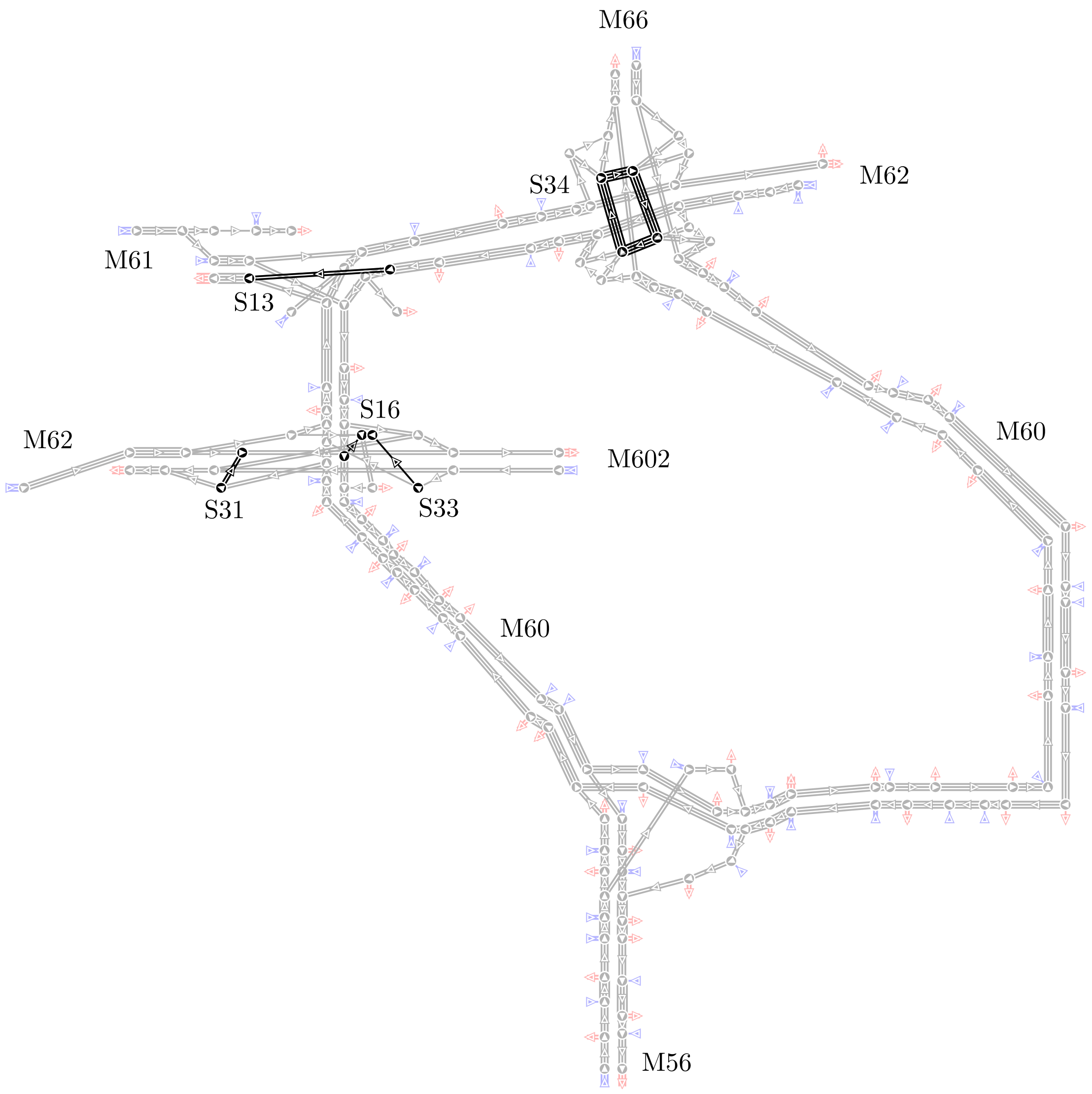


Figure 6.22: Sections 13, 16, 31, 33 and 34 of the Manchester network.

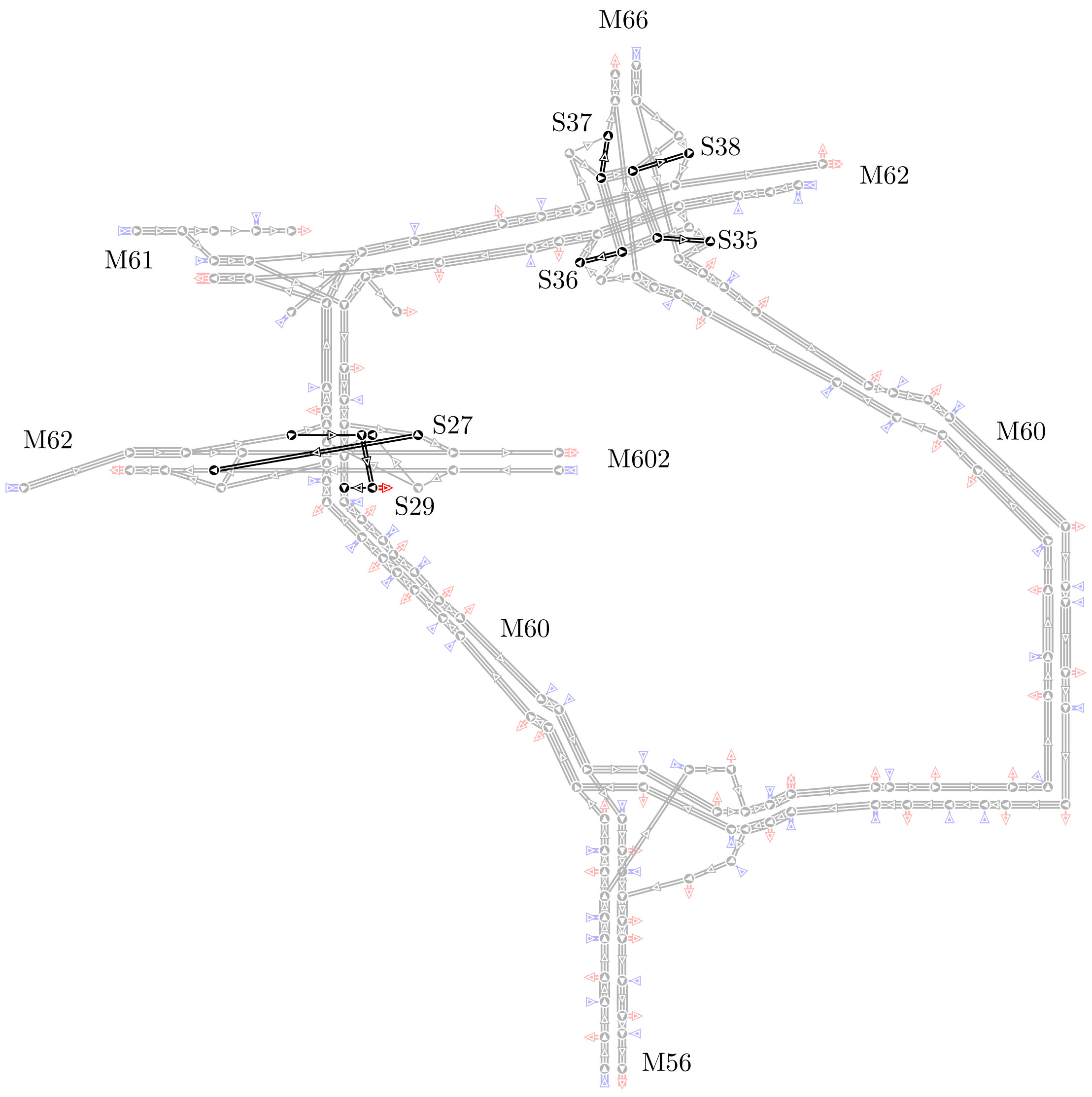


Figure 6.23: Sections 27, 29 and 35–38 of the Manchester network.

Chapter 7

Software

7.1 Calibration Software

The main focus of this chapter is the development of software for the optimisation problem discussed in Chapter 4, with real data processed as discussed in Chapter 6 using the algorithms described in Chapter 5. The algorithms described in Chapter 5 are included in a C library.

To ensure the developed software is expandable the traffic models are not embedded within the optimisation code. They are external executables that must be invoked when required, i.e. for an objective function evaluation. This is shown in Figure 7.1, where the model filled in grey is an external program. In order to run the optimisation algorithm for different traffic models a simple substitution of the external application is sufficient if routines exist for communication, i.e. the blocks 'generate model inputs' and 'read model outputs'.

The developed calibration software is controlled by a series of text files. A sample set of configuration files is given in Appendix B. The first input file `config` controls the library, setting the parameter values for the selected optimization algorithm. The file also defines the limits on the search domain. Where applicable, further input files are supplied for each solver class defining specific attributes. This approach means that calibrating a model using different optimisation algorithms is easy and does not require re-compilation.

The model files are also given as input to the developed system, the software extracts the required information to appropriately set up the problem. Extraction of the relevant data from the MIDAS data files requires a mapping between the loop detectors and model location and this is provided in the `.mna` file. The routines described in Chapter 6 allow for the automatic estimation of flows at points through out the network for missing data. If insufficient information is supplied for these estimations then the user can define custom measurements at the end of the `.mna`

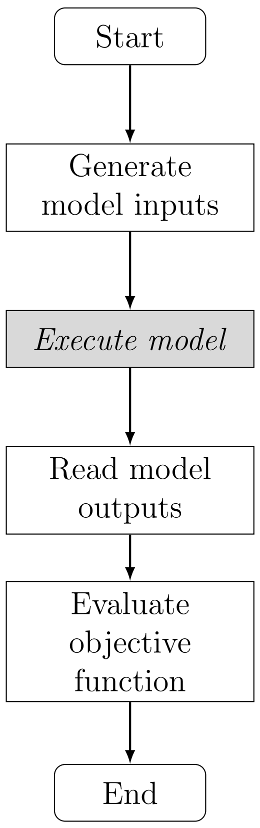


Figure 7.1: Model-optimisation algorithm integration, objective function evaluation.

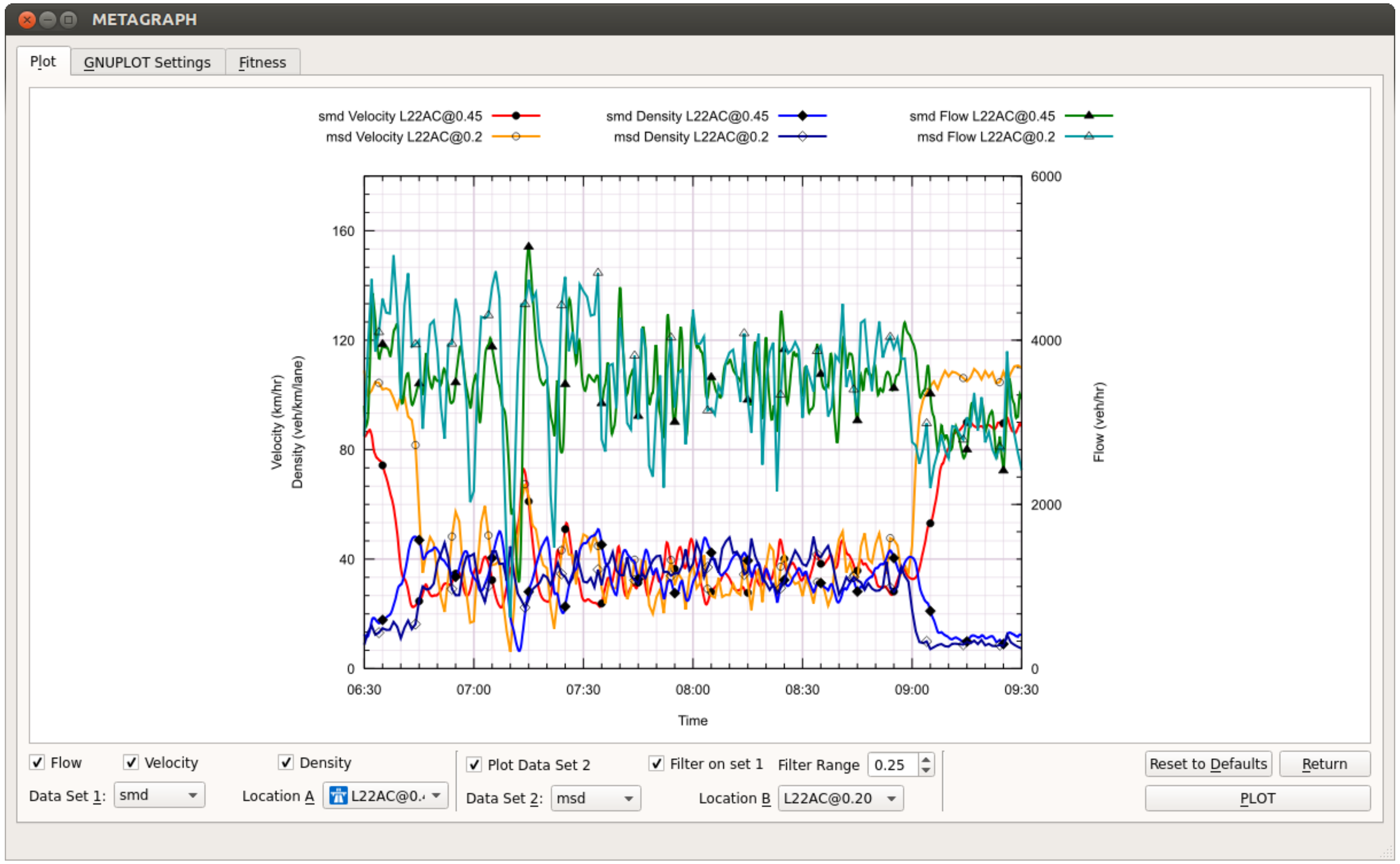
file. By setting up the calibration in this manner the underlying traffic model and sites can easily be changed.

7.2 Analysis Software

The objective function (4.17) gives a measure of a model's accuracy. However, it only provides an overview and does not provide information about the models performance at specific locations. Discussion and analysis of the results usually entails a more in depth approach, such as plotting time profiles of the model and measured data at detector locations. Hence, there is a need for a software tool to extract the relevant data and allow for easy comparison of results in detail.

For ease of use and to allow for different models to be analysed according to their specific needs a graphical user interface (GUI) was developed. With a GUI a user would be able to select the points of interest for the specific site and data under investigation. The GUI was developed in C++ using the Qt library [111], allowing for direct integration of source code from the calibration software.

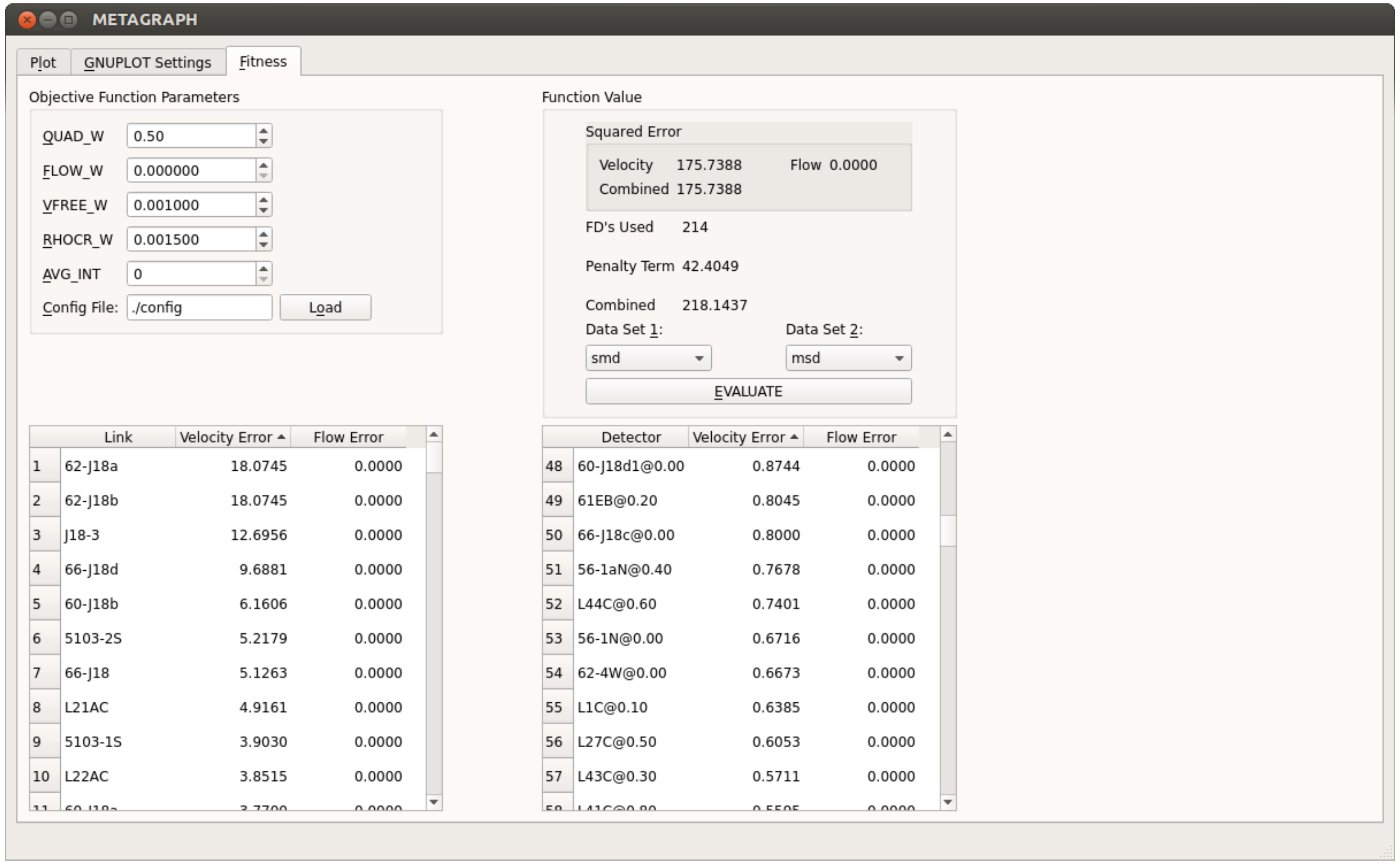
The application uses the same files as used in the model and calibration problem. For a full run down of the software's capabilities and a detailed user guide consult Appendix C. The application has two main modes. 'Data Plot', where the user can



✓ Flow ✓ Velocity ✓ Density ✓ Plot Data Set 2 ✓ Filter on set 1 Filter Range 0.25

Data Set 1: smd Location A L22AC@0.4 Data Set 2: msd Location B L22AC@0.20

Figure 7.2: The main data plot window.



Objective Function Parameters

QUAD_W	0.50
FLOW_W	0.000000
VFREE_W	0.001000
RHOCR_W	0.001500
AVG_INT	0
Config File:	./config

Function Value

Squared Error

Velocity 175.7388 Flow 0.0000
Combined 175.7388

FD's Used 214

Penalty Term 42.4049

Combined 218.1437

Data Set 1: smd

Data Set 2: msd

	Link	Velocity Error	Flow Error
1	62-J18a	18.0745	0.0000
2	62-J18b	18.0745	0.0000
3	J18-3	12.6956	0.0000
4	66-J18d	9.6881	0.0000
5	60-J18b	6.1606	0.0000
6	5103-2S	5.2179	0.0000
7	66-J18	5.1263	0.0000
8	L21AC	4.9161	0.0000
9	5103-1S	3.9030	0.0000
10	L22AC	3.8515	0.0000

	Detector	Velocity Error	Flow Error
48	60-J18d1@0.00	0.8744	0.0000
49	61EB@0.20	0.8045	0.0000
50	66-J18c@0.00	0.8000	0.0000
51	56-1aN@0.40	0.7678	0.0000
52	L44C@0.60	0.7401	0.0000
53	56-1N@0.00	0.6716	0.0000
54	62-4W@0.00	0.6673	0.0000
55	L1C@0.10	0.6385	0.0000
56	L27C@0.50	0.6053	0.0000
57	L43C@0.30	0.5711	0.0000

Figure 7.3: Fitness tab of METAGRAPH.

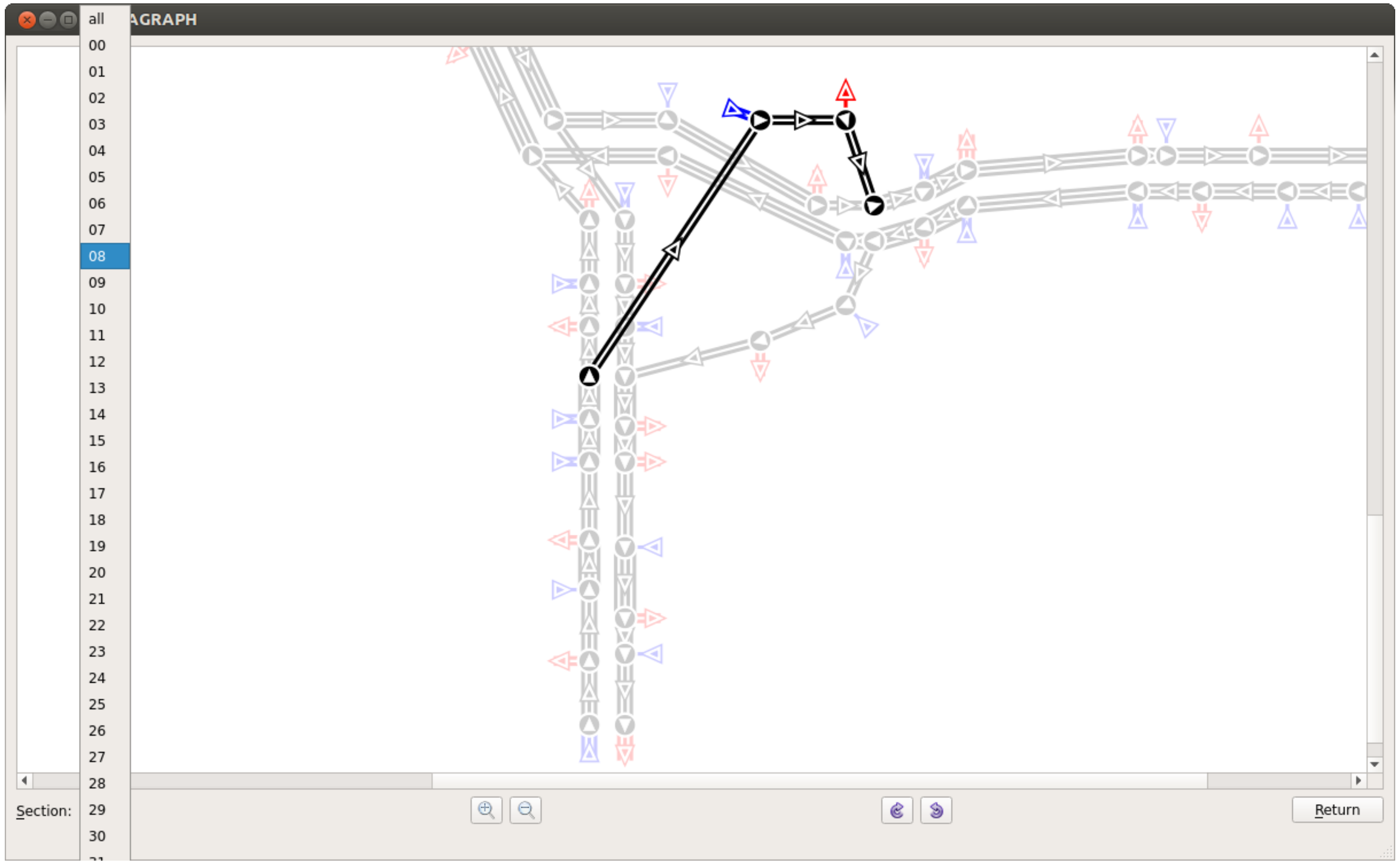


Figure 7.4: The Network plot, individual sections can be highlighted.

compare data at various locations and 'Network Diagram' where the model topology can be analysed. The 'Data Plot' aspect of the program is shown in Figure 7.2, here the user can compare time profiles of various datasets at their chosen location. The graphing is done using Gnuplot [112] and the second tab allows for the plot settings to be customised. The final tab details the objective function and gives a break down of it link by link and for every detector location as seen in Figure 7.3.

The second part of the program creates a plot of the network as seen in Figure 7.4. The schematic aids in the analysis of large models by providing the names of the links, origins, destinations and nodes as tool tips. The program also allows to select a single section as demonstrated in the figure. This provides an easy way to visualise the sectioning done by the algorithm (c.f Section 4.2.3). This developed software is independent of the METAGRAPH software packaged with the METANET simulator. It provides a more up to date interface and can be run on Windows and Linux systems, as well providing information about the objective function value.

7.3 Automatic Differentiation

Automatic differentiation (AD) is a way of augmenting a piece of source code or function within a program for the computation of derivatives [89]. The process works by calculating derivatives of the simple base operations and combining these together via the chain rule. Therefore, the complexity of calculating the gradient scales proportionally to the original function calculation. AD is not numerical differentiation as it calculates the derivative from the operations that make up the function. It is not symbolic differentiation either as the derivatives are calculated piece by piece. As such, function expansion is avoided when programs run through loops that involve products. As the method works by subsequent applications of the chain rule it calculates accurate derivatives to the precision of the machine performing the calculation.

The main premise of AD is to calculate the derivative in tandem with the code that evaluates the function value. Each line of code will contain basic operations for which a derivative can easily be calculated and stored. These are combined with previous results to obtain the full derivative. Suppose we take the following two equations as some hypothetical source code:

$$z = f(x, a) = x^2 + g(a) \tag{7.1}$$

$$y = h(x, z) = z^2 + xz \tag{7.2}$$

where the partial derivative $\frac{\partial y}{\partial x}$ needs to be obtained for $x = 2$, $g(a) = 3$. To do this symbolically one could substitute the expression for z into y and calculate the expression for $\frac{\partial y}{\partial x}$ which gives,

$$\frac{\partial y}{\partial x} = 4x^3 + 3x^2 + 4g(a)x + g(a). \tag{7.3}$$

AD works line by line so it would evaluate z and its partial derivative with respect to x,

$$z = 2^2 + 3 = 7 \tag{7.4}$$

$$\frac{\partial z}{\partial x} = 2x = 2 \times 2 = 4. \tag{7.5}$$

The second equation (7.2) uses the auxiliary variable z and is also a function of x so is more complicated. For the first term AD needs to make use of the chain rule, the second requires the product rule. As before AD will work alongside the function evaluation,

$$y = 7^2 + 2 \times 7 = 63 \tag{7.6}$$

$$\begin{aligned} \frac{\partial y}{\partial x} &= \frac{\partial z}{\partial x}\frac{\partial}{\partial z}z^2 + z\frac{\partial}{\partial x}x + x\frac{\partial}{\partial x}z \\ &= 2z\frac{\partial z}{\partial x} + z + x\frac{\partial z}{\partial x} \\ &= 2 \times 7 \times 4 + 7 + 2 \times 4 \\ &= 71. \end{aligned} \tag{7.7}$$

As the code already has z and $\frac{\partial z}{\partial x}$ calculated it just needs to use the stored values. If we compare what would be done analytically to calculate the value and its derivative (7.1)–(7.3) to what would be done by AD (7.4)–(7.7) it can be seen that function expansion has been avoided and the number of basic operations reduced.

There are two methods to implement AD, Source Code Transformation (SCT) and Operator Overloading (OO) [89]. In SCT the AD routine creates a new source code from the original. The new source code will add extra lines to calculate the derivative for each line of the original code. The new code will look similar to the example above, this is then be compiled into a new program that calculates derivatives as well as performing its original function. OO involves the use of a library included into the original source code. The independent and dependant variables are doubled up with one containing the variable value and the other information pertaining to the calculation. Each operation is then overloaded, this means that as well as performing the original operation other functions are called at the same time. For example for the multiplication operation the product rule will also be implemented. The advantage of SCT is that the compiler can create optimised code for the derivative calculation. However the process of creating the differentiated program is more involved than for OO. OO is simple to implement and the code remains readable and requires no major alterations although data structures need substituting to types that support OO. Due to OO compiler optimisation becomes more difficult a disadvantage compared with SCT [89].

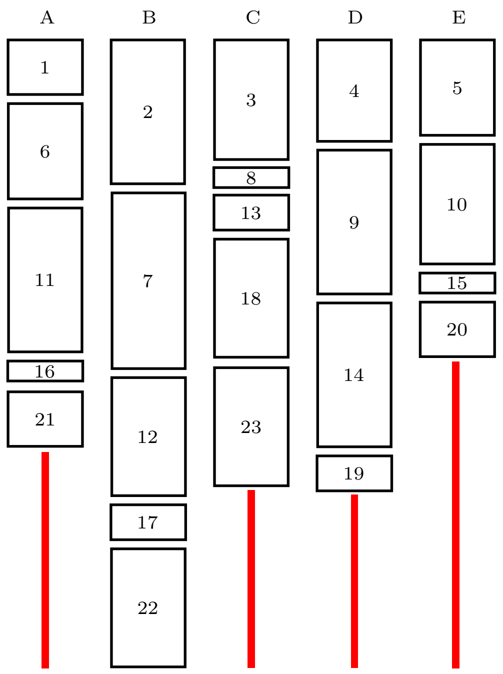
There are several tools available to implement AD but this work uses ADOL-C (Automatic Differentiation by OverLoading in C++) [113]. This AD package was selected due to the functions and support available. Creation of new source code is straightforward and comparable to the original. Using this library model variants were created that calculate the model Jacobian at the same time as the output. In this thesis AD is used to obtain the gradients required for the RPROP optimisation algorithm. To keep the black-box architecture only the model Jacobian is obtained by AD and the objective function is not supplied to the AD algorithm. As discussed in Section 5.5 it is possible to apply the chain rule further and obtain the required gradients for the optimisation from the Jacobian. This final step is done analytically as the expression (eqn. (5.43)) is simple and the overhead for AD is unnecessary.

7.4 Parallel Computing

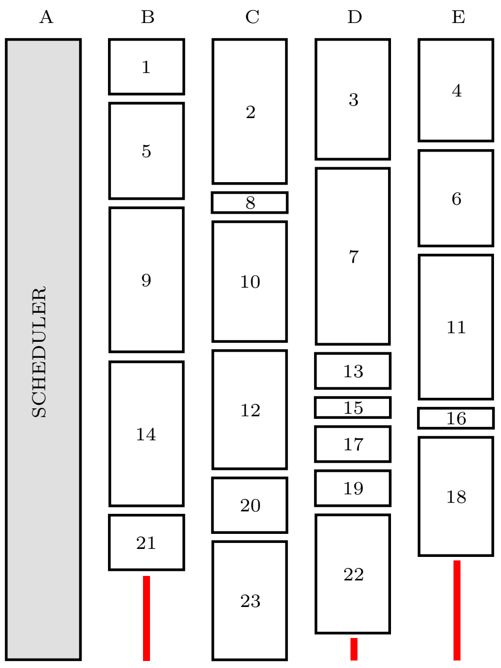
All of the algorithms under investigation have a series of solution vectors that are evaluated at every iteration. To minimise the total time taken to evaluate the parameter vectors simulations are executed in parallel. This means that the problem can be run on a supercomputer or make full use of a desktop system's multi-core processors.

There are various forms of parallel computing. For the problem at hand we are interested in task parallelism [114]. At each iteration we have a collection of parameter sets for which the objective function must be analysed. The function remains the same but with different input. Parallel computing introduces many potential problems and the data must be accessed safely and structures transferred to the various threads [115]. To communicate between the sub-threads of the program the Message Passing Interface (MPI) [116] is used. This allows for the developed software to be portable and run on many systems with different computing architectures. MPI is well suited to this problem as the required data transfer is small. The information that needs to be transferred between processes is the parameter vector to be analysed and the result of the objective function.

Since it is not possible to guarantee a parallel task's run time, a simple equal division of the tasks over the available processors is not necessarily the optimal method of managing this computation. If a parameter set provides an invalid model it could terminate the model's execution prematurely. A processor could therefore be



(a) Equal division of tasks.



(b) Tasks assigned by scheduler.

Figure 7.5: Task assignment methods, example with 23 variable length time tasks.

sitting idle whilst others are still running tasks as shown in Figure 7.5a. The figure shows 23 random length processes with the runtime represented by the height of the box distributed equally across 5 processors. The red bar indicates the amount of idle time processors A,C,D and E have to wait for B to complete its work. Another option is to dedicate one processor to act as a scheduler distributing the tasks amongst the remaining processors as shown in Figure 7.5b. As each processor returns the results to the scheduler it receives a new task to execute until all tasks have been completed. As seen the Figure 7.5 by assigning the same task by a scheduler the total wait time is reduced. Using a scheduler has the advantage of being more adaptable and scalable. The disadvantage is that a processor has been lost to managing the tasks and is no longer used directly in task execution. It was decided to dedicate a processor to distributing the parallel tasks as this provides the most scope for extension of the work to large problems.

7.5 Interprocess Communication

7.5.1 Introduction

As discussed earlier, the optimization requires data transfer between the algorithm and the model invoked (c.f Figure 7.1) which are separate executable programs. This is called Interprocess Communication (IPC). The communication between separate executables is important and can have a significant impact on runtime. The calibration software developed for this thesis is written in the *C* language, and as such a variety of portable IPC methods are available through the Portable Operating System Interface (POSIX) [117]. There are three main methods for IPC:

File system processes can communicate by reading and writing data to the disk.

Message passing processes sending and receiving messages between themselves through pipes [115].

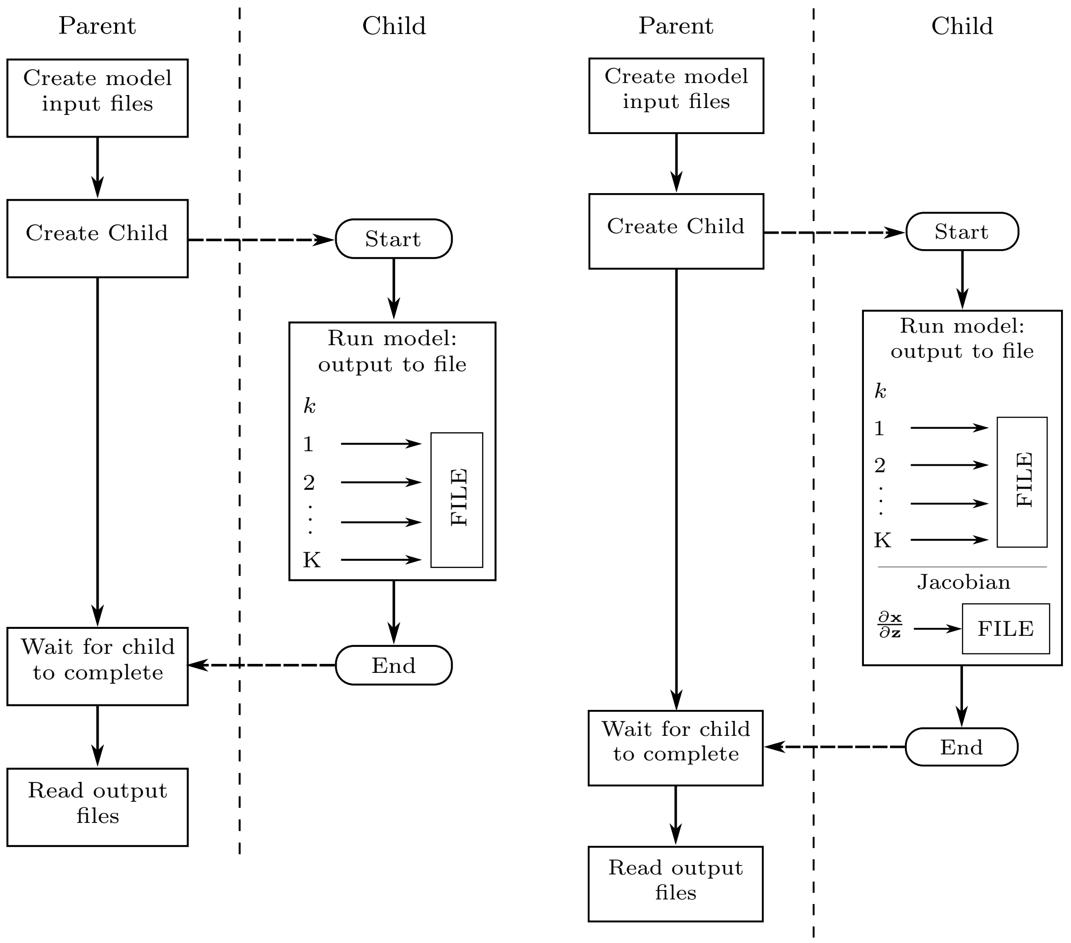
Shared memory both processes can access a region of memory, making the information available to both [115].

In this section the discussion revolves around calibrations with and without gradient information. When a gradient is not required the amount of data to be transferred is considerably less. When gradient information is included the size of the model Jacobian and its structure have to be passed from the simulator to the optimisation algorithm. For this reason the two scenarios are treated independently.

7.5.2 File System

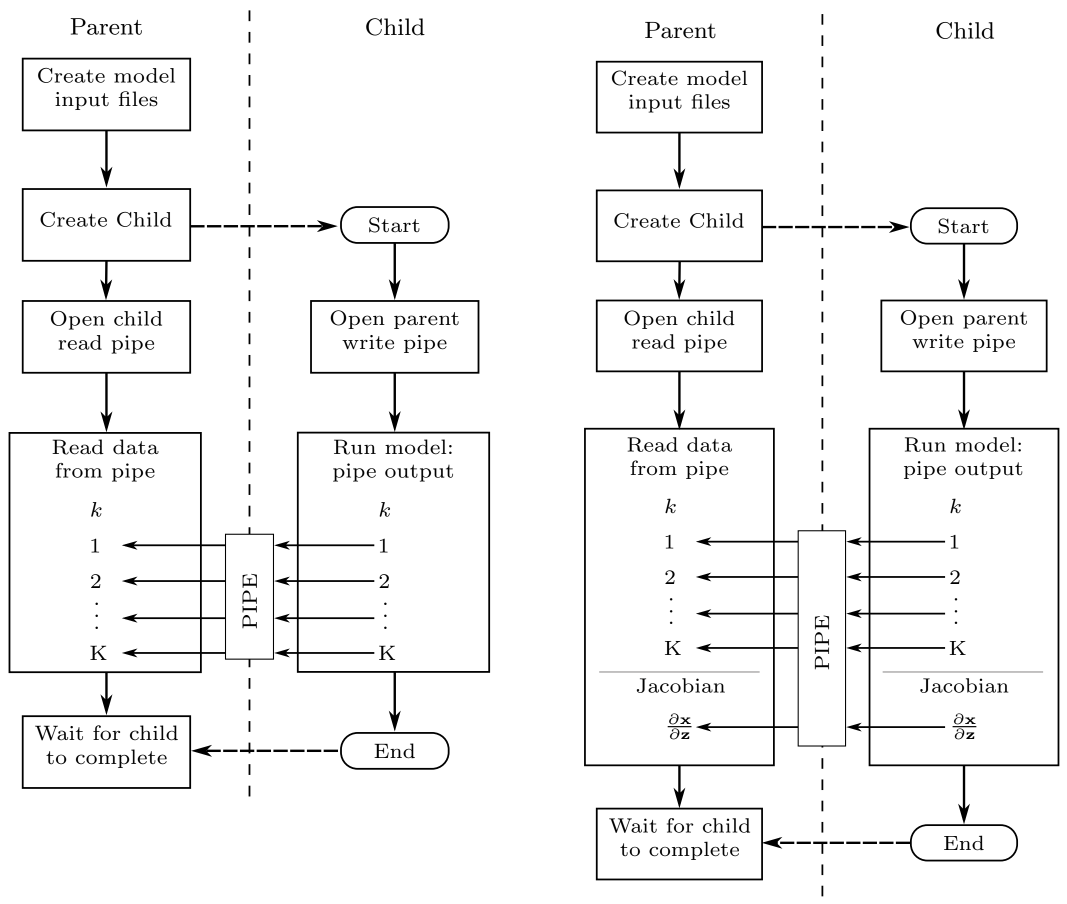
The simplest approach is to use the file system as shown in Figure 7.6. When the optimisation algorithm (the parent process) needs to evaluate a parameter vector it creates the appropriate input file for the model. A child process is spawned that then runs the external simulator executable. The simulator by design receives its input from a set of files and saves its output to file. In doing so the program reduces its memory requirement by only storing the current state and the calculation of the next state in memory. Once a state has been calculated for time step $k+1$, the state at time k is not required for further calculations and the result is saved to file and the memory address storing it is overwritten with the new value after the next iteration. By using the file system no alterations to the model code are required. Routines are required for the optimisation algorithm to parse the output files and load them into memory.

Using the file system for IPC is a straightforward approach but time consuming. At each iteration the simulator opens the output file, saves the current state and then closes the file. Once the child has completed and the parent can then access those files safely. At which point all of the data are reloaded into the memory. This results in a lot of writing and reading of data to disk, which is a very slow operation [115]. The creation of the simulation input files is simple and only small files need to be generated. The data structures however are large and grow with model size. Therefore reading and writing to disk of this data should be avoided.



(a) Without gradient calculation. (b) With gradient calculation.

Figure 7.6: File IPC between optimisation algorithm and model.



(a) Without gradient calculation. (b) With gradient calculation.

Figure 7.7: Pipe IPC between optimisation algorithm and model.

7.5.3 IPC based on Pipes

One possible method to avoid file writing operations is to use a pipe as shown in Figure 7.7. A pipe is a form of message passing that allows the output of a program to be redirected to the input of another. This way the writing and reading of data to disk is avoided.

Each model state $\mathbf{x}(k)$ is written to the pipe as it is calculated and the communication happens in a serial fashion during the runtime of the program, rather than as a block at the end of the model execution. The gradient calculation through ADOL-C is done at the end of the model execution after all states have been calculated. This data is still transferred as a block at the end of the simulator, see Figure 7.7.

7.5.4 IPC using Shared Memory

Even with pipes the output is still written to a file-descriptor and read from one, so read-write operations have not been bypassed entirely. This can be achieved by using shared memory accessible to both programs. The largest data structure to be communicated is the model Jacobian and as mentioned previously for minimal memory storage the simulator does not keep copies of the state $\mathbf{x}(k)$ after it has been outputted, it is overwritten in the calculation of the next state $\mathbf{x}(k+1)$. This is why pipe communication is retained for the simulator state and focus is initially on the Jacobian transfer. The Jacobian exists as a memory block in both sub-processes (optimisation algorithm and simulator) so it can easily be stored in a SHM block as shown in Figure 7.8. Here a SHM segment is created by the parent, the child runs as before communicating the simulator states via a pipe. The Jacobian however is calculated and stored in the SHM. No extra writes of these data are required and when the simulator finishes it is possible to safely accesses the data. This removes the requirement for adding any locks or semaphores for safe access [115] of the SHM segment, the parent does not access the memory while the child is active.

Even though the model state is not persisted by the simulator a SHM region could

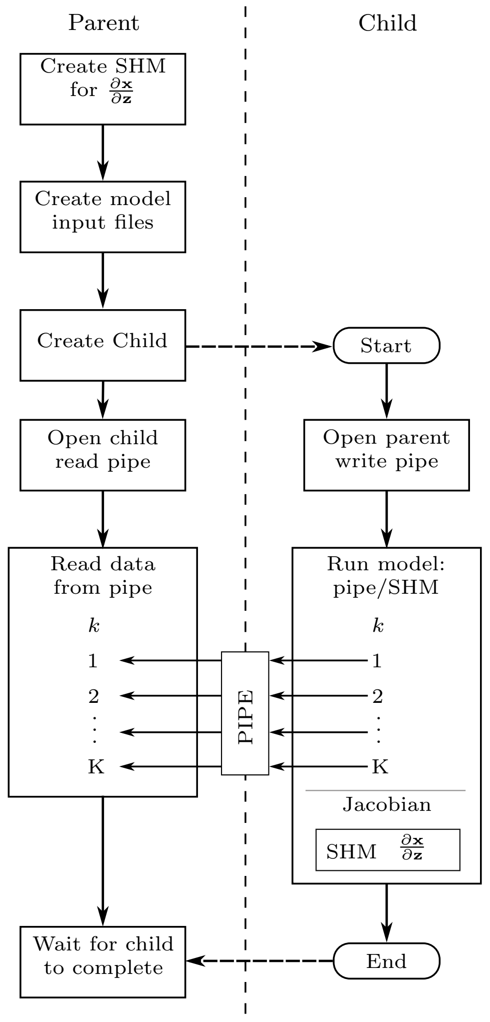
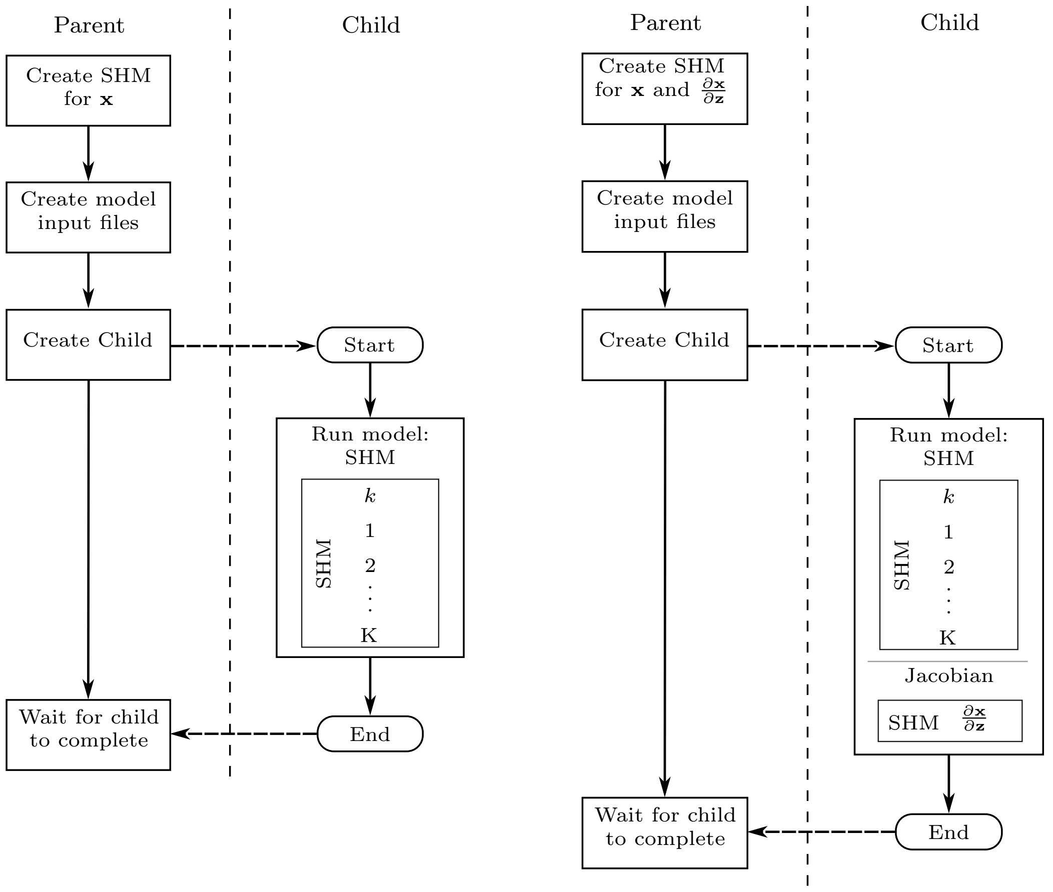


Figure 7.8: Schematic of objective function evaluation using Pipe/SHM IPC w/ gradient.

still be used to communicate this data to the optimisation algorithm by saving the current state $\mathbf{x}(k)$ to the shared region at the end of each step as illustrated in Figure 7.9. The overall memory used by the optimisation algorithm-simulator system remains the same because the optimisation algorithm already required this memory space and now is sharing it with the simulator.


Parent
Child
Create SHM for $\mathbf{x}$
Create model input files
Create Child
Start
Run model: SHM
SHM
k
1
2
K
Wait for child to complete
End
Create SHM for $\mathbf{x}$ and $\frac{\partial \mathbf{x}}{\partial \mathbf{z}}$
Jacobian
SHM $\frac{\partial \mathbf{x}}{\partial \mathbf{z}}$

(a) Without gradient calculation. (b) With gradient calculation.

Figure 7.9: SHM IPC between optimisation algorithm and model.

7.5.5 Runtime

All of the timings obtained in this chapter were obtained using an Intel Core i5-2400 @ 3.10 GHz processor with four cores. Figure 7.10 shows the average time taken for an objective function evaluation over different size sites. The model size is the total length of all links in the sites discussed in 6.4. The shortest site at 8.8 km is Heathrow, then 23.1 km and 186.9 km for Sheffield and Manchester, respectively. The effect of removing disk operations can be seen by comparing the red and black lines for an objective function only using squared error terms and the pink and green lines for an objective function with gradient calculation. This leads to a speed up of around 30% and 50% independent of the model size for the optimisation with and without gradient calculation, respectively, as shown in Tables 7.2 and 7.1. For the largest site this results to a reduction of the runtime from 3.2 CPU-seconds to 1.6 CPU-seconds when gradients are not calculated. When they are times are reduced from 590.4 CPU-seconds to 416.1 CPU-seconds. These results also show the overhead of the gradient calculation and how it scales with model size, a larger model requires more processing effort and scales by a power law.

Table 7.1: Average run times from 100 trials for various IPC methods using objective function without gradient calculation.

IPC method	Site size (km)	Avg. run time (CPU-seconds)	Standard deviation (CPU-seconds)	Speed up (CPU-seconds)	Speed up (%)
File					
	8.8	0.2088	0.0018	–	0
	23.1	0.5812	0.0059	–	0
	186.9	3.2008	0.5483	–	0
Pipe					
	8.8	0.0930	0.0017	0.1158	55.46
	23.1	0.2645	0.0099	0.3167	54.49
	186.9	1.6414	0.3192	1.5594	48.72
SHM					
	8.8	0.0187	0.0020	0.1901	91.04
	23.1	0.0478	0.0026	0.5334	91.77
	186.9	0.3133	0.0034	2.8875	90.21

Further reductions in runtime are achieved by use of the shared memory for the model Jacobian when gradients are calculated (orange line). Compared to using the file system this provides a speed up of 64 to 94%. The speed up now depends on the model size. This is because the model Jacobian is a large data structure that scales by a power law. Recall the Jacobian's structure (5.37) which is repeated here for convenience;

$$\mathbf{J} = \frac{\partial \mathbf{x}}{\partial \mathbf{z}} = \begin{bmatrix} \frac{\partial \mathbf{x}(1)}{\partial \mathbf{z}_1} & \frac{\partial \mathbf{x}(1)}{\partial \mathbf{z}_2} & \cdots & \frac{\partial \mathbf{x}(1)}{\partial \mathbf{z}_A} \\ \frac{\partial \mathbf{x}(2)}{\partial \mathbf{z}_1} & \frac{\partial \mathbf{x}(2)}{\partial \mathbf{z}_2} & \cdots & \frac{\partial \mathbf{x}(2)}{\partial \mathbf{z}_A} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{x}(K)}{\partial \mathbf{z}_1} & \frac{\partial \mathbf{x}(K)}{\partial \mathbf{z}_2} & \cdots & \frac{\partial \mathbf{x}(K)}{\partial \mathbf{z}_A} \end{bmatrix} \tag{7.8}$$

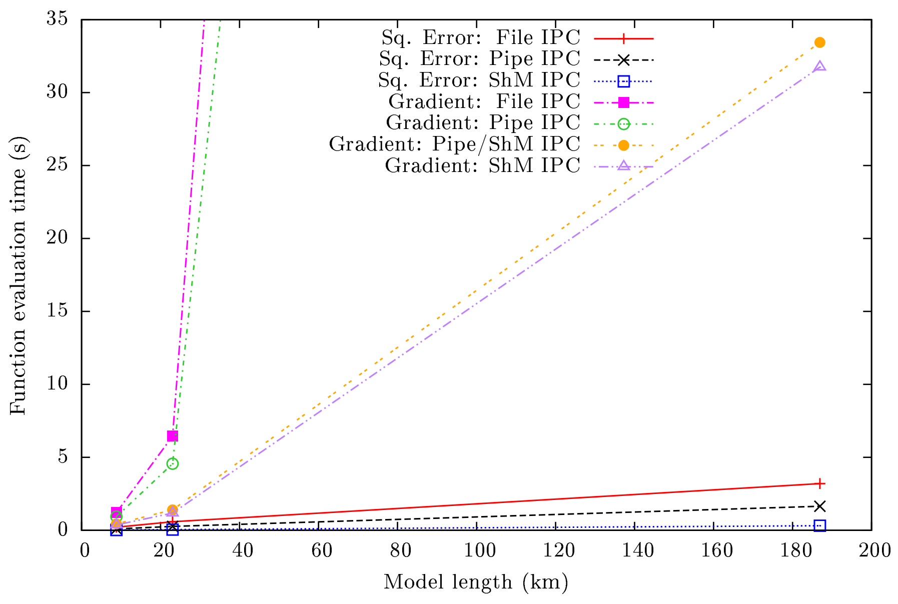
Each extra link increases the state vector $\mathbf{x}$, i.e. the number of rows rows of the matrix; it also adds new parameters to the traffic model increasing the size of $\mathbf{z}$, i.e extra columns. Therefore, as the model increases in size its Jacobian increases in size rapidly. The transfer of this large structure has a significant impact on the total runtime of the objective function. For the longest site tested the reduction of the

Table 7.2: Average run times from 100 trials for various IPC methods using objective function including gradient.

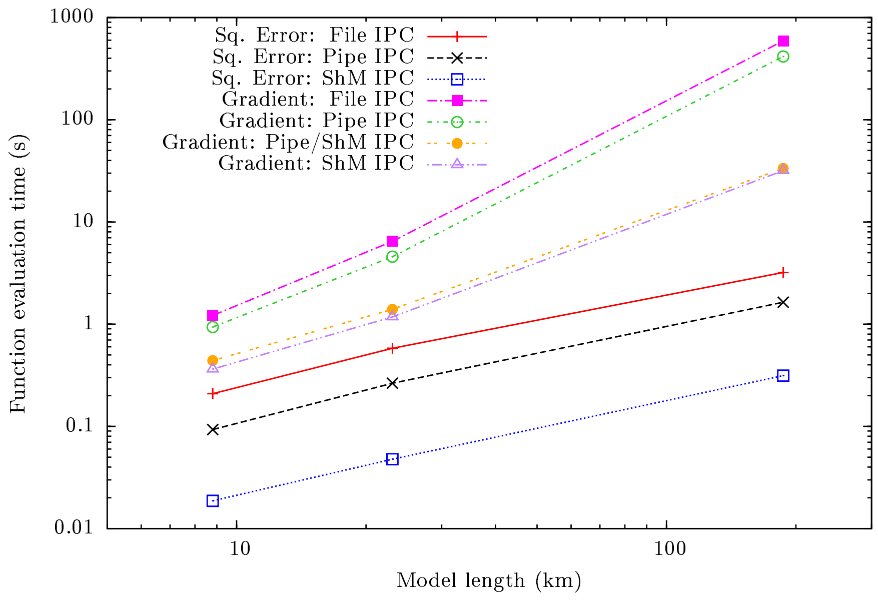
IPC method	Site size (km)	Avg. run time (CPU-seconds)	Standard deviation (CPU-seconds)	Speed up (CPU-seconds)	Speed up (%)
File					
	8.8	1.2188	0.2100	–	0
	23.1	6.4528	0.9199	–	0
	186.9	590.4040	56.4823	–	0
Pipe					
	8.8	0.9362	0.1979	0.2826	23.19
	23.1	4.5639	0.9366	1.8889	29.27
	186.9	416.1110	43.5811	174.2930	29.52
Pipe/SHM					
	8.8	0.4412	0.0045	0.7776	63.80
	23.1	1.3971	0.0144	5.0557	78.34
	186.9	33.4375	0.6799	556.9665	94.34
SHM					
	8.8	0.3636	0.0069	0.8552	70.17
	23.1	1.1763	0.0561	5.2765	81.77
	186.9	31.7439	0.4368	558.6601	94.62

average run time compared to simple file writing IPC is 558.6 CPU-seconds (95%). This means in the original case the runtime of the model including the calculation of the gradient takes less than 5% of the total runtime. Another benefit of using SHM is the reduction of the variability in the run times, for simple file IPC and pipe IPC the standard deviation in the measured runtime is significant (Table 7.2), 10–15% of the runtime, with SHM it is less than 2%.

The greatest speed up is achieved by using SHM for all data. The blue line in Figure 7.10 shows the objective function calculation without gradients and the violet line shows the result when the Jacobian is calculated. Without Jacobian matrix calculation the speed-up is constantly around 90% irrespective of the model size, and for the largest site the runtime is reduced from 3.2 CPU-seconds to 0.3 CPU-seconds. However using shared memory for the state vector in the gradient based objective function does not provide a significant reduction in the runtime over the pipe/SHM method. The runtime is reduced by 70–95% over using the file system, again depending on the model size. The improvements over the pipe/SHM architecture are small and diminish as the model size increases. This is because the costly operations (Jacobian transfer) have already been optimised. Furthermore, out of the time it takes to run the model with ADOL-C the proportion of time spent communicating the model output via pipe is small compared time taken to perform the Jacobian calculation and simulation. Due to the scaling of the Jacobian



(a)

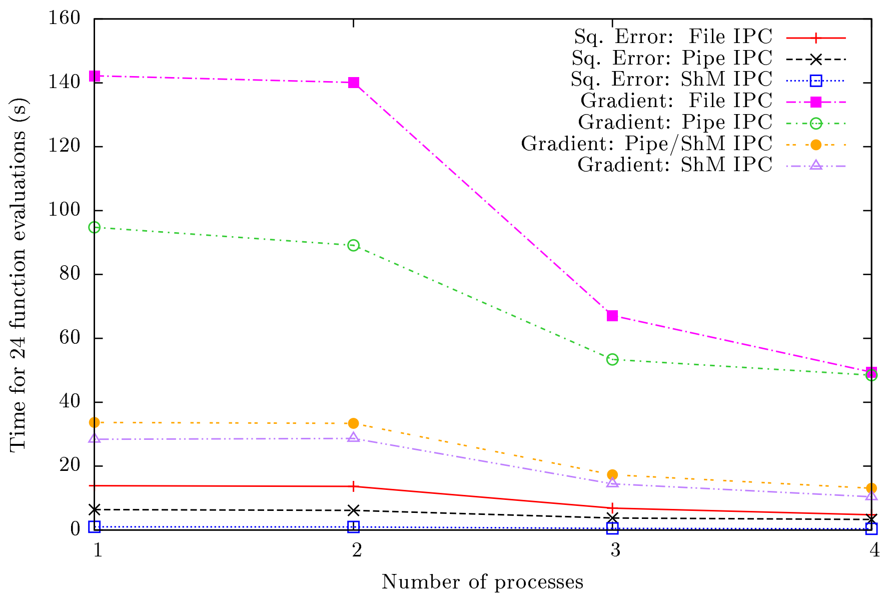


(b)

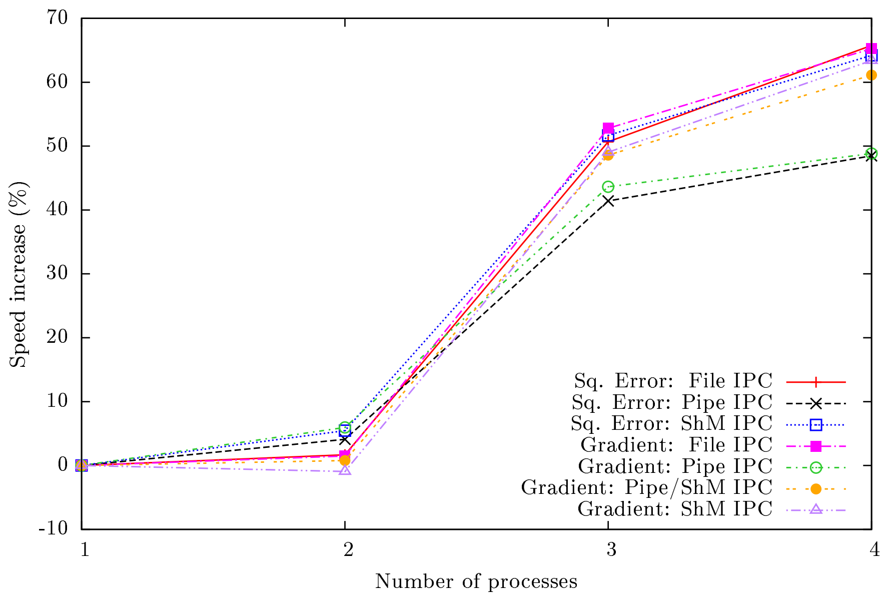
Figure 7.10: Average objective function runtime from 100 trials for various IPC methods with respect to model length. (a) is a normal plot, (b) on a log-log scale.

in respect to model size the percentage of the total runtime spent communicating the model state reduces as model size increases. As the Jacobian has already been optimised to use the SHM it means the improvements in total time are small. There is still however an improvement by using SHM for all of the data transfer; the standard deviation of the runtime is reduced. The reduction in time for the largest site by using SHM over pipes for the objective function without gradients is 1.3 CPU-seconds. The reduction between the pipe/SHM and complete SHM method with gradient calculation is 1.7 CPU-seconds. Hence by moving the state vector to SHM a similar runtime reduction is achieved. However, as a percentage the improvement in runtime is small as the majority of time is spent on the gradient calculation.

Figure 7.11 shows the performance of the IPC methods as these operations are run in parallel across an increasing number of processors for the Sheffield site. Note



(a)



(b)

Figure 7.11: Average runtime for 24 objective function evaluations from 5 trials for various IPC methods with respect to number of processes used for the Sheffield model. (a) is the average time taken, (b) the percentage speed increase.

that in Figure 7.11 IPC performance for one processor is similar to two processors due to the implementation of the parallelism, one node always acts as a scheduler and does not contribute to the calculation as discussed in Section 7.4. As such with this framework the optimal expected speed up is 50% for 3 processors and 66.66% for four. These results obtained in Figure 7.11 were obtained from an Intel Core i5-2400 @ 3.10GHz processor with four cores. For this reason the number of processes was only tested up to four but in a supercomputing environment a larger number of processes could be used.

In the parallel implementation file system IPC actually exceeds the expected speed up rate using 2 and 3 processes (Figure 7.11b red and pink lines). As the test was run on a quad-core processor there is still spare Central Processing Unit (CPU) capacity which the Operating System (OS) can use to speed up the process. When four processes are in use the full capacity of the CPU is utilised and the speed up is slightly less than the expected value. These expected values however assume a perfect system where no overheads exist, which is not the case. The speed up performance is similar for both variants of the objective function. Using pipes for IPC has a detrimental affect on the speed up factor independent of the objective function calculation being used. When all four cores of the CPU are in use there is a significant difference to the expected speed up of 17%. Overall even with lower than expected speed up factors pipe IPC is still quicker than using files. Looking at the process in detail (see Figure 7.7) the cause for this is that on each core the program branches into two paths meaning that the number of processes is doubled. As the child process is created to run the model an extra thread is created, and both threads have tasks to complete during the simulations execution, the simulation (child process) has to output its data while concurrently the parent thread reads it. It is also true that extra threads are created in file system IPC, but as can be seen in Figure 7.6 all the parent process does while the child is running is a wait operation, as such it is not consuming CPU resources. Effectively there is a sequential program and neither thread runs any routines whilst the other is operating. For pipe IPC this is not the case as child and parent compete for resources which has to managed by the OS. This is why the speed up performance is below that expected. The OS has no spare CPU capacity for the parallel tasks when the CPU is fully loaded which is why the speed up rate makes little improvement using four processors over three.

As discussed earlier for the gradient based objective function the transfer of the Jacobian takes a significant portion of the runtime. By using SHM for this data while still communicating the state vector by pipes the speed up rate improves, even so the speed up observed is less than for the other IPC methods. When SHM is used for all data structures (Figure 7.9) as in file system IPC (Figure 7.6) the child and parent thread do not run routines in parallel. Hence the speed up by using additional processors is similar to using the file system and a little below the expected value. By using SHM we have removed the requirement for the child and parent to compete for CPU resources and also remove the slow disk read and write operations.

Overall using SHM is the best method, it scales well as the number of parallel processes is increased and provides the fastest run times. The results in Figure 7.10b also show that the communication of the data has a dramatic effect on the run time of the algorithm. The runtime of objective functions with SHM are also more consistent than the other methods tried. As such there are no downsides to using a SHM model for this application and it can provide a reduction in runtime of up to 95%.

The communication protocols are important part of the software system to solve the validation problem. Another consideration should be the wider framework in which the software is utilised. The whole system should be robust and future-proof, a possible way to achieve this is through autonomic computing and this topic is considered in the next section.

7.6 Future Work: Autonomic Framework

The Autonomic Computing Manifesto [118] outlines the vision of autonomic computing, with the aim to develop complex computational systems with high level interfaces that allow for the administrator to adjust policies that then causes the whole system to modify automatically. This requires the system to be have, by design, a number of self-$*$ properties, such as self-configuration, self-healing, self-management, self-protection, self-optimisation and so on.

The idea behind Autonomic computing comes from biological systems, specifically, the Autonomic Nervous System (ANS). The ANS controls visceral functions, such as the heart rate, digestion, respiration, salivation, perspiration and sexual arousal. It is also responsible for reflex reactions as part of it's self-preservation when presented with extreme external stimulus. The ANS modulates the body within a wide range of environmental conditions as well as own states, furthermore it operates without any conscious input. This last fact is the most important property. The autonomic co-ordination effort and marshalling of resources allows the conscious mind to focus on high level issues.

Designing an Autonomic system to achieve similar results requires the definition of a set of self-$*$ properties. The number of properties required and the meaning associated to them is domain and application specific. Autonomic systems have been used in a variety of domains, including energy management systems [119], communication networks [120], financial markets [121] and spacecraft operations [122].

This section takes a look at defining an autonomic system for Intelligent Transport Systems (ITS). Initial works in field are proposed in [123], a real-time traffic management architecture is outlined in [124] and the optimisation of decentralised autonomic systems for traffic control is considered in [125].

Valid models have a variety of uses such as; traffic forecasting, route choice analysis, traffic management strategy evaluation amongst other ITS applications. Each of these systems have different requirements and may need a model that is valid and maintained over a period of time. The developed software in this thesis could be extended with these potential end uses in mind so that it can easily be integrated within a wider framework. Ideally these systems should be autonomous and exhibit self-$*$ properties [6, 8]. The self-$*$ properties include self-healing, adaptability to incomplete or incorrect data; self-configuring, setting up the model or simulation as required for control strategy or incidents; and self-optimising, recalculating parameters based on relevant metrics. This way the system will provide data to users reacting to changing conditions to give reliable results without external interaction. Autonomic programs implement this by the use of the Monitor-Analyse-Plan-Execute (MAPE) loop as shown in Figure 7.12.

In this framework the traffic model is the managed resource. The manager needs

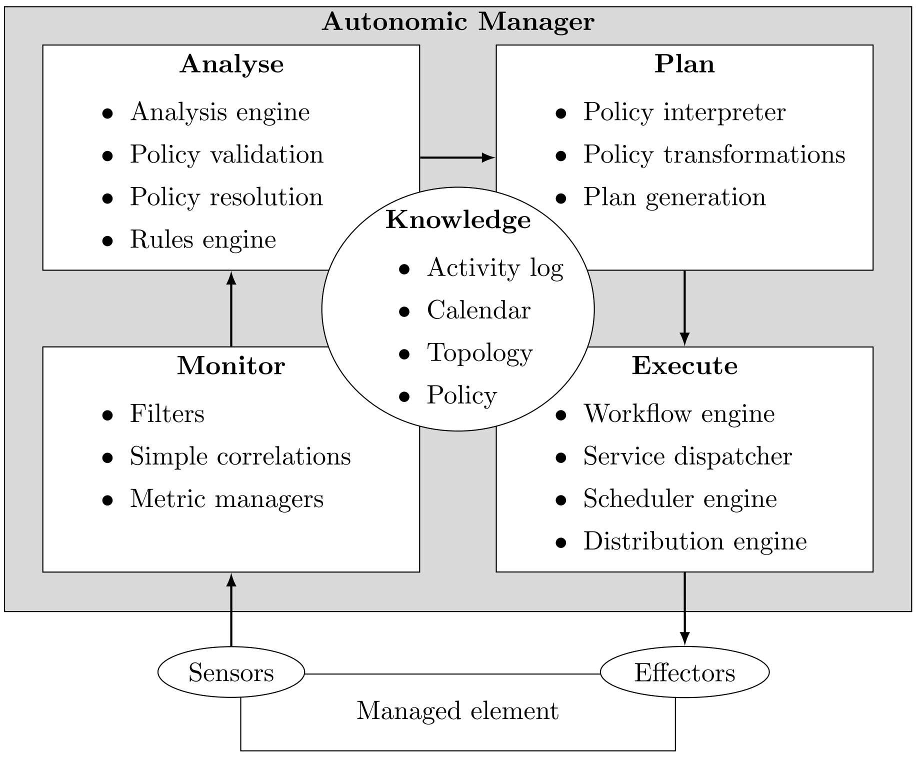


Figure 7.12: The autonomic control loop, adapted from [118].

to be aware of the model's accuracy for the network over time. The application should have the ability to run *self-assessment* at periodic intervals, using recently archived data from a live feed. Thus checking the model is an accurate representation of the current conditions. Based on the results of the self-assessment analysis will be done to determine if the current parameter set is relevant or if a detailed validation of the model is required. The software developed could be expanded with the MAPE loop in mind so it can easily be called in a modular fashion when required. With this format it is possible to use the developed software to maintain an up to date model that evaluates its performance with the newly acquired and past data. A second managed resource could be the optimisation algorithms used for obtaining the optimal parameter set $\mathbf{z}^*$. Each of the optimisers presented in this thesis have parameters that determine how they search the solution space. Autonomic planning platforms [126] could be applied to these parameters, the system will have to collect results over a mid- to long-term time scale and learn how the optimisation algorithms react to adjustments in their parameters in order to be able adjust them in a beneficial manner.

7.7 Conclusion

This chapter has provided the overall software architecture and implementation of the methods discussed in the thesis. A library of optimisation algorithms has been implemented using sequential and parallel computing. Two key areas have been addressed. First the AD tool for extracting gradient information from a traffic model has been selected (ADOL-C) and applied to the model source code. Second a study into the communication methods between the optimisation solver executable and the traffic simulator has been undertaken. It has been shown that in all cases a SHM memory model is the most efficient. It gives the quickest run times and does not adversely affect speed up in a parallel implementation. The Chapter also developed a GUI tool for aiding in the analysis of various traffic models and introduced the notion of autonomics for this ITS application.

Chapter 8

Evolutionary Algorithm Performance

8.1 Introduction

In this chapter the performance of the EAs described in Chapter 5 are investigated for the two shorter sites, Heathrow and Sheffield. This chapter provides a study of the EA optimisations for both the CTM and METANET models. The chapter is composed of two independent studies the first detailed in Section 8.3 is a investigation into the performance of EA algorithms when applied to the METANET model. Firstly the relative performance of each algorithm and the convergence profile is analysed across both sites. This is followed by an analysis of the obtained parameters, including FD assignment patterns and validation results.

The second study follows the same format as the first but applies the optimisation algorithms to the CTM instead of METANET. The next section then compares the obtained results from the two different models, looking at the performance of the identified parameters in relation to the measured data for both calibration and verification.

8.2 Problem Formulation

8.2.1 Optimisation Problem

The calibration problem involves finding a set of parameters to make the model output match as set of measured data and is discussed in detail in Chapter 4. The problem is summarised in Figure 4.8 (see page 52) which can be summarised as,

$$\min_{\mathbf{z}} J\left[\mathbf{x}(k), \mathbf{y}(k); \mathbf{z}\right] \tag{8.1}$$

subject to

$$\mathbf{x}(k+1) = \mathbf{f}\left[\mathbf{x}(k), \mathbf{d}(k); \mathbf{z}\right], \ \mathbf{x}(0) = \mathbf{x}_0 \tag{8.2}$$

$$\mathbf{z}_{\min} \leq \mathbf{z} \leq \mathbf{z}_{\max} \tag{8.3}$$

with J as in (4.17) which is repeated here for convenience;

$$J(\mathbf{z}) = \sum_{k=1}^{K} \sum_{j=1}^{Y} \left(w_v E_{v,j}\left[\mathbf{x}, \mathbf{y}, \mathbf{z}, k\right] + w_q E_{q,j}\left[\mathbf{x}, \mathbf{y}, \mathbf{z}, k\right] \right) + w_p \sum_{m=1}^{M} \sum_{\substack{\mu \in O_{a_m} \\ \mu \notin \Phi}} (\mathbf{z}_{\mathrm{L},m} - \mathbf{z}_{\mathrm{L},\mu})^\top w_{\mathrm{L}} (\mathbf{z}_{\mathrm{L},m} - \mathbf{z}_{\mathrm{L},\mu}). \tag{8.4}$$

Both sites have the same search range defined in Table 8.1, and use the same parameters in the objective function with $w_v = 1.0$, $w_q = 0.0001$, $w_p = 2000$ and

$$w_{\mathrm{L}} = \begin{bmatrix} w_{\mathrm{L},v} & 0 & 0 \\ 0 & w_{\mathrm{L},\rho} & 0 \\ 0 & 0 & w_{\mathrm{L},\alpha} \end{bmatrix} = \begin{bmatrix} 0.04 & 0 & 0 \\ 0 & 0.05 & 0 \\ 0 & 0 & 1.00 \end{bmatrix}. \tag{8.5}$$

The values selected for w_L mean that a difference of 5 km/h in v_f and 4.5 veh/km/lane in ρ_{cr} and 1.0 in α are approximately represented equally in the penalty term.

Table 8.1: Variable limits.

Variable	τ	κ	ν	$v_{\min}$	$\rho_{\max}$	δ	ϕ	α_m	$v_{f,m}$	$\rho_{cr,m}$
Maximum	60	90	90	8	190	4	3.0	5.00	130	40.0
Minimum	1	5	1	5	160	0.001	0.1	0.40	80	18.0

All algorithms were initialised using Latin hypercubes following a uniform distribution within its range, as defined in Table 5.3. For the PSO and GA optimisation algorithms the population size was set to 30. This was selected based on preliminary investigations balancing the computation time for a single iteration against the convergence rate and final solution. A low population number was selected to allow for the same population size to be used for all sites investigated. For CS and MCS a population of 30 was also used and the percentage of nests to be abandoned was set equal to 25% and 75% respectively. For the other optimisation parameter values see Chapter 5.

8.2.2 Site Description

This chapter uses two of the sites discussed in Chapter 6, Heathrow and Sheffield. For readability the site schematics introduced in that chapter are repeated here. The Heathrow site is as shown in Figure 8.1 and Sheffield in Figure 8.2.

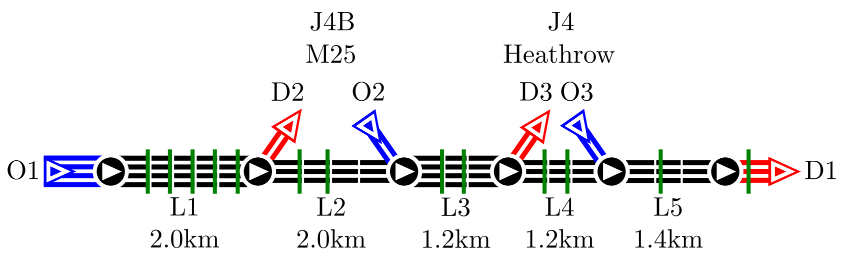


Figure 8.1: Schematic of the Heathrow model. A line represents a lane of traffic, breaks indicate segments. Origins are shown in blue and destinations in red. Locations of available loop detectors are shown by green bars.

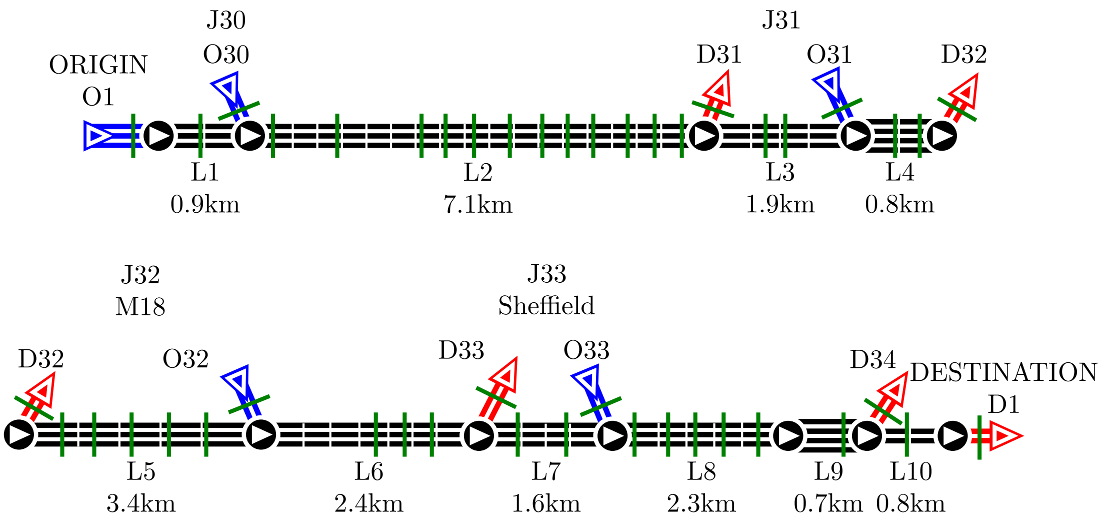


Figure 8.2: Schematic of the Sheffield model, including the location of available loop detectors.

To model Heathrow a parameter vector $\mathbf{z}$ with 27 and 29 parameters for METANET and CTM respectively is required. METANET has 7 parameters defining global attributes and then 3 per link (15 total for this model) that define the fundamental diagram and a further 5 parameters are required for the length of FDs as needed for AAFD $(7 + 15 + 5 = 27)$. CTM has no global characteristics but requires FDs to be defined at destinations so it has 15 parameters to model the links a further 9 to define the destinations (3×3) and the 5 as needed for AAFD. This actually means that the CTM has more parameters even though it is only a first order model and has a single PDE defining it. For Sheffield the METANET model has 47 parameters, 3 × 10 links +7 global +10 for AAFD. The CTM is defined by 58 parameters, 3 × 10 links +3 × 6 destinations +10 for AAFD.

8.3 METANET Model Application

8.3.1 Baseline Genetic Algorithm Optimisation

The first algorithm investigated is a simple and straightforward GA based on a real-coding scheme [96]. The chance of crossover occurring was set at 0.7 and the mutation probability at 0.05. These values were chosen as they allowed for most mutations to occur without disrupting convergence and stability.

In [77] it is shown that by using a combination of crossover operators the GA's ability to search the solution space is improved. The four operators discussed in 5.2.3; i.e. single point, mask, whole arithmetical and heuristic crossover are evaluated.

Figure 8.3 shows the evolution of the objective function of the iteration optimum against the number of function evaluations required by the GA for the Heathrow site for evaluated crossover schemes. The best results were obtained when arithmetical and real coded crossover operators are used. Single point heuristic and mask crossover perform the worst and a combination of all schemes did not improve convergence. Single point and mask crossover are combinatorial and do not affect the value of individual decision variables. When these two operators are used the GA relies on mutation to alter the values to move to new regions of the search. This could explain the poor performance of these two methods compared to heuristic and arithmetical crossover which create child solutions that have parameter values in-

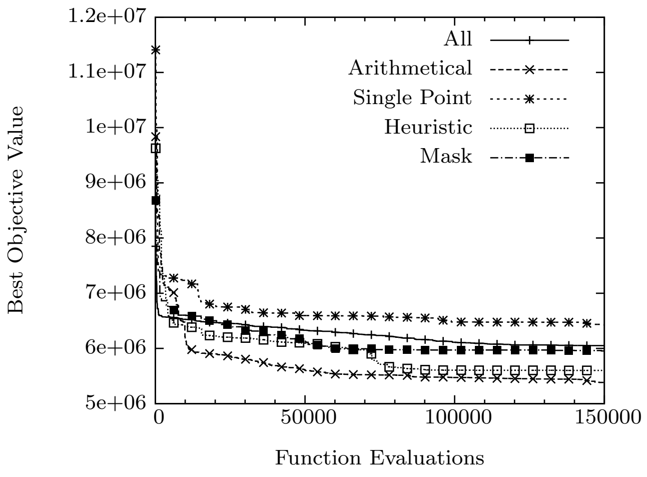


Figure 8.3: GA crossover operators comparison for Heathrow model calibration.

between those of the parent solutions. These operators therefore have to ability to search each dimension more freely and do not solely rely on mutation. For the rest of this thesis the GA will be based on arithmetical crossover only as this provided the best results for this problem.

8.3.2 Algorithmic Performance

As discussed in Chapter 5, ten different EAs have been selected for solving the optimisation problem as shown in Figure 4.10. The number of objective function evaluations that significantly reduce and the minimum value achieved, are used to quantify the algorithms' performance. To evaluate the performance of each algorithm the number of objective function evaluations is used rather than the number of iterations as some algorithms require additional objective function evaluations for certain routines, such as mutation in the GA. Evaluating the objective function requires a model run, this is computationally costly compared to the manipulation of the solution vector performed by any of the investigated algorithms providing a better indication of algorithmic efficiency.

Table 8.2 provides the best objective function values found from three repeats of each of the 10 algorithms for both the Heathrow and Sheffield sites. These include the penalty term for minimising the number of FDs. For brevity, the table headers are labelled as H*date* and S*date* for Heathrow and Sheffield, respectively; thus, H8th refers to the data of the Heathrow site collected on the 8th of February 2010 whereas S8th refers to the data collected from the Sheffield site on the 8th of June 2009. This convention will be used throughout. Each row gives the minimum found by each of the ten algorithms using the data from the corresponding site and date indicated by the column. The full results for each of the repeats of the PSO algorithms are shown in Figure 8.4.

It can be seen from Table 8.2 that the PSO algorithms outperform both the simple

Table 8.2: Summary of best objective value of METANET model for each heuristic (H stands for Heathrow and S for Sheffield).

	H 8th	H 15th	H 22nd	S 1st	S 8th	S 15th	Avg.	Std. Dev. ($\times 10^6$)
GA	5384538	5858308	6739297	6595545	7024323	7619637	5658108	0.81
GPSO	3792653	3142996	2466972	4387156	5644626	3913927	4241388	1.09
LPSO	3785607	**1997553**	1893783	3956224	**4399481**	**3902504**	3322525	1.09
APSO-09	5108443	2613920	**1882825**	4530514	5542426	4414949	4015513	1.45
APSO-12	6541304	2934669	3478906	4841882	7178995	5257622	5038896	1.66
APSO-14	4357332	5823782	4222196	4366692	7530088	5295245	5265889	1.28
HEPSO	**2585451**	2389230	1928796	**3797889**	4453520	4061613	3202750	1.03
CEPSO	3801993	3215278	2092989	4567079	5299468	4881406	3976369	1.19
CS	5733080	3806055	6468042	6273483	6293645	6649381	5870614	1.06
MCS	5440922	3988003	6483786	7278220	6840414	7581395	6268790	1.34

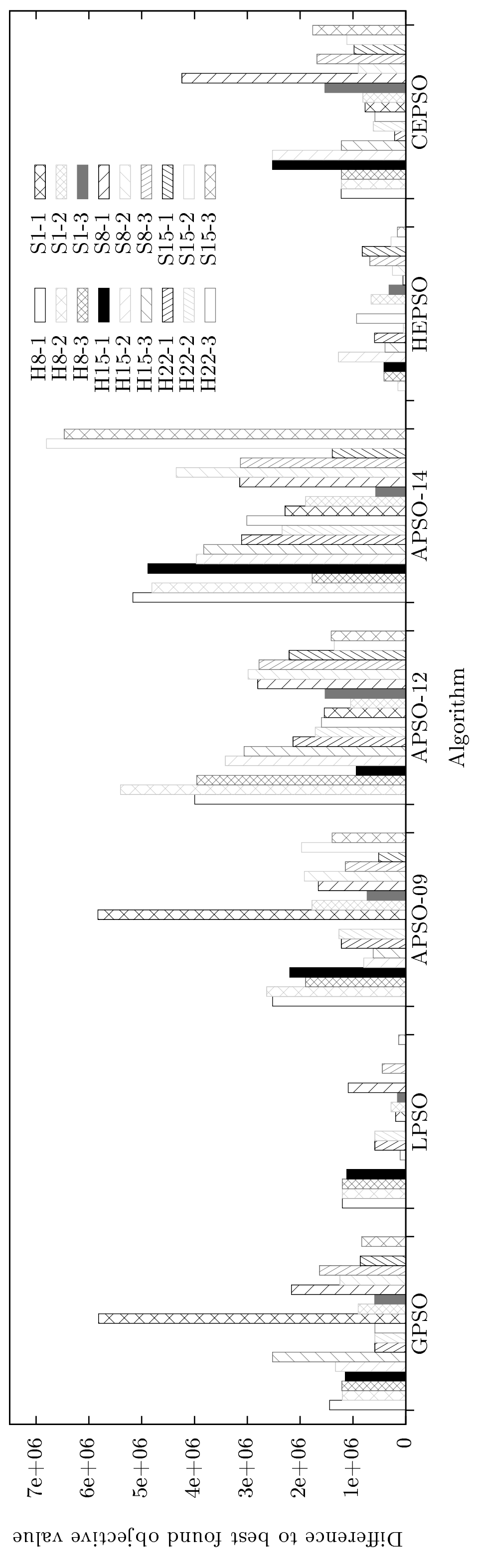


Figure 8.4: Difference to best objective value found by any calibrator for every dataset of METANET model.

GA and the two CS variants. Comparing CS and GA, CS provides better solutions to the dataset of H15th and H22nd. The GA solution for the H8th dataset is superior to that obtained by either CS algorithm. CS consistently performs better than both GA and MCS when applied to the Sheffield site. Interestingly, the unmodified CS performs better than the MCS for all Sheffield and for two of the three Heathrow sets of data. Hence, it can be concluded that for this particular problem the original CS is more efficient from the modified one and the baseline GA for this particular problem.

The PSO algorithms generally find the solutions with the lowest objective function values. Comparing with the GA, the worst performance by any PSO variant is that of APSO-12 for the H8th and S8th sets of data, where the solution is 21.48% and 7.2%, respectively, worse than the corresponding GA solution. For all of the remaining data sets, the worst PSO outperforms the GA. For the H15th dataset this improvement is marginal; for the rest there is at least a 25% improvement in the objective function value obtained by the GA.

Table 8.2 indicates that the worst PSO variant tested is APSO-14. With the best solutions being obtained on average by HEPSO, and then LPSO. These two algorithms have a very similar performance. LPSO found the best solution for a particular dataset on three occasions, H15th, S8th and S15th. HEPSO found the best on two, H8th and S1st. For the H22nd dataset APSO-09 finds the best solution. This is the only occasion where the best result is obtained by an algorithm other than LPSO and HEPSO. Even so the improvement over the solution found by LPSO is only at 0.58%. Hence, it can be argued that LPSO consistently performs better than APSO-09.

The consistency of the algorithms can be seen in Figure 8.4. The best performing variants, LPSO and HEPSO, also appear to the be the most consistent. The figure also shows that APSO-09, CEPSO and GPSO are able on occasion to find good solutions, but typically with larger objective functions values than those obtained by LPSO or HEPSO. However these algorithms lack consistency. This is backed up by the standard deviation in the best objective values found, the lowest is for the HEPSO algorithm at 1.03×10^6, LPSO's standard deviation is 5% larger as is GPSO, CEPSO and APSO-14 show an increase of 15% and 23% respectively. With the most erratic results coming from APSO-09 and APSO-12, where the standard deviation increases by 40% and 60%, respectively, in comparison to HEPSO.

Figure 8.5 gives more details about the consistency of each variant for every dataset. The bar height represents the average error across the three repeats to the best result found by any variant for that dataset (bold value in Table 8.2). The figure confirms the consistency of HEPSO and highlights that LPSO is fairly consistent but

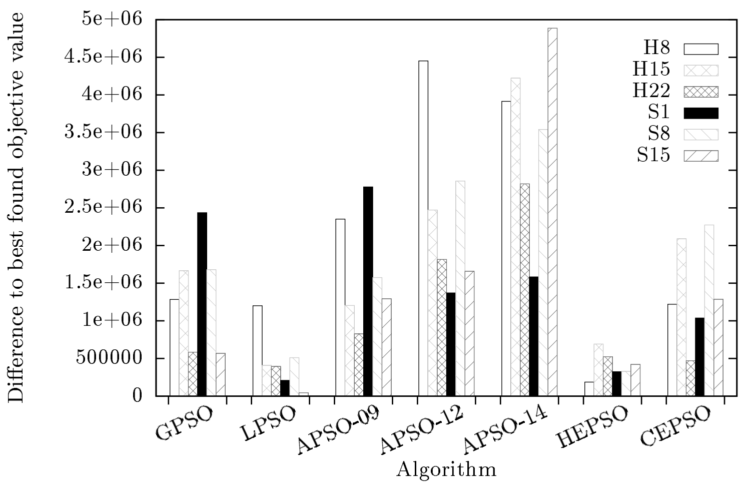


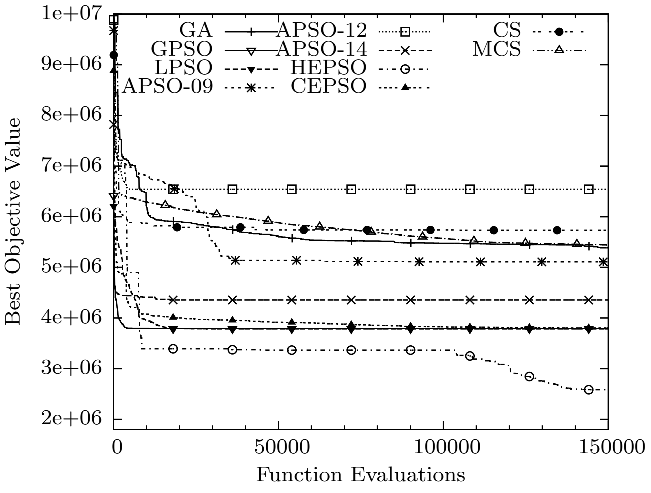
Figure 8.5: Average difference to best objective value found for each dataset of METANET model.

is unable to find the optimal solutions for the H8th dataset. APSO-12 and APSO-14 have a similar pattern and both algorithms fail to find solutions close to the best obtained. Although as mentioned earlier the most erratic algorithms in terms of standard deviation are GPSO and APSO-09 they do achieve reasonable results for each dataset so as on average these algorithms actually provide more reliable and better results than APSO-12 and APSO-14 as shown in the figure.

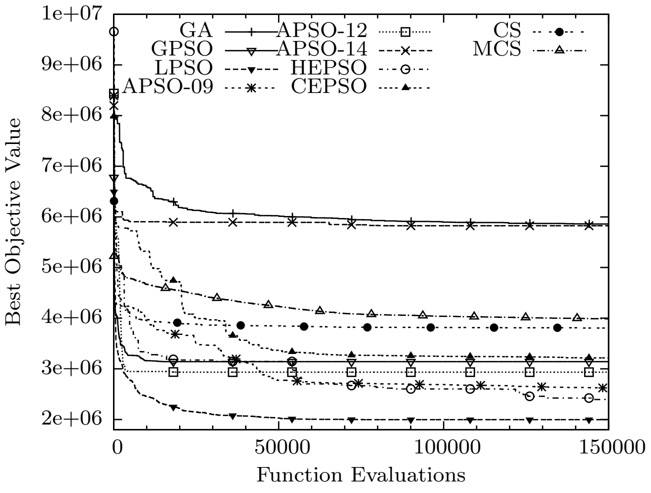
The results shown in Table 8.2 and Figures 8.4–8.5 are interesting since LPSO, which is a basic variant of the PSO algorithm using local information, provides some of the best performances for the calibration problem. The modifications suggested for the more recent variants are not able to provide improved performance, with the exception of HEPSO, which incorporates elements of GAs and artificial bee colony optimisation [82].

The poor performance of APSO-12 and APSO-14 could be due to the nature of their adaptive parameters. These two algorithms attempt to refine the search by adapting parameters as a function of the number of iterations. As such, these algorithms may restrict the search prematurely and struggle to identify a good minimum. APSO-09 performs slightly better, as it adapts the acceleration coefficients based on the state of the search, determined by the Euclidean distance between the particles. The aim of this adaptation is to refine the search when the particles are clustered and expand it when they are spread apart. CEPSO with its chaotic and cyclic parameter values can on occasion find good solutions but is not consistent.

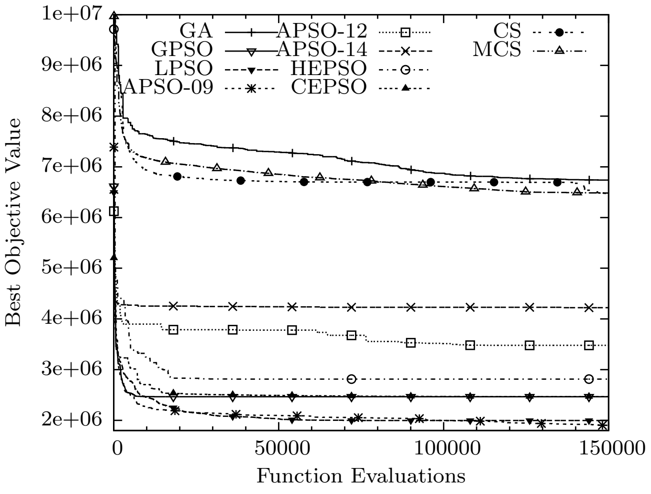
Figures 8.6 and 8.7 depict the convergence profiles, i.e. the best objective function value over the number of function evaluations, where a function evaluation requires a METANET model run. After a short initial period of rapid decrease the GA, CS



(a)

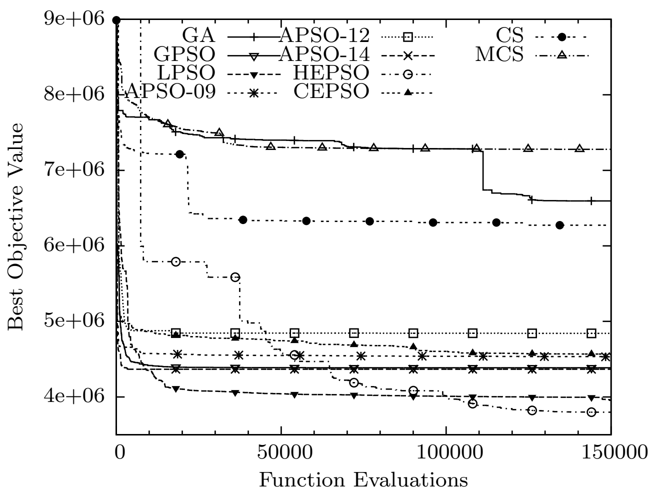


(b)

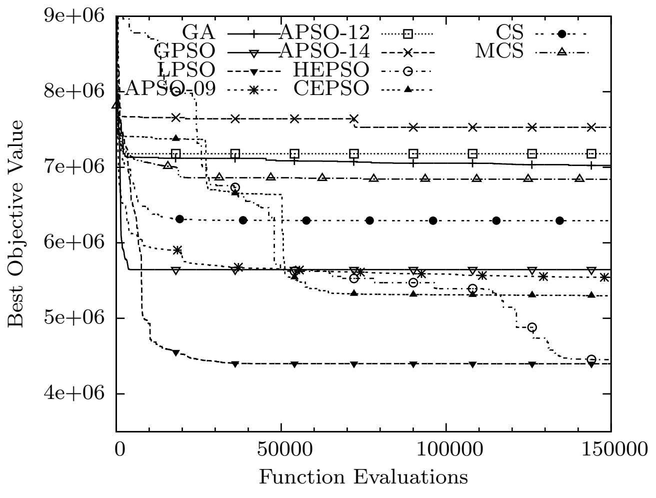


(c)

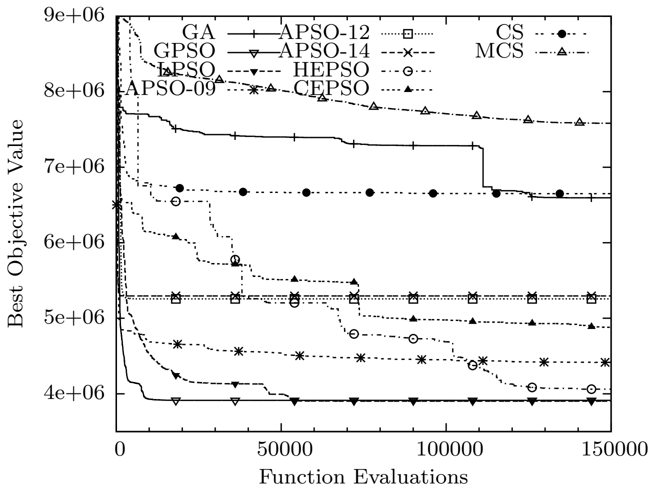
Figure 8.6: Convergence profiles for calibration of Heathrow METANET model: (a) H8th (b) H15th (c) H22nd.



(a)



(b)



(c)

Figure 8.7: Convergence profiles for calibration of Sheffield METANET model: (a) S1st (b) S8th (c) S15th.

and MCS settle on a bad minimum, whereas the PSO algorithms bundle around smaller values. LPSO's behaviour is consistent in the sense that it tends to settle to the minimum quite early on and then improve upon it. HEPSO manages to find solutions with similar objective function values but displays a slower convergence. With its ABC and GA based operators it tends to continually make improvements and appears to be less likely to become trapped in local minima.

The convergence plots also provide corroborating evidence that the APSO-12 and APSO-14 algorithms are being hampered by their adaptive parameters. In both of these algorithms the convergence profiles typically exhibit rapid initial convergence to a point from which the algorithm rarely makes any improvement. In contrast CEPSO, with a cyclic parameter change due to the sinusoidal mapping, makes steady improvements throughout the search. APSO-09 shares similar properties with the other APSO variants but either due to the mutation operator or the state estimation it is able to make some improvements throughout the search or to converge to lower values during the early phase of the search.

Another interesting point shown from the convergence plots is the comparison of GPSO and LPSO, these algorithms only differ in the connectivity of the swarm and the amount of information shared between the particles. In GPSO the each particle knows about every other particle so it has a very steep initial descent as the whole swarm can move quickly towards the direction of the current optimum. But often the search appears to get stuck at a local minima and improvements stop rapidly. This is also the case for some of the Adaptive PSOs. LPSO has less information shared so the initial convergence is slower and the slope of the initial convergence is less steep. However, in most cases at the end of the linear convergence the slope decreases gradually giving a smoother curve with incremental improvements. This could be due to the fact that by having reduced information shared amongst particles the swarm will take longer for every particle to travel towards the current optimum and allow for the particles to be less clustered, slowing the initial search but allowing for a more thorough search of the solution space.

Figures 8.6–8.7 also give some insight into the working of the CS variants. The MCS appears to have disrupted the initial convergence of the CS algorithm, but makes improvements throughout the search. The modifications may have improved the algorithm's ability to combine information from other surrounding particles but the reduction in the initial convergence means that CS overall tends to find better solutions.

The run times for all of the different algorithms was very similar. Using an Intel Core i5-2400 @ 3.10 GHz processor with four cores the run time for the Heathrow site was between 22–25 minutes. There is a resonable variation between runs but no algorithm is significantly faster than the other as the update mechanism is not very costly compared to the objective function calculation (model run) that takes up the majority of the run time. The GA is an exception to this as the mutation operator adds in extra function evaluations outside of the optimised parallel environment (see Ch. 7) this made the GA take 2–3 minutes longer in overall runtime. The same was seen for the Sheffield site with all the PSO variants, CS and MCS taking 60–62 minutes and the GA 63–66 minutes.

8.3.3 Heathrow Site

Analysing the results obtained for the Heathrow site in detail, Table 8.3 gives the optimal values of the global model parameters and Table 8.4 the used FD parameters and their spatial extension. The calibration process identified that three different FDs should be used for each of the three datasets. These are shown in Figure 8.8, where it can be seen that they are arranged into two bundles. The highway capacity pattern identified in all three cases follows a low-high-low pattern, with the area of links 2–4 having high capacity.

Table 8.3: Optimal solutions found for Heathrow METANET model calibration.

	τ	κ	ν	$v_{\min}$	$\rho_{\max}$	δ	ϕ
8th	39.74	5.12	70.26	7.56	160.00	1.06	0.01
15th	14.42	5.04	43.49	8.00	161.77	4.00	0.17
22nd	31.078	5.00	48.59	6.39	189.75	0.002	0.44

The extension of the FD is also the same for all solutions. The reason for this kind of assignment is the fact that link 1 (see Figure 8.1) experiences a spill back effect from junction's J4b off-ramp. Congestion spilling back from an off-ramp, either due to reduced capacity or due to traffic lights at the surface network further downstream, results in an aggregate local behaviour different from the rest of the highway, hence the need for a dedicated FD, covering the affected area. The critical

Table 8.4: Heathrow, optimal solutions of METANET FD parameters.

	ρ_{cr}	v_f	α	Start link	End link
8th					
FD 1	19.65	126.42	1.5192	1	1
FD 2	34.36	126.36	1.4474	2	4
FD 3	20.76	122.35	2.2059	5	5
15th					
FD 1	22.26	122.65	1.4733	1	1
FD 2	31.96	125.17	1.4244	2	4
FD 3	23.75	111.87	1.8305	5	5
22nd					
FD 1	21.79	129.72	1.2579	1	1
FD 2	30.07	129.20	1.3122	2	4
FD 3	20.06	126.33	1.1600	5	5

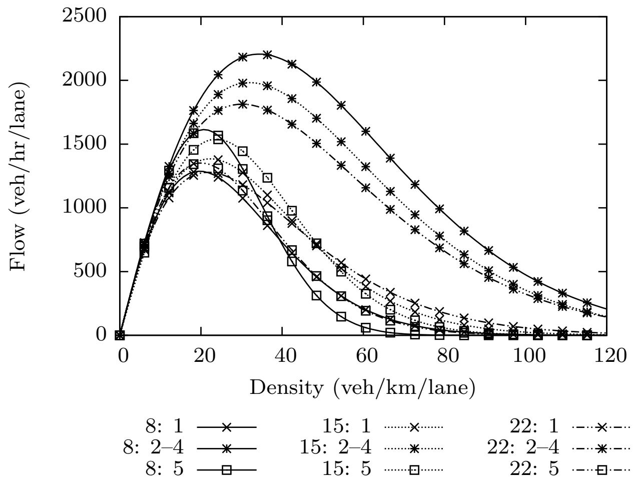


Figure 8.8: Optimal FDs for the Heathrow METANET model; in the legend, the first number indicates the date of data and the second the link number it applies to.

density of this FD is significantly lower than that of links 2–4 FD, reflecting the different and more cautious behaviour of drivers leaving the system via J4b off-ramp. The spill back effect is interpreted by the optimisation as a reduced capacity area, although geometrically the road is similar.

The second region's FD covering a length of 4.4 km (links 2–4) has higher critical density reflecting the fact that the flow continuing past the first off-ramp is no longer impeded by any spill back.

The final link's FD has also small capacity following the trend of the first FD. At the end of this final link the main boundary condition of the highway exit is applied, which is a main influence on the system behaviour. However, there is no congestion there during the entire time horizon, hence the full range of possible traffic conditions are not provided as local information to the optimisation problem. Furthermore, link 5 has an on-ramp at its beginning, which means there are inflows to the mainstream and therefore a reduction of the mean speed due to merging. This should be viewed in conjunction with the use of the merging term $-\delta T q_\mu(K) v_{m,1}(K)/(L_m \gamma_m(\rho_{m,1}(k)+\kappa))$ added explicitly to the speed equation (3.22), see page 29. The optimisation just considers the measurements and converges to a solution where a change in the FD parameters for reproducing the drop and recovery of speed is preferred to a solution that changes one of the global model parameters, i.e. δ. This can be seen from the solutions shown in Tables 8.3 and 8.4. For the two data sets H8th and H22nd where the optimal critical density of link 5 FD is about 20 veh/km/lane, the optimal δ has a low value; for the solution based on H15th the critical density is 23.75 veh/km/lane but the drop of speed is compensated by assigning δ the maximum value allowed.

Table 8.5: Heathrow METANET verification total square error.

	8	15	22	Avg.
Calibrated 8	2539830	3060580	4057189	6952806
Calibrated 15	7404680	1967917	3080164	4150920
Calibrated 22	6614281	3517627	1867561	3377303
22 adjusted	4023508	2395942	2421725	2947058

The overall result, however, is the desired one, as shown in Figure 8.9 where the model flow and speed trajectories are compared to the corresponding MIDAS measurements. The model outputs are obtained by using the optimal parameter set determined by the best algorithm indicated in Table 8.2. Figure 8.9 depicts the calibrated model output against the data set used.

The flow and speed trajectories shown are from the last segment of link 4 and the first segment of link 5, where there is a change in the FD. When the optimal parameter sets are used, the model is able to capture accurately the dynamics of congestion in terms of predicting its onset and the subsequent speed recovery. The resulting total square error, i.e. the values of the objective function without the penalty terms, for the calibrated models are given by the diagonal elements of Table 8.5.

The off-diagonal elements of the table are the total square error when the optimal parameter set determined using the data from the date indicated at the row (found by the corresponding algorithm mentioned in Table 8.2) is applied to the METANET model of Heathrow using input boundary conditions from the date indicated by the columns. In other words, Table 8.5 provides a measure of how the calibrated model based on a particular data set is generalised to the rest of the available data. Reading the table row-wise and averaging the elements, the quality of the corresponding optimal parameter set can be evaluated.

It can be seen that the best and most consistent set of parameters is the one identified by HEPSO with the calibration running with data from H8th. The second best in that respect is the solution based on H22nd. Figure 8.10 compares the model flow and speed trajectories to measurements when the model inputs are from H15th and the model parameters used are the optimal solutions identified using data sets H8th and H22nd. These should also be compared with Figure 8.9(b), i.e. the calibration case of H15th. As expected, METANET with model parameters based on H15th is the best fit, since it is calibrated with them, however, performance of the other two parameter sets is quite good and the main elements of congestion dynamics are captured, as can be visually confirmed from Figure 8.10. This is also reflected in the corresponding error values in Table 8.5.

A further issue Table 8.5 raises is the fact that the best calibration result does

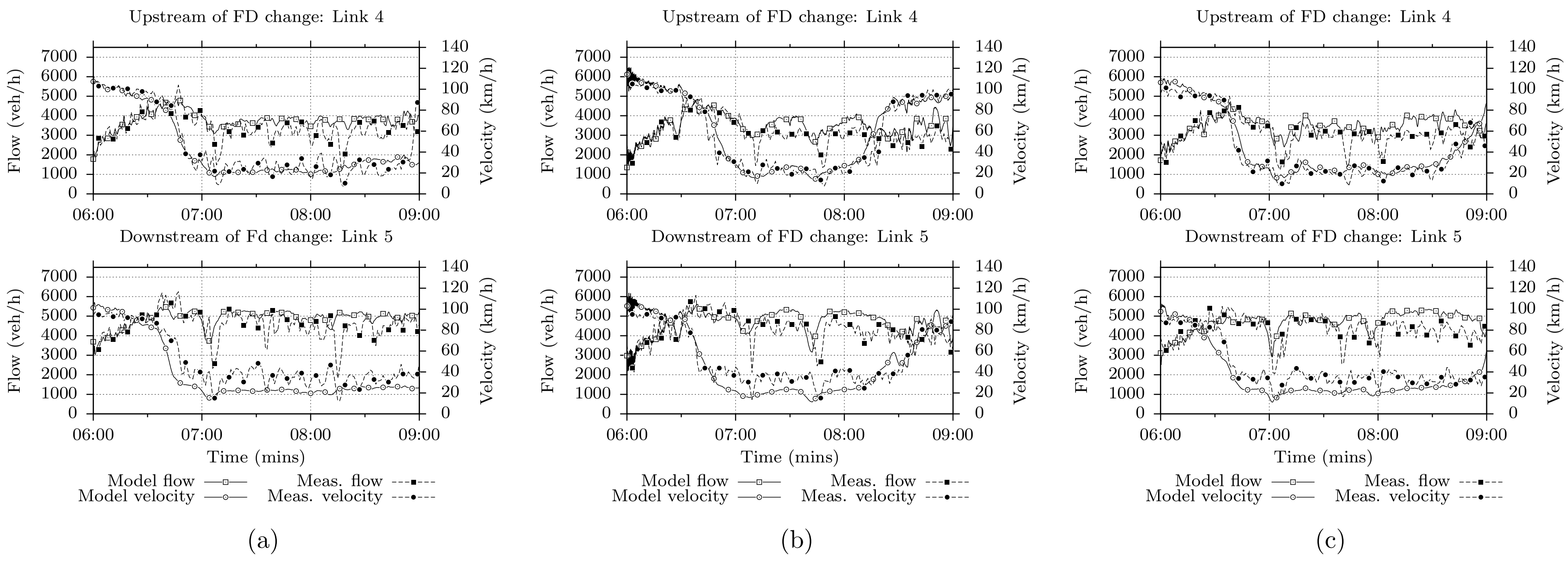
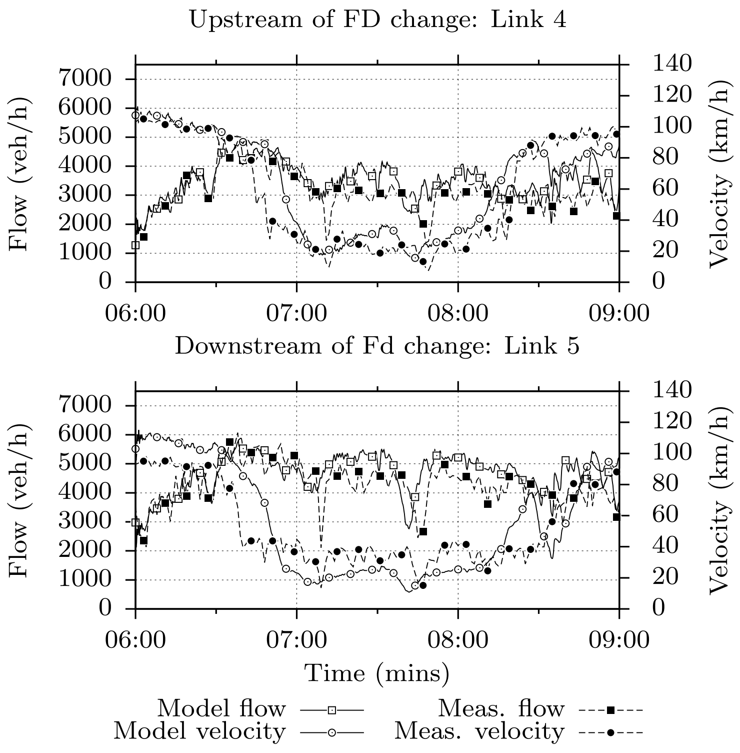
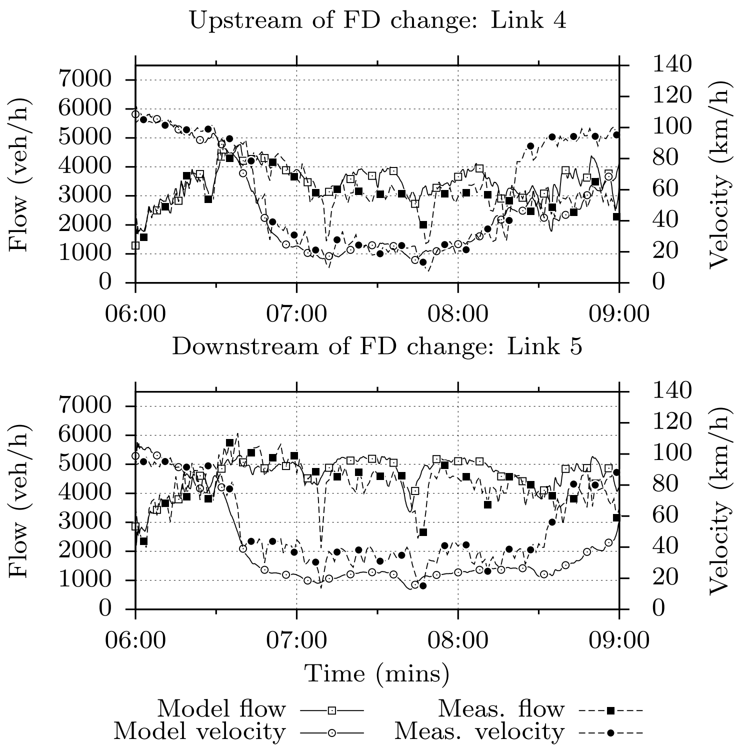


Figure 8.9: Sample of time profiles for the calibrated Heathrow METANET model: (a) H8th (b) H15th (c) H22nd.



(a)



(b)

Figure 8.10: Verification of the Heathrow METANET model using input data from set H15th and using the optimal parameter set based on (a) the H8th data set and (b) the H22nd data set.

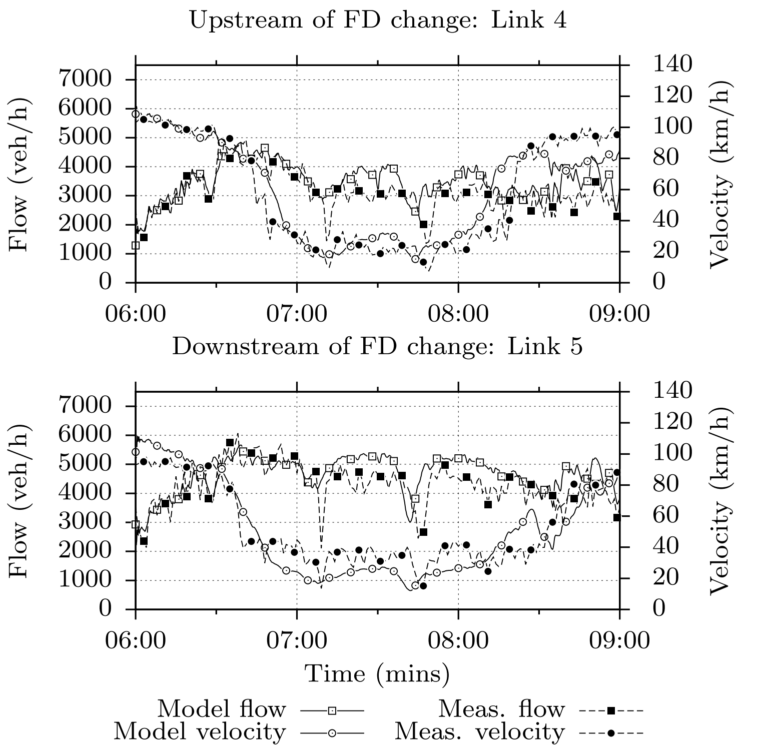


Figure 8.11: Verification of Heathrow 15th using manually adjusted METANET model from the 22nd.

not provide the most general solution. The solution identified by LPSO using the H22nd data set for calibration is outperformed by the solution found by HEPSO using the H8th data set. The H22nd based optimal parameter set provides a good accuracy, with the exception of the latency of recovering the speed at links 4 and 5 at the last half hour of the time horizon, as shown in Figure 8.10(b).

Investigating further, the critical density of link 5 in the parameter set of the H22nd was manually increased from 20.06 to 20.56 veh/km/lane. The outcome of this adjustment is summarised at the last row of Table 8.5. When the new parameter set is again applied to the three data sets, the average total square error is the smallest of all cases, indicating that the adjusted solution is more relevant. This improvement can also be seen in Figure 8.11, where now the speed recovery in link 5 is restored for the last half hour. This result demonstrates the increased model sensitivity with respect to the critical density, something which is reported in the literature as well, [26].

8.3.4 Sheffield Site

The optimal parameter sets identified during calibration for the Sheffield site for each dataset are given in Tables 8.6 and 8.7. The longer Sheffield site allows for a

Table 8.6: Optimal solutions found for Sheffield METANET model calibration.

	τ	κ	ν	$v_{\min}$	$\rho_{\max}$	δ	ϕ
1st	32.34	28.63	55.90	7.48	183.13	0.844	0.038
8th	10.22	20.06	20.00	7.99	184.24	0.001	0.360
15th	19.09	5.89	23.39	8.00	173.38	0.001	0.122

Table 8.7: Sheffield optimal solutions for METANET FD parameters.

	ρ_{cr}	v_f	α	Start link	End link
1st					
FD 1	27.23	122.36	2.6760	1	7
FD 2	30.19	105.35	2.3494	8	9
FD 3	26.68	109.75	1.1386	10	10
8th					
FD 1	31.53	122.22	2.5587	1	3
FD 2	28.50	113.77	1.8865	4	9
FD 3	38.02	104.72	1.0724	10	10
15th					
FD 1	28.43	115.96	2.1077	1	7
FD 2	31.88	103.43	2.0904	8	9
FD 3	35.17	104.73	1.1459	10	10

larger variety of FD assignments. Out of the possible ten different FDs that can be used, the optimisation algorithms converge to solutions that make use of only three. However, their spatial extension is different. As can be seen from Table 8.7 there is always a separate FD for the final link 10. The other two FDs are applied from link 1 to 7 and 8 to 9 for the calibration solutions based on S1st and S15th; the solution based on S8th assigns a single FD from link 1 to 3 and then a different one from link 4 to 9. The resulting optimal FDs are shown in Figure 8.12. It can be seen that the site is split into high and low capacity areas. However, the calibration results follow different patterns. Based on S1st data the site split follows a low-high-low pattern; using S8th data results to a high-low-high pattern whereas S15th to a low-high-high.

This is a problematic result, since there is a fundamental disagreement between the capacity patterns emerging from the S8th data set and the other two. The difference between the capacity pattern in the results based on S1st and S15th is the last link's capacity. Just as in the case of Heathrow, there is no congestion in the measured speed trajectories at link 10, hence, the discrepancy is attributed to the lack of information of the full spectrum of traffic conditions, critical and congested.

The low-high-low and low-high-high capacity patterns are not very different as suggested by Table 8.8, where the total square error is shown for all the calibration and verification cases. The calibration error for the S1st and S15th is comparable and when the calibrated parameters based on S1st are applied using S15th input data and vice-versa, the results are again similar. This is not the case for the S8th data set, which generally shows larger errors. The results based on S8th are not consistent with the rest of the data and do not generalise well.

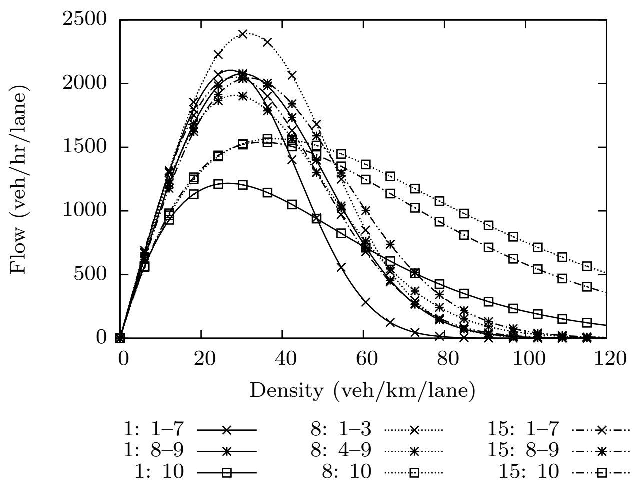


Figure 8.12: Optimal FDs for the Sheffield METANET model; in the legend, the first number indicates the date of data and the second the link number it applies to.

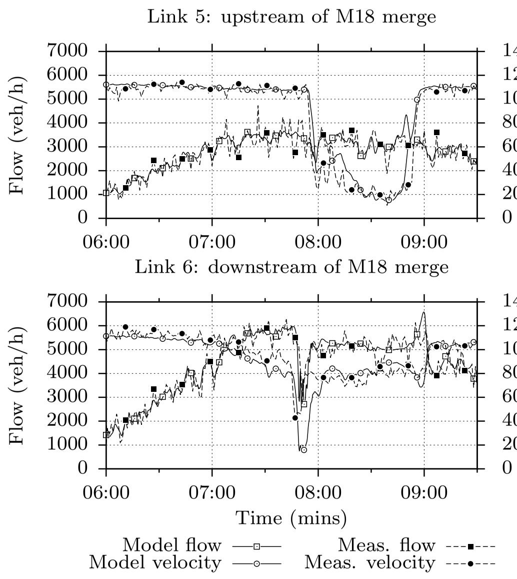
Table 8.8: Sheffield METANET verification total square error.

	S1st	S8th	S15th	Avg.
Calibrated 1	3782550	9741511	7764030	7096030
Calibrated 8	14388131	4382314	20064014	12944819
Calibrated 15	7578581	10574672	3858505	7337253

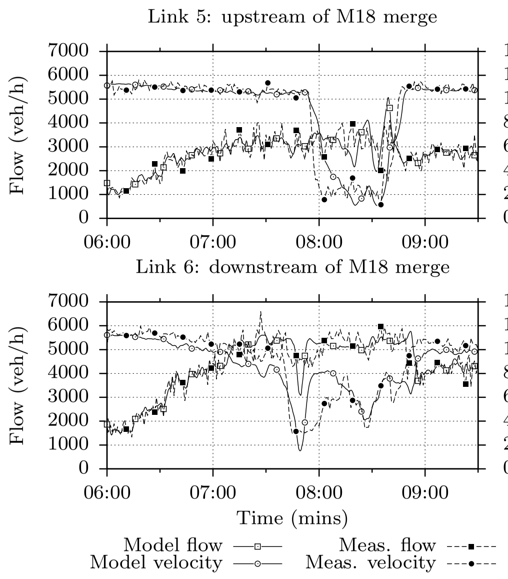
Figure 8.13 provides samples of the calibration results comparing model outputs with measurements. It allows for the accuracy of the solutions reported in Tables 8.6 and 8.7 to be appreciated, including the good behaviour of the solution based on S8th. However, the verification results shown in Figure 8.14 show its poor quality generalisation.

Figure 8.14 shows the model output and corresponding measurements, when S15th data are used along with the optimal solutions obtained from S1st and S8th. The model with the parameter set based on S1st is able to correctly reproduce the traffic conditions of the 15th. This is not true when the optimal parameter set based on S8th is used. The mean speed shown in Figure 8.14(b) does not follow measurements in a satisfactory way. This result shows the difficult nature of the calibration problem. It is difficult, if not impossible, in view of the noisy nature of the available measurements, to formulate an optimisation problem that does not suffer from local minima.

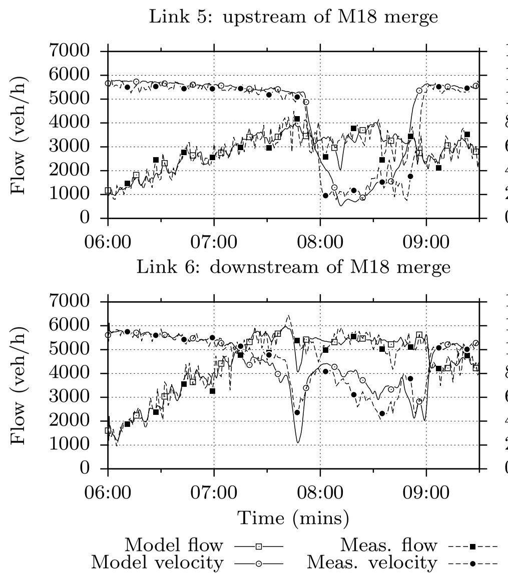
As a final comment, it should be noted that it is the solution provided by HEPSO, using the S1st data set, just as in the case of the Heathrow site provides the best generalised model.



(a)

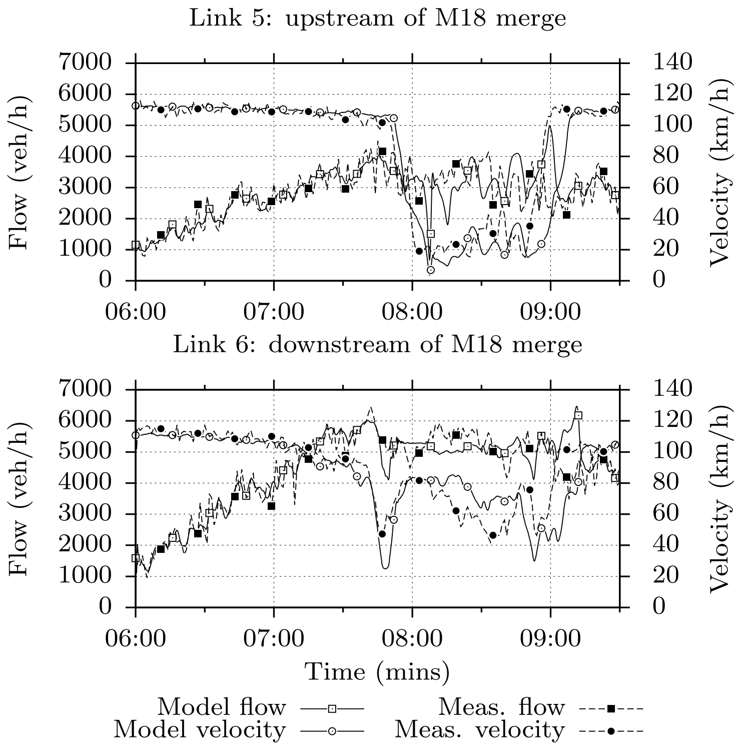


(b)

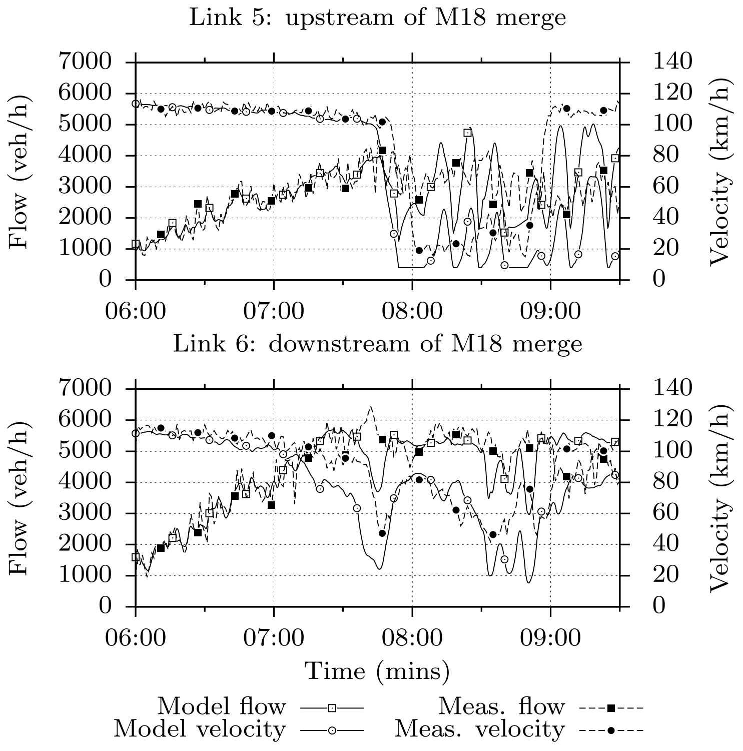


(c)

Figure 8.13: Sample of time profiles for the calibrated Sheffield METANET model: (a) S1st (b) S8th (c) S15th.



(a)



(b)

Figure 8.14: Verification of the Sheffield METANET model using input data from set S15th and using the optimal parameter set based on (a) the S1st data set and (b) the S8th data set.

8.4 Cell Transmission Model Application

8.4.1 Algorithmic Performance

From the investigations involving METANET it can be seen that PSO algorithms provide the best results and significantly improve on the evaluated GS and CS methods. As such for the CTM investigation only algorithms are considered. The same two sites are investigated using the same data, the best objective function values obtained by each algorithm are shown in Table 8.9. The difference to the best result found for each dataset is presented in Figure 8.15.

From Table 8.9 it can seen that the worst PSO algorithm on average is APSO-09, but with a similar value to APSO-14 which was the worst algorithm when METANET was being calibrated. In this case however APSO-14 did find the best parameter set for the H15th. For the best PSO variants a similar pattern to the METANET calibration is observed with HEPSO and LPSO obtaining the best parameter sets on average and the difference between the average objective function value found being 5.1%. For CTM it is clear that HEPSO is able to obtain better parameter sets than LPSO. HEPSO finds the best result for three of the datasets whilst LPSO only finds the best result for one dataset. For the S8th dataset the best set of parameters is obtained by the CEPSO algorithm.

As well as finding the solutions with the smallest square errors HEPSO and LPSO are also the most consistent as seen in Figure 8.15. The figure also shows that every variant can on occasion find solutions close to the best found. Still the same trends as seen for the METANET calibration are observed, Figure 8.4 and 8.15 exhibit the similar profile across experiments and datasets. In both cases HEPSO and LPSO have consistently low bar heights. The adaptive PSO algorithms (APSO-09, APSO-12, APSO-14) show worse consistency as evidenced by the uneven bar heights. CEPSO shows consistent performance for the CTM calibration but typically is not able to find a parameter set better than the other variants. This is also shown by the standard deviation, where for HEPSO it is 0.31×10^6 which increases by 20%

Table 8.9: Summary of best objective value for each heuristic (H stands for Heathrow and S for Sheffield) for calibration of the CTM.

	H 8th	H 15th	H 22nd	S 1st	S 8th	S 15th	Avg.	Std. Dev ($\times 10^6$)
GPSO	5459568	3344361	3266385	3184315	2001146	2945463	3366873	1.14
LPSO	2875403	2605589	2761755	3021043	1994393	**2325524**	2597284	0.38
APSO-09	5555052	2486453	3233378	4977353	2543779	3382440	3696409	1.28
APSO-12	3815504	2519997	2272938	3555410	2432378	3339509	2989289	0.66
APSO-14	2842989	**2426759**	3211044	4544151	4299485	3526727	3475192	0.82
HEPSO	**2698322**	2700748	**2187223**	**2707326**	1976136	2559699	2471575	0.31
CEPSO	5511412	2666628	3245769	3763338	**1940203**	2846570	3328987	1.23

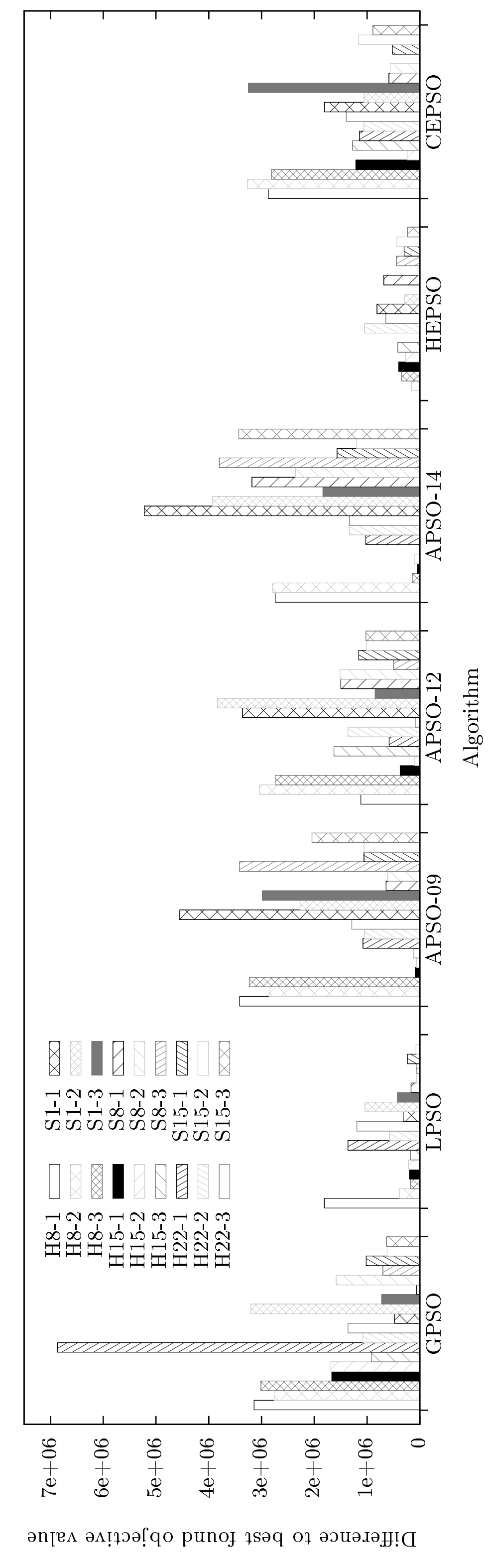


Figure 8.15: Difference to best objective value found by any calibrator for every dataset of CTM.

for LPSO. Unlike the METANET calibration as each algorithm does find a value close to the best, rather than consistently finding lower quality results. Still the consistency of the adaptive algorithms is poor and the standard deviations relative to HEPSO increase rapidly, 110 and 160% increases are observed for APSO-12 and APSO-14 respectively, CEPSO increases by 291% and APSO-09 by 309%.

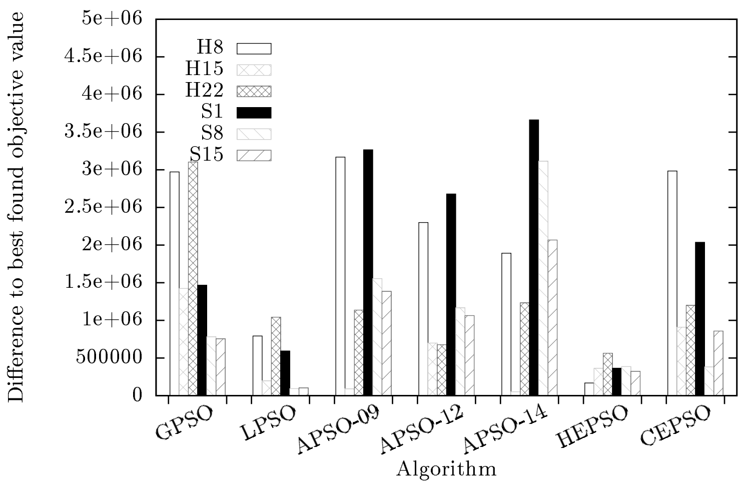
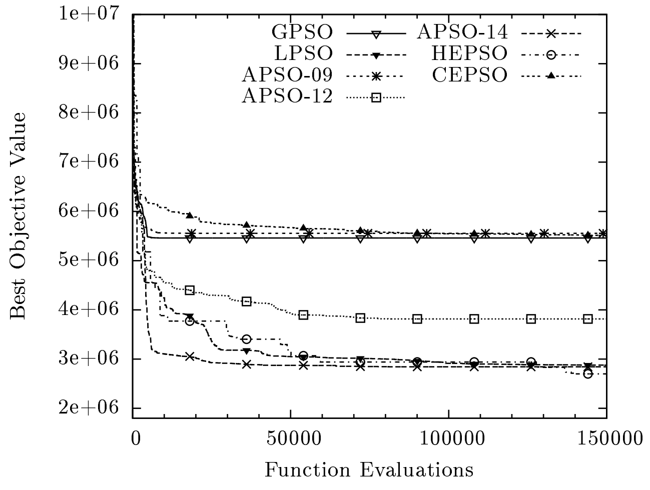


Figure 8.16: Average difference to best objective value found for each dataset of CTM.

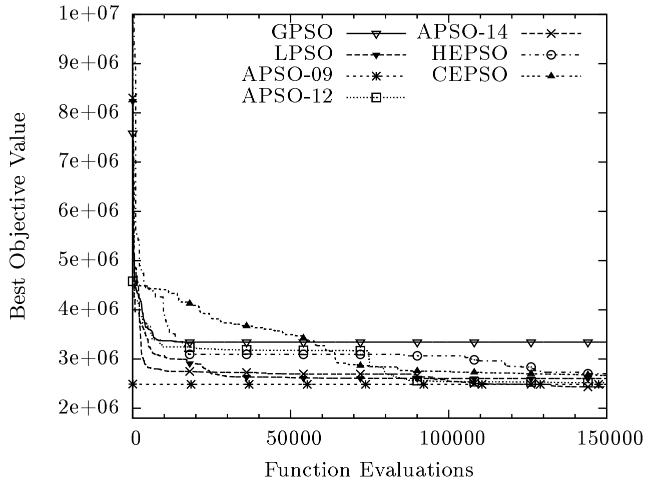
Figure 8.16 shows the average difference to best objective function value found by any of the variants. Again the same pattern is exhibited as in the METANET calibration (c.f. Figure 8.5), LPSO and HEPSO are the best performers finding the values closest to the best consistently. HEPSO is a bit more consistent than LPSO with the bars all having similar heights. Again the adaptive PSOs seem to be the worst, with the largest variance. Apart from APSO-09 and APSO-14 for the H15 dataset the average distance from the best is always worse than that achieved by HEPSO.

It is not surprising that the same patterns in the algorithms performance is shown for the two models regardless of the fact that METANET is second-order with an explicit speed calculation and CTM being first order. Both model the same dynamics, capturing and propagating the various types of shockwaves and furthermore both models have similar parameters through the FD. There are differences with CTM having extra parameters for destinations (Receiving function: v_f, ρ_{cr}, α) and METANET extra global constants (τ, κ, ν, δ and ϕ); this may be the reason why some of the adaptive algorithms are able to match or provide the best result for individual datasets in the CTM but can not find those solutions for the METANET model.

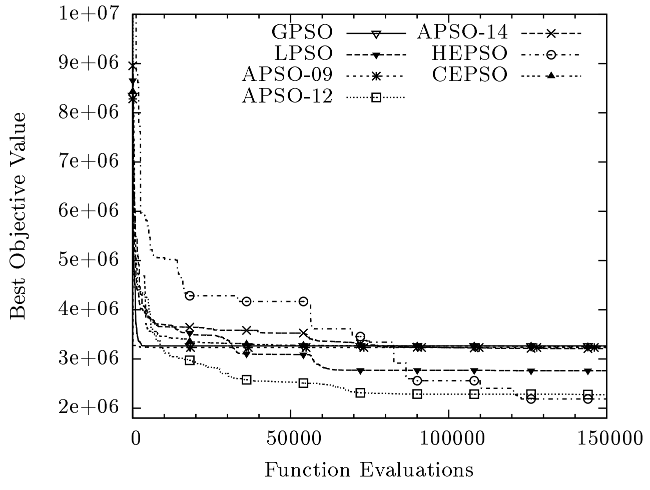
Figures 8.17 and 8.18 show the convergence profiles of the PSO algorithms for



(a)

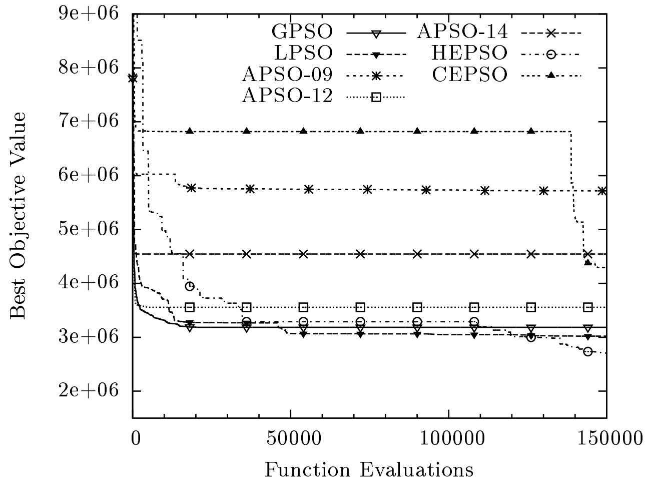


(b)

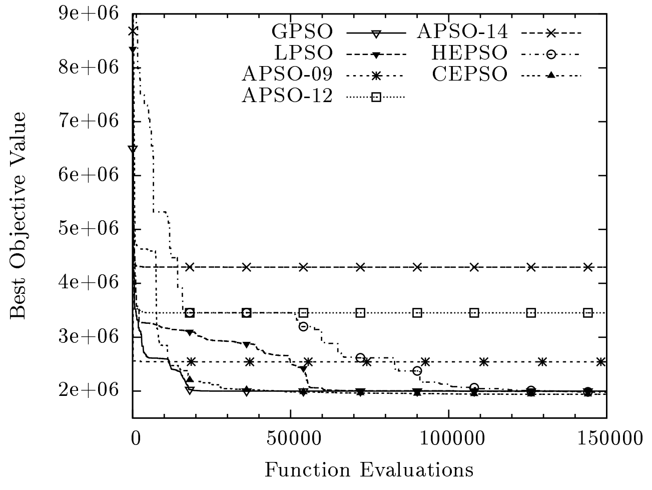


(c)

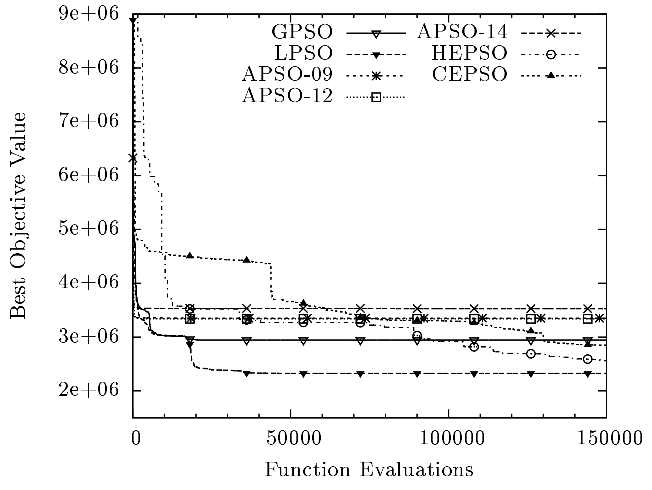
Figure 8.17: Convergence profiles for calibration of Heathrow CTM: (a) H8th (b) H15th (c) H22nd.



(a)



(b)



(c)

Figure 8.18: Convergence profiles for calibration of Sheffield CTM: (a) S1st (b) S8th (c) S15th.

the best parameter set obtained. The same profiles are seen as for the METANET model with most of the APSOs making large improvements in the early stages, but these algorithms appear to not find improvements from the point at which they first converge to and rarely make improvements after 1000 function evaluations. HEPSO is one of the slowest algorithms in terms of the initial search but it makes improvements throughout the search with its extra operators allowing for particles to move away from the current local minima. GPSO again appears to stop finding improvements before LPSO so its fully connected topology appears to be to focused on the initial convergence speed and often leads to a final solution with larger square errors than obtained by LPSO.

The same pattern for run times is seen in the CTM calibration with all the tested algorithms having similar run times. The Heathrow site took between 13–16 minutes and Sheffield 32–36 minutes.

8.4.2 Heathrow Site

The optimal parameter values found from the CTM calibrations are shown in Table 8.10. The calibration process identified that three different FDs should be used for every data set, however the spatial extension is different on the 15th. The FDs found are shown in Figure 8.19. From the figure we can class the identified FDs into three groups, two of which are quite similar. The distinct group is a high capacity, high critical density set of FDs where the capacity exceeds 1800 veh/hr/lane and the critical density is larger than 40 veh/km/lane. This class contains two FDs both of which are only applied to link 2 of the model for data sets of the 8th and 22nd. The second group contains the FDs of links 3–5 for the data set of the 8th and 15th and the FD for link 1 of the model of the 22nd. The final group contains four FDs. The FD for link 1 of the data set of the 8th, links 1–2 of the data set of the 15th (which is a close match to link 1 FD for the 8th and hard to see in the figure), the FD for links 3–4 of the 15th and also the FD for link 5 of the data set of the 15th. Both of the final groups cover a similar range of critical densities and capacities.

The grouping shows that the models of the 8th and 22nd are selecting a high capacity FD for link 2 to capture the free flow dynamics of the link that typically has lower flows than most of the site as it resides between an off- and on-ramp. For the model of the 8th the parameter set identified does not have this jump and the first FD is applied to both links 1 and 2. The models of the 8th and 22 follow similar patterns, but for the model of the 22nd the FD that covers link 1 is in the group with a steeper shape (larger α). This is a difference between these two datasets and in the model of the 15th the FD applied to link 1 is also in the final group, and is nearly identical to that used for the 8th, even though it also applied to link 2 also.

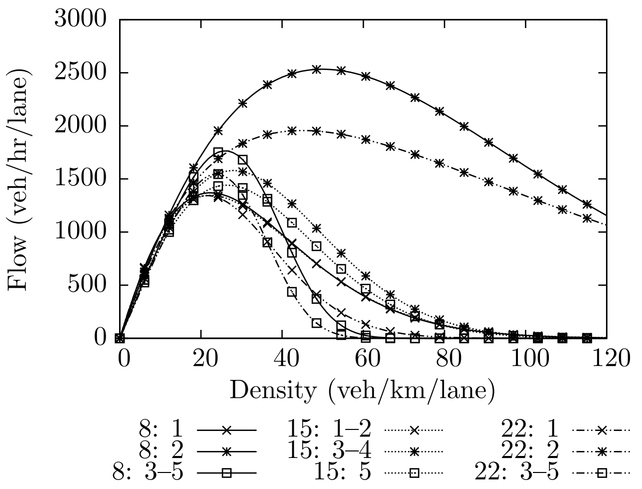
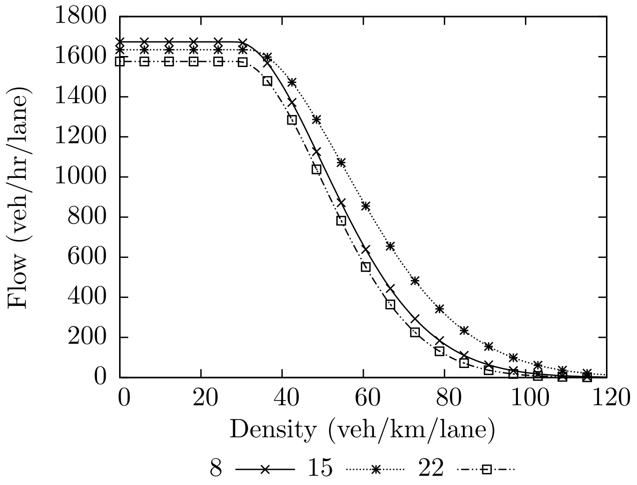


Figure 8.19: Optimal FDs for the Heathrow CTM model; in the legend, the first number indicates the date of data and the second the link number it applies to.

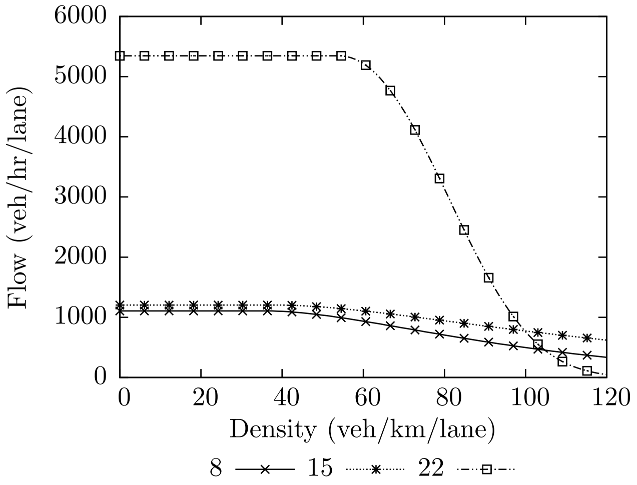
The CTM unlike the METANET model has a set of parameters to govern the boundary conditions at destinations. The receiving functions (c.f section 3.4.2) are shown in Figure 8.20. For the main downstream destination of the model the parameters selected by each calibration are very similar (Figure 8.20a) with capacity around 1600 veh/km/lane and critical densities 38–40 veh/km/lane. For D2 the destination at the end of link 1 there are similarities for the parameters selected by the calibrations for the datasets of the 8th and 15th, the values selected by the calibration 22nd are unusual. The capacity is very high and not realistic in a real

Table 8.10: Optimal solutions found for Heathrow CTM calibration.

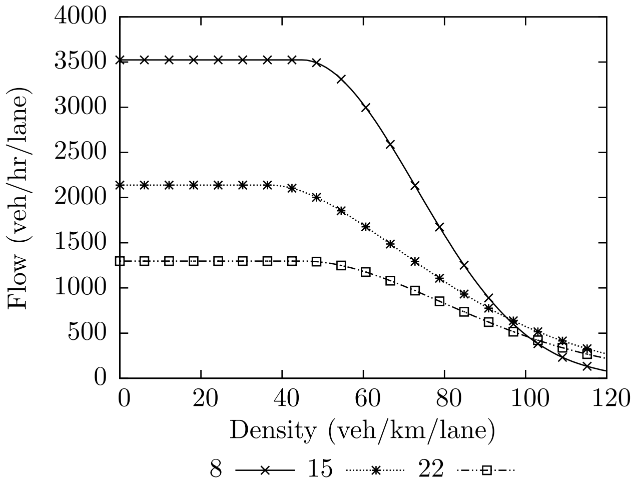
	ρ_{cr}	v_f	α	Start link	End link
8th					
FD 1	22.30	122.02	1.4571	1	1
FD 2	50.00	107.22	1.3340	2	2
FD 3	25.98	90.96	3.4271	3	5
D1	28.69	96.80	1.9743	–	–
D2	35.62	82.56	1.0231	–	–
D3	44.74	115.01	2.6432	–	–
15th					
FD 1	22.33	120.64	1.4427	1	2
FD 2	27.89	96.09	1.8911	3	4
FD 3	26.43	93.99	1.8376	5	5
D1	31.40	88.32	1.8910	–	–
D2	38.49	117.79	0.7539	–	–
D3	36.77	113.56	1.4947	–	–
22nd					
FD 1	21.55	103.79	1.9633	1	1
FD 2	45.00	129.68	0.9149	2	2
FD 3	23.79	86.87	3.4503	3	5
D1	29.08	86.41	2.1424	–	–
D2	54.08	127.71	3.9043	–	–
D3	45.02	49.60	1.8430	–	–



(a) Destination: D1.



(b) Destination: D2.



(c) Destination: D3.

Figure 8.20: Optimal boundary receiving functions for the Heathrow CTM model.

world situation, at 5200 veh/km/lane traffic has a time headway less than 0.7 s. This however, is only part of the model and the flow received by destinations is also dependant on the supply so in this model the destination parameters are such that it can accept all of the demand. Finally the destination D3 has a different profile for each calibration. The values selected do not restrict the flow for the data sets of the 8th or 15th.

Comparing the spatial impact of the FDs and the effect of the destinations receiving functions the solutions obtained for the CTM do not match the patterns found in the METANET model. Here the obtained results do not form a very specific clear pattern that is consistent across datasets, which was the case for the METANET model. In the METANET model each parameter set found contained a set of FDs following a low-high-low pattern. The low capacity regions account for modelling spill back from congested off-ramps. CTM has more control over destinations and this then has had an impact on the values obtained for the mainline FD parameters. The parameters found for the CTM for the dataset of the 8th and 22nd still follow the low-high-low pattern but the high region is restricted to link 2 where as in METANET this extended to Link 4.

Sample time profiles for the calibrated models are shown in Figure 8.21 from the same locations as used in the METANET model. These plots show a good correlation between the model and measured data, in all cases however the onset of congestion in link 4 is predicted with a delay. The effect of using the METANET FD, instead of the typically used trapezoid, can be seen at the early stages of the model time horizon. With a trapezoidal FD the predicted speed would be constant during free flow conditions which is not the case here.

Table 8.11 shows the squared errors when the calibrated parameter sets are applied to the initial conditions and demands of the other data sets. The overall verification result seems to be very poor especially compared with the results obtained by METANET. It appears that the CTM parameters selected are unable to properly capture the correct dynamics for the other datasets. The trends identified for the different data sets for the FD parameters for each data set do appear similar. The main difference between the individual calibration attempts comes from the destinations parameters. The time profiles for the verification using data from the 15th are shown in Figure 8.22. For the parameters trained on the 8th the model predicts the speed recovery too soon, while for the 22nd the recovery does not happen at all. Looking at Figure 8.20c a possible cause is identified, the boundary upstream of the shown plots varies wildly. The max capacity of the off-ramp for the calibration of the 15th lies in the middle of those identified for the other data sets. When the model of the 8th is applied to the 15th it predicts recovery too soon, comparing the

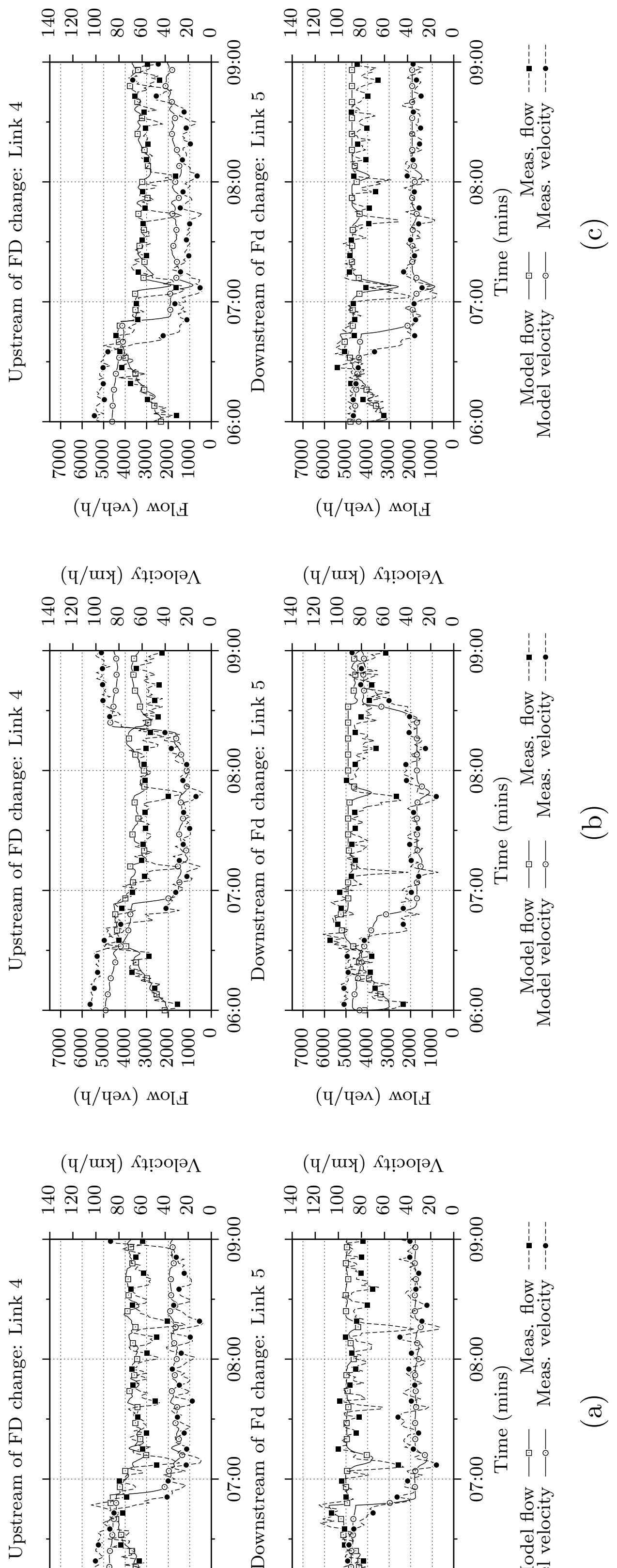


Figure 8.21: Sample of time profiles for the calibrated Heathrow CTM: (a) H8th (b) H15th (c) H22nd.

Table 8.11: Heathrow CTM verification total square error.

	8	15	22	Avg.
Calibrated 8	2573182	12924260	18299454	11265632
Calibrated 15	9757216	2250603	10091894	7366571
Calibrated 22	13307439	6102528	2275194	7228387

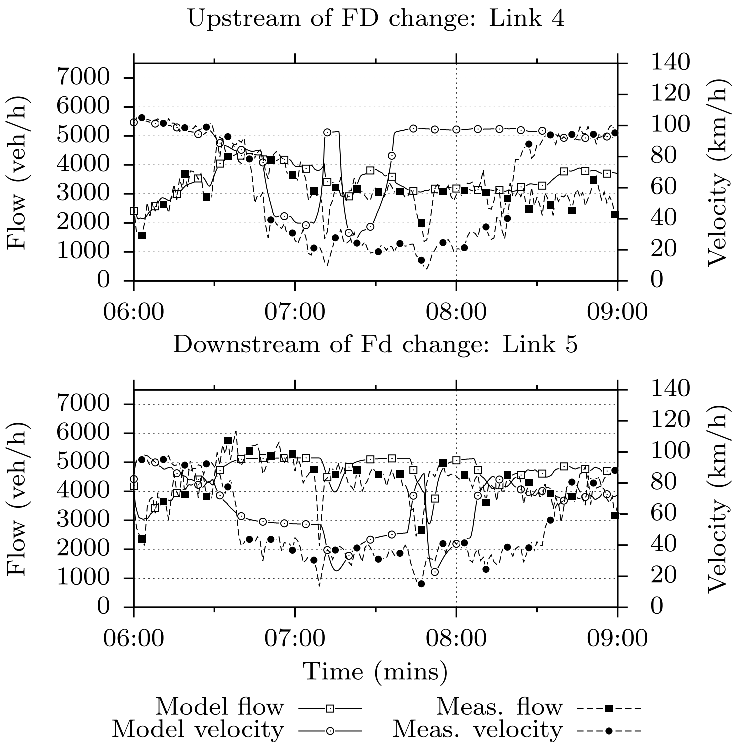
parameters obtained for both data sets the capacity for the off-ramp D3 is much greater on the 8th and therefore the destination is not causing spill backs into the model so the speed recovers prematurely. The contrary is true when the parameters trained on the 22nd are applied to the data set of the 15th. This time the capacity of the destination is less and the impact of spill back is too great.

The Heathrow site is a difficult test as the available data for destinations D2 and D3 and Link 3 not being based off direct measurements. For D2 downstream of the modelled destination the road diverges in two to allow access to both directions of the M25. The only working loop detector available in this area is just downstream of this split, so node balancing as discussed in section 6.3.3 is used to calculate the destinations demand. As for Link 3 and D2 the loop detectors are split with the left hand lane being measured independently, these are combined with the main highway to get the required values across all lanes. It appears for METANET that these issues have a lower impact, but CTM needs to determine the receiving functions at the destinations and a consistent pattern across the three datasets was not obtained.

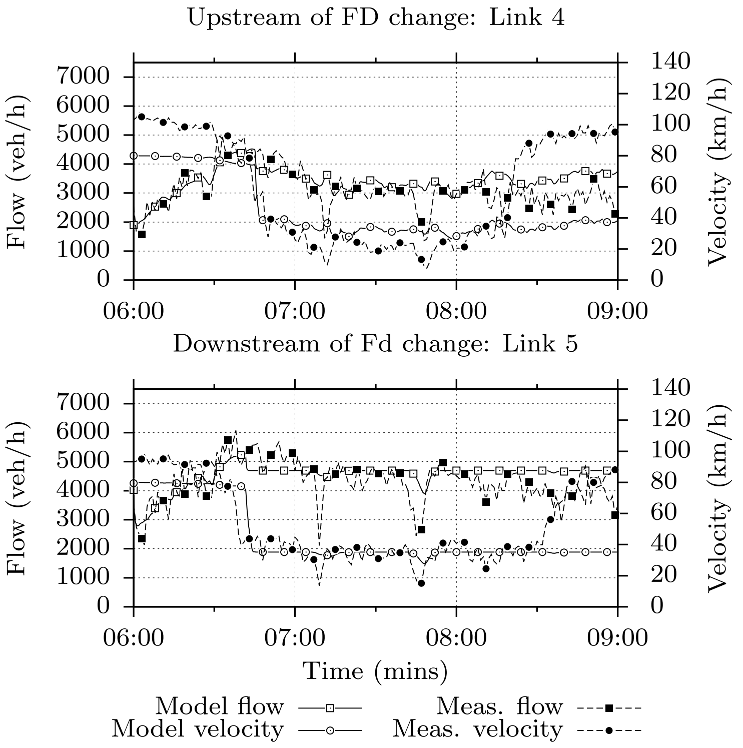
8.4.3 Sheffield Site

Table 8.12 shows the identified FDs and the spatial configuration; these are visualised in Figure 8.12 for the Fundamental diagrams and in Figure 8.24 the receiving functions at destinations. As in the MEANTET calibration the solutions identified only utilise 3 FDs out of the 10 available. Also similar to the METANET calibration the extent of the FDs is different, but the similar patters are found. Both the calibration of the 1st and 15th use a single FD for links 1–5 inclusive, however the second FD for the claibration of the 15th is a link longer. The final FD for all three data sets is short in the calibrations of the 1st and 15th it is only used for Link 10 but the calibration of the 15th it is also applied to Link 9. The spatial pattern found for the dataset of the 8th matches the pattern obtained for the METANET model for the datasets of the 1st and 15th.

Looking at the figure of the FDs found for the datasets of the 1st and 15th it can be seen that the first FD from Link 1 to Link 5 has its parameters set so that model will predict free flow conditions. Due to this only the slope at low densities is important, high densities are not seen in this region. This is why a very large capacity FD has been selected for the dataset of the first with a capacity of 3000 vehicles/hour/lane.



(a)



(b)

Figure 8.22: Verification of the Heathrow CTM using input data from set H15th and using the optimal parameter set based on (a) the H8th data set and (b) the H22nd data set.

Table 8.12: Optimal solutions found for Sheffield CTM calibration.

		ρ_{cr}	v_f	α	Start link	End link
1st						
	FD 1	45.00	111.00	1.9943	1	5
	FD 2	24.36	113.93	2.1769	6	8
	FD 3	30.06	110.40	1.6187	9	10
	D1	24.97	104.54	2.4747	–	–
	D31	44.35	80.79	0.5151	–	–
	D32	24.46	106.36	1.5538	–	–
	D33	36.37	129.68	0.5885	–	–
	D34	20.18	85.67	3.4909	–	–
8th						
	FD 1	24.27	110.79	3.4942	1	7
	FD 2	29.48	101.83	2.6604	8	9
	FD 3	35.32	84.26	2.3635	10	10
	D1	26.56	108.57	2.4585	–	–
	D31	28.15	106.94	0.5077	–	–
	D32	37.9	108.79	2.885	–	–
	D33	33.84	109.27	0.6946	–	–
	D34	31.73	127.57	2.6202	–	–
15th						
	FD 1	30.62	112.15	2.4691	1	5
	FD 2	24.99	107.58	2.4664	6	9
	FD 3	34.29	85.53	2.3133	10	10
	D1	32.59	103.57	1.9155	–	–
	D31	32.38	90.42	1.7292	–	–
	D32	25.78	112.24	1.3719	–	–
	D33	28.79	101.08	1.9452	–	–
	D34	41.72	118.52	2.9149	–	–

The middle FD for this dataset covers the region that experiences congestion and has a lower critical density and low capacity, the calibration obtained very similar FDs for each dataset. The final FD for all sites is similar and it has a middle level capacity but with a high critical density and lower free speed. The FD found for this part of the model for the dataset of the 1st has a higher free speed because it covers

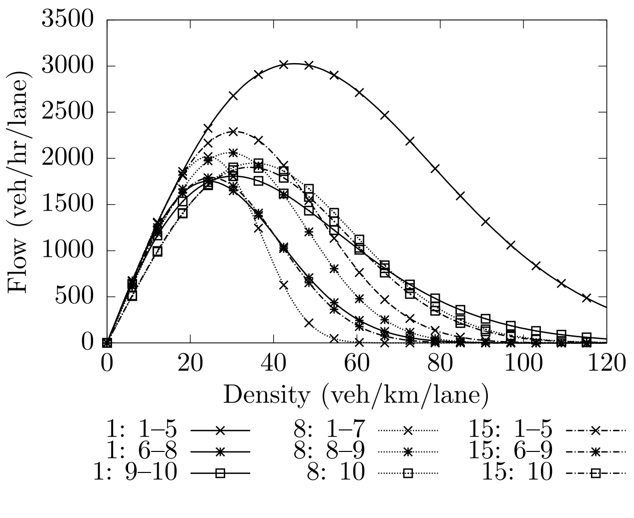


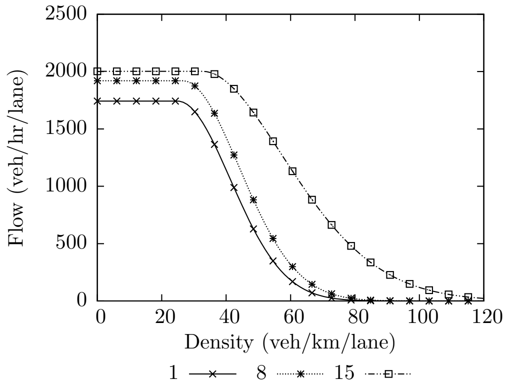
Figure 8.23: Optimal FDs for the Sheffield CTM model; in the legend, the first number indicates the date of data and the second the link number it applies to.

an additional link and has aggregated the properties over the region. The calibration of the 8th is slightly different, the first FD covers both the free flow region and the congested area, it has a higher capacity than the FDs found for the congested region for the datasets of the 1st and 15th but with a similar critical density. The next FD then covers a free flowing region after the recovery from congestion and has a slightly increased capacity and larger critical density.

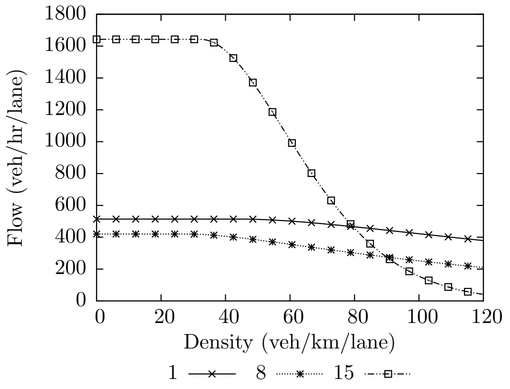
The receiving functions as in the Heathrow site show a wide amount of variation between the different datasets as shown in Figure 8.24. For the main downstream destination of the model D1 all solutions have selected a receiving function with a high critical density. The peak capacity for the model of the 1st does have a lower capacity. The destinations D31 and D34 show a lot of variation, however the demand at these destinations is low. The parameters selected for each dataset do not have an effect on the outflow. This means that the impact of these parameters on the objective value is small and the variance of the identified receiving functions can be explained. For D32, the destination that leads to the M18 motorway, the obtained parameters of the 1st and 15th are similar. The parameters based on the 8th result in a destination with unrestricted flow. Destination D33 is important as this off-ramp is congested and the spill back is what starts the congestion. The calibration of the 1st and 8th both select very similar receiving functions with a capacity of 850 veh/hr/lane and a high critical density and very little drop off, the function selected for the dataset of the 15th is significantly different and will only restrict it at very high densities. A capacity of 850 veh/hr/lane is quite low but the destination at D33 is shortly followed by traffic a set of traffic lights controlling the flow into urban network, this could be the reason for receiving function found with a low capacity and little drop off at high densities.

Sample time profiles for the three calibrated models are shown in Figure 8.25. Each model is able to provide a very good match to the data, looking at Link 6 the bottom panel it can be seen that in each data set there is a gradual drop off in speed before a sharp shock. For the dataset of the 15th in Link 6 the model reached capacity flow in the main line creating congestion. This is why the receiving function for the congested destination is free flowing; the parameters of the mainline are set to create the speed drop caused by the spill back. This creates some errors in the predicted speed in Link 6 between 6:00 and 7:45.

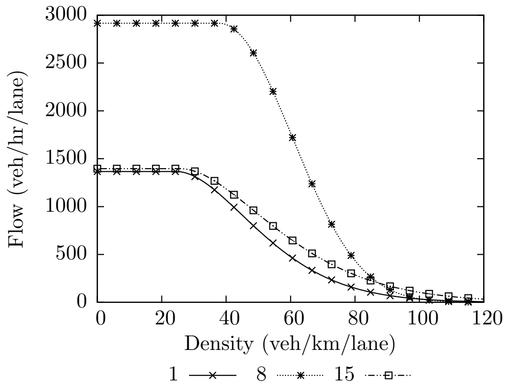
In verification the models all provide a good correlation to the measured data but the model of the 8th is the best on average as evidenced by the squared error values, Table 8.13. It is interesting to note that the model of the 8th applied to the 15th actually provides a better result than the model calibrated using that data set. The verification time profiles for the parameters of the 8th are shown in Figure 8.26. It



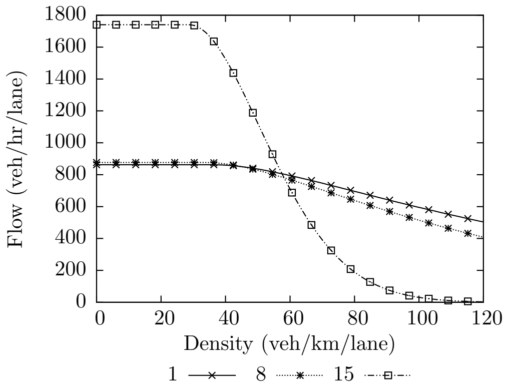
(a) Destination: D1.



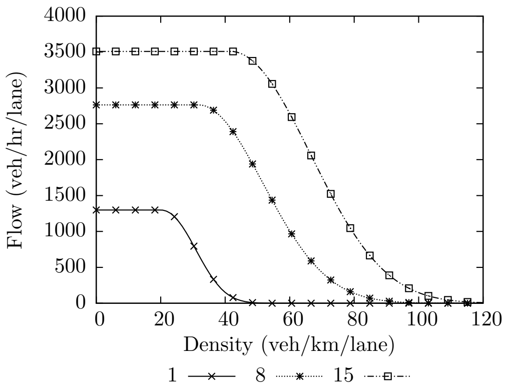
(b) Destination: D31.



(c) Destination: D32.



(d) Destination: D33.

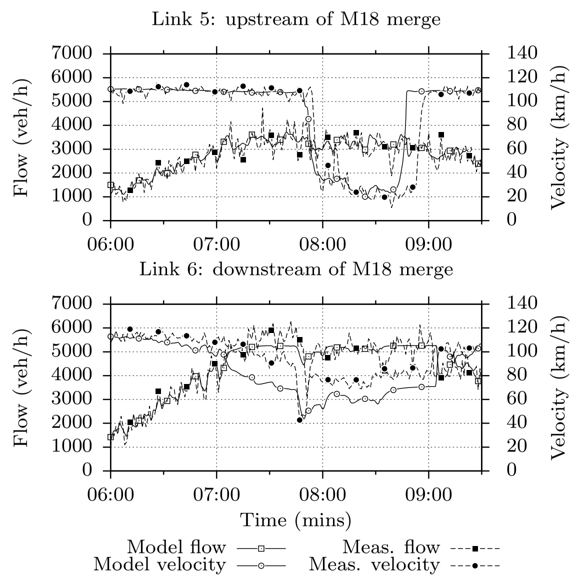


(e) Destination: D34.

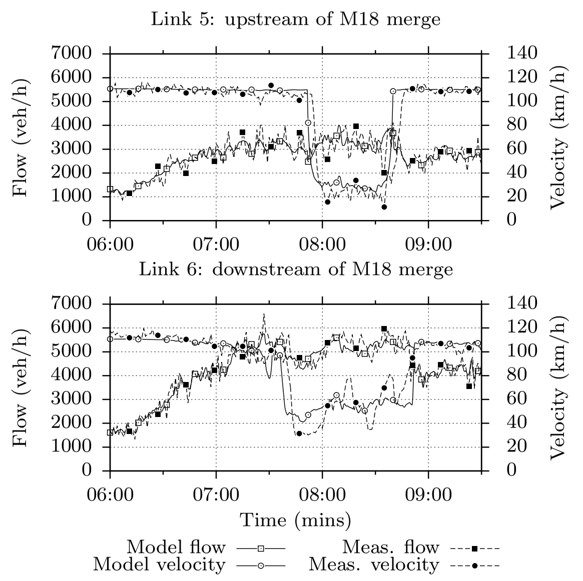
Figure 8.24: Optimal boundary receiving functions for the Sheffield CTM.

Table 8.13: Sheffield CTM verification total square error.

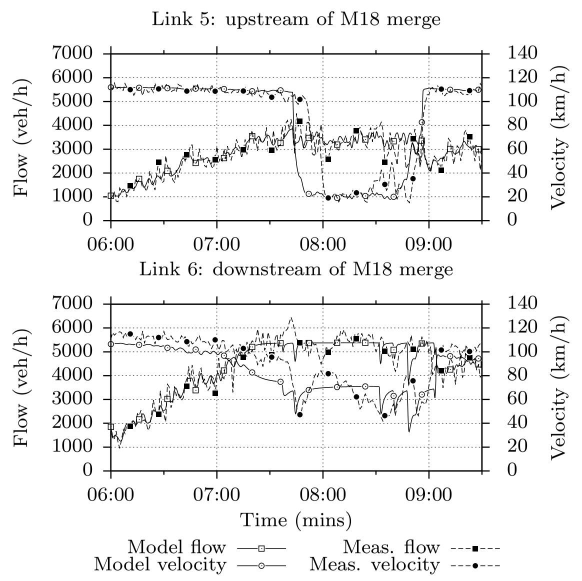
	1	8	15	Avg.
Calibrated 1	2982186	5051910	4296714	4110270
Calibrated 8	5279704	1893166	2119759	3097543
Calibrated 15	4977353	6370133	2564983	4637490



(a)

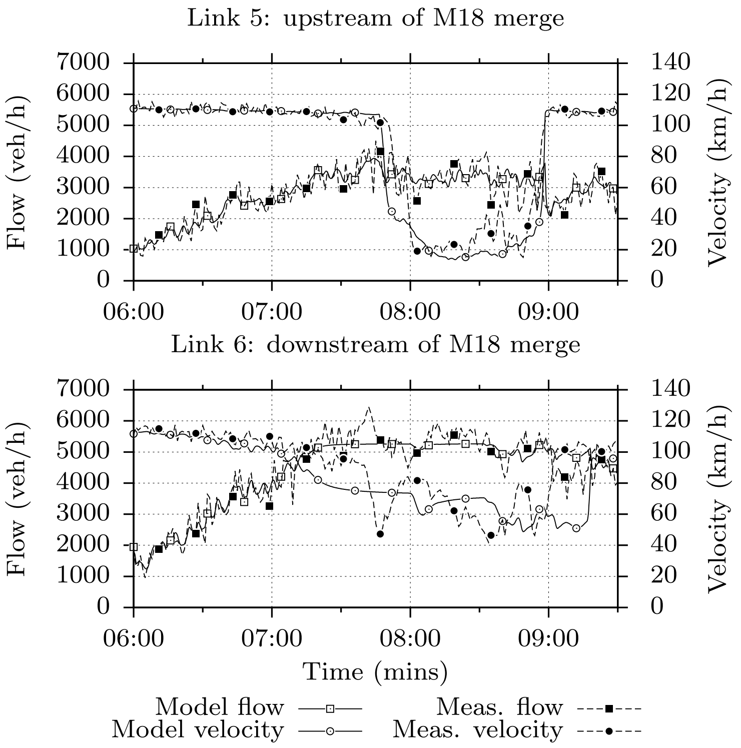


(b)

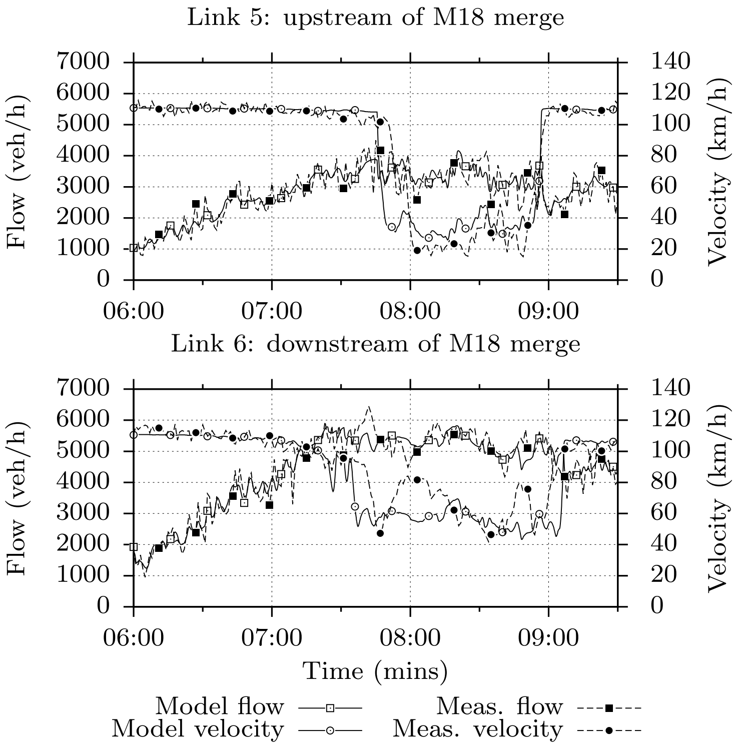


(c)

Figure 8.25: Sample of time profiles for the calibrated Sheffield CTM: (a) S1st (b) S8th (c) S15th.



(a)



(b)

Figure 8.26: Verification of the Sheffield CTM using input data from set S15th and using the optimal parameter set based on (a) the S1st data set and (b) the S8th data set.

can be seen when used for data of the 15 the model of the 8th correctly predicts the speed during the start of the time horizon and during the congestion region the flow is not capped. The model of the 1st does not perform as well when applied to the 15th, it correctly predicts the spill back of congestion upstream of the merge but the initial speed drop in Link 7 is not a close match, the capacity of the link is too low and the speed drop is predicted too early and the sharp peak drop in speed at 7:45 is smoothed out, after this time the predicted speed at this location is too low and the recovery comes 10 minutes late. The model of the 8th does correctly predict the speed drops during the congested region when applied to the 1st but it propagates the congestion further upstream hence the increase in the objective value.

The CTM of the 8th provides a good generalisation to the Sheffield site. This is very different to Heathrow where the CTM failed in verification. The key difference is that there are good data for all destinations. Without getting the destinations correct it will be very difficult to find a set of parameters that gives a model that could be used in any meaningful manner. The calibration of the 15th is not a good set of parameters although it gives a reasonable result it actually is worse than the parameter set of the 8th a model that was trained on different data. This highlights the complexity of the solution space, which also changes depending on the data applied.

8.5 Model comparison

This chapter has looked at two different models for two different sites. The Heathrow site was able to captured by METANET and with a small adjustment a verified model was produced. For CTM it could not cope with the difficulties imposed by the lack of suitable data to set its receiving functions, and a valid model was not able to be obtained. This section will look in detail at comparing the outputs of the two models obtained for the Sheffield site where the CTM was able to generate a model that generalised well.

In calibration both models performed well but the difference in the objective functions is considerable (see Tables 8.2 and 8.9) with METANET having a 75–100% larger objective values than the corresponding CTM. Note that for Heathrow METANET produces better results in calibration due to the issues CTM faced with the data. This makes it appear that the CTM is a much better model but in fact the differences are not as significant as it appears from the objective value alone. Space time profiles for the calibration of the 8th is shown in Figure 8.27 and the 15th in Figure 8.28. Remember that the white bars in the data panel show regions where no measurements were available. From these it can be seen that the congested off-ramp has a considerable impact on CTM model for the data of the 8th, for the model

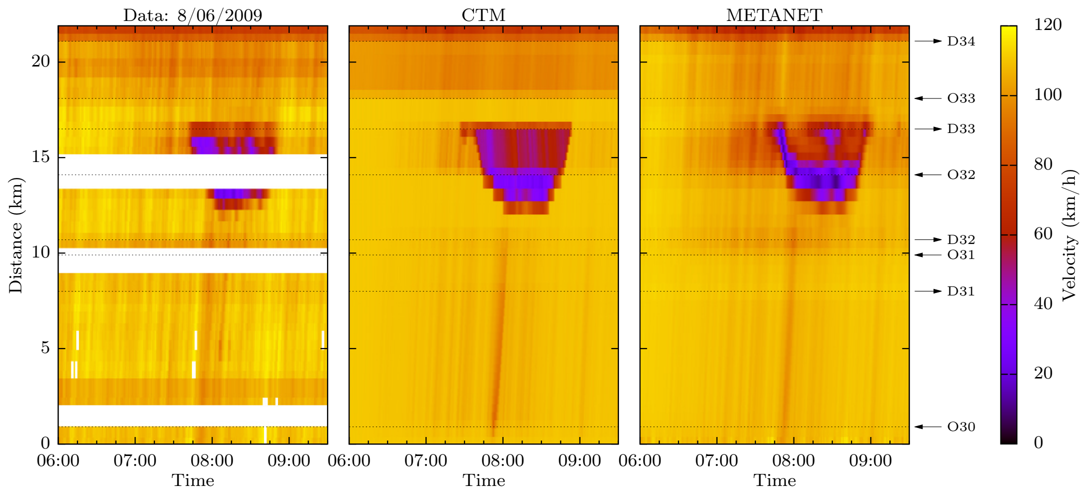


Figure 8.27: Space-Time profile of calibrated Sheffield model using data of the 8th.

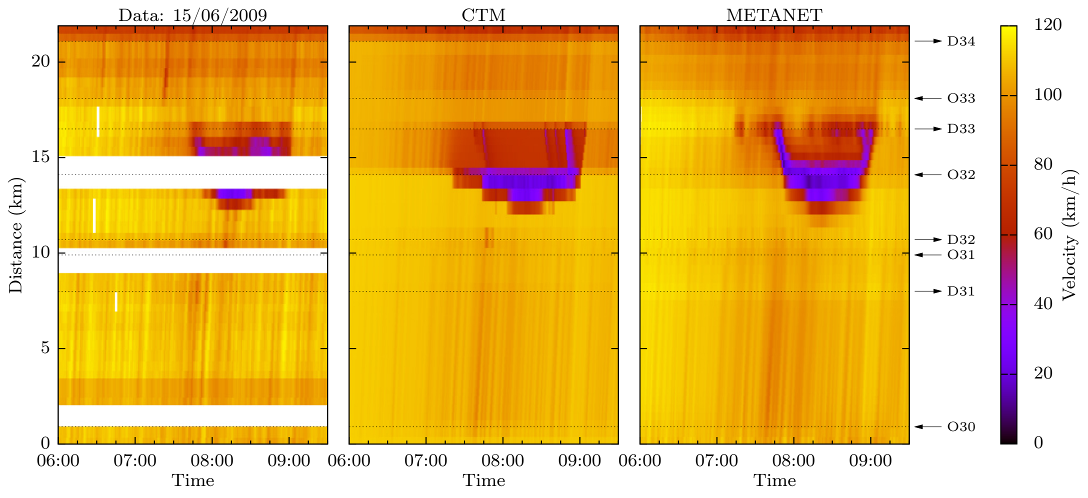


Figure 8.28: Space-Time profile of calibrated Sheffield model using data of the 15th.

of the 15th where the destination is not restricting the flow but the link parameters the speed reduction for the congested region does not form a sharp block and speed changes are more gradual. In the corresponding METANET model there is only an impact of this congested off-ramp when very high densities were present in the boundary data, so the there is not a solid block of speed reduction but shocks are started, with speed recovery in-between. The difference in the objective functions values comes from small error between the predicted and measured speed throughout that end up in a significant difference in the final objective function value. In the METANET model the speed recovery in-between the sharp shocks caused by spill back is large. The plot of the 8th shows clearly the FD change locations with the step change in the colour, moving from yellow to orange at the O33 node and orange to red after D34. In the METANET model the FD changes are not sudden step changes in speed; due to its explicit speed equation, gradual transitions occur.

The verification plots for the calibration of the 8th are shown in Figures 8.29 and 8.30. In verification the obtained Sheffield model was not able to replicate the correct dynamics. In both cases the model correctly starts predicting speed drops but once the merge of O32 happens very large speed drops and stop-and-go waves form. In the CTM model this problem does not occur. CTM handles merges by use of merging rates (see section 3.4.3) to control congested flow. These were set to accept all of the incoming flow and restrict the mainline if required. For METANET

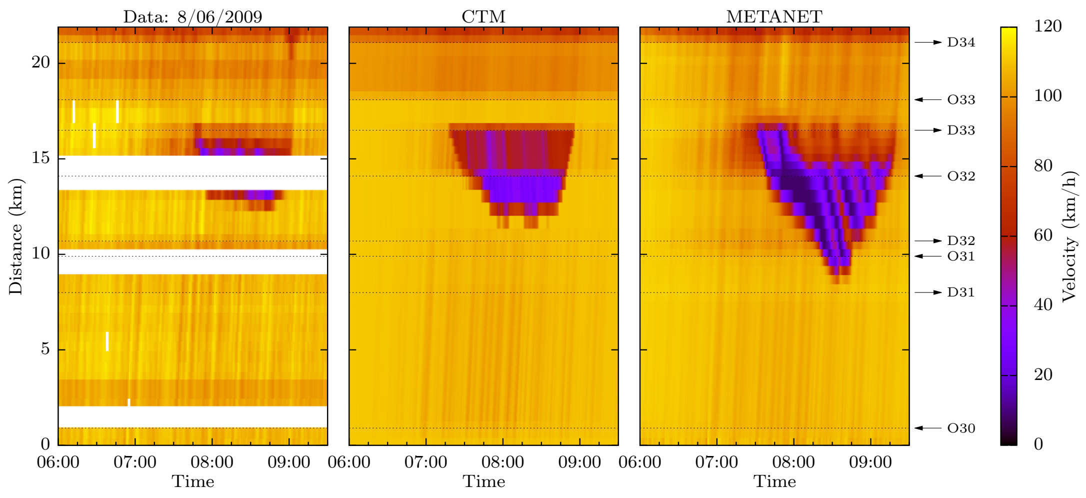


Figure 8.29: Space-Time profile of Sheffield verified against data of the 1st calibrated using data of the 8th.

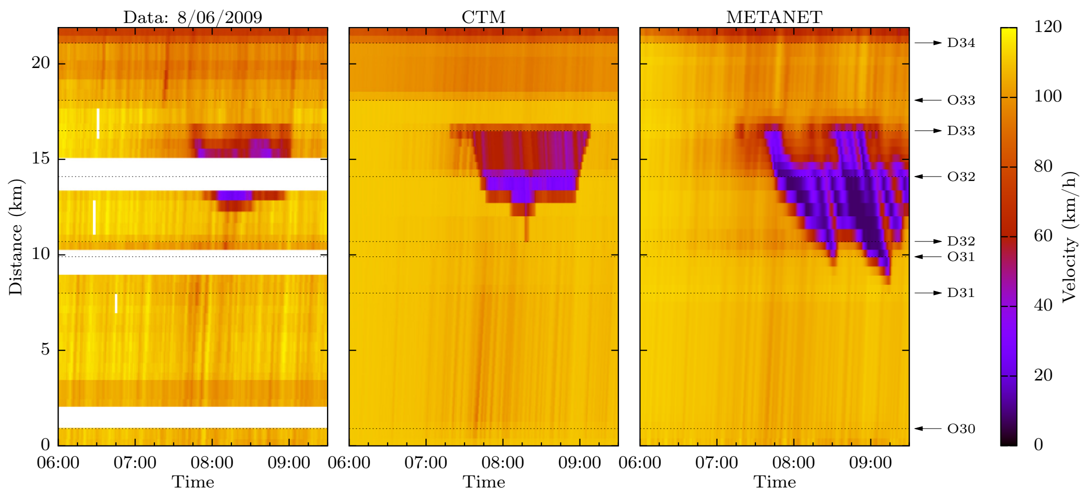


Figure 8.30: Space-Time profile of Sheffield verified against data of the 15th calibrated using data of the 8th.

there is an explicit merging term that further reduces the speed, which is controlled by the parameter ϕ that interacts on the first segment downstream of the merge. This is the point at which the stop-and-go wave occur that cause the congestion to propagate upstream and persist for a longer time period.

In Figures 8.31 and 8.32 the verification results for the models calibrated on the 15th are shown. When applied to the data of the 1st CTM is able to get the rough properties of the model correct but as already discussed this parameter set is not correct. The predicted speed decreases early at a gradual rate, rather than being a sudden deceleration at a later time as shown in the data. The METANET model

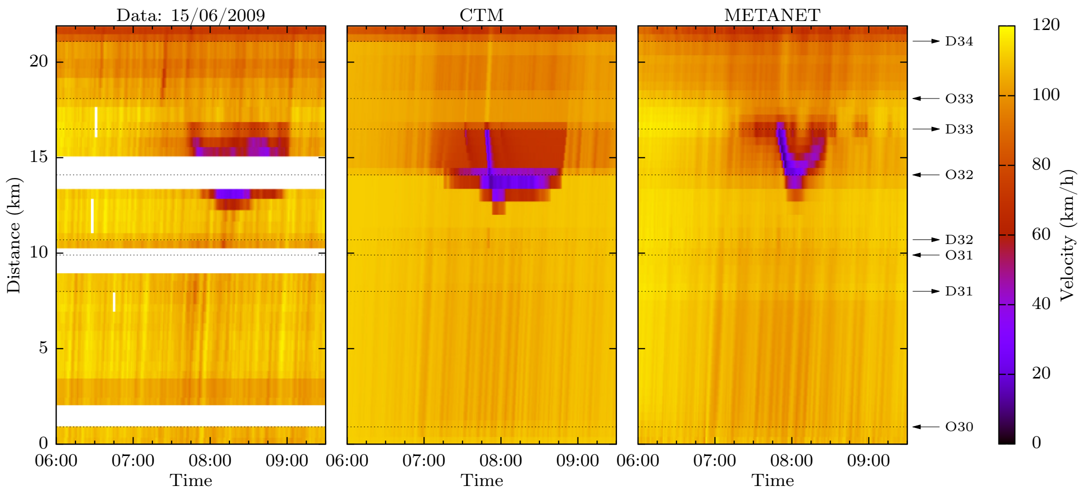


Figure 8.31: Space-Time profile of Sheffield verified against data of the 1st calibrated using data of the 15th.

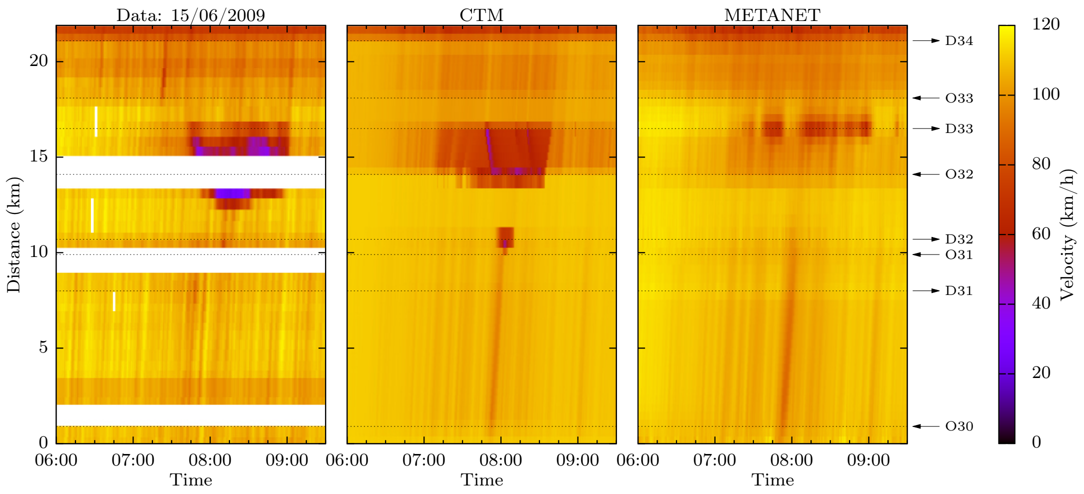


Figure 8.32: Space-Time profile of Sheffield verified against data of the 8th calibrated using data of the 15th.

only shows a small period of congestion and does not persist for as long as measured. This parameter set has a small ϕ compared to the parameter set from the 8th and does not create further congestion at the merge. When applied to the 8th both models fail to correctly predict the congestion. The CTM model still shows a slow down but the amount of speed reduction is too small whereas METANET has some small slow downs at the destination. It is also interesting to note that in this case the selected receiving function for D32 causes a small speed drop at 8:00, this is probably due to the verification data having larger demands.

Another difference between the two models is the run time. The CTM only has a single governing equation to evaluate as such it runs in nearly half the time of the METANET model. The calibration of the CTM for Heathrow was around 15 minutes while METANET took 23 minutes. For Sheffield the CTM took around 34 minutes and METANET around 64 minutes.

8.6 Conclusion

This chapter has discussed the application of population based optimisation methods to the problem of macroscopic traffic flow model calibration and verification. The mapping of FDs shows promise and the models produced create solutions that have physical meaning based on the real world network. This allows expert engineering opinion to be removed from the initial calibration effort.

Ten different algorithms have been implemented and evaluated for the METANET model. A simple GA, two Cuckoo Search variants and seven PSO algorithms. It is clear that the PSO family outperforms the rest. Surprisingly, one of the most efficient of the PSO algorithms is LPSO, which manages to converge faster to very good solutions for the stand alone calibration problem compared to more recently proposed PSO variants. However, it is the solutions provided by HEPSO that generalise better for the particular verification problem.

Due to PSO algorithms providing the best results only these were analysed for the CTM. The same patterns emerged with the best variants being HEPSO and LPSO. These algorithms find solutions with the lowest objective function values and also show consistency across datasets and experiments having low standard deviations. The adaptive PSOs appear to restrict the search and do not provide an improvement over LPSO. HEPSO does have adaptive parameters but it appears that its extra search operators allow it perform well.

For the Heathrow METANET calibration all results achieved a similar FD assignment pattern through AAFD without user input. In verification of these datasets a manual adjustment of the critical density of the final link was required to create a valid model. At the Sheffield site again a consistent pattern is seen for the FD assignment pattern. The obtained parameter values, however, do not provide a valid model and in verification congestion is over predicted or not present.

For the CTM at both sites a consistent FD assignment is achieved but the receiving functions at destinations does not show the same consistency. For the Heathrow site the identified parameters do not produce a valid model as a set of generalised BC parameters is not identified. For the Sheffield site this is not the case. The key destinations in the model do show consistency in the receiving functions, and those that show high variation have low demands. These low demand destinations so do not experience the full range of traffic conditions therefore the optimisation algorithms are unable to identify consistent parameters. For this site the models obtained from each dataset do give have a valid set of parameters and reproduce the congestion seen in the other datasets.

When comparing the two models it can be seen from the space-time plots that both work equally well in the calibration datasets. The METANET models do show higher square error values due to small changes throughout time and space adding up to what appears to be a significant difference. This shows there is scope in revising the objective function. In verification the CTM is dependant on correct BCs being found which did not happen at the Heathrow site so a valid model was not obtained, the METANET model did not appear to have this issue. The situation is reversed at Sheffield, in the CTM a verified set of parameters is obtained. While for the METANET model the congested destination D33 and merge at O32 is not correct as in verification the identified models produce no congestion or stop-and-go waves are produced the propagate too far upstream and dissipate too late.

This chapter managed to create a valid METANET model for the Heathrow site and a valid CTM for Sheffield without prior selection of FD assignment. The Heathrow site failed for generalisation in the CTM model due to data not being available to the optimisation algorithm to reliably identify correct receiving functions at the destinations. Due to the limitations of the Heathrow site it is not considered further in this thesis. The METANET model for Sheffield also failed in generalisation and this will be addressed in the next chapter. Errors seemed to come in from the way the METANET model with the parameters identified handled the congested off-ramp and the merge into the congested traffic. As well as looking at the Sheffield model in more detail the next chapter will also expand to gradient based optimisation.

Chapter 9

Sheffield METANET Model Validation: Particle Swarm Optimisation and Resilient Back-Propragation

9.1 Introduction

In the previous chapter a variety of EAs were analysed for two different traffic simulators at two sites. The focus of this chapter is try and improve the results of the Sheffield METANET model and also to develop and test a gradient based optimisation of that particular combination of simulator and site. This chapter details some model changes for the Sheffield site to try and allow for a METANET model that works in calibration and also capture the generalisation property so that it correctly predicts the site conditions for data other than that used for calibration. A further consideration is the alteration of the objective function to assist in this regard and also reduces the complexity of gradient calculation. Details are presented on the best selection of the restart parameter for RPROP that is important to help it escape from local minima and search the entire solution space. Results are then given for a selection of PSO algorithms and these are compared with solutions obtained from a gradient based optimization utilising the RPROP algorithm. The PSO algorithms being considered in this chapter are LPSO, HEPSO and APSO-09, the latter is included as it was one of the better APSOs tested in the previous chapter even though it did not perform as well as the other two variants being used here. From this point forward APSO-09 as the only adaptive PSO considered will be referenced to as APSO.

The chapter starts off with an investigation into two restart methods for the

RPROP algorithm which are detailed in 5. The investigation involves adjusting the interval at which restarts occur using two different algorithms. The best result from this is then used for the RPROP algorithm that is used in the remainder of the chapter. This study is followed by the problem formulation used including the objective function and alterations to the Sheffield site.

The main results of the calibration using RRPOP and PSO algorithms are then detailed with analysis of the convergence profiles, identified parameters, overall solution quality and the additional information that can be obtained from the model by using AD. Finally a discussion is given about how the models generated by both classes of algorithms perform in verification for data sets that were not used by the optimisation algorithms.

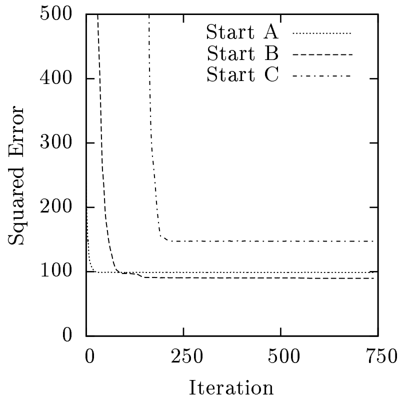
9.2 RPROP Restart Parameter

One important aspect of the RPROP algorithm is its restart parameter. This controls how often the search is reset to the best location found and the parameters for the step lengths reduced to improve the fine tuning of the search. A small scale investigation was performed using a multi-start algorithm with 3 initial points.

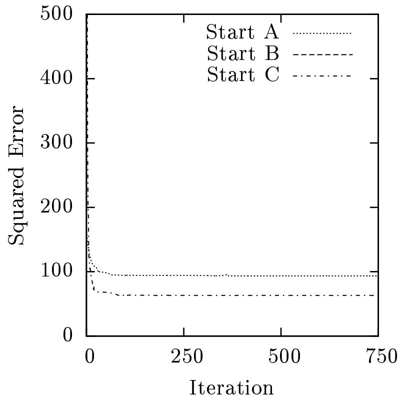
In this study only the squared error is considered and AAFD is not applied and each link is supplied its own parameters. Figure 9.1 shows the convergence profile of a Sheffield site for varying values of $\mathfrak{c}_r$ which is the number of iterations between restarts (c.f. Section 5.5.4, pp. 85). Figure 9.1 shows for three randomly selected starting locations the objective function value at the current iteration for each point. In Figures 9.1a–9.1c the it can be seen that for each point after initial convergence rarely shows an observable improvement in the current objective function value. There are a couple of exceptions, in Figure 9.1a Starting location C makes some small improvements. For scaling purposes the graph does not show the full range of squared errors but the solution converged for start C at 100 iterations before an improvement occurred. In Figure 9.1b no improvements are seen beyond the initial convergence of the algorithm and the restarts have no influence on the algorithm. Start point B converges to a suboptimal solution with a squared error $\approx$ 2200 of and never manages to escape that local minimum. With a longer restart interval less interruptions are made to the algorithm. In 9.1c gradual improvements are seen for start B which did not happen for the restarts using a shorter interval. However, all the start locations converged to a reasonable solution, with a restart interval of 30 one of the start points became stuck and showed no improvement. A compromise needs to be achieved between a long restart interval to try and avoid disruptions to the algorithm and a short interval that allows for a more refinement within the search.

The chaotic restart period (5.50) is shown in Figure 9.1d with values $\mathfrak{c}_{r,\min} = 10$ and $\mathfrak{c}_{r,\max} = 40$. Start point B still exhibits the gradual improvements whilst start point A ended up in exhibiting a phenomenon. The solver could not get to the minimum of the region it had found and entered an oscillatory phase. It is thought this is due to the nature of the solution space at this specific location as even after restarts (some can be seen in graph as spikes in the squared error) this pattern remained. For a brief period the current objective value was stable, before a restart occurred restarting the oscillating pattern.

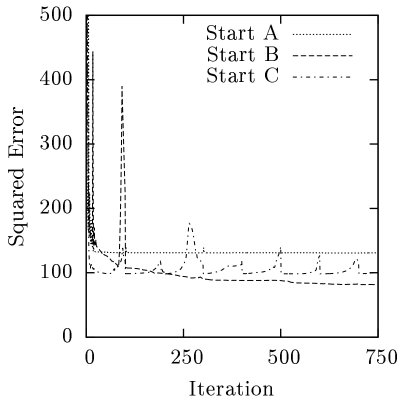
In all of tests shown in Figure 9.1 the final solution value is not of extreme importance. The study is aimed at trying to ensure the algorithm is capable of exploring the whole of the solution space. The best solution was actually obtained by start point C in the trail with $\mathfrak{c}_r = 30$. In this case it was mainly due to initial convergence to a good solution and the restarts then helping make the search more fine tuned. The current restart scheme is good at exploiting current optima but it does not move the solution away from the current point sufficiently. This can



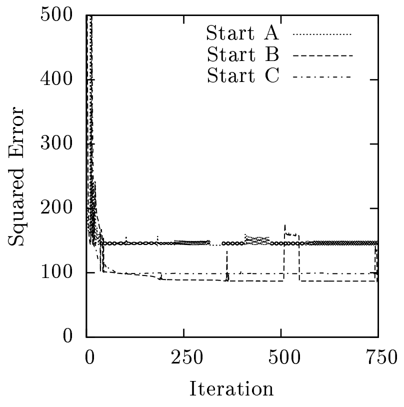
(a) $\mathfrak{c}_r$=10.



(b) $\mathfrak{c}_r$=30.



(c) $\mathfrak{c}_r$=100.



(d) Chaotic $\mathfrak{c}_r$.

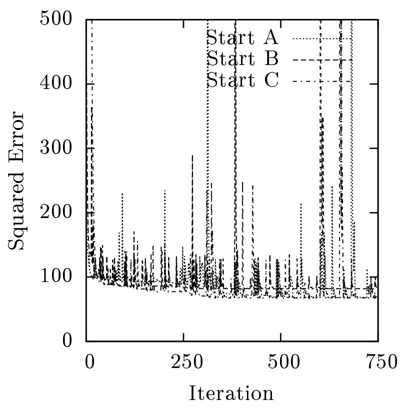
Figure 9.1: RPROP convergence profile of Sheffield model for various values of $\mathfrak{c}_r$. Restart using Algorithm 5.5.

be seen as very few spikes can are observed in the plots showing a change caused by a restart. A few do exist i.e. start point C for $\mathfrak{c}_r = 100$ and start point B of the chaotic restart. This restart scheme does not allow for escaping from a local minimum and is more suited to a fine and precise search.

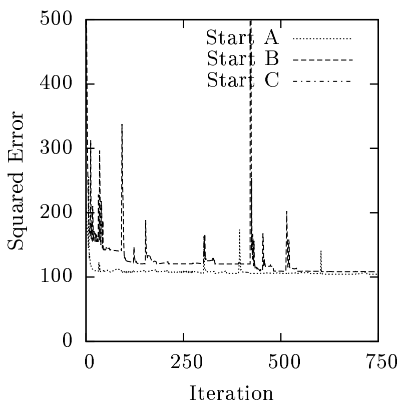
The results for $\mathfrak{c}_r = 10$, $\mathfrak{c}_r = 30$ and $\mathfrak{c}_r = 100$ compared to a chaotic interval using the restart scheme as outlined in Algorithm 5.6 are shown in Figure 9.2. For all of the restart intervals it can be seen that the new restart method (c.f section 5.5.4) is more aggressive. The original restart method only allowed for a contraction of the search and does not help in moving the solution out of local minima. By randomly resetting the step length back to the initial value δ_0 (see pp. 84) it is possible that the search can move out of its current position and away from the current optimum as shown by the spikes in the objective value when a restart occurs. The new method allows for a sub-optimal solution to move into the correct region for start point C in the chaotic restart test, initial converge occurred with a squared error in the region of 5400 but after the restart around iteration 280 the solution improved such that it can be seen in the plot. Through a series of successful restarts it manages to move towards a better local minimum. In the trial using a restart interval of $\mathfrak{c}_r = 30$ the solution from start point C initially converges to a squared error value ≈ 5000, this is far from the ideal solution and over time with various restarts the algorithm does move towards more optimal solutions the best value found is ≈ 2000. This still does not find the correct region of the domain to search however unlike the original reset scheme improvements are made. Some starting points may be far from the optimal and this is why a multi-start approach is are used. But the result does highlight the current scheme's ability to move the solution out of a local minimum. All of the results shown in Figure 9.2 show less stagnation than those in Figure 9.1 but also less variation in the objective values found at each iteration. As with the original restart period no definite best value for the restart period exists. As such the chaotic scheme will be used in all further schemes to try and use the associated benefits of a short/long restart intervals even though the results here show no definitive improvements.

9.2.1 Problem formulation

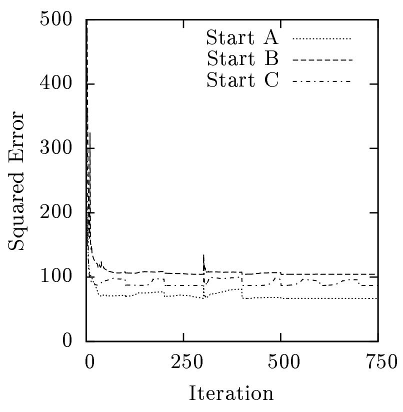
In the previous chapter the objective function (4.17) considers the squared error between each of the macroscopic traffic variables. However as these are related, w_ρ was set to zero and densities were not considered in the optimisation. Here the objective function being used only considers speed as shown in (4.21) and repeated



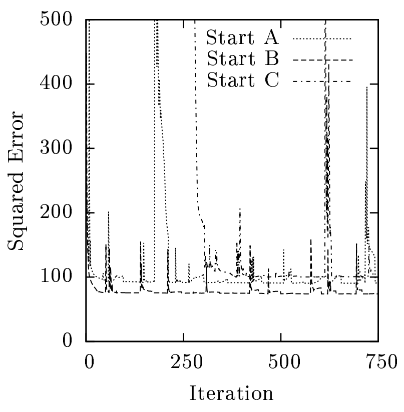
(a) $\mathfrak{c}_r$=10.



(b) $\mathfrak{c}_r$=30.



(c) $\mathfrak{c}_r$=100.



(d) Chaotic $\mathfrak{c}_r$.

Figure 9.2: RPROP convergence profile of Sheffield model with modified restart algorithm for various values of $\mathfrak{c}_r$. Restart using Algorithm 5.6.

here for ease of reference,

$$J(\mathbf{z}) = \frac{1}{KY}\sum_{k=1}^{K}\sum_{j=1}^{Y}\left(E_{v,j}\left[\mathbf{x},\mathbf{y},k\right]\right)+w_p\sum_{m=1}^{M}\sum_{\substack{\mu\in O_{a_m}\\ \mu\notin\Phi}}\left(\mathbf{z}_{\mathrm{L},m}-\mathbf{z}_{\mathrm{L},\mu}\right)^{\top}w_{\mathrm{L}}\left(\mathbf{z}_{\mathrm{L},m}-\mathbf{z}_{\mathrm{L},\mu}\right). \quad (9.1)$$

The reasons for using this formulation are discussed in Section 4.2.1. It has scaling based on the model size. This may make models of different sizes and time horizons be directly comparable. Furthermore, it only considers speed meaning that for RPROP only the speed terms of the Jacobian need to be calculated. The density and flow terms in vector $\left\{\frac{\partial J}{\partial \mathbf{x}}\right\}$ are set to zero and have no impact. Therefore,

$$\frac{\partial J}{\partial \mathbf{z}} = \mathbf{J}_v \frac{\partial J_{\mathrm{s}}}{\partial \mathbf{x}_v}^{\top} + \frac{\partial J_{\mathrm{p}}}{\partial \mathbf{z}}. \quad (9.2)$$

where $\mathbf{x}_v$ is the speed state vector and $\mathbf{J}_v$ the Jacobian of the model's speed states. This reduces the problem significantly as $\mathbf{J}_v$ is one third the size of $\mathbf{J}$, with the

Table 9.1: Traffic flow model parameters upper and lower limits.

Variable	τ	κ	ν	$v_{\min}$	$\rho_{\max}$	δ	ϕ	α_m	$v_{f,m}$	$\rho_{cr,m}$
Maximum	40	30	80	8	190	4	4	3.5	130	45.0
Minimum	1	5	1	0.5	160	5E-5	5E-5	0.5	60	18.0

calculation of the model Jacobian using AD being one of the most intensive parts of the calculation.

For all algorithms considered in this chapter the parameters' upper and lower bounds are given by Table 9.1. The objective function being used is that which is given by (9.1) where the components of w_{L}: $w_{\mathrm{L},v}$, $w_{\mathrm{L},\rho}$ and $w_{\mathrm{L},\alpha}$ are set to 0.001, 0.0015 and 1.0. As this objective function is now scaled differently to the previous chapter the penalty term weight is reduced and w_p is set to 5.0.

APSO, LPSO, HEPSO and RPROP are initialised using Latin hypercubes. Each decision variable is assumed to have a uniform distribution within its range, as defined in Table 9.1. For APSO, LPSO and HEPSO those limits are handled by moving solutions that are outside the domain, back to the boundary. PSO algorithms have a population size set equal to 30. Other parameters are those given by the papers they were proposed in. The probabilities for the artificial bee colony and GA operators used by HEPSO are the same as those in [75]. For LPSO the results were obtained by using the parameter values suggested in [103]. These values are listed in Table 5.4.

RPROP starts from 6 initial points which are updated simultaneously, as if it was a population based algorithm. The number of starting locations was selected so that an iteration of each algorithm (RPROP vs PSO) takes a similar amount of time. The initial step length δ_0 is set to be 1/5 of the variable limits shown in Table 9.1.

9.2.2 Sheffield Model Configuration

The Sheffield model as described in Section 6.4.3 was used in the previous chapter. This model when used with the CTM was successfully verified but the performance for the METANET simulator was not as good. The calibrated model was worse, and when the identified parameters from calibration were applied to other datasets the model did not replicate the correct conditions. The main issues stemmed from the congested destination D33 and the origin O32 feeding in flows from the M18 motorway. On further inspection of the loop detectors and the real-world road layout at these locations it is evident that the flows from the M18 motorway are actually fed into the main highway at two different locations with a long filter lane running parallel to the main highway. This may cause an adverse impact to the METANET model due to the merging term that reduced the speed in the first segment of link m upstream of merging link μ by $-\delta T q_\mu(k) v_{m,1}(k)/(L_m \lambda_m(\rho_{m1,1}(k) + \kappa))$. The speed drop caused by a single high flow merge is greater than a split across two separate merges. The first merge will drop the speed and consequently cause an increase in density which reduces the impact of the second. The loop detector at D33 is also a short distance from the diverge from the main carriage way (approx 200 m) and captures propagating queues from the downstream urban network (a signalled roundabout). The high densities measured therefore impact directly on the mainstream without the remaining space on the off-ramp being considered. This was not an issue for CTM as it could select a receiving function for the destination to provide the correct dynamics at the node.

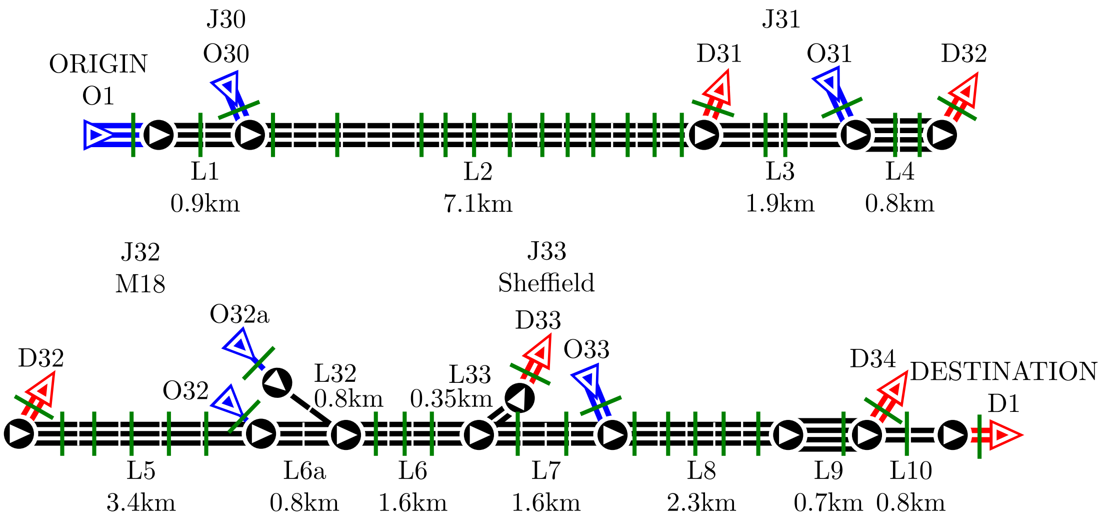


Figure 9.3: Schematic of the Sheffield model, including the location of available loop detectors.

A schematic of the updated model is shown in Figure 9.3, the origin O32 from the M18 motorway has been split in two, as has Link 6. With the flows from the right hand lane O32 being input at the start of the Link 6a which starts where Link 6 did in the original model. The left hand lane of the origin O32a is merged at the new node, after travelling along a new link L32 that is included to account for the travel time of the flow along the filter lane. The off-ramp at Junction 33 is also included in the model so that the additional capacity of the destination can be accounted for due the measurement of the Boundary being offset from the divergence point of the main carriage way.

The addition of the extra links to apply the correct BCs means that AAFD (see 4.2.4) will now split the highway into three linear sections. One that starts at L1 and ends at D1. Another for the on-ramp at Junction 32 with the M18 slip road which is a single link, L32. Again another single link section, L33, is included for the off-ramp at Junction 33.

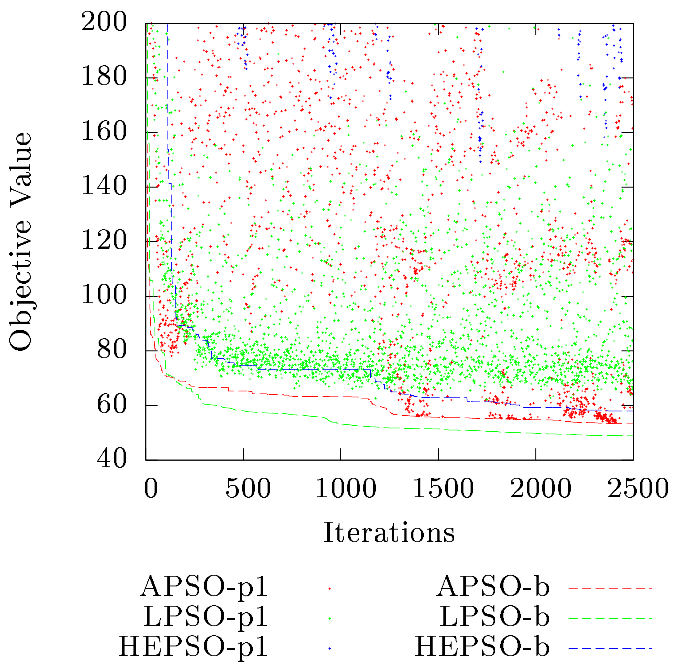
This change to model increases the number of parameters required to define the model by 6 as two extra links have been added. This means that for the PSOs the problem vector consists of 53 parameters. Now due to the implementation of RPROP the length parameter is set as 1 and not used so it actually has a smaller problem vector as the 10 parameters to define the FD length for AAFD can be omitted, so for RPROP the dimension of the problem is 43.

9.3 Model Calibration

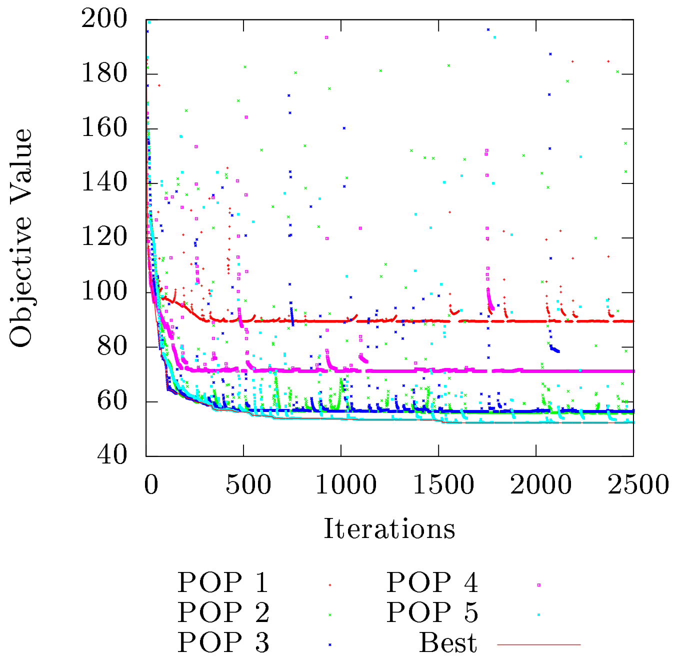
9.3.1 Convergence Profiles

Three data sets were selected for calibrating the model, from July 1st, 8th and 15th, 2009. The same METANET model was calibrated using APSO, LPSO, HEPSO (EAs) and RPROP. Figures 9.4(a) and (b) depict the convergence profiles for the EA and RPROP, respectively, when solving the calibration problem using the 15th's data. Subfigure (a) shows the objective function value over the number of iterations of particle number 1 in the swarm for each of the three EA. The corresponding algorithm's best of the whole swarm is depicted using lines. Particle number 1 achieves the best only for APSO after about 1,300 iterations, hence the concentration of points near the best line. For the other two algorithms, this specific particle does not achieve the optimum; the solid line (best particle) is produced by one of the untraced particles in the swarm. The erratic nature of the EA's search is clearly shown.

The LPSO and HEPSO algorithms required 75,030 function evaluations; for APSO this number is not static and approximately 76,500 were required for 2,500 iterations. Using a parallel implementation on a Intel i5 3.1 GHz quad core processor LPSO and HEPSO took approximately 30 minutes run time; due to the extra function evaluations needed by APSO 33 minutes were required. RPROP with 6 initial points requires 15,012 function evaluations for 2,500 iterations, which takes approximately 92 minutes to complete. The ADOL-C routines were not optimised and speed up of the implementation is possible, and has been discussed in Chapter 7. The difference in run times is due the complexity of the objective function evaluation; it takes 8.4 seconds for RPROP to perform 24 objective function evaluations using a parallel implementation on the quad core system. For the same number of evaluations PSO algorithms require 0.36 seconds on average. METANET with ADOL-C is slower when performing a single evaluation because the gradients are computed along with the model outputs and there is the large matrix multiplication



(a) APSO, APSO and HEPSO particle 1 of the swarm and best value (*algorithm*-b).



(b) Multistart RPROP from five initial points and the current best.

Figure 9.4: Convergence profiles for the data of the 15th. "Best" lines how the minimum value of the objective function found up to the current iteration.

Table 9.2: Calibration results for J and J_s.

	Algorithm							
Calibration	**APSO**		**LPSO**		**HEPSO**		**RPROP**	
day & repeat	J	J_s	J	J_s	J	J_s	J	J_s
1st								
Repeat 1	56.54	**55.84**	55.17	**54.20**	66.76	66.00	48.50	39.66
Repeat 2	74.34	73.40	57.51	56.22	54.85	**53.14**	49.21	**36.96**
Repeat 3	59.04	57.93	59.43	58.86	67.33	65.59	51.37	40.47
8th								
Repeat 1	58.67	**55.63**	45.05	42.09	72.70	66.41	43.27	36.96
Repeat 2	59.34	56.21	46.00	42.99	74.41	70.87	42.91	36.55
Repeat 3	58.80	55.94	40.69	**38.36**	78.81	**66.12**	43.36	**36.53**
15th								
Repeat 1	53.22	**51.45**	52.74	51.08	76.48	75.12	52.17	33.73
Repeat 2	57.41	56.49	48.86	**47.62**	65.04	62.51	52.37	**32.13**
Repeat 3	53.31	51.66	51.66	50.13	58.00	**55.36**	62.21	38.92

performed for obtaining the gradient.

On the other hand, RPROP has a smoother, more targeted profile as can be seen in Figure 9.4(b). This reflects the nature of the gradient based search performed. The spikes observed at each trajectory, are due to the restart. Because of the chaotic restart used, they occur at seemingly random iterations. RPROP does not require the large variance sampling that the EAs use and is able to converge to the neighbourhood of the optimum faster.

The total run time of the two different algorithm classes varies significantly. The PSOs took around 32 minutes with no differences detectable between the variants. RPROP is a lot more involved with the gradient computation increasing the runtime significantly as shown in Chapter 7. The total runtime for RPROP was around 75 minutes.

9.3.2 Identified Parameters

Table 9.2 shows the objective function value (4.21) for each run of each day's data set and the corresponding square error averaged over time and number of detector stations, i.e. the first term of (4.21). Since J_s is a square error, the resulting mean absolute error is in the order of 6 to 7 km/h.

With the exception of HEPSO for the data of the 8th, the best result for J achieved by any of the PSOs is also the best result with respect to the mean square error J_v. This is not true for RPROP as the best J_v does not correspond to the best J. The penalty term for the optimal solution is larger in RPROP. This is to be expected, since RPROP has a larger number of degrees of freedom in the form of available decision variables representing a more detailed profile of model parameters. RPROP has 13 different fundamental diagrams at its disposal, which can be tuned, whereas the swarm algorithms are restrained to a maximum of 7 for the mainstream plus 2 separate ones for L32 and L33. In other words, in addition to the global parameters, which are the same for both types of algorithms, RPROP has 39 parameters (v_f, ρ_{cr} and α) for the available fundamental diagrams and PSO a maximum of 27. In fact, PSO algorithms converge to a solution where not all available fundamental diagrams are used. The difference in performance is quite small, since the mean absolute error remains in the area of 6 to 7 km/h.

The optimal solutions achieved by each algorithm for each of the data sets indicated by the bold font at Table 9.2 are given by Tables 9.3–9.6. The resulting spatial distribution of the capacity, free speed, critical density and α given by the optimal solutions obtained using the data of the 15th are depicted in Figures 9.5–9.7.

Table 9.3: Global parameter set part of the optimal solutions with respect to J_v at Table 9.2.

Algorithm and day	τ (s)	κ (veh/km/ln)	ν (km^2/h)	$v_{\min}$ (km/h)	$\rho_{\max}$ (veh/km/ln)	δ (h/km)	ϕ (h/km)
1st							
APSO	34.619	29.989	69.294	0.506	176.394	1.22392	0.38401
LPSO	27.184	24.421	62.902	6.075	163.780	0.00292	0.97145
HEPSO	25.273	30.000	49.954	0.500	182.882	1.21320	0.44915
RPROP	22.667	22.399	56.681	6.175	182.746	0.07998	0.00005
Average	27.436	26.702	59.708	3.314	176.451	0.63001	0.4511
8th							
APSO	10.014	22.692	37.396	6.947	171.584	1.19801	1.13019
LPSO	9.000	29.994	37.052	0.655	189.805	0.00012	0.37390
HEPSO	9.641	15.478	29.947	7.718	166.337	0.70086	0.00005
RPROP	11.389	29.174	41.964	7.998	173.297	0.00008	0.00005
Average	10.011	24.334	36.590	5.830	175.256	0.47477	0.37605
15th							
APSO	20.400	26.024	41.006	7.332	181.031	0.00017	0.68438
LPSO	20.943	29.688	48.248	5.875	189.843	0.00025	0.34124
HEPSO	25.120	13.157	41.305	8.708	155.088	0.57899	0.65611
RPROP	18.572	24.714	40.342	8.000	177.834	0.09266	0.00005
Average	21.259	23.396	42.725	7.479	175.949	0.16802	0.42045

It can be seen from Tables (9.4)–(9.6) and Figures 9.5–9.7 that APSO(1st), HEPSO(1st) and HEPSO(15th) converge to solutions that result to the same capacity assignment pattern. Two different FD are used to model the mainstream; the first one starts from link L1 (beginning of the site) and extends to link L7. The second one starts from L8 extends up to L10 (the site's end). HEPSO(1st)

Table 9.4: Fundamental diagram parameter parts of the optimal solutions for the data set of the 1st.

Algorithm	**FD start** (name)	**FD end** (name)	v_f (km/h)	ρ_{cr} (veh/km/lane)	α (–)	**Capacity** (veh/h/lane)
APSO	L1	L7	114.66	24.714	2.677	1951
	L8	L10	107.88	22.574	2.657	1671
	L32	L32	114.57	29.396	2.663	2314
	L33	L33	116.77	30.447	2.681	2449
LPSO	L1	L5	112.21	28.197	2.668	2175
	L6a	L10	111.19	23.967	2.704	1841
	L32	L32	112.90	28.463	2.663	2208
	L33	L33	115.84	32.422	2.736	2606
HEPSO	L1	L7	113.88	25.671	2.609	1993
	L8	L10	106.08	24.256	2.543	1737
	L32	L32	110.88	20.454	2.930	1612
	L33	L33	103.23	27.696	2.650	1960
RPROP	L1	L1	112.71	27.365	2.698	2129
	L2	L2	112.63	28.958	2.683	2247
	L3	L3	108.39	28.165	2.696	2107
	L4	L4	103.60	28.092	2.706	2011
	L5	L5	106.61	28.631	2.717	2112
	L6a	L6a	125.79	23.938	2.702	2080
	L6	L6	113.10	23.510	2.698	1835
	L7	L7	118.38	26.472	2.691	2161
	L8	L8	100.59	26.493	2.690	1838
	L9	L9	113.76	26.508	2.700	2082
	L10	L10	83.51	24.009	2.696	1384
	L32	L32	102.98	25.920	2.696	1842
	L33	L33	125.79	31.860	2.696	2766

Table 9.5: Fundamental diagram parameter parts of the optimal solutions for the data set of the 8th.

Algorithm	**FD start** (name)	**FD end** (name)	v_f (km/h)	ρ_{cr} (veh/km/lane)	α (–)	**Capacity** (veh/h/lane)
APSO	L1	L9	112.23	23.399	2.864	1852
	L10	L10	91.08	24.829	2.724	1567
	L32	L32	115.55	24.264	2.838	1971
	L33	L33	104.55	27.719	3.062	2091
LPSO	L1	L3	112.53	29.889	2.574	2281
	L4	L7	113.50	23.657	2.467	1790
	L8	L9	104.69	27.843	2.446	1937
	L10	L10	90.09	27.265	2.336	1601
	L32	L32	116.99	27.265	2.479	2131
	L33	L33	112.13	25.847	2.625	1980
HEPSO	L1	L2	112.67	35.766	2.275	2596
	L3	L3	111.11	29.986	2.407	2199
	L4	L7	113.71	22.923	2.556	1763
	L8	L10	101.18	31.830	2.274	2075
	L32	L32	107.78	31.362	2.062	2081
	L33	L33	105.37	30.530	2.125	2009
RPROP	L1	L1	116.52	26.531	2.265	1988
	L2	L2	115.52	30.264	2.236	2235
	L3	L3	113.54	27.525	2.252	2004
	L4	L4	109.12	24.626	2.261	1727
	L5	L5	114.99	21.082	2.276	1562
	L6a	L6a	111.42	23.672	2.278	1700
	L6	L6	115.31	23.696	2.285	1764
	L7	L7	117.50	26.335	2.267	1991
	L8	L8	105.29	27.230	2.261	1842
	L9	L9	111.81	28.433	2.260	2042
	L10	L10	83.41	28.078	2.258	1504
	L32	L32	123.15	26.015	2.277	2065
	L33	L33	125.79	26.085	2.279	2116

Table 9.6: Fundamental diagram parameter parts of the optimal solutions for the data set of the 15th.

Algorithm	**FD start**	**FD end**	v_f	ρ_{cr}	α	**Capacity**
	(name)	(name)	(km/h)	(veh/km/lane)	(–)	(veh/h/lane)
APSO	L1	L7	112.24	28.155	2.651	2167
	L8	L9	110.47	24.808	2.641	1877
	L10	L10	95.44	23.078	2.448	1464
	L32	L32	113.82	27.641	2.656	2159
	L33	L33	107.41	28.510	2.534	2064
LPSO	L1	L4	111.95	28.376	2.673	2185
	L5	L7	112.17	24.383	2.693	1887
	L8	L10	101.93	23.843	2.645	1665
	L32	L32	111.00	27.813	2.703	2133
	L33	L33	112.17	24.383	2.693	1887
HEPSO	L1	L7	114.91	24.666	2.710	1960
	L8	L10	102.07	25.583	2.682	1799
	L32	L32	115.20	30.074	2.730	2402
	L33	L33	103.97	30.539	2.330	2067
RPROP	L1	L1	114.10	28.843	2.221	2098
	L2	L2	115.76	30.951	2.200	2274
	L3	L3	112.05	28.667	2.212	2044
	L4	L4	108.08	25.811	2.216	1777
	L5	L5	118.44	22.997	2.219	1736
	L6a	L6a	128.23	22.976	2.230	1882
	L6	L6	108.89	26.499	2.252	1851
	L7	L7	119.48	29.163	2.225	2223
	L8	L8	101.22	29.104	2.215	1876
	L9	L9	112.03	29.115	2.214	2076
	L10	L10	80.00	27.140	2.212	1381
	L32	L32	103.81	28.328	2.240	1882
	L33	L33	69.09	36.751	2.231	1622

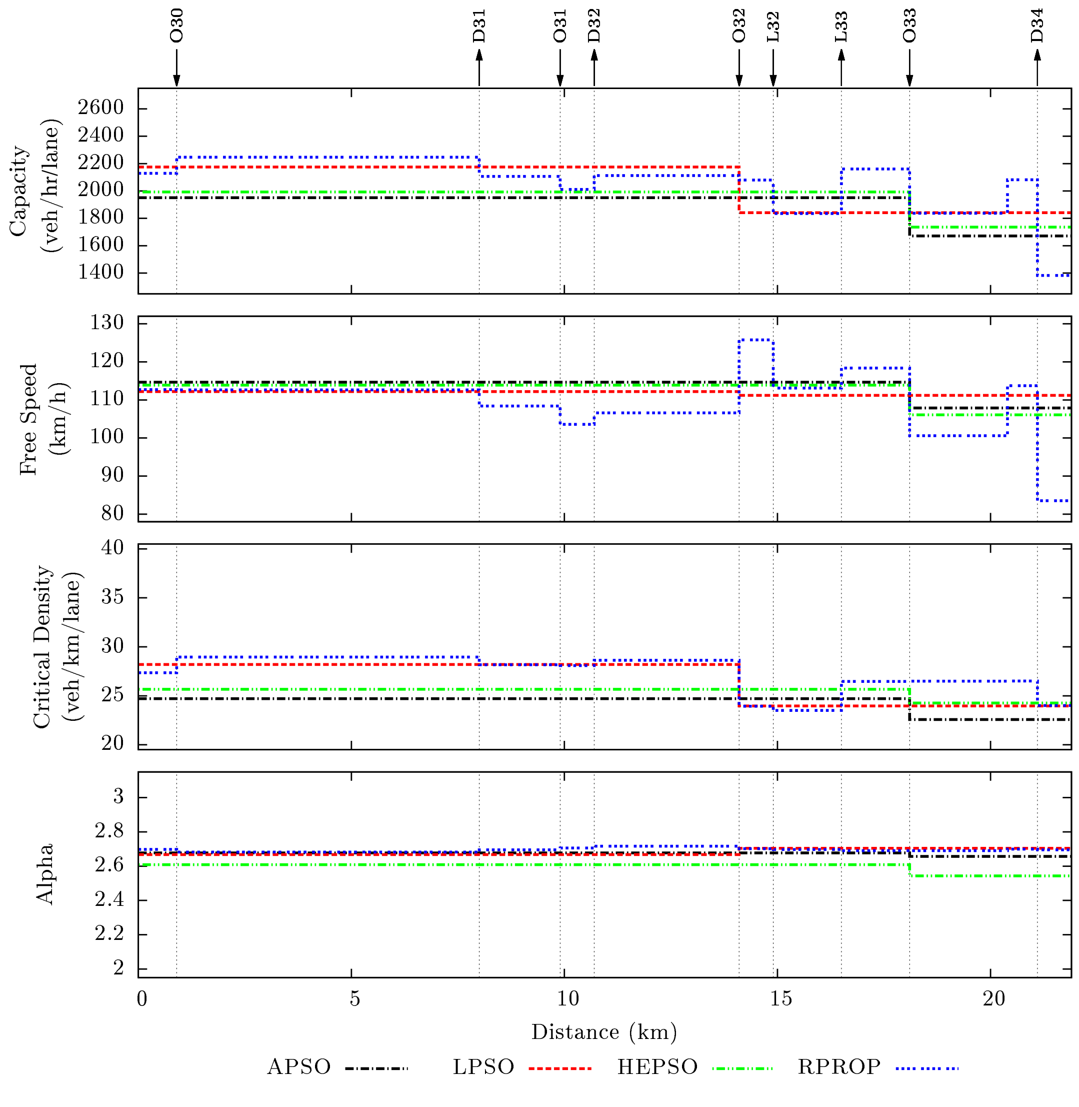


Figure 9.5: FD parameters' spatial distribution from the calibration optimal solutions found by the four optimisation algorithms using data of the 1st.

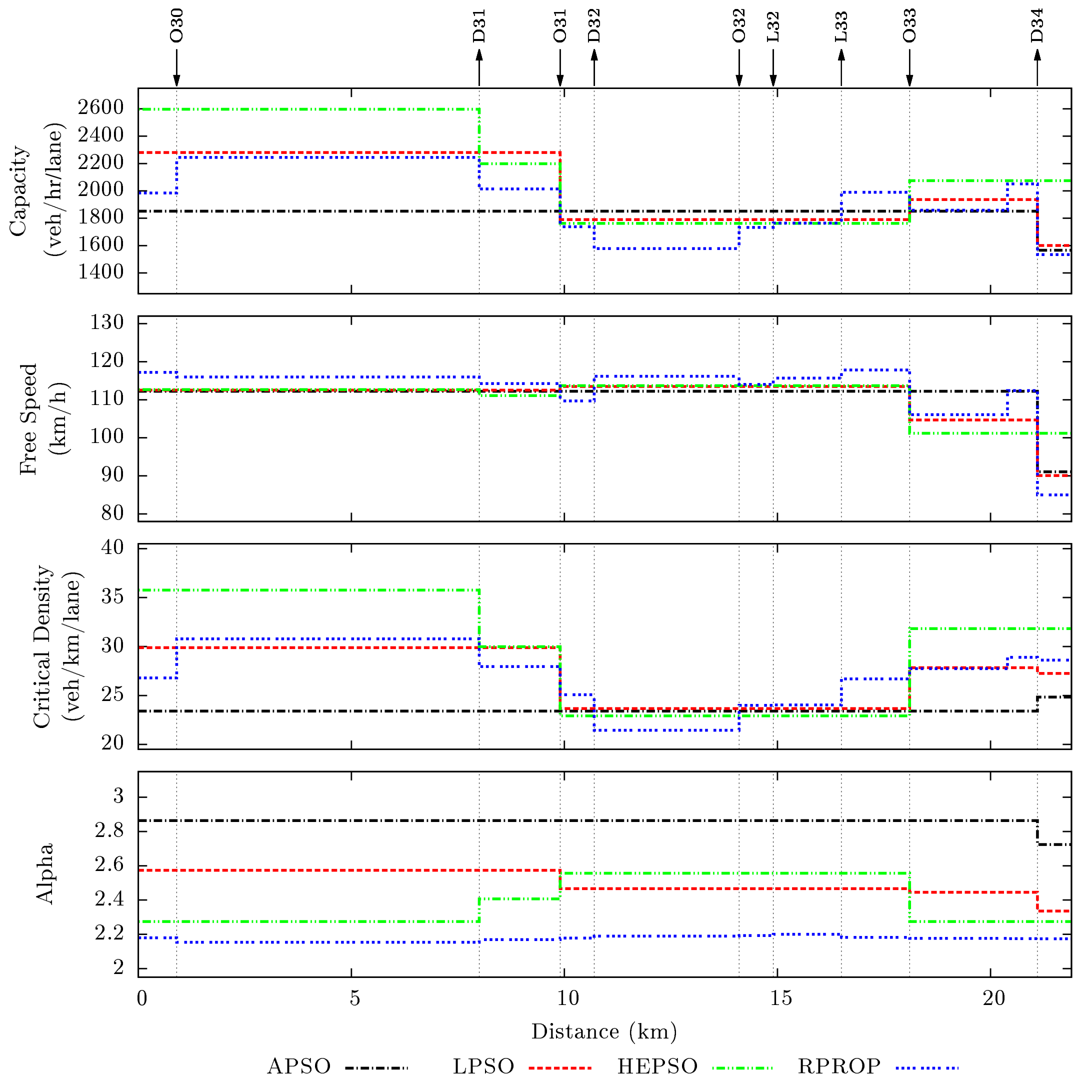


Figure 9.6: FD parameters' spatial distribution from the calibration optimal solutions found by the four optimisation algorithms using data of the 8th.

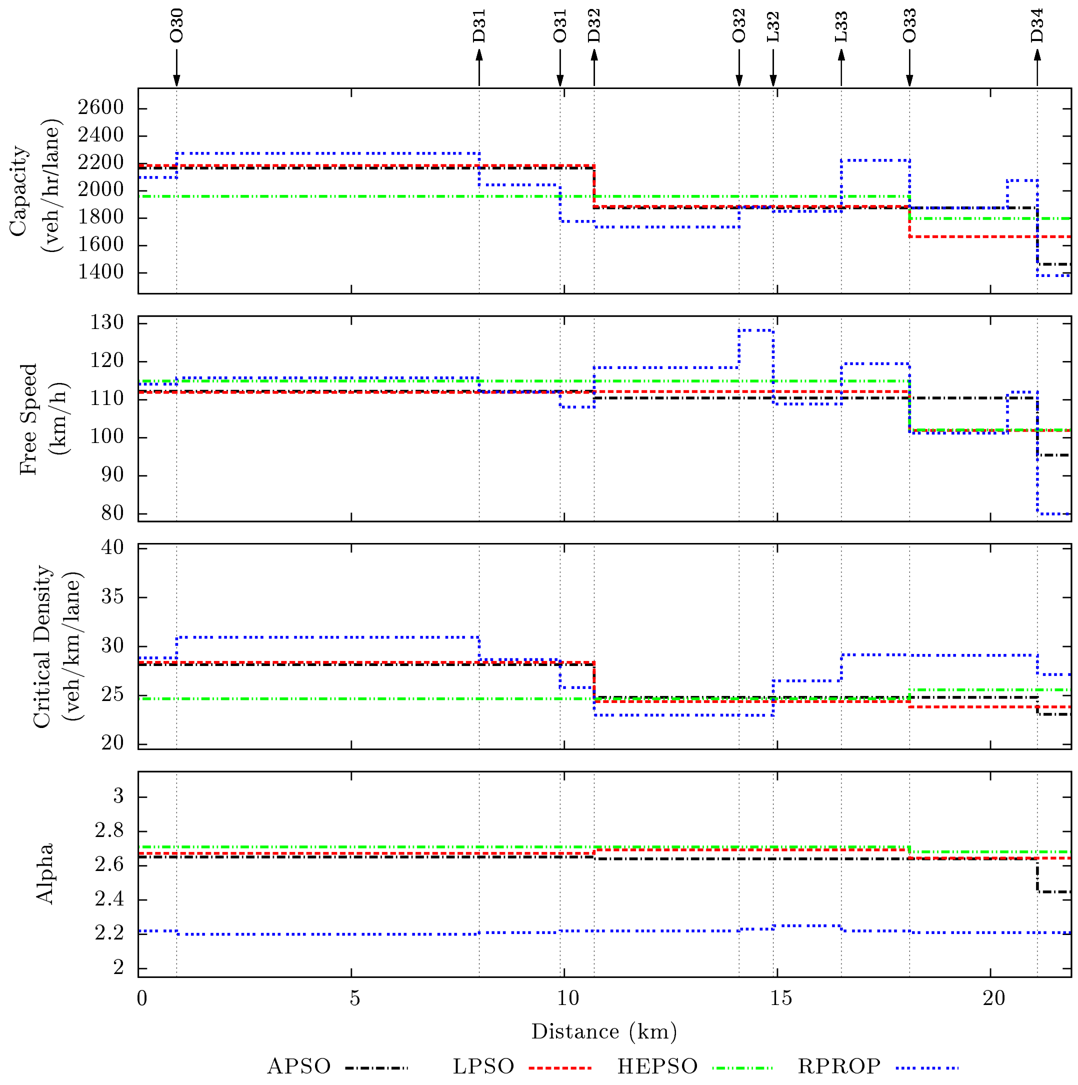


Figure 9.7: FD parameters' spatial distribution from the calibration optimal solutions found by the four optimisation algorithms using data of the 15th.

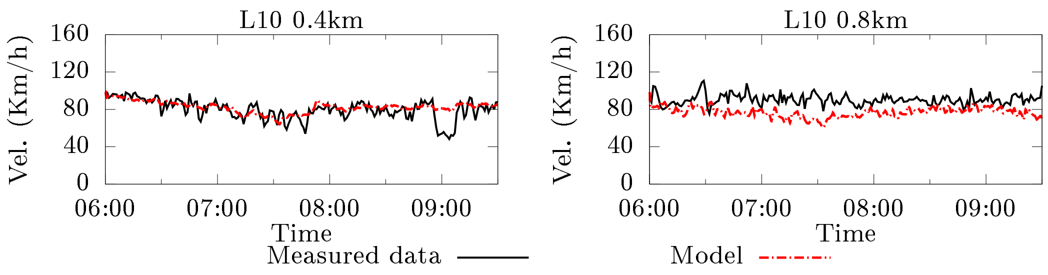
and HEPSO(15th) split the mainstream into two sections with the upstream one to have high capacity and the downstream with lower. A slightly different pattern is followed by APSO(15th) where an additional FD is used for L10. LPSO(1st) also follows a two FD pattern but with different extension. The split takes place at L5 rather than L7.

RPROP(1st) and RPROP(15th) have a more complicated capacity pattern due to the larger number of available FD. The capacity pattern for section L7-L10 is the same and the two solutions differ on the mainstream part from L1–L5. These patterns reflect the fact that upstream of the L6 (onset of congestion) is all free flowing and therefore the full condition of traffic are not observed, as long as the selected parameters allow for unrestricted flow with the correct free speed the specific values critical density and capacity are not important.

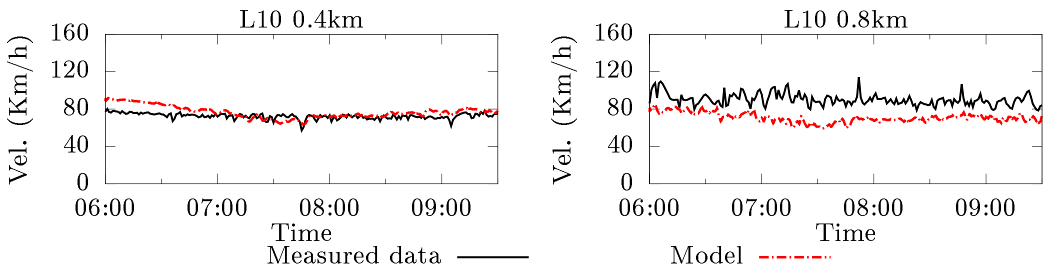
A common feature of all solutions' capacity patterns is that the section that contains the last link L10 has always relatively smaller capacity. This is a very pronounced feature at the RPROP solutions as well as in APSO(15th). This tendency of the optimisation is not misplaced since the speed levels predicted by the model are close to those measured. This can be seen in Figure 9.8, where the measured and calibrated model speed trajectories based on RPROP(1st) and RPROP(15th) for the two segments of L10. It can be seen that the selection of the link's capacity is a trade-off between the two speed trajectories from the two segments, hence the small bias observed at the model speed for the second segment sending flows out of the system.

Another feature of the solutions are links L32 and L33, which have their own FD. They are auxiliary highway links where dynamics are important for the overall model and they need to be described in more detail than the simple queueing of the origin links or the destinations' discharge. For RPROP(1st, 15th) L32's capacity is very similar to that of L6, into which it is feeding its traffic volume. Although this observation does not hold for the EA the RPROP solutions clearly support this.

No such connection can be identified for L33. This small link models a problematic interface with the surface street network, where congestion spilbacks occur. An explanation for the different capacities at the solutions of the 1st and 15th can be based on a closer examination of destinations density as shown in Figure 9.9 and the subdiagram showing the density at destination D33 where L33 leads. During the period 8:55–9:10 am there is a pronounced difference between the destination density trajectories given as boundary conditions for D33. This density on the 15th remains high whereas on the 1st a sharp downwards peak can be seen. The hypothesis is that this difference on the boundary conditions is the cause for leading the capacity at the high level of 2,766 veh/km/lane for the 1st and at the level of



(a) Calibrated using data of the 1st.



(b) Calibrated using data of the 15th.

Figure 9.8: Measured and model speeds for the two segments of L10 calibrated using RPROP.

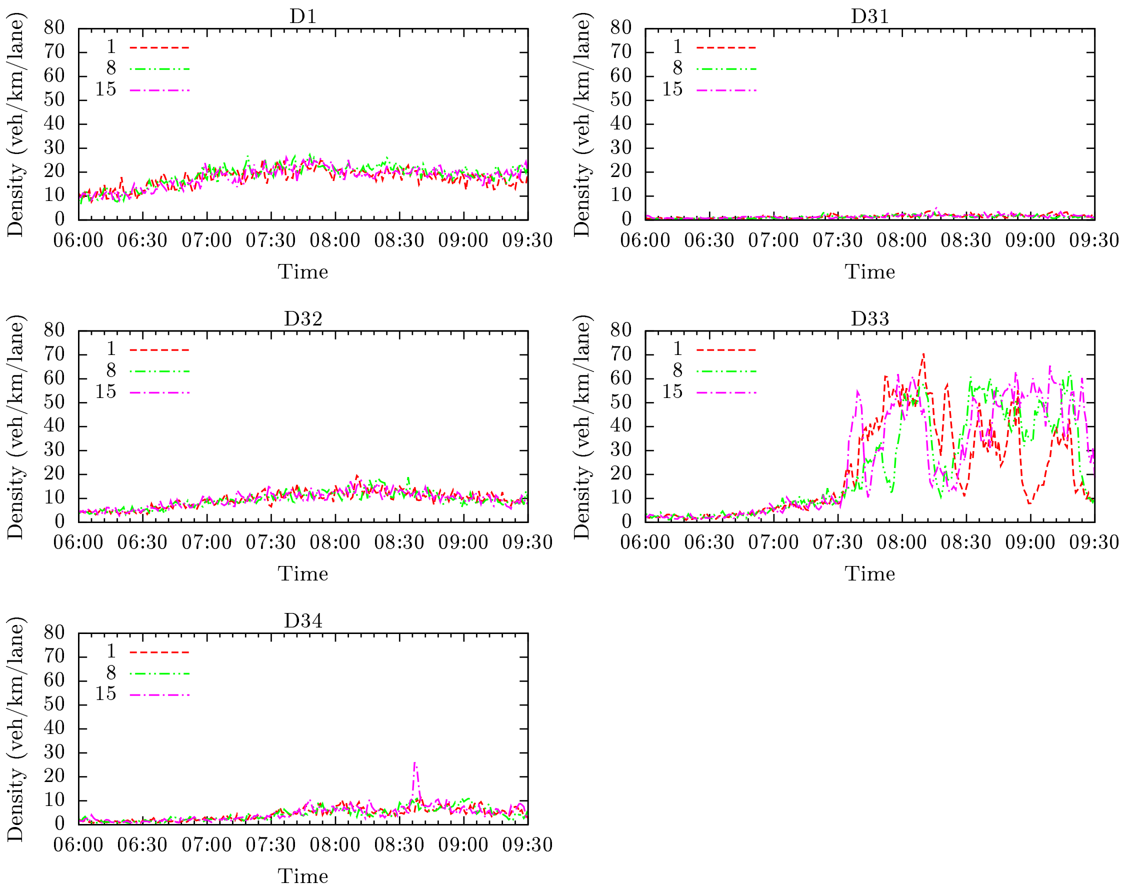


Figure 9.9: Density applied to boundaries for all data sets.

1,622 veh/km/lane for the 15th. Further evidence supporting it are the solutions delivered by the EA for L33.

9.3.3 Solution quality

Overall, all algorithms provide parameter sets that are able to reproduce the traffic dynamics with very good accuracy. Figure 9.10 provides the calibrated model, based on the optimal solutions found by each algorithm, versus speed measurements diagrams. Figure 9.11 depicts the distance-time diagrams of the calibrated METANET model of the 15th using the parameter sets resulting to the minimum J_v on Table 9.2 for each of the four optimisation algorithms. Empty spaces indicate areas where there are no data available.

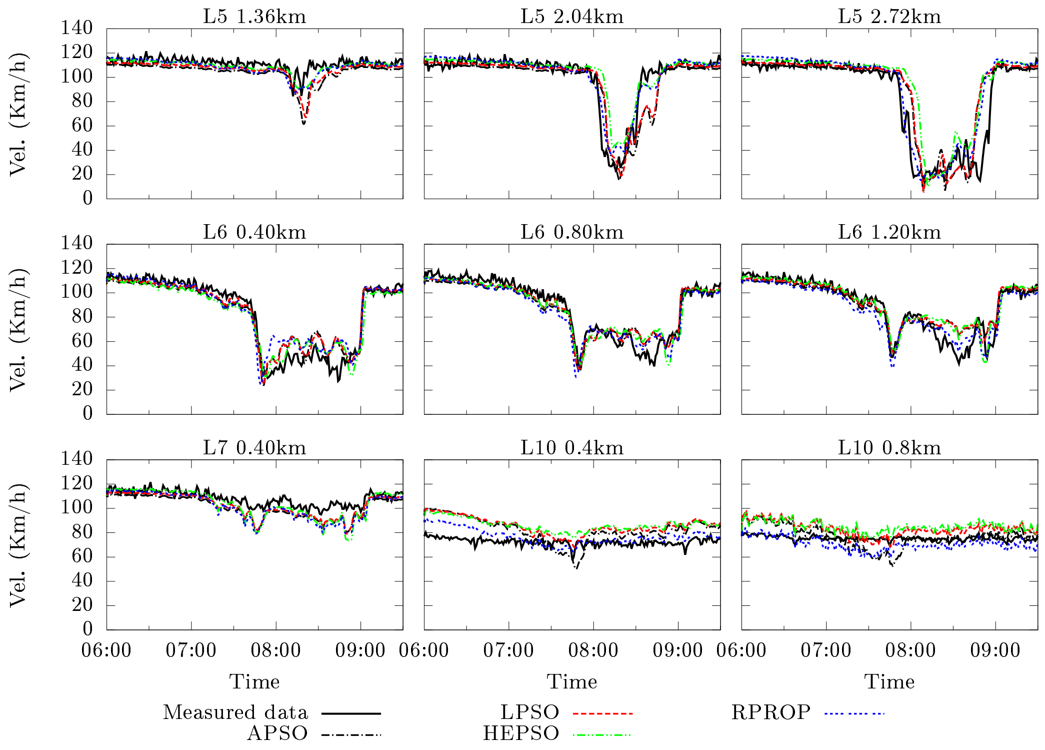


Figure 9.10: Calibrated model versus measured speeds for the data of the 15th.

9.3.4 Model Sensitivity Diagrams

An additional insight on how the mean speeds are related to the model parameters for which the calibration problem is solved, can be obtained by the information contained in the Jacobian matrix $\partial \mathbf{x}/\partial \mathbf{z}$. In effect this matrix provides the sensitivities of the segments' mean speed at every point in time with respect to all variables included in $\mathbf{z}$. These sensitivities can be displayed on distance-time diagrams similar to Figure 9.11. The result is an additional insight on systems dynamics and how

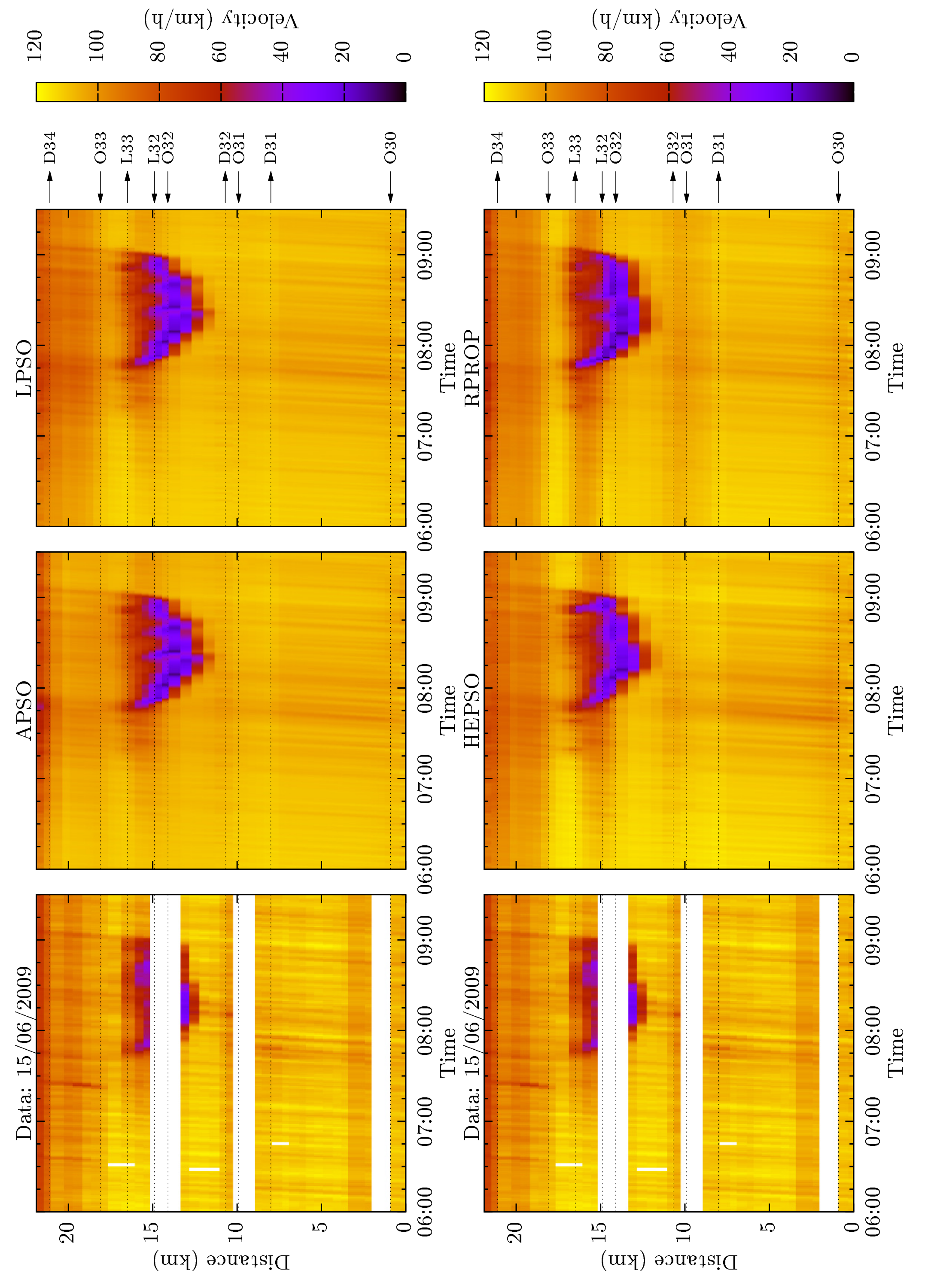
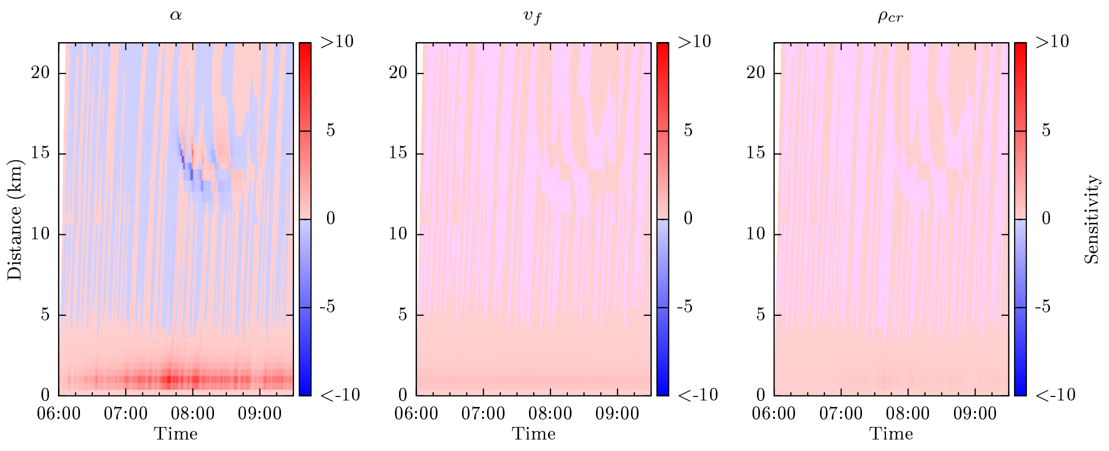


Figure 9.11: Distance-time diagram for calibration using the data of the 15th.

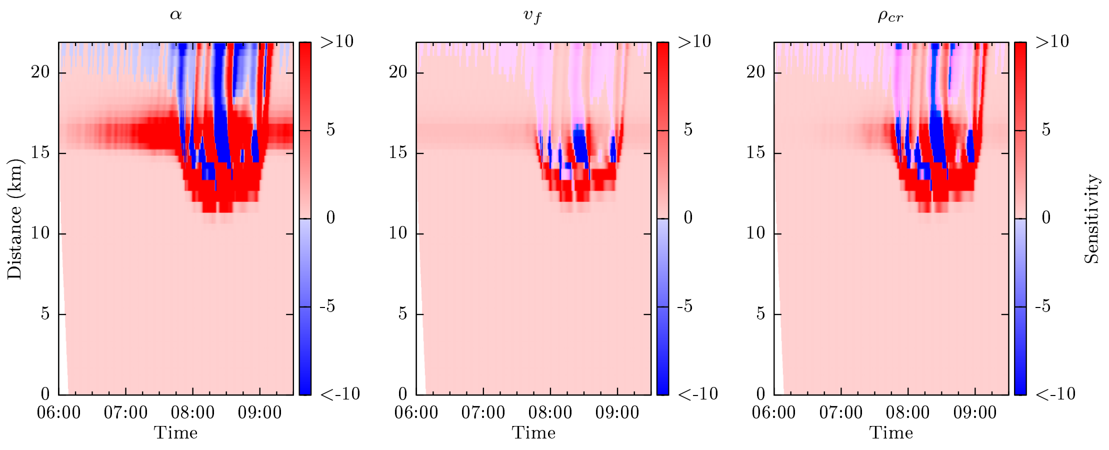
sensitivities propagate inside the network.

Figure 9.12 shows the sensitivity of the network's speeds over the whole time horizon with respect to the FD parameters of links L1, L6 and L10 for the solution obtained by RPROP(15th). Sensitivity shockwaves can be seen depending on the links' locations. Forwards moving shockwaves can be observed for the sensitivity to L1's parameters, from the site's start towards its end. Backwards shockwaves can be seen for the sensitivities with respect to the L10 FD parameters propagating from the site's end towards the start. A more complicated pattern in the area where congestion forms can be seen on link's L6 diagram. Figure 9.13 depicts the speed sensitivities with respect to the global parameters. All sensitivities are evaluated for the specific boundary conditions of the particular day at the solutions given by Tables 9.3–9.6.

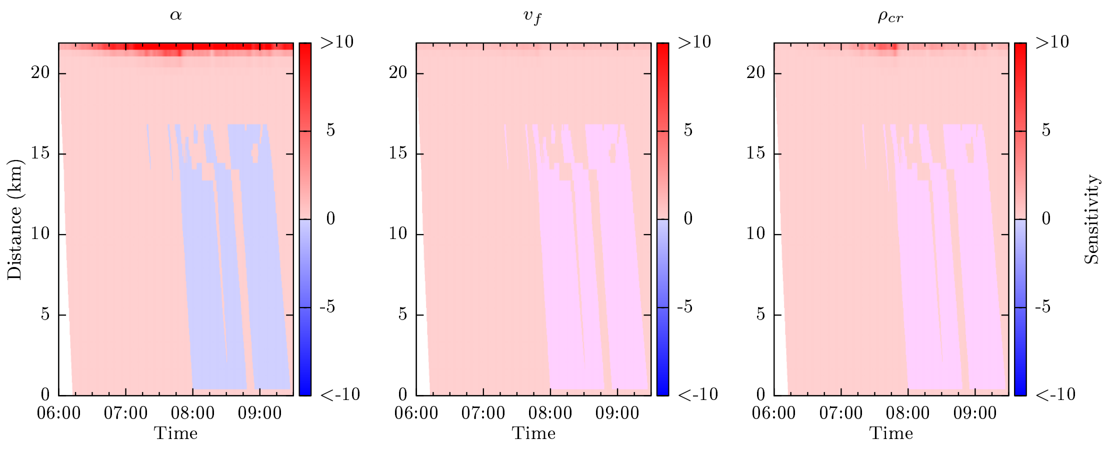
This information can be used for systematic analysis of highway features, such as bottlenecks over time and space. Control strategy design can benefit as well as infrastructure improvement projects. It is interesting to note that shock waves generated at the congested region invert the value of the upstream influence of L10. Also the Shock waves generated at L6 are more obvious in the sensitivity plot than in the observed speed.



(a) L1 sensitivities.



(b) L6 sensitivities.



(c) L10 sensitivities.

Figure 9.12: Network mean speed sensitivities to the FD parameters calculated by RPROP(15th).

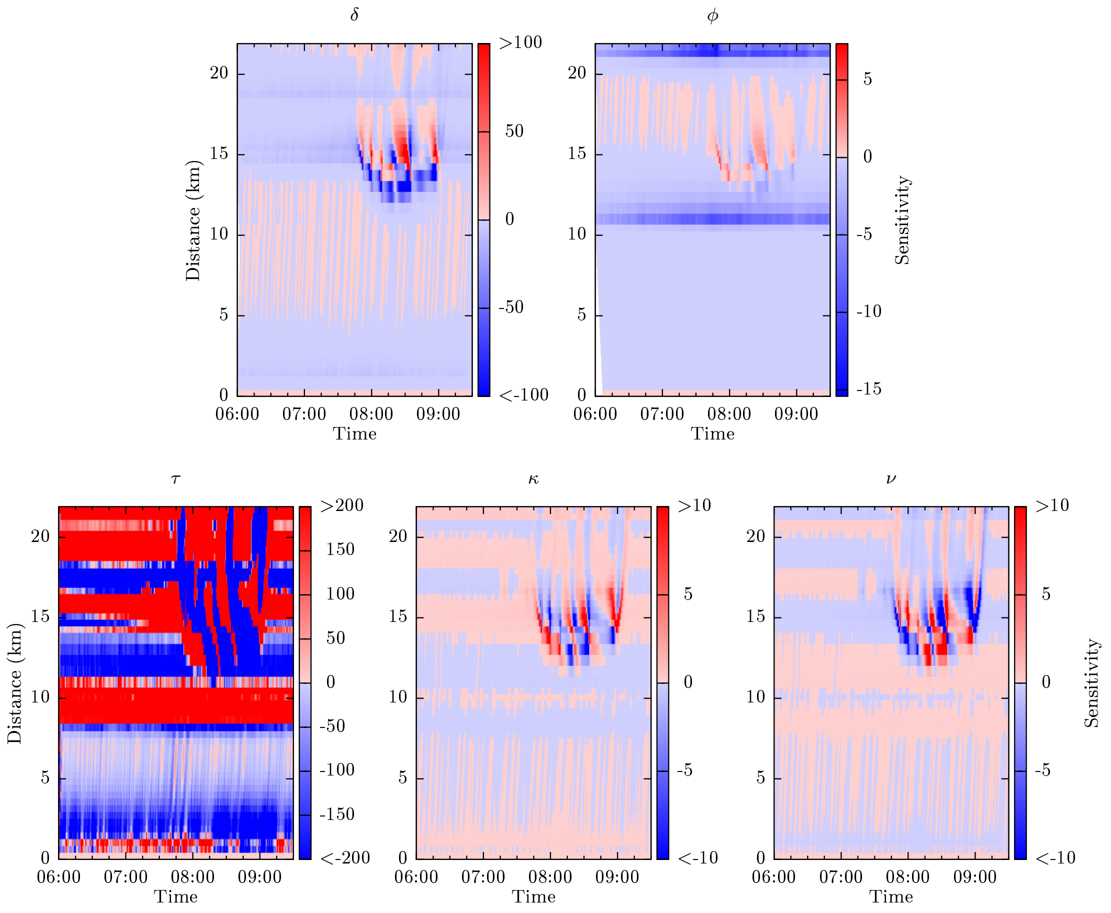


Figure 9.13: Network mean speed sensitivities to the global model parameters calculated by RPROP(15th).

Looking at Figure 9.12 notice there is a vacuum state which can be seen as the white region on the left side of each plot. For link 1 this vacuum ends once the sensitivity has propagated from the its origin in L1. For L6 in the middle of the model there is a forward and backwards vacuum regions, the forwards region is small and cannot be seen in the plots. For L10 only a upstream vacuum state exists as its the final link in the model. Looking at L6 as it has both an upstream and downstream vacuum state, with respect to sensitivity, we can calculate those two wave speeds an sample of the raw data for the first few time steps of the α parameter is shown in Table 9.7. Each row in the table shows a segment in the model and the distance from the site start is given and the columns are the time steps.

The average forward propagation speed of information for L6's α parameter is 186.92 km/h and the backwards propagation speed is 103.15 km/h. Looking in detail at Table 9.7 and the METANET system of equations (3.20–3.22), repeated

Table 9.7: Speed sensitvity to α for Link 6 $\frac{\partial v}{\partial \alpha_{\mathrm{L6}}}$, extract from $k = 0$, 6:00.

Distance	Time Step														
(km)	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$k=10$	$k=11$	$k=12$	$k=13$	$k=14$
12.06	0	0	0	0	0	0	0	0	0	0	0	0	0	1.86E-11	7.65E-11
12.74	0	0	0	0	0	0	0	0	0	0	0	1.98E-09	6.57E-09	1.24E-08	1.81E-08
13.42	0	0	0	0	0	0	0	0	0	2.10E-07	5.29E-07	7.85E-07	9.61E-07	1.09E-06	1.18E-06
14.10	0	0	0	0	0	0	0	2.24E-05	3.83E-05	4.12E-05	4.23E-05	4.41E-05	4.53E-05	4.65E-05	4.81E-05
14.50	0	0	0	0	0	1.28E-03	1.52E-03	1.31E-03	1.34E-03	1.40E-03	1.41E-03	1.45E-03	1.50E-03	1.55E-03	1.60E-03
14.90	0	0	0	0.05	0.05	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
15.30	0	1.15	0.89	0.71	0.77	0.77	0.76	0.77	0.78	0.80	0.81	0.83	0.84	0.85	0.85
15.70	0	1.01	1.63	1.47	1.22	1.24	1.25	1.24	1.24	1.26	1.28	1.30	1.32	1.34	1.36
16.10	0	1.32	1.72	1.95	1.85	1.59	1.55	1.56	1.54	1.54	1.55	1.56	1.59	1.61	1.63
16.50	0	1.32	2.05	2.06	2.05	1.97	1.74	1.66	1.64	1.62	1.61	1.62	1.64	1.66	1.68
16.90	0	0	0.83	1.28	1.18	1.13	1.12	0.97	0.91	0.92	0.92	0.93	0.94	0.95	0.96
17.30	0	0	0	0.53	0.84	0.70	0.62	0.64	0.54	0.49	0.51	0.52	0.54	0.55	0.56
17.70	0	0	0	0	0.34	0.57	0.44	0.34	0.36	0.30	0.25	0.27	0.29	0.30	0.32
18.10	0	0	0	0	0	0.22	0.40	0.30	0.19	0.20	0.17	0.12	0.12	0.14	0.16
18.56	0	0	0	0	0	0	0.11	0.24	0.21	0.13	0.11	0.09	0.05	0.04	0.05
19.02	0	0	0	0	0	0	0	0.06	0.14	0.15	0.09	0.06	0.05	0.02	0.01
19.48	0	0	0	0	0	0	0	0	0.03	0.08	0.10	0.07	0.04	0.03	0.01
19.94	0	0	0	0	0	0	0	0	0	0.01	0.05	0.07	0.06	0.04	0.02
20.40	0	0	0	0	0	0	0	0	0	0	7.20E-03	2.73E-02	4.64E-02	4.84E-02	4.00E-02
21.10	0	0	0	0	0	0	0	0	0	0	0	2.41E-03	1.04E-02	2.01E-02	2.36E-02
21.50	0	0	0	0	0	0	0	0	0	0	0	0	1.23E-03	6.56E-03	1.56E-02
21.90	0	0	0	0	0	0	0	0	0	0	0	0	0	5.53E-04	3.68E-03

here for readability,

$$\rho_{m,i}(k+1) = \rho_{m,i}(k) + \frac{T}{L_m \lambda_m}[q_{m,i-1}(k) - q_{m,i}(k)] \tag{9.3}$$

$$q_{m,i}(k) = \rho_{m,i}(k) \cdot v_{m,i}(k) \cdot \lambda_m \tag{9.4}$$

$$\begin{aligned} v_{m,i}(k+1) = v_{m,i}(k) &+ \frac{T}{\tau}\{V[\rho_{m,i}(k)] - v_{m,i}(k)\} && \text{(relaxation)} \\ &+ \frac{T}{L_m} v_{m,i}(k)[v_{m,i-1}(k) - v_{m,i}(k)] && \text{(convection)} \\ &- \frac{\nu \cdot T}{\tau \cdot L_m} \frac{\rho_{m,i+1}(k) - \rho_{m,i}(k)}{\rho_{m,i}(k) + \kappa} && \text{(anticipation)} \end{aligned} \tag{9.5}$$

the reason for the high propagation speeds becomes apparent. In the table L6 covers from 15.3–16.5km, the region between the horizontal lines, hence why the sensitivity covers that extent at $k = 1$, all sensitivities are 0 at $k = 0$ as that is the initial condition and the model parameters have not been applied. The sensitivity then propagates downstream every single time step. The upstream propagation always takes two time steps. In the speed equation (9.5) the convection term references the speed of the upstream segment, this means that the speed from that segment is used in the current segments calculation and any speed sensitivity will be propagated from the upstream cell to the current one, this also happens with the convection of flow. This means that regardless of time step or cell length the sensitivity will propagate downstream one segment every time step. The upstream propagation is more complex, the anticipation term in (9.5) uses the the density of the downstream cell, this means that at each time step the density sensitivity will propagate one segment upstream. In the next time step that density effect and associated sensitivity is used directly in the speed calculation. This makes the upstream propagation of the speed sensitivity take two time steps. The two step propagation through the density means that upstream sensitivity is considerably smaller than those propagated downstream. So the speeds seen for the initial propagation of sensitivities is large, and greater than the traffic flow speed, but the same will be true for the CTM model. In the CTM it is expected that the upstream propagation will take one time step rather than two as the model only has a single dynamic equation.

The network parameters τ, κ and ν are applied to every link and therefore do not have any vacuum region. The parameters δ and ϕ are applied when specific conditions are met so small vacuum regions do exist in-between the application areas. For δ these are very small and cannot be seen in Figure 9.13, however for ϕ a small region can be seen in the at 0.0 km.

Table 9.8: Speed sensitvity to ρ_{cr} for Link 6 $\frac{\partial v}{\partial \rho_{cr,\mathrm{L6}}}$ extract from $k = 0$, 6:00.

Distance	Time Step														
(km)	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$k=10$	$k=11$	$k=12$	$k=13$	$k=14$
12.06	0	0	0	0	0	0	0	0	0	0	0	0	0	6.90E-12	2.96E-11
12.74	0	0	0	0	0	0	0	0	0	0	0	7.34E-10	2.57E-09	5.19E-09	8.08E-09
13.42	0	0	0	0	0	0	0	0	0	7.78E-08	2.11E-07	3.43E-07	4.53E-07	5.44E-07	6.18E-07
14.10	0	0	0	0	0	0	0	8.28E-06	1.58E-05	1.95E-05	2.18E-05	2.37E-05	2.51E-05	2.62E-05	2.71E-05
14.50	0	0	0	0	0	4.75E-04	6.52E-04	6.77E-04	7.23E-04	7.69E-04	7.99E-04	8.22E-04	8.43E-04	8.61E-04	8.76E-04
14.90	0	0	0	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
15.30	0	0.43	0.42	0.41	0.43	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44
15.70	0	0.43	0.67	0.67	0.65	0.67	0.68	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69
16.10	0	0.42	0.66	0.81	0.81	0.78	0.80	0.82	0.83	0.83	0.83	0.83	0.83	0.83	0.83
16.50	0	0.42	0.66	0.77	0.86	0.87	0.84	0.84	0.85	0.86	0.87	0.87	0.87	0.87	0.87
16.90	0	0	0.27	0.41	0.45	0.49	0.50	0.47	0.47	0.48	0.49	0.50	0.50	0.50	0.50
17.30	0	0	0	0.17	0.27	0.27	0.28	0.29	0.27	0.26	0.27	0.28	0.28	0.29	0.29
17.70	0	0	0	0	0.11	0.18	0.17	0.17	0.17	0.15	0.14	0.15	0.16	0.16	0.16
18.10	0	0	0	0	0	0.07	0.13	0.12	0.10	0.11	0.09	0.08	0.08	0.08	0.09
18.56	0	0	0	0	0	0	0.04	0.08	0.08	0.06	0.06	0.05	0.04	0.03	0.04
19.02	0	0	0	0	0	0	0	0.02	0.05	0.05	0.04	0.04	0.03	0.02	0.01
19.48	0	0	0	0	0	0	0	0	0.01	0.03	0.04	0.03	0.03	0.02	0.01
19.94	0	0	0	0	0	0	0	0	0	4.61E-03	0.015	0.023	0.023	0.019	0.016
20.40	0	0	0	0	0	0	0	0	0	0	2.32E-03	8.80E-03	0.016	0.018	0.018
21.10	0	0	0	0	0	0	0	0	0	0	0	7.75E-04	3.34E-03	6.68E-03	8.56E-03
21.50	0	0	0	0	0	0	0	0	0	0	0	0	3.95E-04	2.11E-03	5.14E-03
21.90	0	0	0	0	0	0	0	0	0	0	0	0	0	1.78E-04	1.18E-03

Table 9.8 shows the sensitivity for the ρ_{cr} parameter at L6. The same properties are observed as for the alpha parameter (Table 9.7). The sensitivity propagates down stream every time step and upstream every two steps. The propagation downstream is stronger with larger values seen, the upstream propagation is much weaker and quickly drops below 0.01 (2 segments). In the downstream direction the sensitivity travels 9 segments to drop below 0.01. For both variables the highest sensitivities are seen at the application of the parameter, i.e. within the link. Tables 9.9 and 9.10, show the sensitivity during congestion at 8:30. The magnitude has increased significantly upstream of L6, in the middle of the congested region. As the sensitivity propagates upstream its magnitude increases leading to larger values than are seen within the link. This is not just the propagation of a large value from within the link as from time steps $k = 1125$ to $k = 1136$ a large increase in the magnitude of the sensitivity is seen as we go from 15.3 km the first segment of the link (row below first horizontal line) to the segment upstream (above the horizontal line). This effect is most pronounced in the α parameter but can also be seen for ρ_{cr}. Even though large values are seen upstream of the L6 as they propagate upstream beyond the congested region they do reduce rapidly back to the near zero values that are seen during the free flowing conditions. In general the alpha sensitivity is larger than that for the critical density by a factor of 10 in the congested region.

The sensitivities for L6's α and ρ_{cr} parameters for free flowing conditions after the congestion has passed are shown in Tables 9.11 and 9.12 respectively. The largest sensitivity values are now seen within the link where the parameters applied. The values fall away quickly as seen at the start of the model, which was also free flowing conditions. Again it only takes a few segments for the upstream sensitivity to drop below 0.01 and it is slower in the downstream direction. There appears to be no residual influence from the earlier congestion but the magnitude of the sensitivities for the α parameter are larger than seen at the start of the model, for ρ_{cr} the values are also larger. So it can be seen that generally the sensitivities are greatest for where the parameters are applied but the values of sensitivities become magnified by the congestion region. This magnification can be significantly large causing the largest observed sensitivity in a time step to be from a location outside of the link.

Table 9.9: Speed sensitvity to α for Link 6 $\frac{\partial v}{\partial \alpha_{\mathrm{L6}}}$, extract from $k = 1125$, 8:30.

Distance								Time Step							
(km)	$k = 1125$	$k = 1126$	$k = 1127$	$k = 1128$	$k = 1129$	$k = 1130$	$k = 1131$	$k = 1132$	$k = 1133$	$k = 1134$	$k = 1135$	$k = 1136$	$k = 1137$	$k = 1138$	$k = 1139$
8.47	6.01E-05	5.75E-05	5.47E-05	5.18E-05	4.89E-05	4.59E-05	4.29E-05	4.01E-05	3.73E-05	3.47E-05	3.21E-05	2.98E-05	2.75E-05	2.54E-05	2.34E-05
8.95	3.92E-04	3.67E-04	3.43E-04	3.19E-04	2.97E-04	2.77E-04	2.57E-04	2.39E-04	2.22E-04	2.05E-04	1.90E-04	1.75E-04	1.61E-04	1.48E-04	1.36E-04
9.42	2.33E-03	2.15E-03	1.99E-03	1.84E-03	1.70E-03	1.58E-03	1.46E-03	1.36E-03	1.26E-03	1.16E-03	1.06E-03	9.78E-04	8.98E-04	8.25E-04	7.58E-04
9.90	0.01	0.01	0.01	0.01	9.34E-03	8.64E-03	7.95E-03	7.28E-03	6.66E-03	6.11E-03	5.61E-03	5.16E-03	4.77E-03	4.44E-03	4.20E-03
10.30	0.07	0.06	0.06	0.05	0.05	0.04	0.04	0.04	0.03	0.03	0.03	0.03	0.03	0.02	0.02
10.70	0.33	0.30	0.27	0.24	0.21	0.19	0.18	0.16	0.15	0.14	0.14	0.14	0.15	0.15	0.16
11.38	3.49	3.20	2.96	2.76	2.61	2.50	2.42	2.39	2.39	2.42	2.48	2.56	2.68	2.81	2.97
12.06	23.20	22.54	22.25	22.30	22.65	23.31	24.24	25.42	26.84	28.46	30.27	32.25	34.39	36.64	39.00
12.74	95.14	104.45	114.95	126.62	139.48	153.49	168.62	184.86	202.13	220.36	239.44	259.28	279.76	300.76	322.09
13.42	138.54	159.15	181.49	205.67	231.79	259.90	290.09	322.43	356.94	393.61	432.34	472.89	514.95	558.05	601.66
14.10	177.27	183.61	188.75	192.52	195.08	196.49	196.73	195.76	193.50	189.87	184.85	178.37	170.38	160.92	150.11
14.50	244.17	238.89	232.13	223.14	211.99	199.07	184.61	168.69	151.35	132.77	113.25	93.07	72.27	50.91	29.14
14.90	216.29	205.98	195.07	183.51	170.71	156.44	140.86	124.16	106.48	87.97	68.83	49.41	30.06	10.94	-7.82
15.30	107.57	99.18	90.71	82.22	73.73	64.95	55.70	46.03	36.08	25.98	15.80	5.68	-4.18	-13.58	-22.42
15.70	52.50	46.03	39.54	33.06	26.67	20.40	14.17	7.89	1.63	-4.54	-10.51	-16.22	-21.61	-26.55	-30.93
16.10	22.01	17.53	13.07	8.63	4.25	-0.04	-4.19	-8.24	-12.20	-16.02	-19.61	-22.87	-25.74	-28.17	-30.09
16.50	15.39	12.56	9.73	6.95	4.19	1.42	-1.35	-4.07	-6.71	-9.22	-11.53	-13.54	-15.09	-16.14	-16.68
16.90	3.17	1.91	0.66	-0.54	-1.69	-2.81	-3.93	-5.04	-6.13	-7.18	-8.19	-9.11	-9.90	-10.47	-10.77
17.30	-1.70	-2.36	-3.03	-3.67	-4.27	-4.82	-5.33	-5.82	-6.32	-6.79	-7.24	-7.66	-8.03	-8.33	-8.49
17.70	-4.41	-4.81	-5.21	-5.61	-5.97	-6.30	-6.58	-6.82	-7.04	-7.25	-7.44	-7.60	-7.73	-7.83	-7.89
18.10	-6.46	-6.77	-7.07	-7.37	-7.66	-7.93	-8.16	-8.35	-8.48	-8.59	-8.67	-8.72	-8.75	-8.74	-8.70
18.56	-8.97	-9.25	-9.53	-9.77	-10.00	-10.22	-10.42	-10.57	-10.68	-10.73	-10.77	-10.79	-10.78	-10.73	-10.64
19.02	-10.29	-10.57	-10.85	-11.11	-11.34	-11.55	-11.74	-11.91	-12.05	-12.14	-12.18	-12.19	-12.20	-12.18	-12.11
19.48	-10.56	-10.85	-11.14	-11.41	-11.67	-11.90	-12.11	-12.31	-12.48	-12.62	-12.72	-12.78	-12.81	-12.83	-12.82
19.94	-9.84	-10.13	-10.41	-10.68	-10.94	-11.19	-11.42	-11.63	-11.82	-12.00	-12.14	-12.26	-12.34	-12.39	-12.42
20.40	-7.99	-8.26	-8.52	-8.77	-9.01	-9.24	-9.45	-9.65	-9.83	-9.99	-10.14	-10.27	-10.38	-10.47	-10.55
21.10	-9.61	-9.96	-10.21	-10.40	-10.58	-10.73	-10.86	-10.96	-11.02	-11.13	-11.31	-11.50	-11.66	-11.80	-11.92
21.50	-12.44	-12.87	-13.31	-13.72	-14.10	-14.47	-14.83	-15.16	-15.45	-15.72	-15.97	-16.24	-16.50	-16.72	-16.91
21.90	-5.86	-6.01	-6.24	-6.49	-6.72	-6.92	-7.12	-7.30	-7.46	-7.55	-7.54	-7.52	-7.51	-7.52	-7.53

Table 9.10: Speed sensitvity to ρ_{cr} for Link 6 $\frac{\partial v}{\partial \rho_{cr,\mathrm{L6}}}$ extract from $k = 1125$, 8:30.

Distance								Time Step							
(km)	$k = 1125$	$k = 1126$	$k = 1127$	$k = 1128$	$k = 1129$	$k = 1130$	$k = 1131$	$k = 1132$	$k = 1133$	$k = 1134$	$k = 1135$	$k = 1136$	$k = 1137$	$k = 1138$	$k = 1139$
8.47	3.09E-06	2.95E-06	2.80E-06	2.65E-06	2.49E-06	2.34E-06	2.18E-06	2.03E-06	1.89E-06	1.75E-06	1.62E-06	1.49E-06	1.37E-06	1.26E-06	1.16E-06
8.95	2.00E-05	1.87E-05	1.74E-05	1.62E-05	1.50E-05	1.40E-05	1.29E-05	1.20E-05	1.11E-05	1.02E-05	9.42E-06	8.66E-06	7.93E-06	7.25E-06	6.62E-06
9.42	1.18E-04	1.08E-04	1.00E-04	9.23E-05	8.51E-05	7.85E-05	7.25E-05	6.70E-05	6.17E-05	5.66E-05	5.17E-05	4.73E-05	4.32E-05	3.94E-05	3.60E-05
9.90	6.46E-04	5.92E-04	5.43E-04	4.99E-04	4.59E-04	4.22E-04	3.86E-04	3.52E-04	3.20E-04	2.92E-04	2.66E-04	2.43E-04	2.23E-04	2.07E-04	1.95E-04
10.30	3.30E-03	3.02E-03	2.76E-03	2.53E-03	2.28E-03	2.05E-03	1.84E-03	1.67E-03	1.51E-03	1.38E-03	1.27E-03	1.20E-03	1.15E-03	1.13E-03	1.12E-03
10.70	0.02	0.01	0.01	0.01	0.01	9.01E-03	8.13E-03	7.43E-03	6.87E-03	6.57E-03	6.51E-03	6.54E-03	6.68E-03	6.92E-03	7.24E-03
11.38	0.16	0.15	0.14	0.13	0.12	0.11	0.11	0.11	0.11	0.11	0.11	0.12	0.12	0.13	0.14
12.06	1.03	1.00	0.99	0.99	1.02	1.06	1.11	1.17	1.25	1.34	1.43	1.54	1.65	1.77	1.90
12.74	4.25	4.73	5.28	5.88	6.55	7.28	8.06	8.91	9.81	10.76	11.75	12.78	13.85	14.95	16.06
13.42	6.81	7.86	9.00	10.24	11.58	13.02	14.57	16.23	18.01	19.90	21.89	23.99	26.16	28.40	30.66
14.10	9.60	9.94	10.22	10.42	10.56	10.64	10.66	10.62	10.50	10.31	10.05	9.70	9.27	8.77	8.18
14.50	13.04	12.76	12.40	11.92	11.33	10.65	9.88	9.04	8.12	7.13	6.10	5.03	3.92	2.79	1.63
14.90	11.43	10.88	10.29	9.67	8.99	8.23	7.40	6.51	5.57	4.58	3.56	2.53	1.50	0.48	-0.52
15.30	5.61	5.16	4.70	4.25	3.79	3.32	2.82	2.31	1.77	1.23	0.69	0.15	-0.38	-0.88	-1.35
15.70	2.60	2.25	1.91	1.56	1.22	0.88	0.54	0.21	-0.13	-0.46	-0.78	-1.09	-1.37	-1.64	-1.87
16.10	0.95	0.71	0.47	0.23	0.00	-0.23	-0.45	-0.67	-0.88	-1.09	-1.28	-1.45	-1.60	-1.73	-1.84
16.50	0.64	0.49	0.34	0.20	0.05	-0.10	-0.25	-0.39	-0.53	-0.67	-0.79	-0.90	-0.99	-1.05	-1.08
16.90	0.10	0.03	-0.04	-0.10	-0.16	-0.22	-0.28	-0.34	-0.40	-0.46	-0.51	-0.56	-0.61	-0.64	-0.65
17.30	-0.12	-0.16	-0.19	-0.23	-0.26	-0.29	-0.31	-0.34	-0.37	-0.39	-0.42	-0.44	-0.46	-0.48	-0.49
17.70	-0.25	-0.27	-0.29	-0.31	-0.33	-0.35	-0.36	-0.37	-0.39	-0.40	-0.41	-0.42	-0.42	-0.43	-0.43
18.10	-0.34	-0.36	-0.38	-0.39	-0.41	-0.42	-0.43	-0.44	-0.45	-0.46	-0.46	-0.47	-0.47	-0.47	-0.46
18.56	-0.47	-0.48	-0.50	-0.51	-0.52	-0.54	-0.55	-0.55	-0.56	-0.56	-0.56	-0.57	-0.57	-0.56	-0.56
19.02	-0.53	-0.55	-0.56	-0.58	-0.59	-0.60	-0.61	-0.62	-0.63	-0.63	-0.63	-0.63	-0.64	-0.63	-0.63
19.48	-0.54	-0.56	-0.57	-0.59	-0.60	-0.62	-0.63	-0.64	-0.65	-0.65	-0.66	-0.66	-0.66	-0.67	-0.67
19.94	-0.50	-0.52	-0.53	-0.55	-0.56	-0.58	-0.59	-0.60	-0.61	-0.62	-0.63	-0.63	-0.64	-0.64	-0.64
20.40	-0.40	-0.42	-0.43	-0.45	-0.46	-0.47	-0.48	-0.50	-0.50	-0.51	-0.52	-0.53	-0.54	-0.54	-0.54
21.10	-0.48	-0.50	-0.51	-0.52	-0.53	-0.54	-0.55	-0.56	-0.56	-0.57	-0.58	-0.59	-0.60	-0.60	-0.61
21.50	-0.61	-0.63	-0.66	-0.68	-0.70	-0.72	-0.74	-0.76	-0.78	-0.80	-0.81	-0.83	-0.84	-0.85	-0.86
21.90	-0.29	-0.29	-0.31	-0.32	-0.33	-0.35	-0.36	-0.37	-0.38	-0.38	-0.38	-0.38	-0.38	-0.38	-0.38

Table 9.11: Speed sensitvity to α for Link 6 $\frac{\partial v}{\partial \alpha_{\mathrm{L6}}}$, extract from $k = 1463$, 9:15.

Distance							Time Step								
(km)	$k = 1463$	$k = 1464$	$k = 1465$	$k = 1466$	$k = 1467$	$k = 1468$	$k = 1469$	$k = 1470$	$k = 1471$	$k = 1472$	$k = 1473$	$k = 1474$	$k = 1475$	$k = 1476$	$k = 1477$
8.47	4.30E-13	4.26E-13	4.22E-13	4.18E-13	4.14E-13	4.10E-13	4.06E-13	4.03E-13	3.99E-13	3.96E-13	3.94E-13	3.92E-13	3.91E-13	3.93E-13	3.96E-13
8.95	3.55E-12	3.50E-12	3.44E-12	3.40E-12	3.35E-12	3.31E-12	3.27E-12	3.23E-12	3.20E-12	3.18E-12	3.18E-12	3.20E-12	3.24E-12	3.30E-12	3.39E-12
9.42	2.86E-11	2.81E-11	2.76E-11	2.72E-11	2.67E-11	2.63E-11	2.60E-11	2.58E-11	2.59E-11	2.61E-11	2.66E-11	2.73E-11	2.82E-11	2.92E-11	3.04E-11
9.90	2.30E-10	2.26E-10	2.21E-10	2.18E-10	2.15E-10	2.13E-10	2.14E-10	2.18E-10	2.23E-10	2.29E-10	2.37E-10	2.46E-10	2.56E-10	2.66E-10	2.74E-10
10.30	1.89E-09	1.86E-09	1.82E-09	1.80E-09	1.80E-09	1.82E-09	1.85E-09	1.88E-09	1.93E-09	1.99E-09	2.07E-09	2.14E-09	2.20E-09	2.26E-09	2.33E-09
10.70	1.41E-08	1.41E-08	1.43E-08	1.45E-08	1.48E-08	1.51E-08	1.54E-08	1.59E-08	1.64E-08	1.66E-08	1.68E-08	1.72E-08	1.76E-08	1.81E-08	1.85E-08
11.38	3.16E-07	3.13E-07	3.11E-07	3.11E-07	3.11E-07	3.13E-07	3.15E-07	3.18E-07	3.21E-07	3.26E-07	3.31E-07	3.37E-07	3.43E-07	3.51E-07	3.59E-07
12.06	4.20E-06	4.21E-06	4.24E-06	4.28E-06	4.33E-06	4.38E-06	4.45E-06	4.51E-06	4.58E-06	4.65E-06	4.72E-06	4.79E-06	4.87E-06	4.95E-06	5.05E-06
12.74	5.86E-05	5.95E-05	6.04E-05	6.14E-05	6.24E-05	6.33E-05	6.43E-05	6.51E-05	6.60E-05	6.68E-05	6.77E-05	6.86E-05	6.96E-05	7.06E-05	7.17E-05
13.42	8.52E-04	8.68E-04	8.82E-04	8.95E-04	9.06E-04	9.16E-04	9.27E-04	9.38E-04	9.49E-04	9.61E-04	9.73E-04	9.86E-04	1.00E-03	1.01E-03	1.03E-03
14.10	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
14.50	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.15	0.15	0.15	0.15	0.15	0.15
14.90	1.22	1.25	1.27	1.28	1.29	1.29	1.29	1.29	1.28	1.28	1.28	1.28	1.29	1.30	1.29
15.30	7.29	7.34	7.38	7.40	7.41	7.43	7.43	7.43	7.44	7.45	7.46	7.47	7.47	7.46	7.45
15.70	10.77	10.88	10.96	11.02	11.07	11.11	11.14	11.16	11.18	11.19	11.21	11.23	11.25	11.25	11.23
16.10	12.19	12.31	12.44	12.53	12.59	12.66	12.72	12.77	12.81	12.83	12.85	12.88	12.90	12.92	12.92
16.50	11.59	11.67	11.76	11.85	11.92	11.97	12.01	12.06	12.10	12.13	12.14	12.15	12.18	12.20	12.22
16.90	6.45	6.49	6.52	6.55	6.60	6.62	6.64	6.67	6.70	6.73	6.76	6.78	6.80	6.82	6.85
17.30	3.69	3.72	3.73	3.72	3.73	3.74	3.75	3.75	3.76	3.78	3.81	3.84	3.85	3.87	3.90
17.70	2.15	2.16	2.18	2.18	2.16	2.14	2.14	2.14	2.13	2.13	2.14	2.16	2.18	2.20	2.21
18.10	1.29	1.30	1.31	1.32	1.31	1.29	1.26	1.25	1.24	1.22	1.20	1.20	1.21	1.23	1.24
18.56	0.73	0.73	0.73	0.73	0.73	0.72	0.70	0.67	0.65	0.63	0.61	0.59	0.58	0.58	0.59
19.02	0.41	0.41	0.41	0.40	0.40	0.40	0.39	0.37	0.34	0.32	0.30	0.27	0.25	0.23	0.22
19.48	0.24	0.24	0.24	0.23	0.23	0.22	0.22	0.21	0.19	0.17	0.15	0.12	0.10	0.08	0.06
19.94	0.17	0.16	0.15	0.15	0.14	0.14	0.13	0.13	0.12	0.10	0.09	0.07	0.04	0.02	0.00
20.40	0.14	0.13	0.12	0.12	0.11	0.11	0.10	0.10	0.09	0.09	0.08	0.07	0.05	0.04	0.02
21.10	0.21	0.20	0.18	0.17	0.16	0.15	0.13	0.12	0.11	0.10	0.09	0.08	0.06	0.05	0.03
21.50	0.30	0.28	0.26	0.24	0.22	0.21	0.19	0.17	0.16	0.15	0.13	0.12	0.10	0.08	0.06
21.90	0.08	0.08	0.07	0.06	0.06	0.06	0.05	0.05	0.05	0.04	0.04	0.04	0.04	0.04	0.04

Table 9.12: Speed sensitvity to ρ_{cr} for Link 6 $\frac{\partial v}{\partial \rho_{cr,\mathrm{L6}}}$ extract from $k = 1463$, 9:15.

Distance	Time Step														
(km)	$k = 1463$	$k = 1464$	$k = 1465$	$k = 1466$	$k = 1467$	$k = 1468$	$k = 1469$	$k = 1470$	$k = 1471$	$k = 1472$	$k = 1473$	$k = 1474$	$k = 1475$	$k = 1476$	$k = 1477$
8.47	3.39E-14	3.37E-14	3.36E-14	3.33E-14	3.31E-14	3.29E-14	3.27E-14	3.26E-14	3.24E-14	3.23E-14	3.21E-14	3.21E-14	3.22E-14	3.24E-14	3.28E-14
8.95	2.84E-13	2.81E-13	2.78E-13	2.75E-13	2.72E-13	2.69E-13	2.66E-13	2.64E-13	2.62E-13	2.61E-13	2.62E-13	2.65E-13	2.69E-13	2.76E-13	2.84E-13
9.42	2.32E-12	2.28E-12	2.25E-12	2.22E-12	2.19E-12	2.16E-12	2.14E-12	2.14E-12	2.15E-12	2.18E-12	2.22E-12	2.29E-12	2.37E-12	2.46E-12	2.57E-12
9.90	1.88E-11	1.85E-11	1.82E-11	1.80E-11	1.78E-11	1.77E-11	1.79E-11	1.82E-11	1.87E-11	1.93E-11	2.00E-11	2.08E-11	2.17E-11	2.26E-11	2.33E-11
10.30	1.56E-10	1.54E-10	1.51E-10	1.50E-10	1.51E-10	1.53E-10	1.56E-10	1.59E-10	1.64E-10	1.69E-10	1.76E-10	1.82E-10	1.87E-10	1.93E-10	1.98E-10
10.70	1.18E-09	1.18E-09	1.20E-09	1.23E-09	1.25E-09	1.28E-09	1.31E-09	1.35E-09	1.40E-09	1.41E-09	1.44E-09	1.47E-09	1.50E-09	1.54E-09	1.57E-09
11.38	2.67E-08	2.66E-08	2.65E-08	2.65E-08	2.66E-08	2.67E-08	2.69E-08	2.71E-08	2.74E-08	2.78E-08	2.81E-08	2.85E-08	2.90E-08	2.94E-08	3.00E-08
12.06	3.59E-07	3.60E-07	3.63E-07	3.66E-07	3.70E-07	3.74E-07	3.79E-07	3.83E-07	3.87E-07	3.91E-07	3.94E-07	3.98E-07	4.01E-07	4.05E-07	4.09E-07
12.74	5.02E-06	5.09E-06	5.15E-06	5.22E-06	5.29E-06	5.34E-06	5.38E-06	5.42E-06	5.44E-06	5.46E-06	5.48E-06	5.51E-06	5.53E-06	5.57E-06	5.62E-06
13.42	7.24E-05	7.34E-05	7.41E-05	7.46E-05	7.49E-05	7.50E-05	7.51E-05	7.52E-05	7.54E-05	7.56E-05	7.60E-05	7.65E-05	7.72E-05	7.79E-05	7.86E-05
14.10	9.89E-04	9.94E-04	9.95E-04	9.94E-04	9.96E-04	9.98E-04	1.00E-03	1.01E-03	1.01E-03	1.02E-03	1.03E-03	1.04E-03	1.05E-03	1.05E-03	1.06E-03
14.50	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
14.90	0.09	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10
15.30	0.55	0.55	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.57
15.70	0.81	0.81	0.82	0.82	0.83	0.83	0.83	0.83	0.83	0.84	0.84	0.84	0.84	0.84	0.84
16.10	0.93	0.93	0.93	0.94	0.94	0.94	0.95	0.95	0.95	0.95	0.95	0.96	0.96	0.96	0.95
16.50	0.91	0.91	0.91	0.91	0.91	0.91	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92
16.90	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.52	0.52	0.52	0.52	0.52
17.30	0.30	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.30	0.30
17.70	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17
18.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10
18.56	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
19.02	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.02	0.02	0.02
19.48	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.01	0.01	0.01	0.01	0.01	0.01
19.94	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.00
20.40	8.00E-03	7.70E-03	7.41E-03	7.16E-03	6.95E-03	6.73E-03	6.50E-03	6.29E-03	6.07E-03	5.79E-03	5.41E-03	4.91E-03	4.37E-03	3.78E-03	3.12E-03
21.10	0.01	9.83E-03	9.18E-03	8.55E-03	7.97E-03	7.44E-03	6.93E-03	6.44E-03	5.96E-03	5.50E-03	5.02E-03	4.48E-03	3.87E-03	3.19E-03	2.45E-03
21.50	0.01	0.01	0.01	0.01	0.01	9.84E-03	9.09E-03	8.40E-03	7.74E-03	7.10E-03	6.48E-03	5.85E-03	5.18E-03	4.42E-03	3.59E-03
21.90	4.34E-03	3.99E-03	3.63E-03	3.30E-03	3.02E-03	2.78E-03	2.58E-03	2.42E-03	2.28E-03	2.23E-03	2.20E-03	2.17E-03	2.12E-03	2.05E-03	1.94E-03

9.4 Model Verification

The discussion on model verification requires testing of the optimal parameter sets obtained from the solution of the calibration problems. In order to facilitate it, the optimal parameter sets obtained by *algorithm* based on data from *date* are denoted as $\mathbf{z}^{*,\,algortihm}_{date}$. When *algorithm* is omitted it refers to the best $\mathbf{z}$ of the day.

Each of the optimal parameter sets given by Tables 9.3–9.6 was used for running a simulation with boundary conditions data (demands, densities and turning rates) from the other two. In each case J_v was calculated according to (4.10). Table 9.13 provides the values of J_s for the all $\mathbf{z}^*$, algorithms and dates.

Table 9.13: J_s for optimal parameter sets from calibration using data from different dates.

Algorithm	**Calib.**	**Verification date**			
	date	1st	8th	15th	8th (incident)
APSO	1st	55.84	154.98	151.99	88.61
	8th	579.97	55.63	604.04	137.45
	15th	77.83	149.90	51.45	64.38
LPSO	1st	54.20	159.83	155.87	82.76
	8th	826.59	38.36	727.41	140.89
	15th	77.44	155.76	47.62	73.45
HEPSO	1st	53.14	155.81	153.88	85.98
	8th	459.95	66.41	514.61	76.25
	15th	83.07	159.63	55.36	73.87
RPROP	1st	36.96	141.00	106.65	55.66
	8th	338.26	36.55	327.48	103.38
	15th	83.22	140.47	32.13	61.65

The diagonal elements in each algorithm block of the table are the minimum row values since they are the calibrated solutions to the specific date. The degradation of the model quality as demonstrated at the other two row entries is to be expected, since the parameters were not optimised for the corresponding set of data.

It can be seen from Table 9.13 that the optimal parameter vectors of the 15th are able to reproduce the congestion pattern of the 1st better than the other way around. When $\mathbf{z}^*_{1st}$ is applied on the data of the 15th, J_s increases by a nearly constant factor of 2.8 times the calibration error of the 1st. When $\mathbf{z}^*_{15th}$ is applied on the data of the 1st, the error increases by a factor of about 1.6 for the EA and 2.6 for RPROP. Hence, it can be concluded that the data of the 15th provide more information about the underlying traffic dynamics and the optimisation algorithms produce more relevant sets of parameters The value of J_s in these cases ranges from

77 to 83 $(\text{km/h})^2$, i.e. from 8.8 to 9.1 km/h per measurement, which is a very small difference. and consistent to what is reported in the literature [51], [41].

The main source of verification error comes from underestimating the congestion's queue tail extension and duration. Figure 9.14 shows the model and corresponding speed measurements when $\mathbf{z}^*_{15th}$ is used for the data of the 1st. The speed dynamics are accurately represented for the site, with the exception of an area outlined by two detector stations that is about 700 meters long. The model predicts for this length of road a faster speed recovery than the one observed, by about 40 minutes. The reasons for this result are not clear, but in view of the fact that solutions based on the 1st do not generalise as well as those based on the 15th, it could be attributed to an exogenous or irregular event factor. This behaviour is consistent with verification results reported in the literature, [51]. However, the overall congestion dynamics are reproduced with sufficient accuracy as can be seen from the resulting distance-time diagram in Figure 9.15.

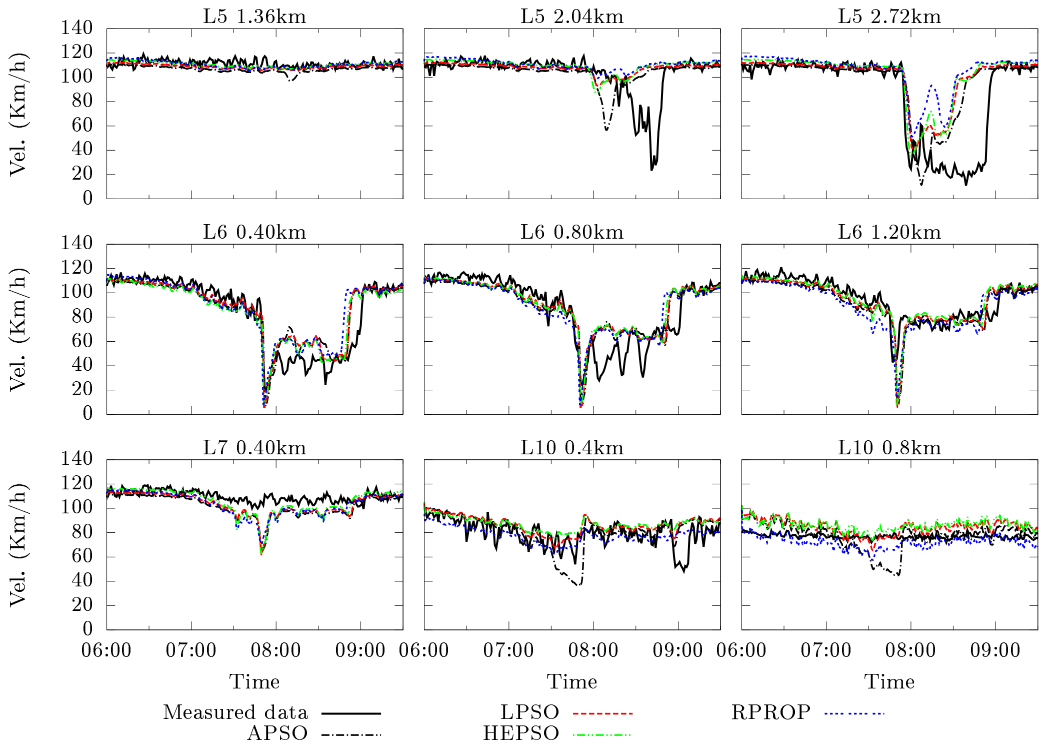


Figure 9.14: Model versus speed measurements when $\mathbf{z}^*_{15th}$ is used with the data of the 1st.

Another pronounced feature on Table 9.13 is the poor generalisation of the parameters obtained based on data from the 8th. Although the calibration error is at the same level as for the other dates, once the $\mathbf{z}^*_{8th}$ of any algorithm is applied to any other date, the error becomes very large. A similar degradation is observed when

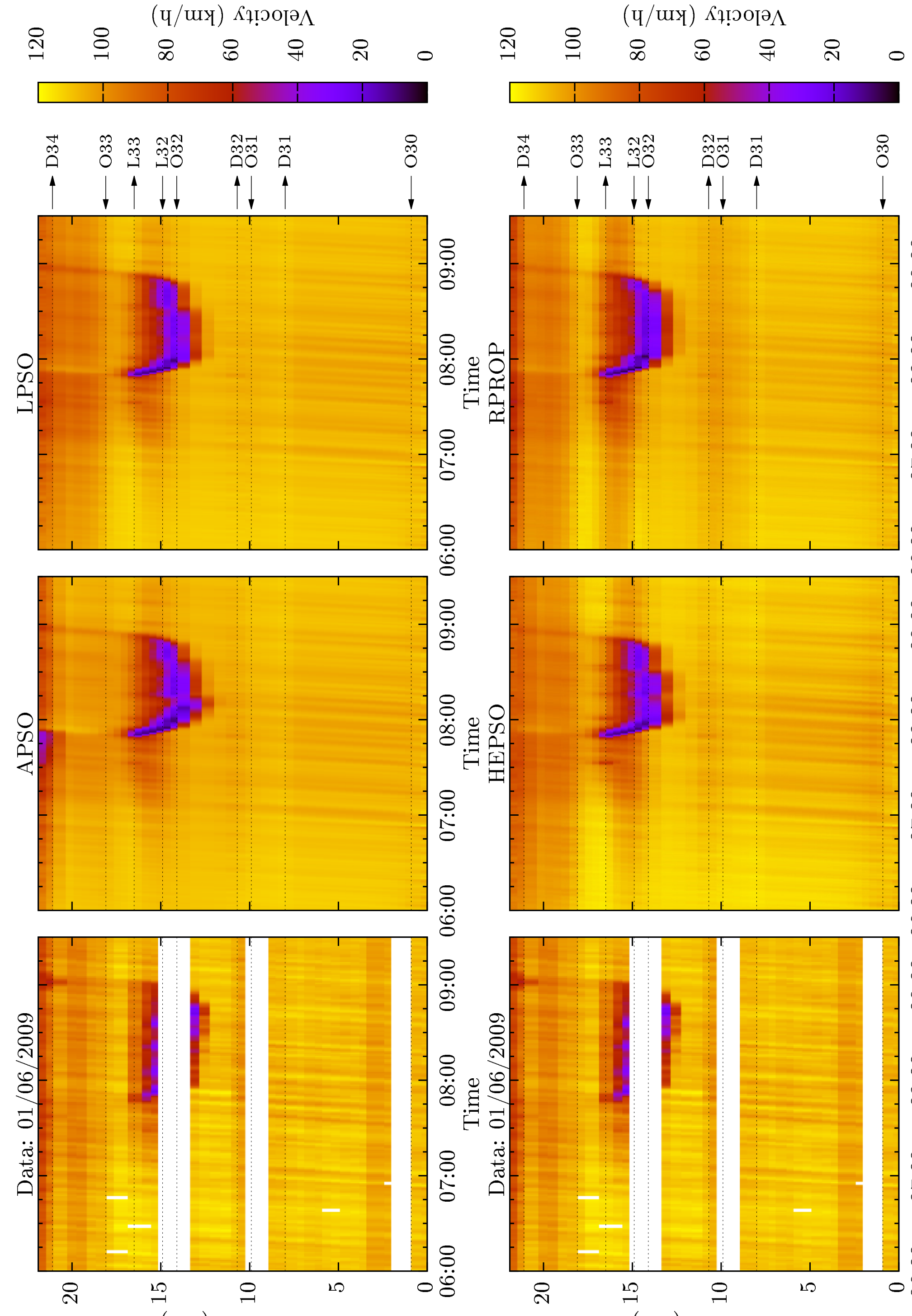


Figure 9.15: Distance-time diagrams of model output when $\mathbf{z}^*_{15th}$ is applied on the data of the 1st.

one of the $\mathbf{z}^*_{1st}$ or $\mathbf{z}^*_{15th}$ is applied on the data of the 8th, signalling a problematic situation on that particular day.

A closer examination of the data leads to the conclusion that there is a prolonged spill back of congestion from the surface are into the highway from D33 for this date. Figure 9.16 depicts the mean speed trajectories of every lane of link L6 for the three days of data. A recurrent drop of speed can be observed. The speed drop on the 8th for the first lane is much higher than in the other days but also it is higher compared with the other two lanes. This is related to a spill back of congestion from L33 and D33 during the short period of 07:40–07:54. However looking at the downstream density demand (Figure 9.9) we see a lower density during this period on the 8th. So something which is not captured in the model dynamics is causing a speed drop in the left hand lane without the destination being as heavily congested. This could be the result of late lane changes resulting in further speed drops in the main line.

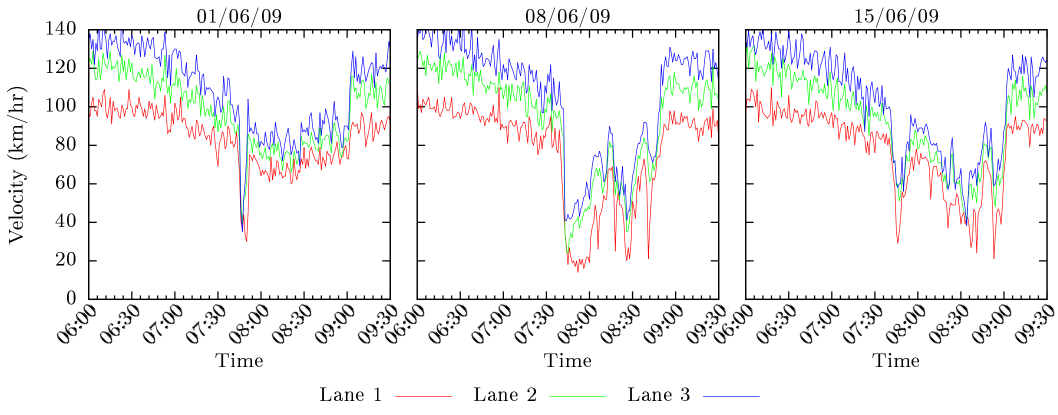


Figure 9.16: Link L6 lane mean speeds.

This effect is exogenous to the traffic dynamics of the system since it is caused by the traffic conditions of the surface streets. In order to compensate for it a small incident was introduced at L33 for the duration of the speed drop. L33's capacity was dropped to 875 veh/hour/lane, or 1,750 veh/hour for both lanes, during this period. The outcome of this intervention is shown on the last column of Table 9.13; the average square mean error is reduced for all parameter sets. Based on the results of Table 9.13, $\mathbf{z}^{*,\,RPROP}_{15th}$ is the best set of parameters identified. It is interesting to note however that the CTM model verified in the previous chapter was based on the calibration of this data set. It appears for this site that the CTMs method of handling destinations is more robust and provides more accurate representation. Part of the reason of adding the extra off ramp link was to try and address this issue.

The solutions generated are better and with a small perturbation has produced a verified model, but the CTM did not require this intervention.

9.5 Conclusion

This chapter has presented a study on METANET model validation using for the first time a gradient based optimisation. This has been done alongside a verification attempt involving the previously found best PSOs. The focus has been on gradient based optimisation but results from evolutionary algorithms were presented as well. The underlying model used is the well-known METANET simulator, which has been combined with automatic differentiation software (ADOL-C) and numerical optimisation (RPROP) for developing a system of traffic flow model parameter estimation. The additional requirement of automatically selecting each fundamental diagram's location and extension is an explicit feature of the evolutionary optimisation algorithms but not of the gradient based where this constraint is implicitly considered in the optimisation problem formulation with variations of the fundamental diagram parameters being penalised.

Data from three different days were used for calibration and verification. It has been shown that not all data given as input for parameter identification were suitable for this purpose. Their ability to guide optimisation algorithms to converge to solutions encapsulating the underlying traffic flow dynamics is not always present. In fact this study raises the question of how to qualify sets of data to be used for calibration. All three data sets used display a recurrent and relatively regular pattern of congestion. However, only one of them proved to be suitable for calibration. A good result from the optimisation algorithm minimising the error is not a sufficient indicator of a valid model. The general rule of thumb for selecting data sets ensuring representative measurements of free, critical and congested conditions is not enough neither. Complex interactions may be overlooked resulting to non-representative parameters and therefore a severe model–reality mismatch. Sufficient criteria and conditions need to be determined in order to automate this qualification process, something which can be based on the notion of persistence of excitation.

The solution of the calibration problem using gradient based optimisation is one of this chapter's original contributions. It has been demonstrated here that the calculation of the necessary partial derivatives is possible by using automatic differentiation software technology. The ADOL-C library employed has proven to be a highly flexible and robust piece of software that can be easily incorporated into the source code of a macroscopic traffic flow simulator. A by-product of the overall gradient calculation, is the calculation of the state's Jacobian matrix with respect to the model parameters. This yields additional information regarding the mean speeds' sensitivities with respect to the infrastructure-vehicle-driver parameters that can be exploited for improving control design and interventions to the traffic system. A more detailed investigation of how these sensitivities can be utilised in a rigorous and systematic manner is a future direction of research.

The fact that gradient information is obtainable via AD opens the way for using well known and established optimisation methods. In this thesis the RPROP algorithm as an optimisation search heuristic has been used. It has been shown that gradient based solvers are capable of solving this highly complicated and demanding problem in a fairly efficient manner, contrasting the comments made in [54]. Its simplicity of implementation has allowed the integration of the three different source codes into a single system, i.e. METANET, ADOL-C and RPROP.

Chapter 10

Large Scale Network Validation: Manchester

10.1 Introduction

This chapter involves the calibration and verification of the large scale network of the M60 orbital around Manchester and the connected motorways. The site is discussed in section 6.4.4 and covers a total length of 186.92 km. Previously in the literature a large scale network has only been validated once in [44, 48]. That work modelled a network of 143 km around Amsterdam, but the implementation required resorting to intuition and engineering expert opinion. This included the location and extension of FDs into the problem. The validation work detailed in this chapter avoids this by using the AAFD method developed as part of this thesis so that the use of experts' knowledge is replaced by the optimisation process. Furthermore, the validation of the Amsterdam network was conducted in two phases, quantitative and qualitative. In the quantitative phase, parts of the network in the form of unidirectional highway stretches were selected and a calibration optimisation problem was solved for each of them, yielding an area specific optimal set of model parameters. These were transferred to the rest of the network based on expert engineering knowledge and intuition during the qualitative model validation phase. The purpose of this second phase was to enable the model to capture the network wide dynamics of congestion propagation. By manually tuning suitably selected parameters and using the optimal parameter sets obtained from the quantitative phase, a fairly accurate representation of the traffic conditions in the whole network during the morning rush hours was achieved [44].

The validation process used in this chapter avoids the identification of sub-networks prior to optimisation and involves calibrating the system as a whole. Two different types of algorithms are considered, PSO [1] and the gradient-based RPROP

[7], [2]. This allows us to determine if the novel method tested in the previous chapter of using a gradient based optimisation using AD through the ADOL-C library [127] is scalable and suitable to large scale model validation. This chapter provides the first reported use of a gradient based solver on a large-scale network. The Manchester network considered here is larger than the previously reported network of Amsterdam. The direct validation of a large scale ring-road network based on real data is the contribution of this chapter.

The chapter starts with a description of the problem to be solved and an overview of the site under consideration. This is followed by a discussion of the obtained calibration results with a look at the convergence profiles, the parameter assignment pattern and the correlation to measured data. Verification results are then discussed, followed by a look at the sensitivity diagrams for the network.

10.1.1 Problem formulation

The calibration process was discussed in Chapter 4 (see Figure 4.8) and follows the same outline as the previous chapters. The objective function used in this chapter only considers the squared errors in speed measurements as shown in (4.21) and repeated here for clarity,

$$J(\mathbf{z}) = \frac{1}{KY}\sum_{k=1}^{K}\sum_{j=1}^{Y}\left(E_{v,j}\left[\mathbf{x},\mathbf{y},k\right]\right) + w_p \sum_{m=1}^{M}\sum_{\substack{\mu\in O_{a_m}\\ \mu\notin\Phi}} (\mathbf{z}_{\mathrm{L},m}-\mathbf{z}_{\mathrm{L},\mu})^\top w_{\mathrm{L}}(\mathbf{z}_{\mathrm{L},m}-\mathbf{z}_{\mathrm{L},\mu}). \tag{10.1}$$

The objective function parameters are the same as the previous chapter with w_{L}: $w_{\mathrm{L},v}$, $w_{\mathrm{L},\rho}$ and $w_{\mathrm{L},\alpha}$ set to 0.001, 0.0015 and 1.0 respectively. The overall penalty weighting is reduced however to account for the increased model size and w_p is set to 0.5. The search domain is restricted to within the limits given in Table 10.1.

Table 10.1: Traffic flow model parameters upper and lower limits.

Variable	τ	κ	ν	$v_{\min}$	$\rho_{\max}$	δ	ϕ	α_m	$v_{f,m}$	$\rho_{cr,m}$
Maximum	40	30	80	8	190	4	4	3.5	130	45.0
Minimum	1	5	1	0.5	160	5E-5	5E-5	0.5	60	18.0

The two PSO variants considered here, are the ones that have achieved good convergence behaviour as reported in the previous chapters, [1] and [2], namely Local PSO [128], [129] and High Exploration PSO [75]. The PSOs use a swarm size of 30 and RPROP is started from 6 different locations simultaneously. The site being validated is that of Manchester which is discussed in Chapter 6. The scope of the optimisation problem is the entire network of 186 km. A single evaluation of the objective function at point **z** involves: (a) the simulation of traffic for three and a quarter hours in all highways using **z** (b) the extraction of the model outputs from the simulation produced files and (c) the comparison of the model speeds with the measured speeds. Despite the fact that a single optimisation problem is solved, because PSO uses a swarm of **z**s and RPROP is used in its multi-start version [88], repeated simulations are performed by invoking the METANET executable. As a result, in each iteration of the optimisation algorithms multiple simulations are performed. For RPROP this is equal to the number of initial points used whereas for the PSO algorithms equals the swarm size.

10.2 Site Overview

For full details on the site being studied see Section 6.4.4, but a summary is included here for reference. The site considers both carriageways of the M60 orbital at Manchester and connecting motorways. The total length of highways modelled is 186.92 km for which there are 56 origin, 55 destination and 192 highway links. The main highway is the M60 motorway encircling the greater area of Manchester. Each of the motorways connecting to the M60 is modelled only for a short length except for the M56, which is modelled for a longer stretch extending past the Manchester Airport access. Also included is a section of the A5103, where no data are supplied since it is not part of MIDAS (the traffic data collection and archiving system operated by Highways England).

By utilising the methods developed in Section 4.2.3 the network was divided into a total of 39 linear sub-networks using Algorithm 4.3. The two main features of the ring-road are sections 2 and 4, which are the anti-clockwise and the clockwise directions, respectively, around the city. Each of these two parallel rings with opposite direction of flow is about 62 km long.

The data used were collected on May 2012, with calibration data sets being from Monday 14th, 21st and 28th. The morning congestion runs between 06:00 and 09:30. A major congestion wave occurs recurrently between the merge of M61 and M62 with shock waves along the joining stretch of the anti-clockwise direction of M60. Further congestion is seen at small localised regions along M60. Another large congestion region occurs on the clockwise M60 direction on the southern side from Junction 27 leading up Junction 4, which is the diverge for the M56. This congestion causes some speed drops along the start of the southbound M56 but it recovers quickly.

The METANET model for this site is defined by a parameter vector with 583 dimensions. That is 3 for each link ($3 \times 192 = 576$) to define the FDs and aa further 7 for the global parameters. This is the size of the problem given to RPROP with its simplied version of AAFD, but for the PSOs the algorithms were allowed to use upto

7 FDs per section requiring the problem vector to be increased by 101 dimensions to allow for the lengths to be defined, note not all sections have more than 7 links hence why the number of extra dimensions is less than the number of sections (39) times 7. So that makes the problem 684 dimensions in total for the PSOs.

10.3 Calibration Results

10.3.1 Objective function values

In order to facilitate the discussion, notation of the optimal parameter set obtained used in the previous chapter is extended to include the run number. The solution of the optimisation problem (calibration) using the data of *date*, when *algorithm* was run for the nr-th time (*nr run*) is denoted as $\mathbf{z}^{*,\ algorithm}_{date,run\ nr}$.

The METANET–RPROP–ADOL-C package and the PSO based system were provided with the three separate data sets and for each one three runs where performed. Since all algorithms are essentially population based with a significant element of exploration, along with exploitation, each run on its own results to a comprehensive search of the solution space.

Table 10.2 provides the objective function value J eqn. (10.1) and the mean square error J_v the first term of (10.1) for the three runs of the three algorithms using the data of the three days. With the exception of the third run of RPROP using the data of the 21st, the three algorithms converge to solutions that achieve an average square error of the speed from 311.65 (km/h)^2 – HEPSO applied on the data of the 21st – to 108.54 (km/h)^2 – RPROP applied on the data of the 14th – i.e. errors on the order of $\pm$17.7 km/h to $\pm$10.4 km/h.

It can be seen that RPROP outperforms HEPSO and LPSO on all data sets. This is to be expected, since AAFD is applied differently to RPROP where every highway link has its own FD providing RPROP with more degrees of freedom. The best calibration result is achieved by RPROP using the data of the 14th. The three RPROP runs with this set of data result to consistently low objective function values. This is true for the solutions based on the data of the 28th, but on average the errors are higher than those of the 14th. The calibration based on the 21st shows higher variance and a failure to converge in the third run. This last run is considered as an exception and is included here in order to show that such results are a possibility, hence the need for more than one run.

Comparing the two PSO algorithms, LPSO systematically outperforms HEPSO. It is difficult to provide an explanation for this result. The problem's scale and stochastic nature of the search makes it very difficult to identify the reasons behind this result. A possible reason could be attributed to the algorithm parameters'

Table 10.2: Calibration results for three runs of each optimisation algorithm for each of the data sets; bold type indicates the best results with respect to J_v.

Date &	**HEPSO**		**LPSO**		**RPROP**	
run nr.	J	J_v	J	J_v	J	J_v
14th						
1	315.77	291.79	182.69	168.78	150.91	**108.54**
2	317.50	290.57	178.92	**164.02**	160.98	111.92
3	310.02	**265.65**	187.12	175.98	160.80	115.73
21st						
1	351.28	318.10	261.60	250.39	204.16	155.64
2	338.77	**311.65**	250.22	236.61	200.02	**151.43**
3	339.31	316.53	251.39	**234.88**	2367.18	2290.37
28th						
1	303.79	277.67	217.57	206.80	171.01	131.39
2	270.71	**250.52**	206.02	194.18	160.85	**128.19**
3	289.76	268.81	204.57	**190.80**	175.84	132.22

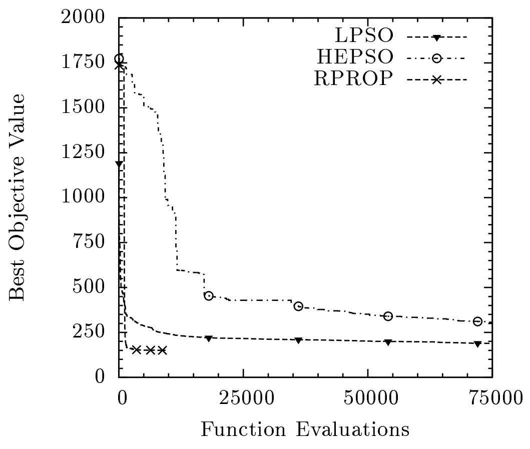
values used. These were selected according to the suggestions provided in the papers they were taken from. A more systematic investigation campaign, based e.g. on hyperheuristics [130] may yield better results, but in view of the problem's complex nature a significant effort would be needed and is left for future research.

10.3.2 Algorithm Convergence

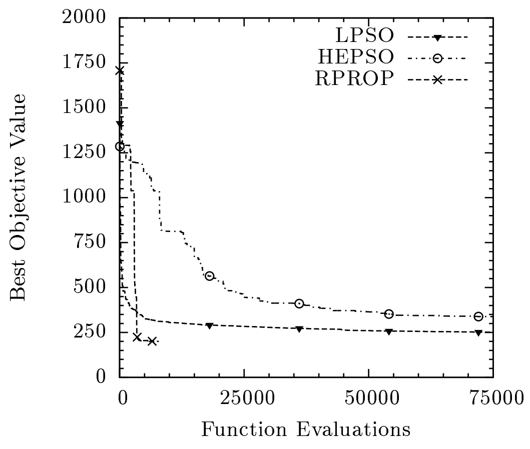
The convergence profiles for the best run of each algorithm are shown in Figure 10.1. This shows clearly the benefit of the additional information that the gradient based METANET–ADOL-C system can achieve with convergence happening much quicker and using less function evaluations. However, each function evaluation is more intensive for the RPROP system and it takes considerably longer for an individual model run when the calculation of the model Jacobian is also required. See Chapter 7 for more details on timings for the individual algorithms.

As in the previous chapters we see that LPSO makes improvements more quickly at the start of the search than HEPSO. Often for the shorter sites HEPSO achieved a better final solution that those found by the simple PSO variant, but this is not the case here. As before HEPSO is able to make improvements throughout the search. The rate of improvements slows down as the search progresses and the sudden drops in the objective function value become less often. The convergence profile tends towards a smooth curve with very small improvements at termination.

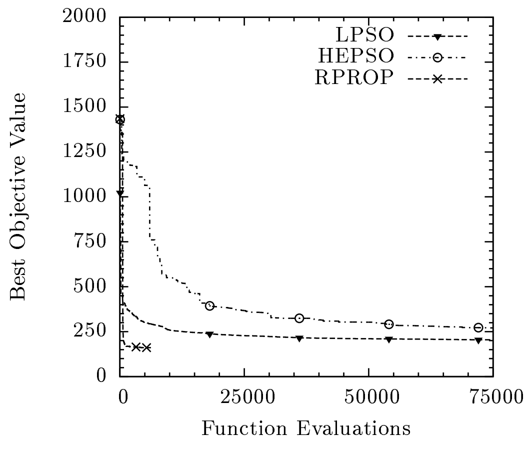
Figure 10.2 shows the convergence profile for each of the RPROP starting locations for a selection of different calibration runs. It can be seen from this that a lot of the starting locations are unable to find a good improvement beyond their initial



(a) 14th.



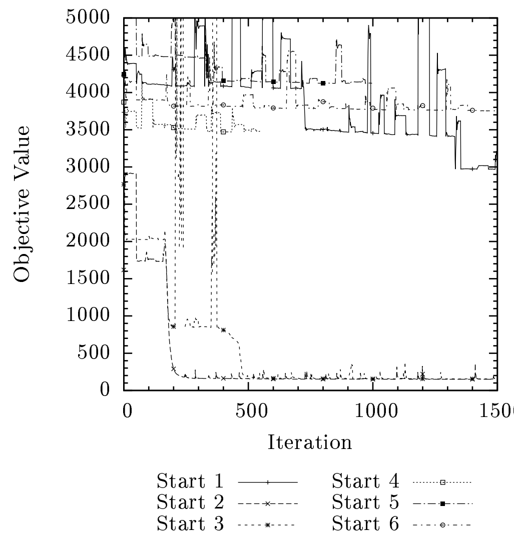
(b) 21st.



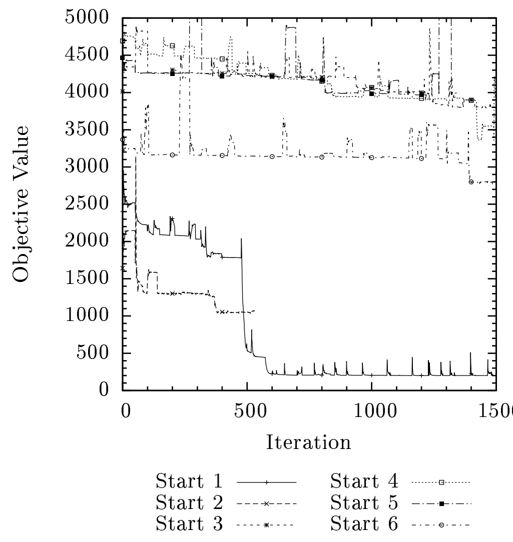
(c) 28th.

Figure 10.1: Convergence profile of the best run for each of the optimisation algorithms.

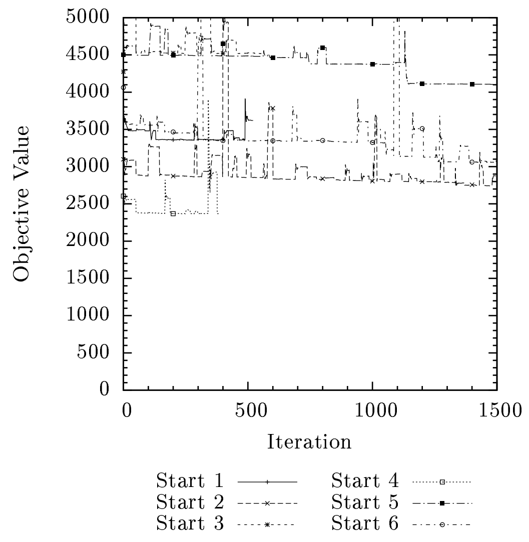
starting point and cannot escape the local minima from which they started. For the three panels where a good solution was obtained (a, b and d) at most two of the six starting locations yielded a good result. In panel (c) the result is shown for run 3 of the dataset of the 21st which failed to find a suitable optimum. Here all of the convergence profiles failed to make an improvement and the best solution found is not much better than the randomly selected starting point.



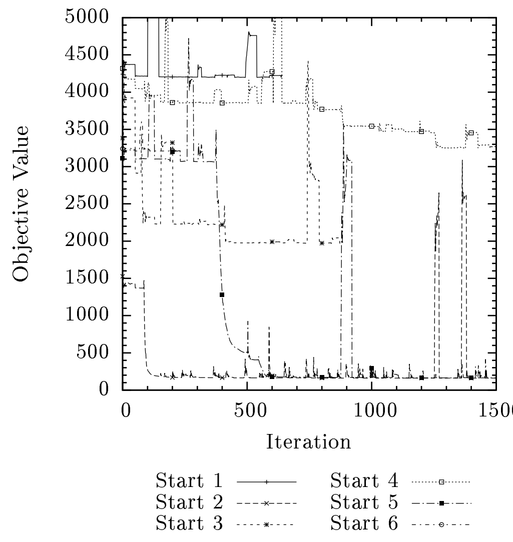
(a) 14th run 1.



(b) 21st run 2.



(c) 21st run 3.



(d) 28th run 2.

Figure 10.2: RPROP convergence profiles for every point in search.

Figure 10.2 shows an interesting set of characteristics the RPROP algorithm is very quick at exploiting and searching the current region and as soon as a restart moves the search to a suitable region of the solutions space the convergence is very

Table 10.3: Network-wide parameters of optimal solutions.

Algorithm	**Calib. date**	τ (s)	κ (veh/km/lane)	ν (km^2/h)	$v_{\min}$ (km/h)	$\rho_{\max}$ (veh/km/lane)	δ (h/km)	ϕ (h/km)
HEPSO	14th	16.661	23.282	27.938	5.45	184.06	0.06339	0.78066
	21st	27.092	20.349	55.509	4.12	170.94	0.43047	0.16419
	28th	27.857	20.594	44.114	4.49	179.93	0.89671	0.53272
LPSO	14th	12.812	29.981	52.189	5.95	169.13	0.00124	0.01483
	21st	23.555	29.956	57.710	3.47	173.67	0.06117	0.00155
	28th	19.427	29.963	48.046	3.25	187.87	0.00107	0.00150
RPROP	14th	19.170	24.649	47.900	0.50	177.35	0.00008	0.00008
	21st	22.436	30.000	55.655	8.00	170.10	0.00008	0.01550
	28th	22.410	30.000	49.823	7.99	178.82	0.00013	0.00005

rapid and is seen in the graphs by the sudden reduction in the current objective function value. It is the restart method that allows the point to break free of it current local minima and this can be seen in the convergence graphs; each of the sudden drops in the objective value is preceded by a peak which is caused by the algorithm moving in the direction of steepest descent at a fixed initial step length value.

As in the previous chapter it can be seen that RPROP adds a lot of intensity even though it was run for fewer function evaluations RPROP calibrations typically took around 9 hours and 30 minutes whilst the PSO calibrations were completed in 3 hours and 30 minutes. The reasons for this and the scaling of the complexity of the run times all comes down to the scaling of the gradient calculation with increasign model length which was discussed in Chapter 7.

10.3.3 Identified Parameters

The optimal parameter sets are given in the form of tables and figures. The global parameters of the model, i.e. those that are applicable to the whole network and are not part of the links' FDs, are given in Table 10.3. The FD parameters that constitute the rest of the optimal parameter vectors are provided graphically as spatially distributed quantities over the different linear subnetworks. Figures 10.3 and 10.4 depict them for the two ring-roads with opposing direction of flow/.

The $x-$axis of the diagrams represent distance and since these are ring roads, the start and the end are geographically close. For each highway link m in the subnetwork a_m, $v_{f,m}$, $\rho_{cr,m}$ and the resulting capacity flow are shown as constants over their length L_m. The large variance over space of the FD parameters is clearly shown for the RPROP solutions. This is to be expected as the optimisation exploits

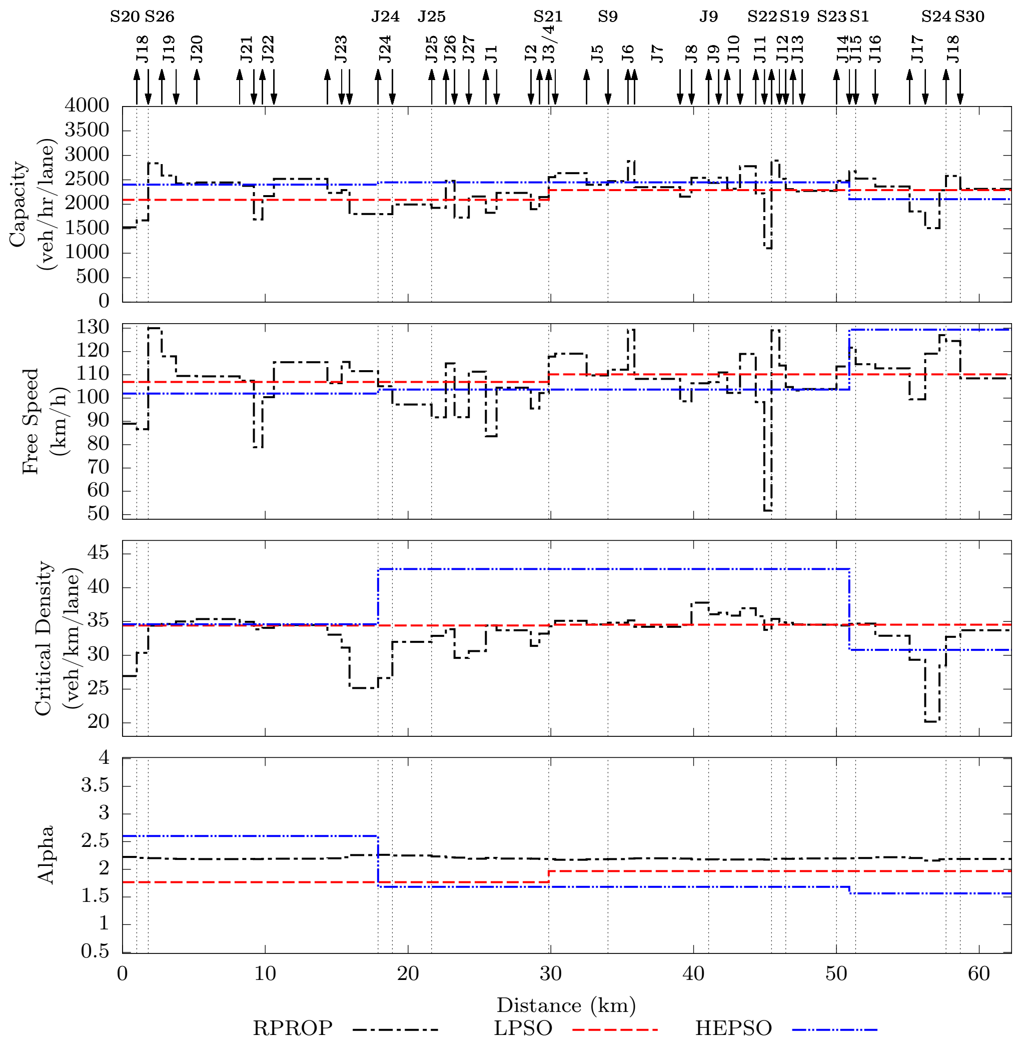


Figure 10.3: Optimal fundamental diagram parameters of M60 for the three algorithms using the data of the 14th: clockwise direction

the potential given by the freedom to use a single FD for each highway link, whereas the PSO algorithms are restrained by constraints of AAFD.

10.3.4 Model–Data correlation

A partial set of the results obtained from the calibrated model for the two ring-road sub-networks is shown on the time-distance diagrams of Figures 10.5 and 10.6. The space mean speed is depicted when $\mathbf{z}^{*,RPROP}_{14th,3}$ is used and $\mathbf{d}$ is populated with measurement from the same date. In the left hand plot the white strips are areas where data were not available. Both sub-figures present a comparison of the speed measurements, on the left, and the corresponding model prediction on the right. The empty spaces in the measurements sub-graphs correspond to areas without any loop detectors, i.e. for those locations there are no data available. The optimisation

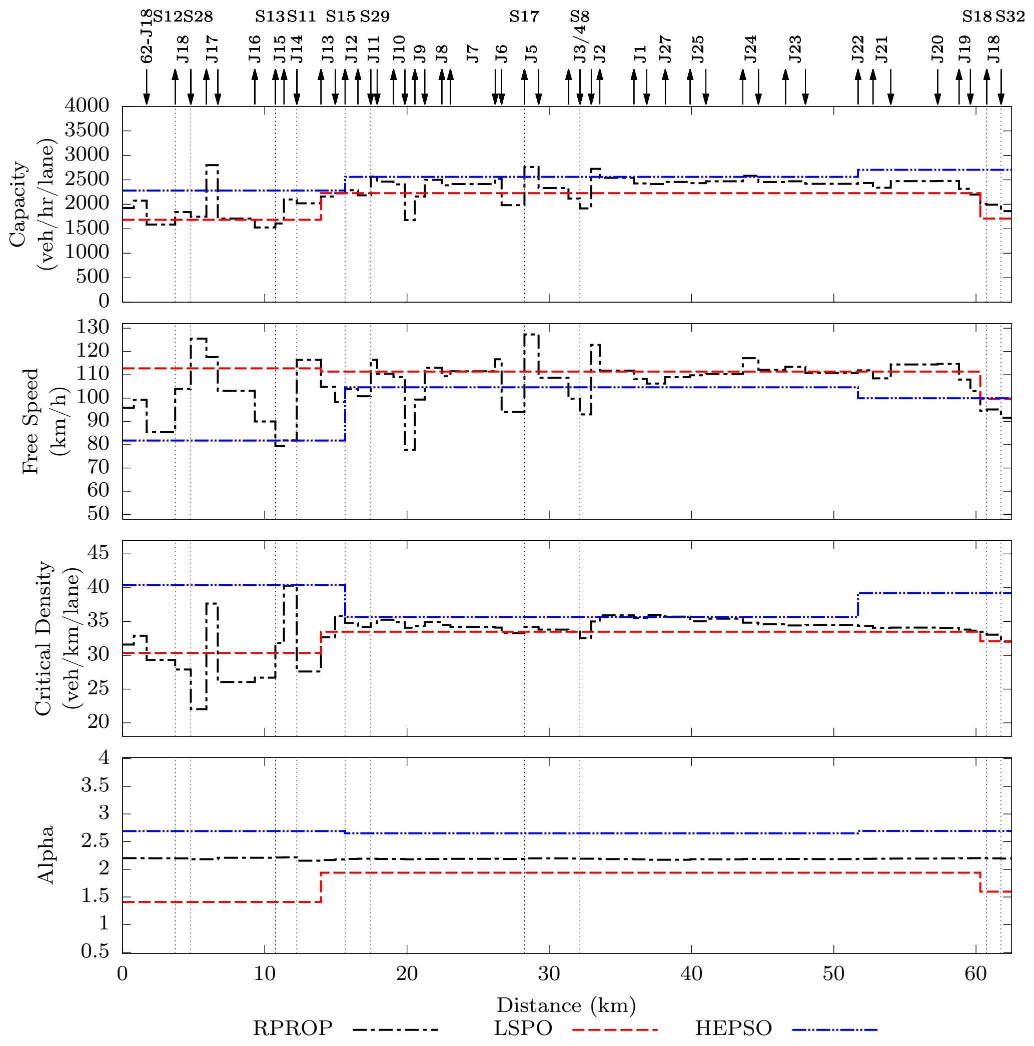


Figure 10.4: Optimal fundamental diagram parameters of M60 for the three algorithms using the data of the 14th: anti-clockwise direction

aims at identifying those model parameters that make the right sub-graph as similar as possible to the left for the areas where data exist; the rest which are not measured are inferred by the traffic flow model dynamics filling in the gaps.

Figures 10.7 and 10.8 show the the correlation between the measured and predicted speed for the two main loops of the model. The reason for the larger error for the HEPSO model becomes apparent, the parameter set identified is not able to reproduce the speed drops for the clockwise direction. The solutions for LPSO and RPROP do capture these. In the LPSO model in the clockwise direction the speed drop at J9 is not fully captured but a larger drop in speed is prediced than in the solution found by RPROP. The RPROP solution is superior in predicting congestion at the diverge to S24 with it showing a reduction in speed while free-flow conditions persist in the LSPO model. In the anti-clockwise direction the congestion predicted

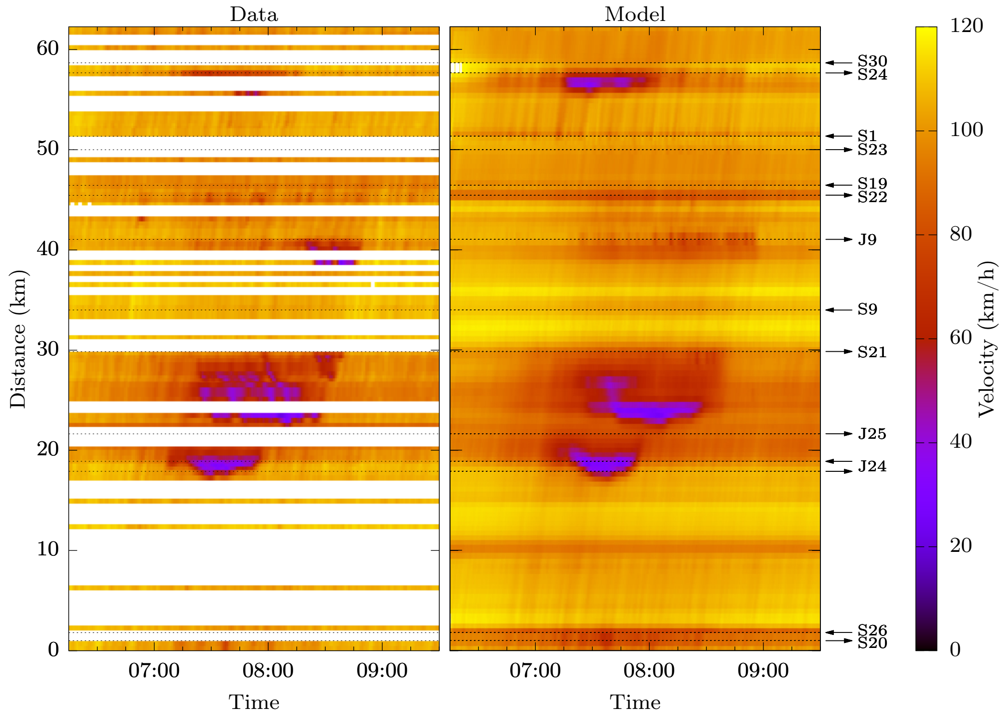


Figure 10.5: Calibration results for $\mathbf{z}^{*,RPROP}_{14th,\,3}$: measurement vs METANET speed time-distance diagram of M60: clockwise direction.

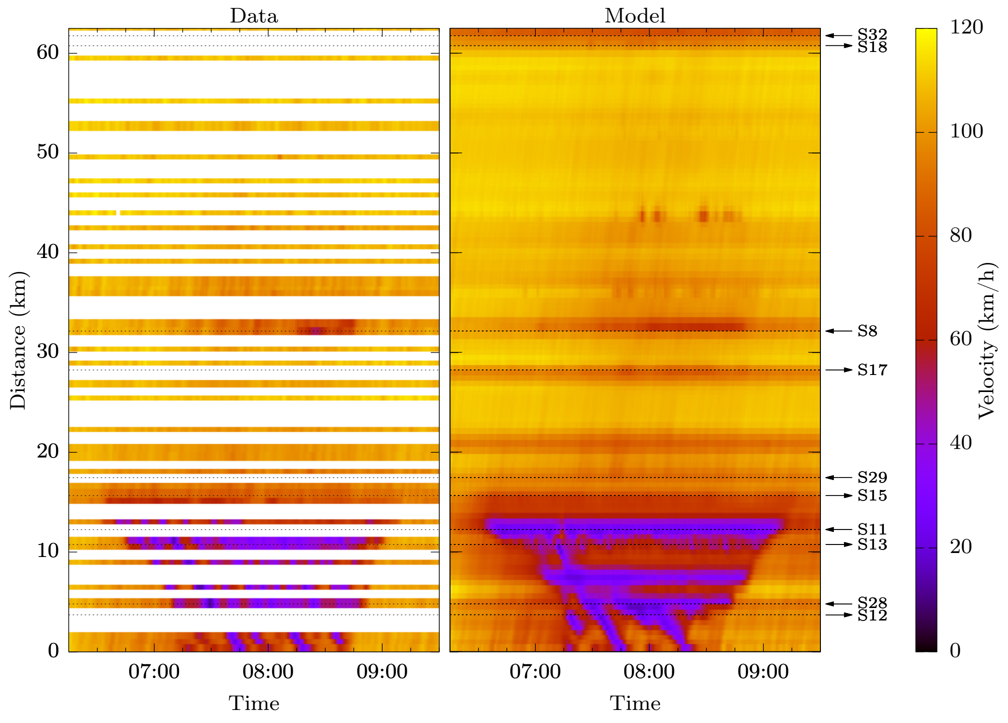
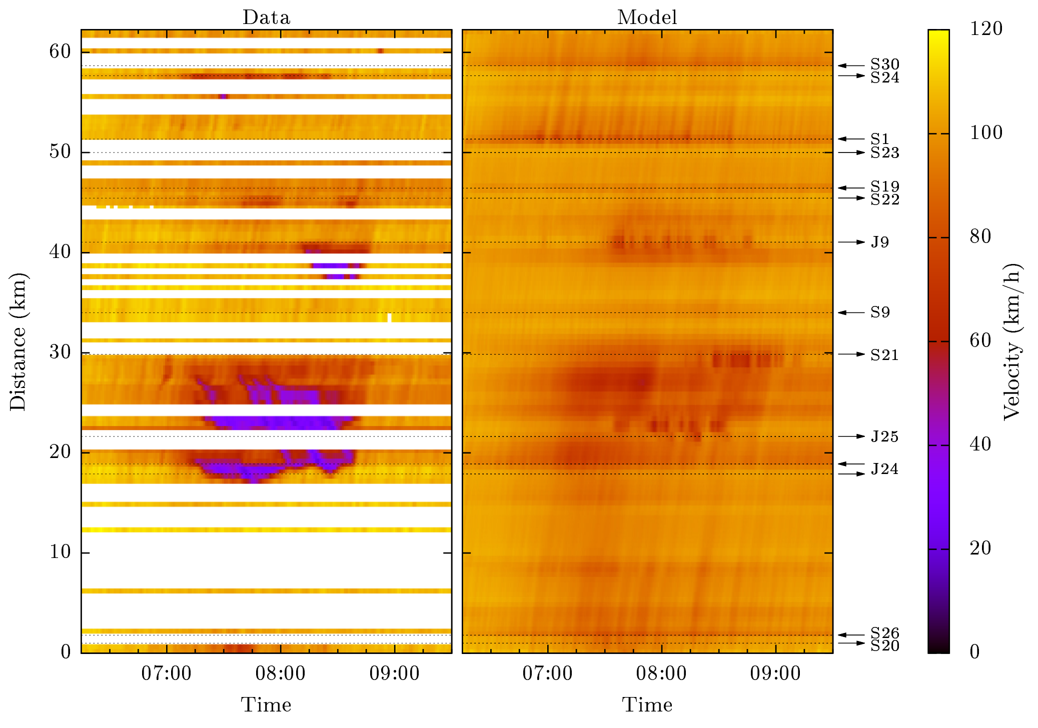
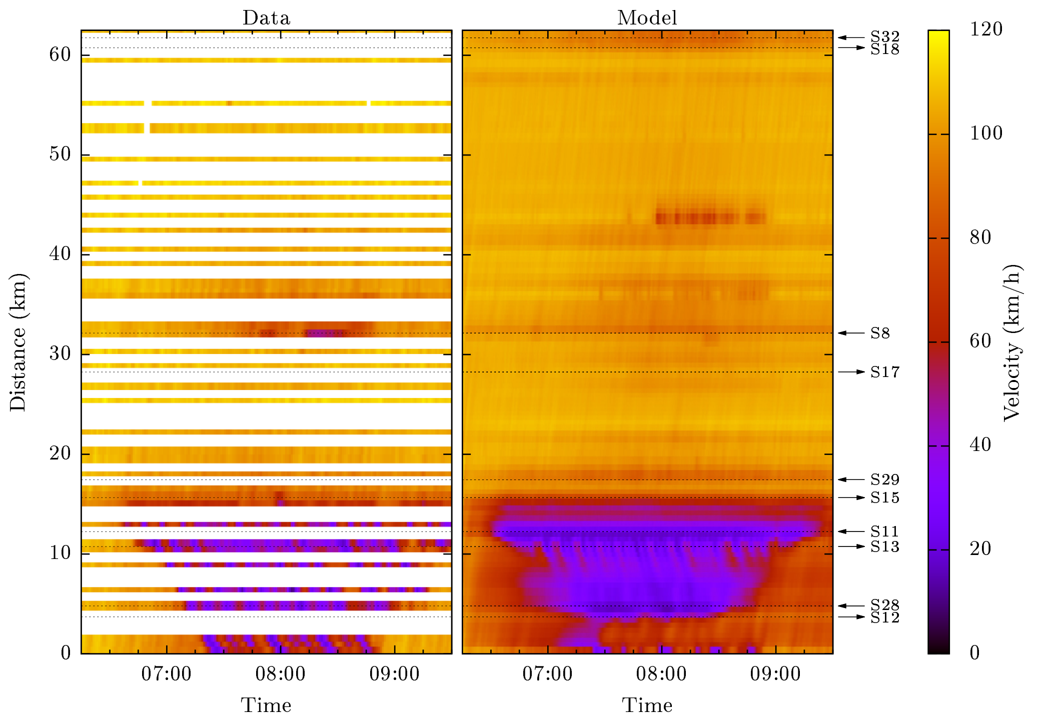


Figure 10.6: Calibration results for $\mathbf{z}^{*,RPROP}_{14th,\,3}$: measurement vs METANET speed time-distance diagram of M60: anti-clockwise direction.

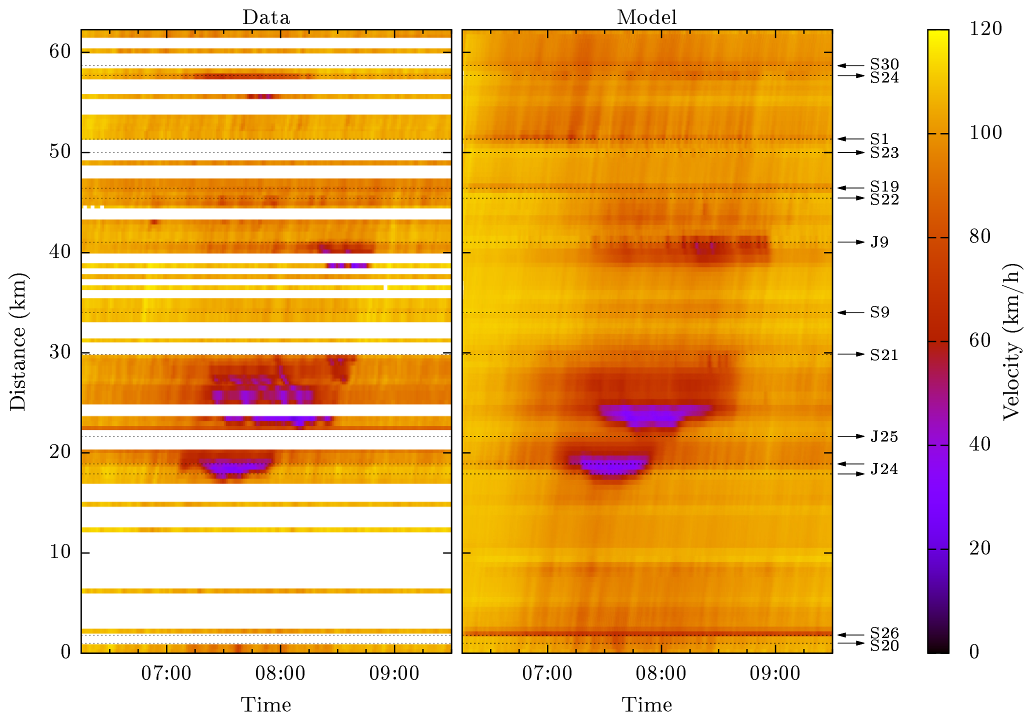


(a) Clockwise direction - section 4.

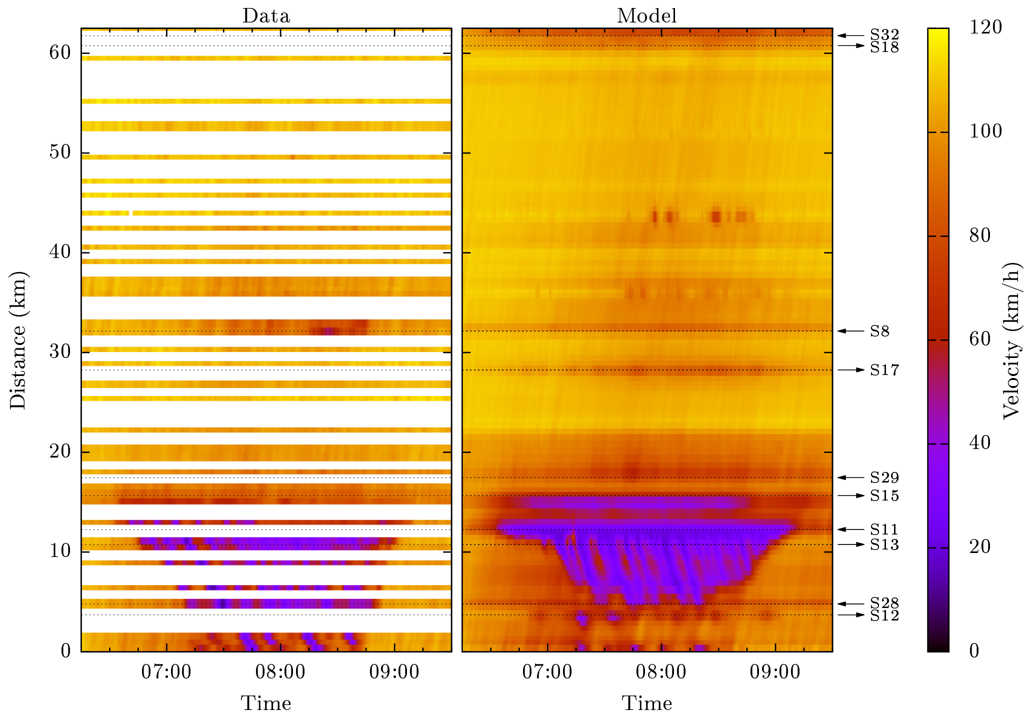


(b) Anti-clockwise direction - section 2.

Figure 10.7: Calibration results for $\mathbf{z}^{*,HEPSO}_{21st,\,2}$: measurement vs METANET speed time-distance diagram of M60.



(a) Clockwise direction - section 4.



(b) Anti-clockwise direction - section 2.

Figure 10.8: Calibration results for $\mathbf{z}^{*,LPSO}_{14th,\,1}$: measurement vs METANET speed time-distance diagram of M60.

at the start of the LPSO model does not propagate up to the boundary as it does for RPROP.

It can be seen that the congestion dynamics are replicated with sufficient accuracy in terms of extension and duration for the RPROP solution. The strength and duration of the congestion at the junction of M60 with M61 and M62 dominates the traffic in the whole network and the optimisation algorithm converges to solutions ensuring this feature's replication, as can be seen in Figure 10.6. The model performs quite good for the clockwise direction as well, although with a small overestimation of speed. The main block of congestion in that direction is not represented adequately in the data, Figure 10.5. For the locations, however, where there are measurements and the mean speed is low, the model is able to replicate the dynamics of congestion.

10.4 Verification

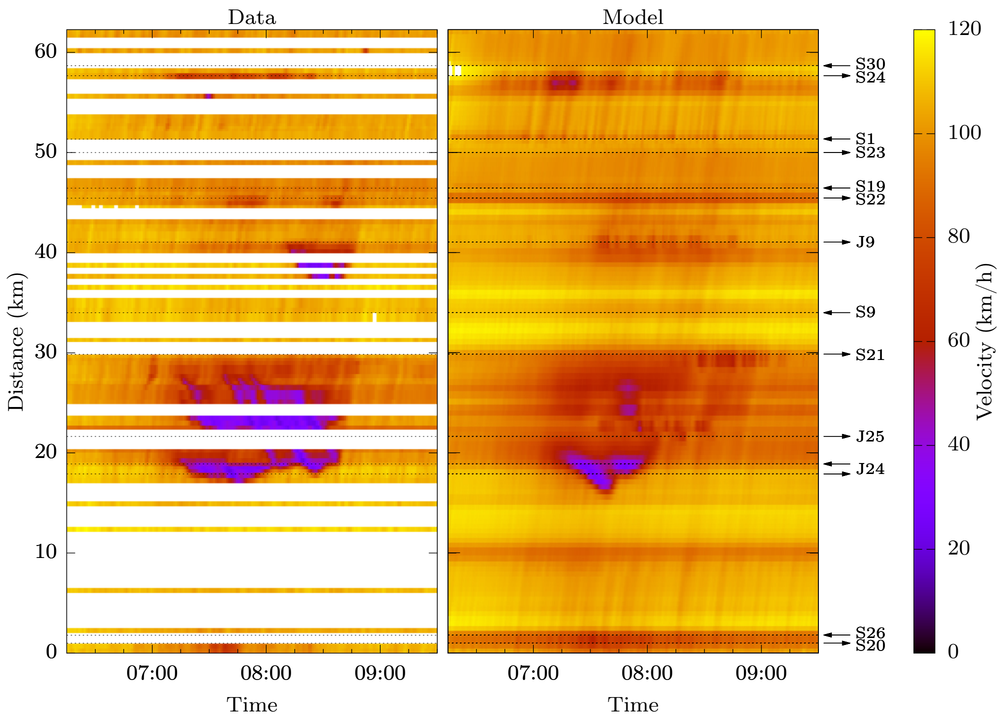
Table 10.4 shows the values of J_v when the first, second and third best set of parameters are used on the data of the three different days (including the set of data used for identifying them). For example, in the 1st best section of Table 10.4, the first row shows the result of applying $\mathbf{z}^{*,HEPSO}_{14th,3}$ on the data of the 14th, 21st and 28th; the value 265.65 at the first column is shown also in Table 10.2 in boldface because it is the first best run of HEPSO on the data of the 14th. The first best run of HEPSO on the data of the 21st is obtained in the second run, and therefore the second line of the 1st best section of Table 10.4 shows the result of applying $\mathbf{z}^{HEPSO,*}_{21st,2}$ at the three days. Scanning the table horizontally, the degradation in the performance of each $\mathbf{z}^*$ when it is applied on data of the days other than those used for identifying it, can be seen. The last column shows the average J_v, including the calibration day, achieved by each $\mathbf{z}^*$.

It can be seen that the best result on average, is achieved by the worst RPROP run using the data of the 14th, i.e. $\mathbf{z}^{*,RPROP}_{14th,3}$. It is the RPROP solution based on the data of the 14th that generalises better, particularly when $\mathbf{z}^{*,RPROP}_{14th,3}$ is applied on the data of the 28th, which results to a mean speed square error equal to 178.57 $(\text{km/h})^2$ and an average error over the three days equal to 173.56 $(\text{km/h})^2$.

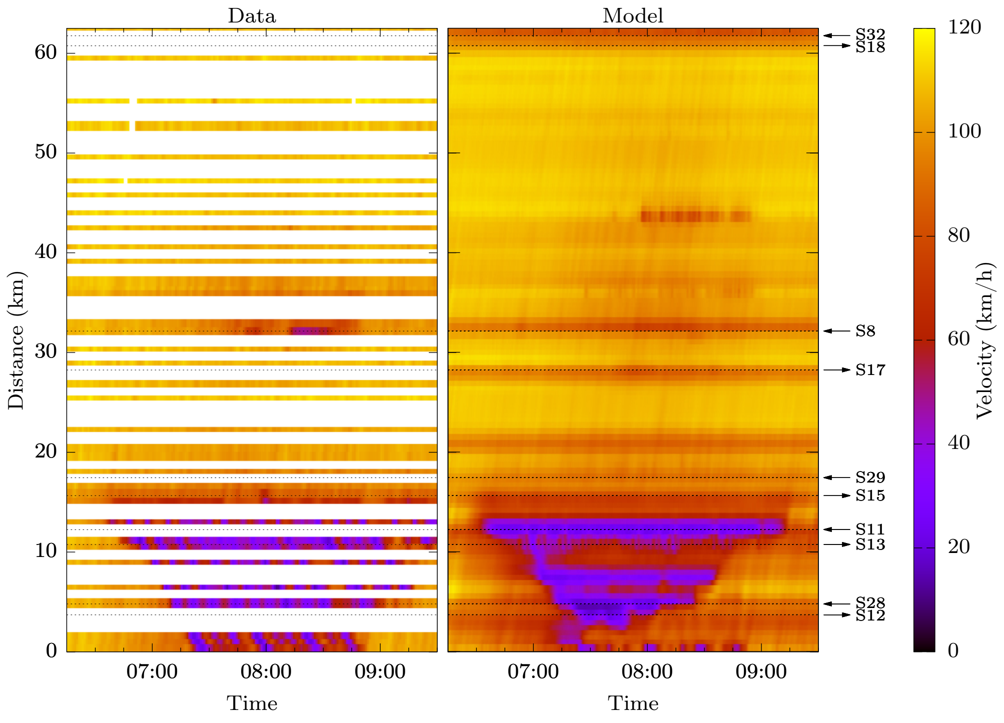
The degradation of a solution when used on verification data can also be viewed in the time-distance diagrams of Figure 10.9. This is a diagram that depicts the spatio temporal evolution of traffic in the two ring-road sub-networks, when $\mathbf{z}^{*,RPROP}_{14th,3}$ is used to run a METANET simulation using the data of the 21st. It can be seen that the main congestion in the anti-clockwise direction is replicated quite well, Figure 10.9(b). The congestion built up in the clockwise direction is underestimated, although a drop of speed is predicted for the correct area of the time-distance diagram, Figure 10.9(a).

Table 10.4: Verification results: J_v $((km/h)^2)$ for each $\mathbf{z}^*$.

$\mathbf{z}^*_{date,run\ nr.}$	14th	21st	28th	avg.
1st best				
$\mathbf{z}^{*,HEPSO}_{14th,3}$	265.65	351.96	273.04	296.88
$\mathbf{z}^{*,HEPSO}_{21st,2}$	273.30	311.65	299.05	294.67
$\mathbf{z}^{*,HEPSO}_{28th,2}$	414.00	456.18	250.52	373.57
$\mathbf{z}^{*,LPSO}_{14th,2}$	164.02	274.43	220.48	219.64
$\mathbf{z}^{*,LPSO}_{21st,3}$	226.32	234.88	276.42	245.87
$\mathbf{z}^{*,LPSO}_{28th,3}$	378.34	478.15	190.80	349.10
$\mathbf{z}^{*,RPROP}_{14th,1}$	108.54	238.84	213.11	186.83
$\mathbf{z}^{*,RPROP}_{21st,2}$	205.33	151.43	371.89	242.88
$\mathbf{z}^{*,RPROP}_{28th,2}$	646.02	901.60	128.19	558.60
2nd best				
$\mathbf{z}^{*,HEPSO}_{14th,2}$	290.57	351.39	324.68	322.21
$\mathbf{z}^{*,HEPSO}_{21st,3}$	294.11	316.53	279.81	296.82
$\mathbf{z}^{*,HEPSO}_{28th,3}$	380.08	504.88	268.81	384.59
$\mathbf{z}^{*,LPSO}_{14th,1}$	168.78	246.86	237.49	217.71
$\mathbf{z}^{*,LPSO}_{21st,2}$	235.51	236.61	308.69	260.27
$\mathbf{z}^{*,LPSO}_{28th,2}$	505.65	584.41	194.18	428.08
$\mathbf{z}^{*,RPROP}_{14th,2}$	111.92	205.45	216.14	177.84
$\mathbf{z}^{*,RPROP}_{21st,1}$	220.28	155.64	319.37	231.76
$\mathbf{z}^{*,RPROP}_{28th,1}$	906.64	979.80	131.39	672.61
3rd best				
$\mathbf{z}^{*,HEPSO}_{14th,1}$	291.79	355.37	303.10	316.76
$\mathbf{z}^{*,HEPSO}_{21st,1}$	294.71	318.10	282.64	298.48
$\mathbf{z}^{*,HEPSO}_{28th,1}$	304.56	366.31	277.67	316.18
$\mathbf{z}^{*,LPSO}_{14th,3}$	175.98	276.71	214.09	222.26
$\mathbf{z}^{*,LPSO}_{21st,1}$	261.29	250.39	321.37	277.68
$\mathbf{z}^{*,LPSO}_{28th,1}$	274.20	343.42	206.80	274.81
$\mathbf{z}^{*,RPROP}_{14th,3}$	115.73	226.38	178.57	**173.56**
$\mathbf{z}^{*,RPROP}_{21st,3}$	2355.91	2290.37	1744.84	2130.37
$\mathbf{z}^{*,RPROP}_{28th,3}$	672.51	831.74	132.22	545.49

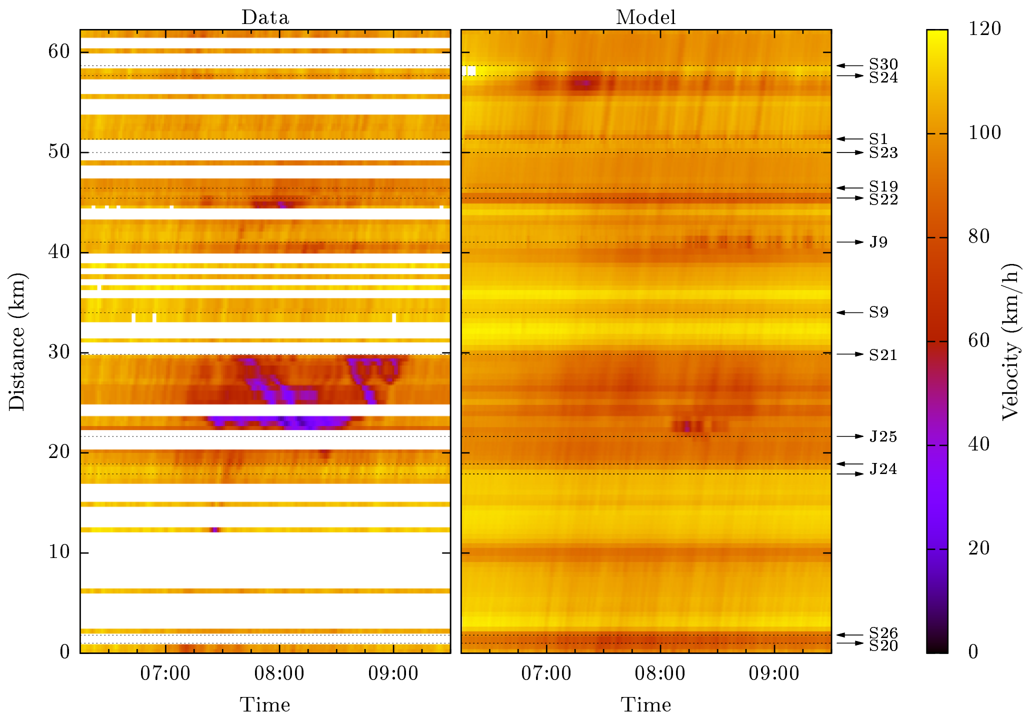


(a) Clockwise direction.

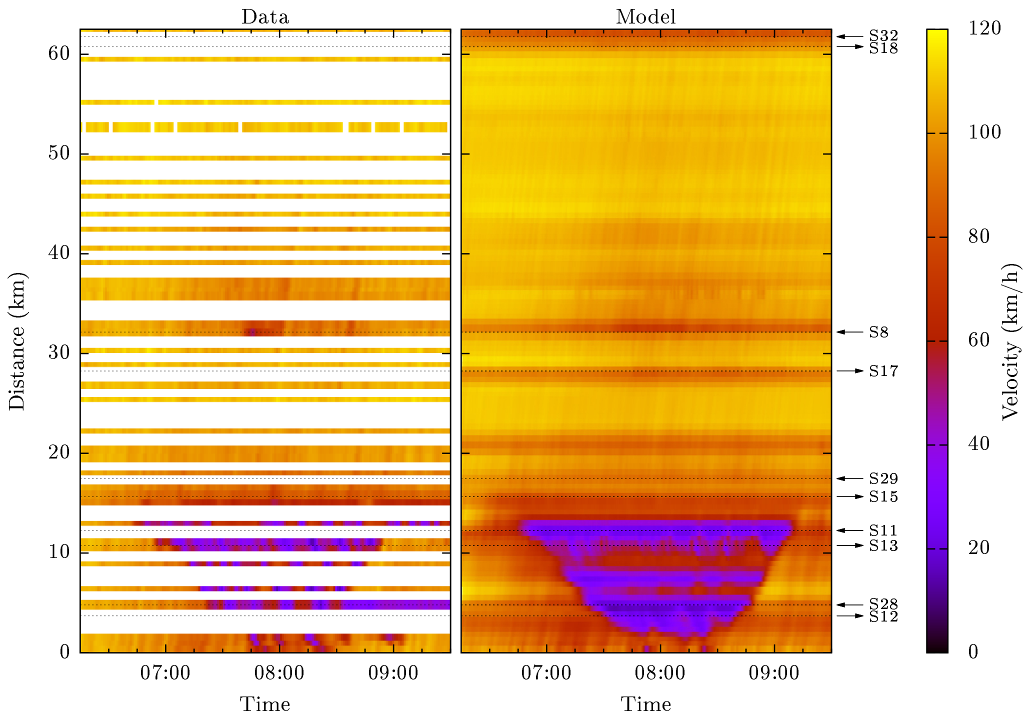


(b) Anti-clockwise direction.

Figure 10.9: Verification of $\mathbf{z}^{*,RPROP}_{14th,3}$: measurement vs METANET speed time-distance diagram for M60 based on the data of the 21st.



(a) Clockwise direction.



(b) Anti-clockwise Direction.

Figure 10.10: Verification of $\mathbf{z}^{*,RPROP}_{14th,3}$: measurement vs METANET speed time-distance diagram for M60 based on data of the 28th.

Figure 10.10 shows the $\mathbf{z}^{*,RPROP}_{14th,3}$ set applied to the data set of the 28th. As the other verification test shows there is a good correlation between the model predictions and the measured data. There is a spill back of congestion from S21 that effects the clockwise carriageway that is not captured by the model. This dynamic appears different to the observed conditions in the other training sets where the traffic slows down upstream of the diverge causing congestion to form a few kilometres away from the junction. The model does have a speed drop at the point that coincides with the central point of congestion observed in the data.

The verification contour plot of the M60 for the best LPSO parameter set is shown in Figure 10.11. This result shares a lot of patterns with the RPROP calibration. Overall the extent of the congestion in time and space is captured correctly, although there is a loss of fidelity. The model captures the oscillatory waves seen at the start of the anti-clockwise direction. Even though the models do not always predict the same amount of slowing in the traffic as observed in the data it never predicts congestion at locations where the data show as free-flowing.

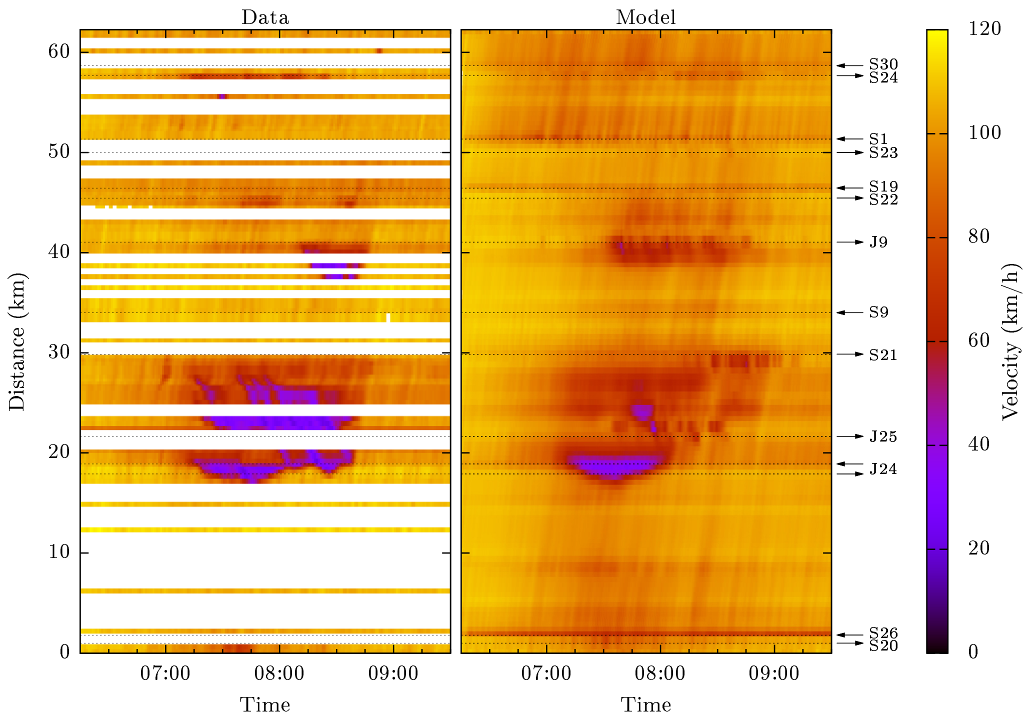
The results presented clearly show the merit and feasibility of performing model validation for large scale highway networks. Instead of testing the validity of a model in a small to medium scale linear network, more complicated topologies are considered. Furthermore, gradient based optimisation combined with a globalisation strategy, even if this is a simple multi-start scheme, is a feasible option for validating such networks, even if the data used as boundary conditions and turning rates are relatively noisy.

10.5 Sensitivity Diagrams

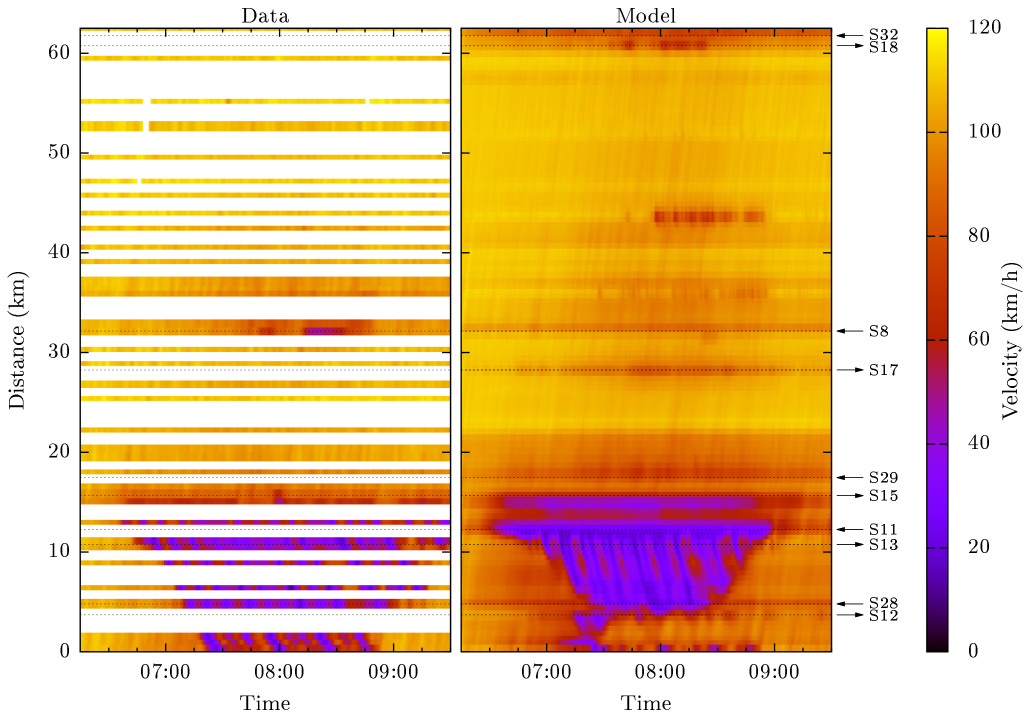
In the process of calculating $\partial J/\partial z_\gamma$, the sensitivity of the speed with respect to every optimised model parameter z_γ, $\partial v_{m,i}(k)/\partial z_\gamma$, is calculated by ADOL-C, [2]. This allows the development of time-distance diagrams of $\partial v_{m,i}(k)/\partial z_\gamma$ for the particular realisation of initial state $\mathbf{x}_0$, boundary conditions and turning rates $\mathbf{d}$ at a point $\mathbf{z}$. This information can be used for investigating the sensitivity of speed, and implicitly of travel times, with respect to the aggregate traffic flow features reflected on the parameters' values.

For example, Figure 10.12 depicts the time-distance diagram of $\partial v_{m,i}(k)/\partial \delta$ and $\partial v_{m,i}(k)/\partial \phi$ for the clockwise and anti-clockwise directions of the two main ring-roads using the data of the 14th at point $\mathbf{z}^{*,RPROP}_{14th,3}$. Both are global model parameters applied to the whole network and sensitivity shock-waves can be seen on both diagrams.

The shock-waves of $\partial v_{m,i}(k)/\partial \delta$ follow the outlines of the congestion shock-waves of Figures 10.6 and 10.5. It is interesting to note that its value is negative at the

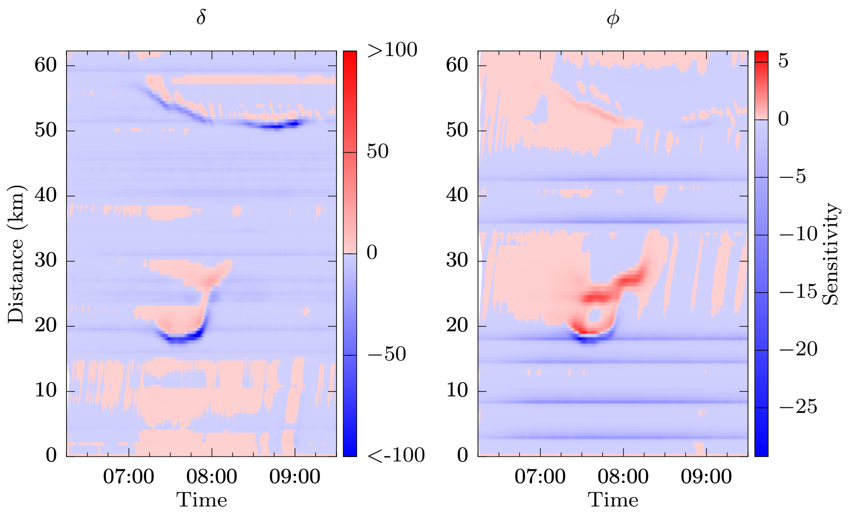


(a) Clockwise direction.

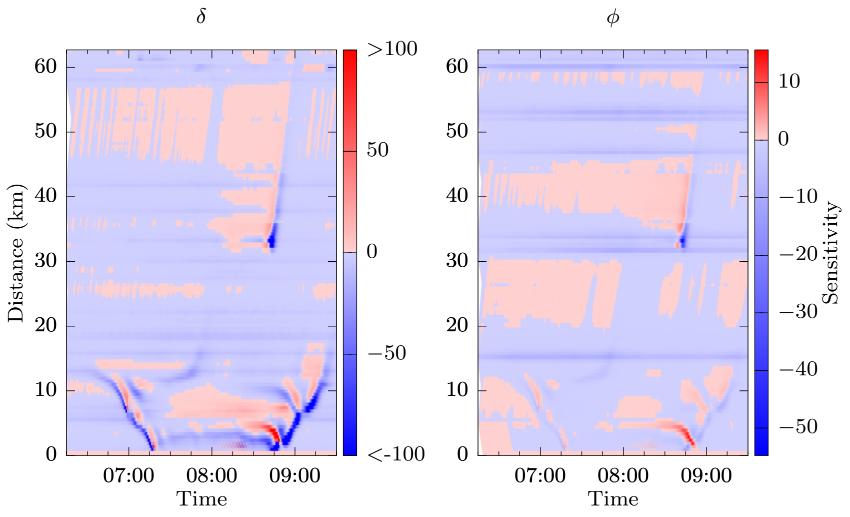


(b) Anti-clockwise direction.

Figure 10.11: Verification of $\mathbf{z}^{*,LPSO}_{14th,1}$: measurement vs METANET speed time-distance diagram for M60 based on data of the 28th.



(a) Clockwise direction.



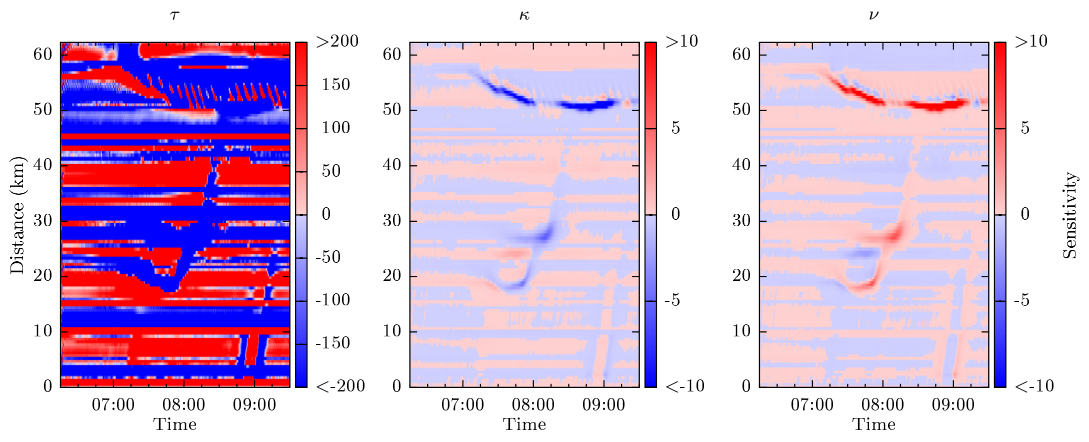
(b) Anti-clockwise direction.

Figure 10.12: Sensitivity of the ring-roads' mean speed with respect to the global model parameters δ and ϕ

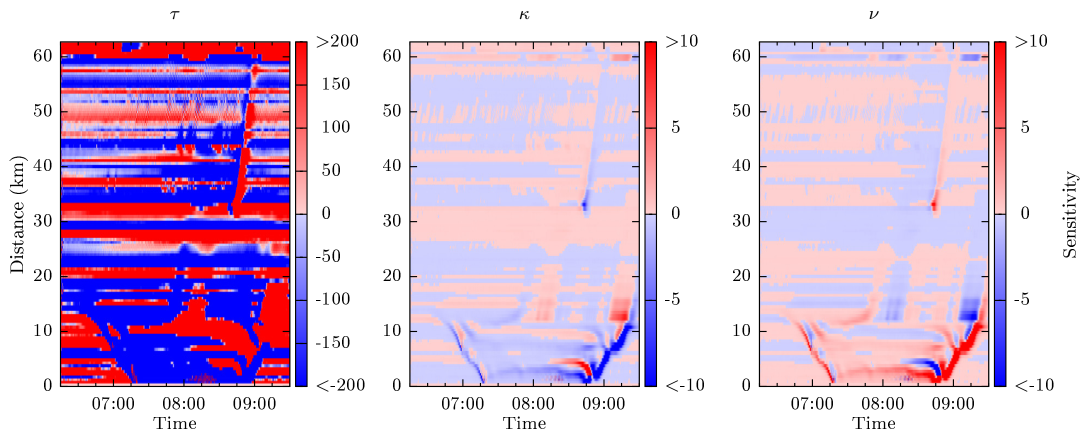
rim of the congestion shock-waves, indicating a decreasing relationship between the mean speed and δ, reflecting the drop of speed due to the on-ramp term included in the speed equation (3.22) (page 29). However, it can be seen at the bottom of Figure 10.12(b) that there is a change of sign in $\partial v_{m,i}(k)/\partial\delta$ inside the core area of the congestion, indicating an increasing relationship between $v_{m,i}(k)$ and δ. A stronger drop of speed due to the merging flows would result to an increase of the downstream speed. This is already reflected on the model output in Figure 10.6, where a relative increase of speed is predicted by the model to take place between the two heavily congested bands shown at the bottom of the diagram. A suitably changing δ could be viewed as a means for describing ramp metering control actions directly influencing the speed of merging vehicles, in view of the capabilities of vehicle-to-vehicle and vehicle-to-infrastructure communication networks. The effect of a speed regulator performing ramp metering, [131], [132], for the merging vehicles could be modelled by the speed equation.

The shock-waves of $\partial v_{m,i}(k)/\partial\phi$ shown in Figure 10.12, are more pronounced at the locations where there the drops in the number of lanes. Stationary shock-waves are formed indicating a more localised action. The interesting feature to note here is the positive sign of $\partial v_{m,i}(k)/\partial\phi$ at the region corresponding to the congestion formed from km 18 to km 30 in the clockwise direction, Figure 10.5. This area is where junction 24 off- and on-ramps are connected to the M67 to the East. The strong negative value following the rim of the congestion models the impact of a drop of lanes from four lanes before to three lanes after the off-ramp. However, there is a strong congestion shock-wave formed further downstream at junction 25, indicating that a further constriction of flow upstream of junction 24's off-ramp would have a positive impact at the mean speed directly downstream the off-ramp and upstream of junction 24's on-ramp. This indicates the potential for speed regulation, e.g. via variable speed limits, upstream of junction 24.

The sensitivities for the network parameters τ, κ and ν are shown in Figure 10.13. The sensitivity of the parameters κ and ν are very similar but inverse of each other. This means that where an increase in κ would cause a corresponding increase to the mean speed a decrease in ν would also achieve the same effect. The most obvious shocks seen in these plots surround the congested region, with a shock seen at congestion onset and a second at recovery. During the congestion itself the values appear to be consistent. The sensitivity profile for τ is unique and shows banding that corresponds to the changes in the FD parameters. Disturbances can propagate through these bands causing a change in the intensity and sign observed in the free flowing conditions. This is due to the fact that the τ parameter not only controls the relaxation term but it is also present in the anticipation term (3.22). There are



(a) Clockwise direction.



(b) Anti-clockwise direction.

Figure 10.13: Sensitivity of the ring-roads' mean speed with respect to the global model parameters τ, κ and ν

more dynamics apparent in these sensitivity diagrams than can be observed in the corresponding mean speed plots Figures 10.5 and 10.6. In the clockwise direction these network parameters show a shock-wave around 60 km near the roundabout junction whereas in the model predicts free flowing conditions with no sign of the structures visible in the sensitivity plot.

The network's mean speed is also sensitive to the FD parameters α_m, $v_{f,m}$ and $\rho_{cr,m}$ of every link m. A sample of the sensitivities with respect to two links' parameters are shown here. These are the L28C, in the clockwise direction, and the L24AC at the anti-clockwise. Both are located in the North and are shown in the zoomed map of the METANET model in Figure 10.14. This is the area where congestion is observed between km 50 and 60 on Figure 10.5.

Figure 10.15(a) depicts the sensitivity diagrams of the mean speed, along the clockwise direction ring-road, with respect to the fundamental diagram parameters

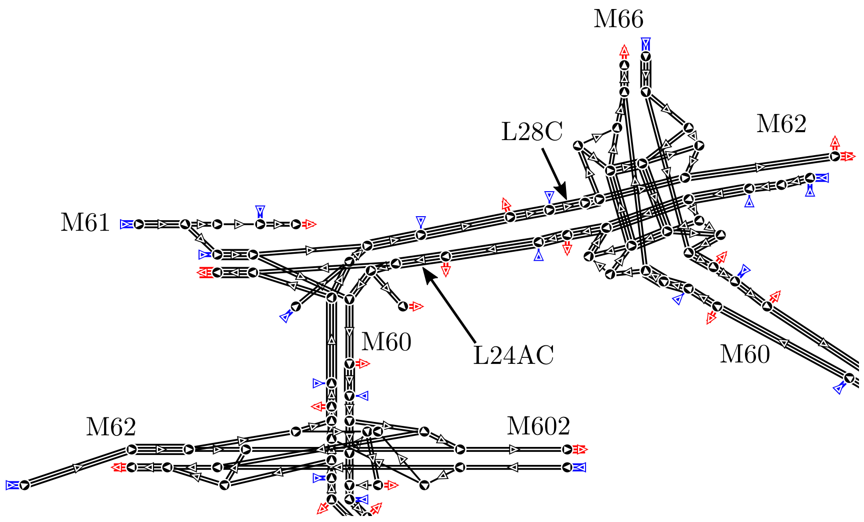
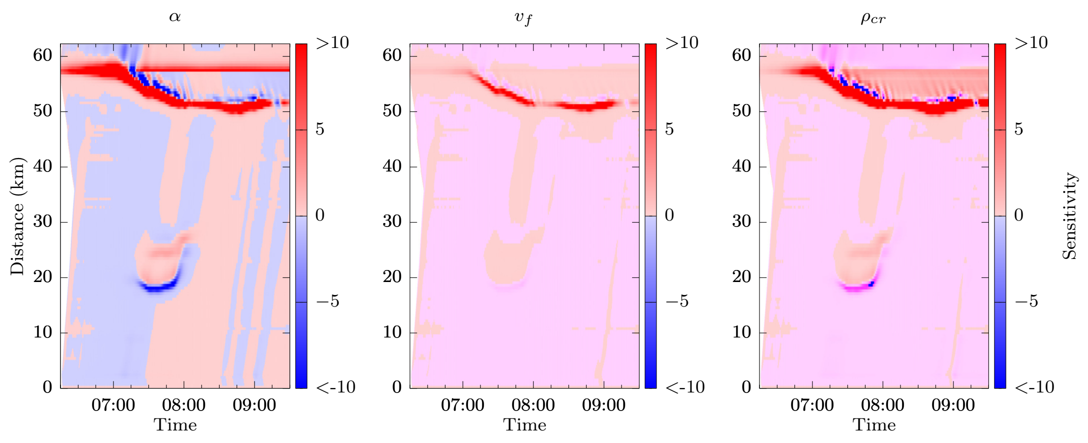


Figure 10.14: Zoomed model graph of the Northern section of the network.

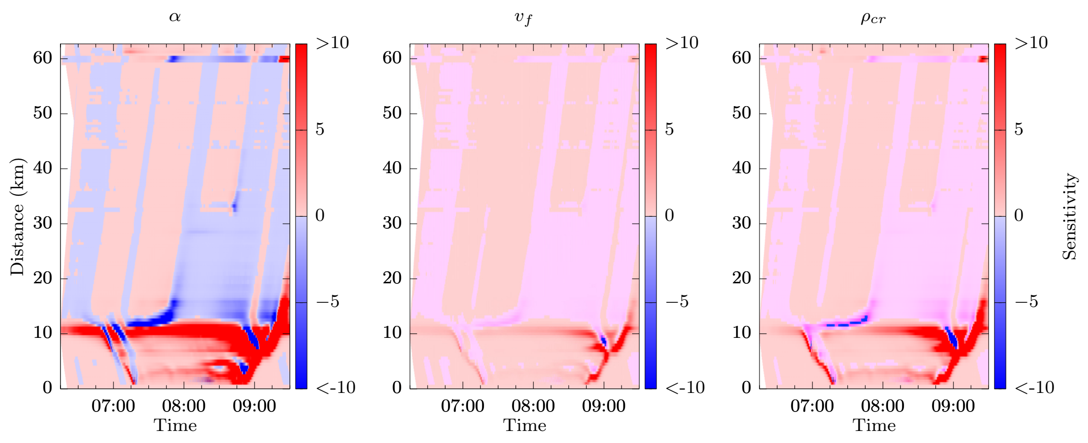
of L28C. It is clearly shown that an increase of the critical density at this link would have positive effects on the mean speed for the network section between the junctions M60 with M61 and M62. A second point to note is that for all parameters' sensitivities there are two expanding fronts, one forwards and one backwards, of $\partial v_{m,i}(k)/\partial z_\gamma$ with the vacuum state, emanating from the start and end of L28C. Both outline the boundary between empty space (no influence) and non-zero sensitivity values, i.e. areas in the time-distance diagram where the parameters influence the mean speed. The first front starts from the end of L28C, propagates forwards, crosses over the top of Figure 10.15(a) and continues from the bottom down with an upwards direction; this kind of propagation takes place because of the ring-road topology. The second front starts from the start of L28C and moves backwards until it meets the first one.

Figure 10.15(a) shows the same information as Figure 10.15(b), but for the anti-clockwise direction of M60 and the sensitivities are calculated with respect to the fundamental diagram parameters of L24AC. This link is also placed at the North, between the junctions of M60 with M61 and M66, Figure 10.14. L24AC splits into two subsidiaries, one sending flow to M61 towards the West and the other continues towards the South in M60. L24AC belongs to the congested area shown between km 16 and km 0 on Figure 10.6. An increase at the link capacity would have a positive influence at the network's mean speed. The two vacuum fronts, with the forwards moving overtaking the backwards moving one, are shown in this diagram as well.

As well as looking at the sensitivity of a links parameters at and around its point of application its effects on the alternate carriageway can also be investigated. Figure 10.16 shows the sensitivity of the parameters applied at L24C on the clockwise



(a) Clockwise ring-road mean speed with respect to highway link L28C



(b) Anti-clockwise ring-road mean speed with respect to highway link L24AC

Figure 10.15: (a) Sensitivity diagrams of link parameters.

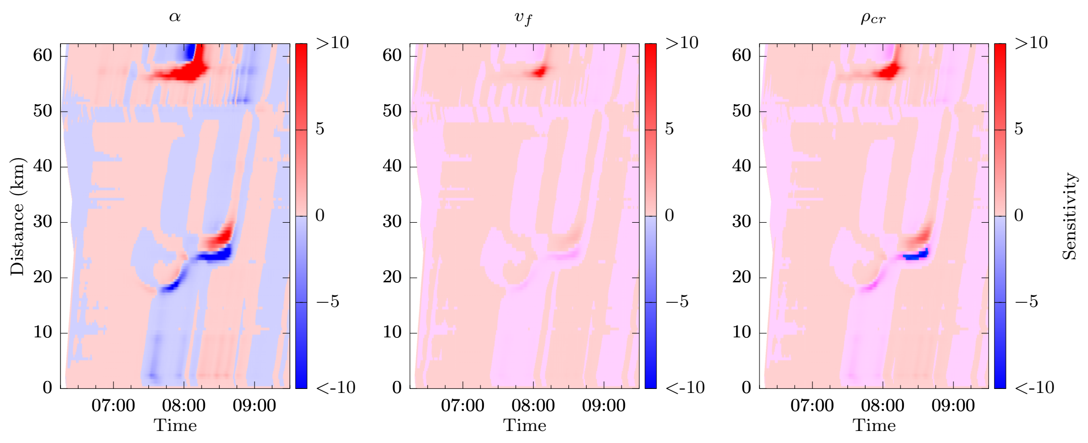


Figure 10.16: L24AC sensitivity on alternate carriageway (clockwise).

sub-network. The overall shape of the patterns seen for the sensitivity show a rough match to those seen for L28 a link on the clockwise sub-network but there are differences. The size of the sensitivity is much smaller but it is interesting to note that the vacuum state is very small still and instead of a very clear forwards and backwards waves a sawtooth like wave occurs. This is due to the propagation of information in the numerical model. As discussed in the previous chapter; it takes one time step for the sensitivity to propagate forwards a single segment and two time steps to propagate backwards. This propagation happens along the sub-network as normal (forwards or backwards) until it reaches a part of the network that joins the two sub-networks. At which point it can propagate along this path to reach the other sub-network, once a sensitivity exists at a point on the other sub-network it propagates as a sensitivity created there would, i.e. every time step it will move downstream a segment while two time steps are required for it to propagate upstream.

To explain the sawtooth vacuum phenomena further consider the extract shown in Figure 10.14. If we consider the forwards propagation of the sensitivity from L24. It immediately encounters a node, and the sensitivity propagates towards the M61 boundary. When this sensitivity propagates past the next node (a merge with traffic originating from the clockwise sub-network) it can then propagate backwards onto the clockwise sub-network. Once there, it can propagate forwards past L28 and also backwards. This makes one of the peaks of the sawtooth. The sensitivity can also cross over from the M62/M602 junction, for it to do so it only requires upstream propagation for one link. While for the sensitivity that was first tracked onto the clockwise link to propagate backwards to the M62/602 requires the traversal of 3 links. This therefore makes a second peak and the forward propagation from from the M62/602 junction will interact from the backwards propagation from the M61 junction, the location at which they meet is the trough of the sawtooth. With downstream propagation being double the speed the sensitivity going further downstream on the anti-clockwise loop will find a path to cross over to the clockwise network faster (geometry depending) than the backwards propagation of any previous sensitivity that crossed over.

10.6 Model Sensitivity

As well as looking at the model Jacobian each parameters sensitivity and the gradient of the objective function are also obtainable via AD. For the parameter set $\mathbf{z}^{*,RPROP}_{14th,3}$ the partial derivatives of the global parameters, and derivatives of the anti-clockwise link parameters that experience the main block of congestion are given in Table 10.5. In the table the dot in the column header should be replaced by the parameter of the row, column $\partial J/\partial.$ for row τ shows the value $\partial J/\partial\tau$. This table also breaks down the partial derivative into the gradient that comes from the squared errors only J_v and also the penalty term J_p. As shown in the table the partial derivatives of the global parameters $\partial J/\partial\tau$, $\partial J/\partial\kappa$, $\partial J/\partial\nu$, $\partial J/\partial\delta$ and $\partial J/\partial\phi$ only have contributions coming from the squared error terms, this is because these parameters are not included in the penalty term. The values for the solution obtained are close to, but not exactly, zero. The links shown in the table have some of the highest gradient values, the majority of links near zero values.

By taking the optimal solution obtained $\mathbf{z}^{*,RPROP}_{14th,3}$ and perturbing individual parameters one at a time and plotting the partial derivative the models sensitivity can be analysed. In [26] a study is made about parameter sensitivity. This is done by plotting the relative change in the objective function value versus the relative change in parameter value. This provides an approximation to the partial derivatives which can now be obtained due to the AD-METANET system. The effect of altering parameters individually on their partial derivative is shown in Figures 10.17 and 10.18. For the global variables although partial derivatives of zero were not obtained Figure 10.17 shows that obtained parameters are indeed at a minimum.

The sensitivity of the τ parameter goes through a rapid change close to the identified value τ^*. The magnitude of the sensitivity with respect to the parameters τ, κ and ν are similar for this solution. Although the most sensitive parameter of the three is τ, then ν with κ the least. Both ν and κ influence the anticipation term of the model, this creates an interesting set of dynamics. It was already noted in the sensitivity diagrams that the effects of these parameters were inverted with each other. This is further supported by Figure 10.17, for values below the optimum $\partial J/\partial\kappa$ has low values while for values above the optimal point $\partial J/\partial\nu$ has a low positive value. The shape of the curves obtained for each plot even follows a similar trend but rotated by 180° around κ^* or ν^* for the κ and ν plots respectively. This trend breaks down at values far away from the minimum obtained. For this solution δ and ϕ have a minimum at 0 and are more sensitive than the other parameters even though they are only applied at specific locations. A hypothesis for this is that the optimisation can incorporate the effects of merging and lane drops into the fundamental diagram, as such it can tune the values obtained for each junction independently rather than applying a global constant.

Figure 10.18 shows the sensitivities for two highway links. As a general trend the α parameter appears to be the least sensitive, and the sensitivity of the free speed v_f appears to be slightly more than the network parameters. For L28C the v_f plot actually crosses the zero 3 times, 2 minimums and a maximum. The additional points are not of interest as one parameter is adjusted the partial derivatives of all

Table 10.5: Parital derivates for $\mathbf{z}^{*,RPROP}_{14th,3}$, with respect to the parameter denoted in the left hand column. Global parameters and selection of links from congested region of anti-clockwise M60.

Parameter		$\partial J/\partial.$	$\partial J_v/\partial.$	$\partial J_p/\partial.$
	τ	-1.2395	-1.2395	0
	κ	-0.1387	-0.1387	0
	ν	0.1235	0.1235	0
	δ	0.5355	0.5355	0
	ϕ	1.2945	1.2945	0
	v_f	-0.0480	-0.0680	0.0200
L19AC	α	-0.0064	0.0003	-0.0067
	ρ_{cr}	-0.0002	-0.0109	0.0108
	v_f	0.5726	0.5773	-0.0047
L20AC	α	0.0147	0.0376	-0.0229
	ρ_{cr}	0.0419	0.0407	0.0013
	v_f	2.7096	2.6489	0.0608
L21AC	α	0.2234	0.2869	-0.0634
	ρ_{cr}	2.2843	2.3226	-0.0383
	v_f	-8.5853	-8.5305	-0.0548
L22AC	α	-0.6168	-0.7023	0.0855
	ρ_{cr}	-1.7659	-1.8076	0.0416
	v_f	-3.9040	-3.8779	-0.0261
L23AC	α	-0.2946	-0.3069	0.0124
	ρ_{cr}	-0.8417	-0.8353	-0.0064
	v_f	-8.9432	-8.9806	0.0373
L24AC	α	-0.6307	-0.6527	0.0220
	ρ_{cr}	-2.9423	-2.9365	-0.0058
	v_f	-1.3778	-1.3765	-0.0013
L25AC	α	-0.1115	-0.1370	0.0255
	ρ_{cr}	-0.4993	-0.4809	-0.0184
	v_f	-1.4332	-1.4528	0.0196
L26AC	α	-0.0657	-0.0281	-0.0376
	ρ_{cr}	-0.1085	-0.1553	0.0468
	v_f	-0.2530	-0.3259	0.0728
L27AC	α	-0.0958	-0.0761	-0.0196
	ρ_{cr}	-1.0876	-1.0431	-0.0446
	v_f	0.0114	0.0098	0.0015
L28AC	α	0.0034	-0.0067	0.0101
	ρ_{cr}	0.0006	0.0006	0.0000
	v_f	0.0124	0.0101	0.0023
L29AC	α	0.0037	-0.0041	0.0079
	ρ_{cr}	0.0005	0.0006	-0.0001

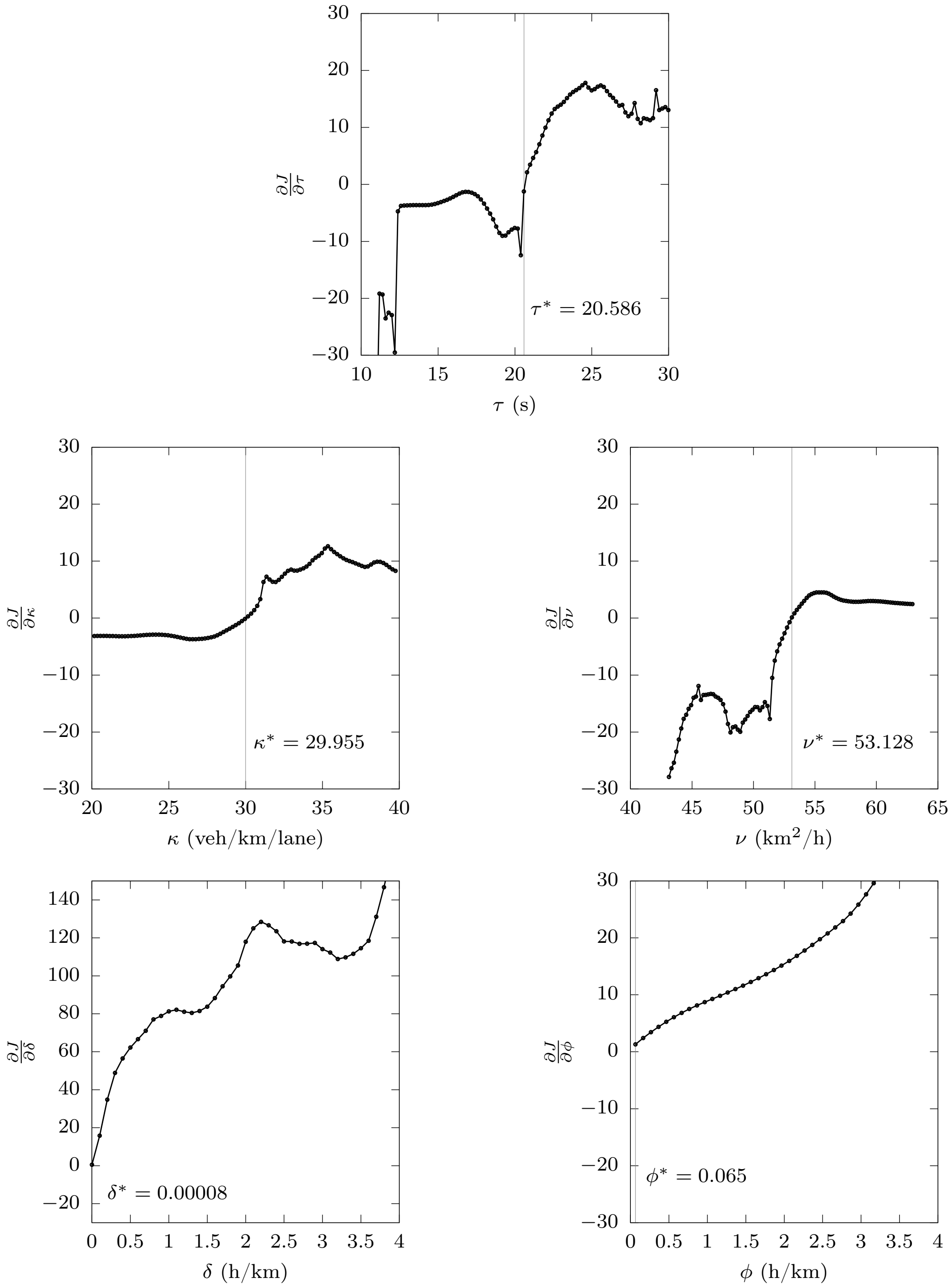
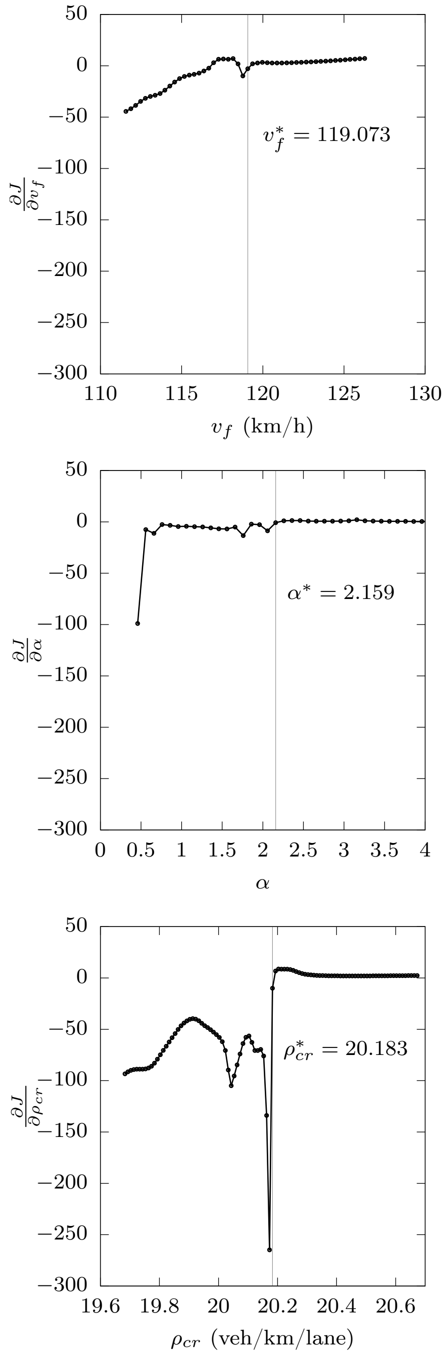
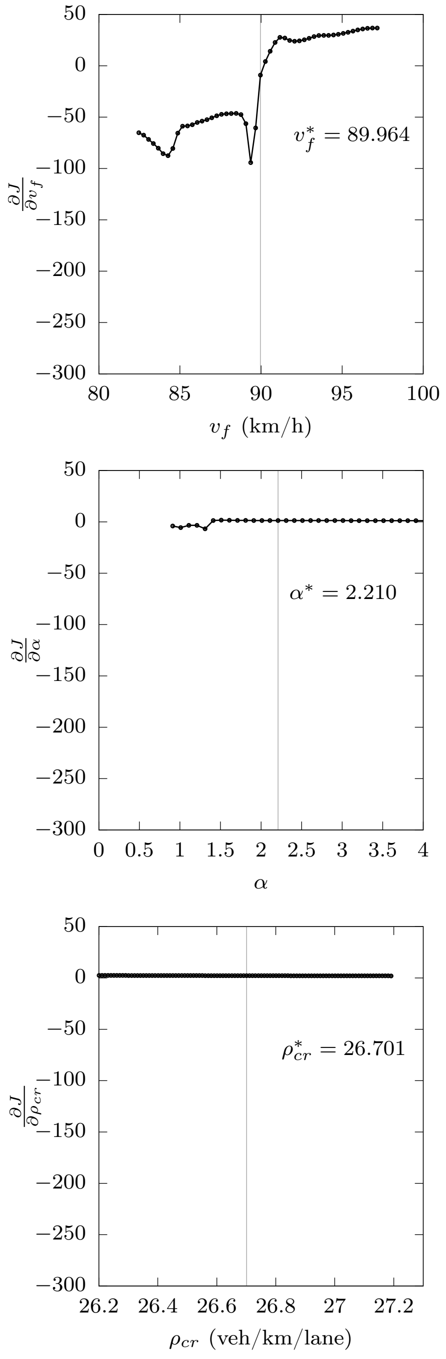


Figure 10.17: Sensitivity of global parameters for $\mathbf{z}^{*,RPROP}_{14th,3}$. Note scale change on $\frac{\partial J}{\partial \delta}$ plot.



(a) L28C on the clockwise M60.



(b) L24AC on the anti-clockwise M60.

Figure 10.18: Sensitivity of sample link parameters for $\mathbf{z}_{14th,3}^{*,RPROP}$.

the other points also change, the whole system needs to be considered. In [26] the sensitivity was investigated for a site at Paris it was concluded that the ρ_{cr} parameter is the most sensitive. Here it can be seen that for the link L28C that is true. A very abrupt and large change in sensitivity is seen near minimum, however for L24AC the sensitivity constant and small. In Paris only a single FD was applied so that makes it easier to identify the sensitivity of a parameter, here a much larger model with many more parameters is investigated, some links will therefore be less sensitive than others as can be seen here. The parameter values for a link upstream of a bottle neck will have little impact on the solution if the dynamics of the congestion spilling back take precedence. Thereby making the bottleneck's parameters determining the upstream state.

Consider the models speed equation (3.22), in free flow conditions the mean speed and density is relatively homogeneous meaning that the convection and anticipation term have a small influence on the speed. The relaxation term is important under these conditions with speed tending towards the value determined by the equilibrium speed function, and as densities are low this is approximately equal to the free speed v_f alterations to the other link parameters, α and ρ_{cr} have a small influence on this while the density is low. When congestion occurs the density rises and speed drops meaning that the influence of α and ρ_{cr} increases, and the importance of v_f reduces. So when a link is being affected by a downstream bottleneck is when v_f has the least impact on the model. As the model here always exhibits free flowing conditions at some time, which is when the free speed parameter has its greatest influence, v_f has a similar sensitivity profile and is not affected by the bottleneck. When a bottleneck present downstream is active the large difference in the upstream and downstream densities causes the anticipation term to reduce the speed of the upstream link, this in turn increases the density at the upstream location. This causes the link to quickly transition from free-flow to congested conditions but not because of the relaxation term. As the link is under capacity and being limited by the downstream conditions the relaxation term will result in value that aims to accelerate the flow. The high density tails of the fundamental diagrams often have similar speeds so the amount of acceleration is small. This does not overcome the effects of the anticipation and leads to the small positive sensitivity as seen in the figure. The sensitivity is constant because the parameters are not reaching a point where the flow is being decelerated due to the relaxation term.

10.7 Conclusions

This chapter has presented the validation (calibration and verification) of the macroscopic traffic flow modelling tool METANET for the large scale highway network orbiting the city of Manchester, UK. A total of 186 km of roads have been considered into a single METANET model and the calibration problem refers to the entire site. A single optimisation problem is formed and solved for identifying the speed equation and FD parameters used by METANET. The system developed from the combination of METANET with PSO algorithms from one side and RPROP and ADOL-C from the other, is able to solve efficiently this problem, yielding the spatial capacity profiles for the network and other important information regarding the traffic flow process. The optimal parameter sets obtained result to a METANET model that reflects accurately the traffic dynamics. These solutions have also good generalisation properties allowing METANET to capture the network wide dynamics of congestion propagation.

This calibration of this large network this was completed in around 3 hours and 30 using PSO algorithms with a problem dimension of 684 on an Intel Core i5-2400 @ 3.10 GHz processor with four cores. To use a gradient based algorithm RPROP the AAFD process had to be simplified so the problem dimensions reduced to 583 but even so due to the complexity of the gradient calculation a considerable increase in run time occurs. RPROP was run for fewer function evaluations and took around 9 hours 30 minutes to calibrate.

The work detailed here shows that large scale network validation can be performed by use of a gradient based algorithm. The development of a system to obtain the required gradient information is a novel contribution and allows for a look at the sensitivity profiles for individual parameters across space and provides information that has not been analysed previously. This chapter also provided the first reported calibration of a network as an entire entity without performing any ad-hoc subdivisions prior to the optimisation process.

Chapter 11

Conclusion

11.1 Summary

This thesis has presented a method that allows for the calibration of networks of varying scales without the need for manual ad-hoc sub-sectioning of the model. This enables the calibration to be performed on a network level with the optimisation algorithm determining the number of Fundamental Diagrams to be used when combined with a meta-heuristic algorithm through the developed Automatic Assignment of Fundamental Diagrams (AAFD) process. The contribution of the work presented here is the extraction of the model Jacobian through the use of Automatic Differentiation (AD). This allowed a gradient based solver to be applied to the problem. Due to the combinatorial nature of the AAFD routine this had to be simplified to a minimisation of FD parameter variation along the sub-networks automatically identified within the model. The use of the AD and the acquired Jacobian also allowed for an analysis of model properties that had not been previously considered in this level of detail, such as the spatial extent of link parameter sensitivities and the overall model sensitivity values and profiles.

Chapters 2 and 3 provided the background and literature review for this thesis and the models being used. In Chapter 4 the AFFD process was introduced, including a methodology to split the large networks into a series of linear sub-networks in an automatic manner. The main calibration metric is the squared-error between the model output and measured data as used in previous works. The AAFD process works by including a penalty function that minimises the changes in the FD parameter values between adjoining links, even if the links are not in the same sub-network. For AAFD to work the parameter vector is extended to include the number of links for which an FD is applied. The decomposition of this extended parameter vector to one typically required by the simulators, that is one FD per link, is also developed.

The optimisation algorithms are detailed in Chapter 5. No alterations are made to the meta-heuristic algorithms and the parameters used are as reported in the literature. The chapter contributed a method to determine the model Jacobian using AD and the subsequent formulation to calculate the gradient as required for the gradient based algorithm. The Jacobian obtained using this method is also analysed and the size and density of the matrix can be monitored, where it very quickly becomes a very dense matrix with few non-zero entries. This means that various techniques for matrix reduction cannot be applied. The formulation of RPROP, the gradient based solver used in this thesis, was also given.

Chapter 6 described the methodologies used to create the required input. The comparison and boundary data from the provided MIDAS source was also considered. This included the transformation of time-mean speed data captured by the loop detectors to the space-mean speed utilised in the models. Methods were also developed to ensure the integrity of the data used and that no vehicles were gained or lost. The three sites used in the thesis were also detailed, the shorter sites of Heathrow and Sheffield are simple stretches of highway with merges and diverges being from origins and to destinations respectively. The Manchester site is a large network 186.2 km in length. The sub-networking algorithm developed as part of Chapter 4 split this into 39 linear sections, with the two major loops being a little over 62 km each.

In Chapter 7 the software used in the thesis was discussed and the various methods of communication between sub-processes was optimised. The work included extending the software to work in a parallel implementation to fully utilise the computing resource available. It was found that the fastest method of communication between processes was the use of shared memory, this also had the additional benefit of reducing the overall memory footprint between the model and optimisation algorithm system. A simple file-based system use of a shared memory communication system lead to a reduction in runtime of up to 95% versus a simple solution involving saving and loading of data files. The investigation was done for each of the three sites. When considering the model that has not been expanded to include gradients, the reduction in runtime for an objective function evaluation was independent of the model size. However when the gradient is also computed the average objective function evaluation time did become dependant on the model size. This is due to the nature of the Jacobian. As the model length increased the Jacobian gains rows and columns meaning it scales as a power law and the data that is required to be transferred grow significantly with the model size. It was also seen that the Jacobian calculation and transfer had a significant impact on the runtime. If the data about the model states is transferred via a pipe and the Jacobian uses shared memory then the speed up for the Manchester site versus the file IO is 94.3%. Switching the state to use shared memory as well only increased this by 0.3%. When the change from pipe to shared memory is done for a model without the Jacobian the corresponding runtime reduction is 40%. The calculation of the Jacobian is a very intense operation. The average run time of an objective function calculation for Manchester using the best IPC method is 0.3s the Jacobian calculation increases this to 31.7s. Also investigated in this chapter was a method of scheduling the operations across multiple processors, the obtained speed up rates when using the fastest IPC methods was close to the optimal value expected. The software developed in this thesis included a graphical interface to assist in the comparison of models and data sets. Finally the chapter also considered how the systems developed may be integrated in the wider framework of autonomic systems.

In Chapter 8 two different models were considered, the CTM and METANET. CTM is a first-order macroscopic model and METANET second-order. Both of these algorithms were used on the shorter sites (Heathrow 7.8 km and Sheffield 21.9 km) with ten different meta-heuristic algorithms. Through the developed AAFD method both sites gave solutions with the number of FDs being less than the maximum allowed. Furthermore, the spatial extent and domain over which FD was applied had a physical meaning to the real-world network. The ten algorithms considered belonged to three classes, GA, CS and PSO. It was clear that the PSO class of algorithms provide the best results and had faster convergence properties. Of the seven considered variants of the PSO algorithm the best results across both sites, and models, was with the LPSO and HEPSO variants. This was an interesting result as LPSO is a basic variant of the algorithm and is not much developed beyond the original proposed PSO algorithm. HEPSO incorporates elements from the ABC and GA algorithms. It appears that this makes the convergence of HEPSO slower than LPSO but often it achieves a better final result. When the best obtained parameter sets were evaluated in verification it was the results obtained by the HEPSO algorithm that provided the best generality.

The two different models performed differently on the sites tested. For the shorter site at Heathrow it was the METANET model that provided the best results while for Sheffield the best results were obtained by the CTM. In calibration each model produced similar results and accurately described the conditions. The differences occurred in the generalisation property. With the CTM model both sites had a consistent set of FDs found though the AAFD process, with each data set yielding results with similar spatial extensions and parameter values. However the FD for the destinations that control the boundary of the model did not show the same consistency for the Heathrow site. It was concluded that this is the reason why the obtained METANET model is better at this site. For the Sheffield site the boundary conditions applied appear to work better in the CTM with the congestion resulting from spill back being accurately captured. The METANET model showed some degradation of performance with this and either under- or over-predicted the intensity of the spill back.

The gradient method was analysed in chapter 9 for the Sheffield site and this was compared to the best performing PSOs of the previous chapter for the METANET model. An initial study was undertaken to set up the RPROP restart method. To help make the search global rather than local, the step length at restarts would be randomly reset to its initial value rather than contracting and fine tuning the search as in the original algorithm. Due to the errors seen in chapter 8 the model was extended to include explicit modelling of on- and off-ramp. Although the restrictions of AAFD were reduced when applied to a model that was optimised by gradients, the parameter profile obtained over space still had a physical relevance and matched the results from the PSO algorithms. Due to the lack of distinct boundaries the profile obtained by RPROP typically had smoother changes between links. The RPROP algorithm had more information available to it and it showed a faster convergence rate, although the additional computation requirement of a single evaluation (shown in Chapter 7) has to be considered. The RPROP solution is the one that has the best performance and generalisation property. This shows the viability of using such a methodology for traffic model validation. The extension of the model to include the congested off-ramp had the required impact and the generality of the obtained solutions was an improvement over the results obtained in chapter 8. For one data set the model still did not generalise well, and on inspection it was shown that the spill back dynamic was different on this date. By including a very short incident on the off-ramp, removing the capacity of one lane, resulted in the desired model response.

The use of AD meant that more insights could be gained from the model such as the sensitivities. The sensitivity diagrams of the various parameters were produced and these diagrams showed additional shocks and structures that are not visible in the speed graphs. A study was also done into the propagation of the FD parameters sensitivities through the METANET model. Sensitivities of these link parameters propagate downstream one segment per time-step and upstream every two. This comes from the nature of the discrete model's speed equation and its coupling with the density equation.

Chapter 10 focused on applying the developed techniques to a large network. Without any external alterations or adjustments a valid model was obtained using both the PSOs and RPROP. This showed that the gradient based approach was scalable and still works when applied to large networks. Unlike the previous chapters, it was apparent that the HEPSO algorithm was unable to find the same quality of solutions as LPSO and RPROP. At some locations spill back does not result in the same level of congestion as seen in the data for the validation data sets.

Sensitivity diagrams were analysed and again additional shock waves are visible in these plots that are not apparent by looking at the speed. The κ and ν parameters are closely related due to them being applied to the same term of the speed equation. Also investigated in this chapter were the sensitivity values and how they change with alterations to the individual model parameters. This shows that the found solution was indeed a minimum as they are all zero. The sensitivity profile obtained corroborated earlier findings reported in the literature, but also showed that bottlenecks can cause blocking that make a link's parameters become insensitive. A blocked link still has a sensitivity with its free speed parameter as this is important in free flow conditions.

11.2 Future Work

More analysis and detailed examination needs to be undertaken into the newly acquired data that is available through the model's Jacobian. The application of this data to traffic control systems could be of interest. It would also be interesting to apply AD to other models and obtain their Jacobians and gradients, as this thesis only considered the METANET model. The main consideration would be how first-order CTM and second-order models that have an anisotropic behaviour (GKT, ARZ models for example) propagate their sensitivities, and whether these structures and shock waves in sensitivity diagrams differ from those presented here. The sensitivity information and its potential uses in control systems is an area that needs to be investigated.

The parameter values for the meta-heuristic algorithms used in this thesis were the same as the papers in which they were proposed or used for an engineering problem. A detailed study into selection of these parameters and their effect on the convergence profiles and the final solutions obtained could also be an area of further work. This may lead to a reason why the HEPSO algorithm could not provide the same performance on the large Manchester network site as it did on the smaller linear sites. The field of hyper heuristics could be applied, adding an additional layer on top of the optimisation algorithm to appropriately select and maintain the optimisation algorithm's parameters.

Another area for consideration is the handling of destinations and nodes in the METANET model. The discrepancies seen in the generalisation of the METANET model occur around congested off-ramps. Further investigation into this is required. Whilst, the current model does provide a valid model there is room for improvement.

In this work the destinations could receive an unlimited flow, something that is not true for the CTM model that could cope with the congested off-ramp at the Sheffield site.

Increasingly with the prevalence of GPS enabled devices there is call for a approach to deal with Big Data in traffic management. It is possible to use the large volumes of data to construct statistical based models. This avoids the issue of model calibration as addressed in this thesis and the real-world system is no longer simulated according to some physical rules. This can be an attractive philosophy but its use is limited. As a statistical model does not have some underlying physical equations it is dangerous to extrapolate and run simulations with conditions outside of the training data. This could lead to some strange phenomena as the model has to represent a complex non-linear system. It is difficult to predict what behaviour will occur when extrapolating. The same is true for models as discussed in this thesis but to a much lesser extent. As the models presented here are discrete forms of PDEs the behaviour predicted will follow the underlying dynamics. This makes the model much more robust. Suppose a lane is closed that has not been seen in any training data set, it would be difficult for a statistical model to perform well. Physical models however will have their capacities reduced for the affected region and a reasonable prediction of the conditions could be obtained.

Instead of using a statical approach work needs to be done to combine all of these additional data sources to support and validate the traditionally used loop detector data. For the Manchester network in this thesis some assumptions had to be made due to faulty loop detectors or lack of sensing technology. With additional data sources real values for these sections could be found and further checking of the quality of the existing data could be undertaken. Additional data sources could be from GPS enabled devices, connected and autonomous vehicles. It will be interesting to see the effect autonomous vehicles have on these models, the reaction time and response of an autonomous vehicle is well defined and should be consistent across manufacturers. However there probably will be differences between manufacturers and there is still a reaction time. This means that traffic will still exhibit the currently observed shock waves, although the critical density will probably rise. The ratio of autonomous to human-controlled vehicles will also be an interesting factor. Will introducing a small number of autonomous vehicles help stabilise the current system? If so, what adoption percentage is required? Or will a small population of human-controlled vehicles in a predominantly autonomous system cause chaos? All of these questions could be researched in detail through a traffic model that incorporates vehicle classes.

The methodologies presented here are aimed at macroscopic models however due to the black-box approach used there is no barrier to applying the methodologies to urban models (micro and meso-scopic). In particular the AD routines and use of gradients could easily be transferred across but this could result in a very slow model. Microscopic models typically have more parameters and the tracing of all the individual vehicles could mean the total computational requirement is prohibitive to current desktop systems. The AAFD routine could be used to keep road parameters similar along connected roads in a urban network. Although the number of linear sub-networks found will be large as intersections will be detected as a new section for the non-priority road.

A final aspect for consideration is the integration of the developed methods into an wider autonomic system as discussed at the end of Chapter 7.

Appendix A

Manchester Model

Table A.1: Links of the Manchester model.

Link Name	Motorway	Direction	Junction	Lanes	Length	Segs	Section
Origins							
O5103-1	A5103	N	N/A	2	–	–	0
O56	M56	E	6–7	3	–	–	0
O56-4N	M56	E	4	2	–	–	0
O56-5N	M56	E	5	2	–	–	0
O56-6N	M56	E	6	1	–	–	0
O61	M61	S	2–3	3	–	–	1
OA666A	A666	S	N/A	2	–	–	1
O60-1AC	M60	ACW	1	1	–	–	2
O60-3AC	M60	ACW	3	2	–	–	2
O60-5AC	M60	ACW	5	1	–	–	2
O60-6AC	M60	ACW	6	1	–	–	2
O60-7AC	M60	ACW	7	1	–	–	2
O60-9AC	M60	ACW	9	2	–	–	2
O60-10AC	M60	ACW	10	2	–	–	2
O60-11AC	M60	ACW	11	2	–	–	2
O60-13AC	M60	ACW	13	1	–	–	2
O60-17AC	M60	ACW	17	1	–	–	2
O60-19AC	M60	ACW	19	1	–	–	2
O60-20AC	M60	ACW	20	2	–	–	2
O60-21AC	M60	ACW	21	2	–	–	2
O60-23AC	M60	ACW	23	2	–	–	2
O60-24AC	M60	ACW	24	2	–	–	2
O60-25AC	M60	ACW	25	1	–	–	2
O62	M62	W	19–20	3	–	–	2
O62-18S	M62	W	18	1	–	–	2
O62-19W	M62	W	19	2	–	–	2
O62E	M62	E	11–12	3	–	–	3
O60-1C	M60	CW	1	2	–	–	4
O60-2	M60	CW	2	2	–	–	4
O60-3C	M60	CW	3	2	–	–	4
O60-6/7C	M60	CW	6/7	1	–	–	4
O60-8C	M60	CW	8	1	–	–	4
O60-9C	M60	CW	9	2	–	–	4
O60-10C	M60	CW	10	2	–	–	4
O60-11C	M60	CW	11	2	–	–	4
O60-13C	M60	CW	13	1	–	–	4
O60-16C	M60	CW	16	1	–	–	4

Table A.1 Links of the Manchester model *Continued*

Link Name	Motorway	Direction	Junction	Lanes	Length	Segs	Section
O60-17C	M60	CW	17	1	–	–	4
O60-19C	M60	CW	19	2	–	–	4
O60-21C	M60	CW	21	1	–	–	4
O60-22C	M60	CW	22	2	–	–	4
O60-23aC	M60	CW	23	1	–	–	4
O60-23bC	M60	CW	23	1	–	–	4
O60-24C	M60	CW	24	2	–	–	4
O60-26C	M60	CW	26	1	–	–	4
O60-27C	M60	CW	27	1	–	–	4
O66	M66	S	3–4	3	–	–	4
O602W	M602	W	1–2	3	–	–	5
O5103	A5103	S	N/A	2	–	–	6
O5103-2	A5103	S	N/A	2	–	–	6
O56-5S	M56	W	5	1	–	–	6
O56-6S	M56	W	6	1	–	–	6
O60-14C	M60	CW	14	2	–	–	7
O56-2N	M56	E	2	2	–	–	8
OA666B	A666	S	N/A	2	–	–	10
OA34S	M56	W	1	1	–	–	21
Destinations							
D5103	A5103	N	N/A	2	–	–	0
D5103-1	A5103	N	N/A	2	–	–	0
D56-5N	M56	E	5	2	–	–	0
D56-6N	M56	E	6	2	–	–	0
D60-1AC	M60	ACW	1	2	–	–	2
D60-2	M60	CW	2	3	–	–	2
D60-3AC	M60	ACW	3	2	–	–	2
D60-7AC	M60	ACW	7	1	–	–	2
D60-8AC	M60	ACW	8	2	–	–	2
D60-9AC	M60	ACW	9	2	–	–	2
D60-10AC	M60	ACW	10	2	–	–	2
D60-13AC	M60	ACW	13	2	–	–	2
D60-16AC	M60	ACW	16	2	–	–	2
D60-17AC	M60	ACW	17	2	–	–	2
D60-19AC	M60	ACW	19	2	–	–	2
D60-21AC	M60	ACW	21	2	–	–	2
D60-22AC	M60	ACW	22	2	–	–	2
D60-23AC	M60	ACW	23	2	–	–	2
D60-24AC	M60	ACW	24	2	–	–	2
D60-25AC	M60	ACW	25	1	–	–	2
D60-27AC	M60	ACW	27	1	–	–	2
D66	M66	N	3–4	2	–	–	2
D602E	M602	E	1–2	3	–	–	3
D60-1C	M60	CW	1	2	–	–	4
D60-3C	M60	CW	3	2	–	–	4
D60-5C	M60	CW	5	1	–	–	4
D60-6C	M60	CW	6	2	–	–	4
D60-7C	M60	CW	7	2	–	–	4
D60-9C	M60	CW	9	2	–	–	4
D60-10C	M60	CW	10	2	–	–	4
D60-11C	M60	CW	11	2	–	–	4
D60-13C	M60	CW	13	2	–	–	4
D60-17C	M60	CW	17	2	–	–	4
D60-19C	M60	CW	19	2	–	–	4
D60-20C	M60	CW	20	2	–	–	4
D60-21C	M60	CW	21	2	–	–	4

Table A.1 Links of the Manchester model *Continued*

Link Name	Motorway	Direction	Junction	Lanes	Length	Segs	Section
D60-22C	M60	CW	22	2	–	–	4
D60-23C	M60	CW	23	2	–	–	4
D60-24C	M60	CW	24	2	–	–	4
D60-25C	M60	CW	25	1	–	–	4
D60-26C	M60	CW	26	1	–	–	4
D62	M62	E	19–20	3	–	–	4
D62-19E	M62	E	19	2	–	–	4
D5103-2	A5103	S	N/A	2	–	–	6
D56	M56	W	6–7	3	–	–	6
D56-4S	M56	W	4	2	–	–	6
D56-5S	M56	W	5	2	–	–	6
D56-6S	M56	W	6	2	–	–	6
DA34N	A34	N	M56 J1	1	–	–	8
DA580E	A580	E	N/A	2	–	–	10
DA580	A580	W	N/A	2	–	–	14
D56-2S	M56	W	2	2	–	–	21
D62W	M62	W	11–12	3	–	–	22
D61W	M61	N	2–3	4	–	–	23
D60-11AC	M60	ACW	11	2	–	–	29
Normal							
5103-1N	A5103	N	N/A	3	0.80	1	0
5103-2N	A5103	N	N/A	3	0.50	1	0
5103-3N	A5103	N	N/A	3	0.80	1	0
56-3N	M56	E	3–4	4	0.60	1	0
56-4N	M56	E	4–5	4	1.10	2	0
56-5N	M56	E	5	3	0.70	1	0
56-6N	M56	E	5–6	4	1.00	2	0
56-7N	M56	E	6	3	0.50	1	0
56-8N	M56	E	6–7	3	0.90	2	0
61-E1	M61	S	1-2	4	1.00	2	1
61-E2	M61	S	2	2	0.70	1	1
61-E3	M61	S	2	3	0.45	1	1
61EA	M61	S	1–2	2	1.00	2	1
62-1W	M62	W	17–18	4	1.10	2	2
62-2W	M62	W	17–18	4	2.00	4	2
62-3W	M62	W	18–19	3	0.90	2	2
62-4W	M62	W	18–19	3	0.80	1	2
66-1N	M66	N	3–4	3	0.75	1	2
66-2N	M66	N	4	2	1.00	2	2
L1AC	M60	ACW	1–2	3	2.40	5	2
L2AC	M60	ACW	2–3	4	0.60	1	2
L3AC	M60	ACW	3	3	0.80	1	2
L4AC	M60	ACW	3	2	0.80	1	2
L5AC	M60	ACW	3–5	3	2.10	4	2
L6AC	M60	ACW	5	3	1.00	2	2
L7AC	M60	ACW	5–6	4	1.60	3	2
L8AC	M60	ACW	6–7	4	0.45	1	2
L9AC	M60	ACW	7	3	3.15	7	2
L10AC	M60	ACW	7–8	3	0.60	1	2
L11AC	M60	ACW	8–9	3	1.20	2	2
L12AC	M60	ACW	9	3	0.70	1	2
L13AC	M60	ACW	9–10	3	0.70	1	2
L14AC	M60	ACW	10	3	0.80	1	2
L15AC	M60	ACW	10–11	3	1.15	2	2
L16AC	M60	ACW	11	3	0.45	1	2
L17AC	M60	ACW	11–12	3	0.90	2	2

Table A.1 Links of the Manchester model *Continued*

Link Name	Motorway	Direction	Junction	Lanes	Length	Segs	Section
L18AC	M60	ACW	11–12	3	0.90	2	2
L19AC	M60	ACW	12–13	3	0.70	1	2
L20AC	M60	ACW	13	4	1.00	2	2
L21AC	M60	ACW	13–14	3	1.70	3	2
L22AC	M60	ACW	14	3	0.90	2	2
L23AC	M60	ACW	14–15	3	0.60	1	2
L24AC	M60	ACW	15–16	4	1.45	3	2
L25AC	M60	ACW	16–17	4	2.60	5	2
L26AC	M60	ACW	17	4	0.80	1	2
L27AC	M60	ACW	17–18	4	1.10	2	2
L28AC	M60	ACW	18–19	5	0.45	1	2
L29AC	M60	ACW	18–19	4	0.70	1	2
L30AC	M60	ACW	19	3	0.80	1	2
L31AC	M60	ACW	19–20	3	1.50	3	2
L32AC	M60	ACW	20–21	3	3.30	7	2
L33AC	M60	ACW	21	2	1.25	2	2
L34AC	M60	ACW	21–22	3	1.05	2	2
L35AC	M60	ACW	22–23	4	3.70	8	2
L36AC	M60	ACW	23	3	1.40	3	2
L37AC	M60	ACW	23–24	4	1.90	4	2
L38AC	M60	ACW	24	3	1.10	2	2
L39AC	M60	ACW	24–25	3	2.60	5	2
L40AC	M60	ACW	25	3	1.10	2	2
L41AC	M60	ACW	25–27	3	1.75	3	2
L42AC	M60	ACW	27–1	3	1.30	2	2
L43AC	M60	ACW	1	3	0.90	2	2
602-E1	M602	E	1	2	0.60	1	3
602-E2	M602	E	1	2	0.45	1	3
602-E3	M602	E	1–2	2	0.60	1	3
62-M0	M62	E	11–12	3	4.00	8	3
62-M1	M62	E	11–12	4	0.90	2	3
62-1E	M62	E	17–18	3	1.00	2	4
62-2E	M62	E	18–19	3	3.60	8	4
66-1S	M66	S	3–4	4	1.00	2	4
66-2S	M66	S	4	2	0.80	1	4
L1C	M60	CW	1–2	3	2.40	4	4
L2C	M60	CW	2–3	4	0.60	1	4
L3C	M60	CW	3	4	0.65	1	4
L4C	M60	CW	3	3	0.45	1	4
L5C	M60	CW	3–4	3	2.20	4	4
L6C	M60	CW	5	3	1.50	3	4
L7C	M60	CW	5–6	4	1.40	3	4
L8C	M60	CW	6–7	4	0.45	1	4
L9C	M60	CW	6–6/7	3	3.20	7	4
L10C	M60	CW	6/7–8	3	0.80	1	4
L11C	M60	CW	8–9	3	1.20	2	4
L12C	M60	CW	9	4	0.70	1	4
L13C	M60	CW	9–10	4	0.60	1	4
L14C	M60	CW	10	3	0.90	2	4
L15C	M60	CW	10–11	3	1.10	2	4
L16C	M60	CW	11	3	0.60	1	4
L17C	M60	CW	11–12	3	0.50	1	4
L18C	M60	CW	12	3	0.55	1	4
L19C	M60	CW	12	3	0.45	1	4
L20C	M60	CW	12–13	4	0.50	1	4
L21C	M60	CW	13	4	0.65	1	4
L22C	M60	CW	13–14	4	2.40	5	4

Table A.1 Links of the Manchester model *Continued*

Link Name	Motorway	Direction	Junction	Lanes	Length	Segs	Section
L23C	M60	CW	14	3	0.90	2	4
L24C	M60	CW	14–15	3	0.45	1	4
L25C	M60	CW	15–16	4	1.37	3	4
L26C	M60	CW	16–17	4	2.40	5	4
L27C	M60	CW	17	4	1.10	2	4
L28C	M60	CW	17–18	4	1.00	2	4
L29C	M60	CW	17–18	5	0.45	1	4
L30C	M60	CW	18–19	4	0.95	2	4
L31C	M60	CW	19	3	1.00	2	4
L32C	M60	CW	19–20	3	1.45	3	4
L33C	M60	CW	20–21	3	3.00	6	4
L34C	M60	CW	21	2	1.00	2	4
L35C	M60	CW	21–22	3	0.60	1	4
L36C	M60	CW	22	3	0.80	1	4
L37C	M60	CW	22–23	4	3.75	8	4
L38C	M60	CW	23	3	1.00	2	4
L39C	M60	CW	23	3	0.55	1	4
L40C	M60	CW	23–24	4	2.00	4	4
L41C	M60	CW	24	3	1.00	2	4
L42C	M60	CW	24–25	3	2.75	6	4
L43C	M60	CW	25–26	3	1.00	2	4
L44C	M60	CW	26	3	0.60	1	4
L45C	M60	CW	26–27	3	1.00	2	4
L46C	M60	CW	27–1	3	1.20	2	4
L47C	M60	CW	1	3	0.75	1	4
602-W1	M602	W	1	2	0.45	1	5
602-W2	M602	W	1	2	0.45	1	5
602-W3	M602	W	1–2	2	0.60	1	5
5103-1S	A5103	S	N/A	3	0.80	1	6
5103-2S	A5103	S	N/A	3	0.50	1	6
5103-3S	A5103	S	N/A	3	0.80	1	6
56-3S	M56	W	4	4	0.70	1	6
56-4S	M56	W	4–5	4	0.50	1	6
56-5S	M56	W	5	3	1.20	2	6
56-6S	M56	W	5–6	4	1.00	2	6
56-7S	M56	W	6	3	0.50	1	6
56-8S	M56	W	6–7	3	1.00	2	6
A580A	A580	W	N/A	2	0.90	2	7
56-1aN	M56	E	1	2	0.45	1	8
56-1N	M56	E	1–2	2	1.00	2	8
56-2N	M56	E	2–3	2	2.00	4	8
J5C-ON	M60/A5103	CW	5	2	0.70	1	9
A580E	A580	E	N/A	2	1.30	2	10
A580E1	A580	E	N/A	2	0.50	1	10
A580E2	A580	E	N/A	1	0.50	1	10
61EB	M61	S	1–2	2	1.20	2	11
62-J18	R'bout			2	0.45	1	12
62-J18a	R'bout			2	0.45	1	12
61WB	M61	N	1–2	2	1.00	2	13
A580B	A580	E	N/A	2	0.80	1	14
602E-K	M60/M602	E	12	2	0.45	1	15
62-K1	M60/M602	E	12	2	0.45	1	15
L18AC2	M60	ACW	11	1	0.45	1	16
J5AC-OFF	M60/A5103	ACW	5	2	1.30	2	17
60-J18	R'bout			2	0.45	1	18
60-J18a	R'bout			2	0.45	1	18
62-M2	M62/M60	E	12	2	0.45	1	19

Table A.1 Links of the Manchester model *Continued*

Link Name	Motorway	Direction	Junction	Lanes	Length	Segs	Section
62-M3	M62/M60	E	12	2	0.45	1	19
66-J18	R'bout			2	0.45	1	20
66-J18a	R'bout			2	0.45	1	20
56-1aS	M56	W	1	2	0.45	1	21
56-1S	M56	W	1–2	2	1.20	2	21
56-2S	M56	W	2–3	2	1.90	4	21
62-L1	M62	W	11–12	3	0.90	2	22
62-L2	M62/M60	W	12	2	0.45	1	22
62-L3	M62/M60	W	12	2	0.45	1	22
61-W1	M61	N	1–2	4	1.00	2	23
61WA	M61	N	1–2	2	0.70	1	23
60-J181	R'bout			2	0.45	1	24
60-J18a1	R'bout			2	0.45	1	24
62-J1	M602/M60/M62	W	12	1	0.70	1	25
62N-J2	M602/M60	N	12	1	0.45	1	25
60-J18d	R'bout			3	0.45	1	26
62-J18b	R'bout			1	0.45	1	26
602W-K	M60/M602	W	12	2	0.80	1	27
60-J18b	R'bout			1	0.45	1	28
60-J18d1	R'bout			3	0.45	1	28
62-M4	M62/M60	E	12	1	1.00	2	29
L16AC2	M60	ACW	11	1	0.45	1	29
L17AC2	M60	ACW	11	2	0.45	1	29
L17AC2A	M60	ACW	11	2	0.45	1	29
62-J18d	R'bout			2	0.45	1	30
66-J18b	R'bout			1	0.45	1	30
602-L2	M60/M602	W	12	2	0.50	1	31
60-J18b1	R'bout			1	0.45	1	32
66-J18d	R'bout			2	0.45	1	32
62S-J2	M602/M60	S	12	1	0.50	1	33
DUM1	R'bout			4	0.00	0	34
DUM2	R'bout			4	0.00	0	34
DUM3	R'bout			4	0.00	0	34
DUM4	R'bout			4	0.00	0	34
J18-1	R'bout			4	0.45	1	34
J18-2	R'bout			4	0.45	1	34
J18-3	R'bout			4	0.45	1	34
J18-4	R'bout			4	0.45	1	34
60-J18c	R'bout			2	0.45	1	35
60-J18c1	R'bout			2	0.45	1	36
66-J18c	R'bout			2	0.45	1	37
62-J18c	R'bout			2	0.45	1	38
				TOTAL	186.92	360	

Appendix B

Calibrator Input Files

Listing B.1: `config` file: defines key parameters for the optimisation.

```
Configuration file for validation algorithm
Name            Value    Description
POP             30       Number of particles in swarm or population in
   GA
ITER            5000     Maximum number of iterations , −ve value means
   max funtion evaluations but only an underestimate
FIX_LKS         0        If true model has single fundamental diagram
FLEX_FUND       1        If true algo tries to find fund_diag positions
    else as dictated by .fund file
QUAD_W          5.000000 Quadratic weight for FLEX_FUND if less than
   zero use stats
FLOW_W          0.000000
VFREE_W         0.0010
RHOCR_W         0.0015
SEARCH_RAD      40       min % that initial search can occupy
EXP_ALPHA       0.7      Value of alpha for smoothing data in
   exponential smoothing
META            1        If true algo uses metanet else AR
ALGO            3        Algo used 0−GA 1−PSO 2−NM 3−RPROP
STOP            30000    Number of iterations in stopping criteria
STOP_TOL        2000     Tolerance in Stopping Crit
RESET           5000     Number of iterations before introducing new
   pop members
RES_PERC        100      Reset percentage
REDUCE_RATIO    0.85     Proportion of POP used in next iteration
REDUCE_CAP      30       Floor value for population
FIT_RATIO       0.0      Proportion of population initialised with FD's
    from curve fitting
FD_MAX_LEN      11       Maximum number of links in a single FD
AVG_INT         −1       Time in seconds for moving avg
```

```
MAX/MIN
TAU        KAPPA     NUE        VMIN       ROMAX     DELTA     PHI        PHI/M     GAM
           VFREE     ALPHA      ROCR       SaF_CAP   SaF_vf    SaF_t
40         30.0      80         8.0        190       4.020     4.006      1.0       1.0
           130       3.50       45.0       2500      100       30
1          5.005     10         0.5        160       0.00008   0.00005    0.0001    0.0
           80        0.50       20.0       1000      50        10

GA-PROPERTIES
C_RATE           0.7       Crossover rate
M_RATE           0.10      Mutation rate
EL_COUNT         3         Elitism count
B                5.0       Factor in non-uniform mutation
M                2.0       Factor in sig-trunc scaling

PSO-PROPERTIES
SIG_MAX          0.5       Maximum value of std dev in guassian
    perturbation
SIG_MIN          0.2       Minimum value for std dev
ACC_W            1         Weighting to acceleration scalars c1,c2
VEL_W            0.2       Percentage of range velocity limited to in pso

DATA-LOCATIONS
DATA_SETS        1         Number of data sets, one is minimum and is in
    cwd
FILE_PATHS                 Paths of extra data sets- put full path or ./
    for extension of cwd. separate by comma

I/O Settings
EMAIL            1         If true Email at start and end of program
XTERM            0         If true start gnome terminals with tail, 0 all
    to stdout, -1 to file no terminal tracking
SAVE             20        Interval of save in minutes
LOAD             0         If true load a previous session
```

Listing B.2: `SI_conf` file: defines some aspects of the PSO algorithm to be used if selected.

```
# This file defines the PSO variant to be used in optimisation
SI_VARIANT       1         Defined Variant to use: 1 APSO 2009, 2 IAPSO
    2012, 3 APSO 2014, 4 SPSO 2011, 5 CEPSO, 6 HEPSO
SI_TOPP          1         Defines neighborhood, 0 SI default, 1 Ring, 2
    Global
SI_MUTATION      0         0 default, 1 my fd mutation
```

Listing B.3: Extract of Manchester `.mna` file: defines loop detector locations and additional detectors.

```
M62/1608B        O62      −10    7
M602/6013B       O602W    −10    7
M61/2039B        O61      −10    7
M60/9025K        O60−2     0     1
M56/8128K        O56−5S   0.0    1
M56/8144K        O56−6S   0.0    1
M56/8088M        O56−2N   0.0    1
M56/8114M        O56−4N   0.0    1
M56/8125M        O56−5N   0.0    1
M60/9018A        L1C      1.0    7
M60/9021A        L1C      1.3    7
M60/9025A        L1C      1.7    7
M60/9028A        L2C      0.2    1
M60/9032A        L3C      0.0    2
M60/9032J        D60−3C   0.0    1
M60/9035A        L3C      0.3    2
M61/2031L        61−E2    0.6    1
M61/5213B        61−E1    0.6    1
M61/5209B        61EB     0.2    1
M62/1429B        D62W     0      1
M602/6012A       D602E    0      1
M66/4123B        D66      0.0    1
M66/4128A        66−1S    0.5    1
M66/4134A        66−2S    0.0    1
M66/4134J        66−J18   0.0    1
M60/9192M        L17AC2A  0.2    1
### SPECIAL ### − 1st col detector loc to be created , 2nd col csv text
    string coefficient then detector (if more than one use @ to input
    distance L1@0.2) all created values use 15min avgs last value in
    string proceeded by # gives distance of created data
O5103−1          1200,CONST,#0.0,?2
O5103−2          800,CONST,#0.0,?2
D5103−2          1200,CONST,#0.0,?2
D5103−1          800,CONST,#0.0,?2
O60−3C           1.0,L5C@0.8,−1.0,L4C@0.0,#0.0,?2
O60−3AC          1.0,L2AC@0.0,−1.0,L3AC@0.4,#0.0,?2
O60−6/7C         0.5,L11C@0.1,−0.5,L9C@2.9,#0.0,?2
O60−8C           0.5,L11C@0.1,−0.5,L9C@2.9,#0.0,?2
D60−19C         1.0,L30C,−1.0,L31C,#0.0,?2
O60−19C         1.0,L33C,−1.0,L31C,500,CONST,#0.0,?2
L32C            1.0,L31C,1.0,O60−19C,#0.7,?3
D60−20C         1.0,L32C,−1.0,L33C,#0.0,?2
5103−1N          1.0,56−4N|60,1.0,O56−4N,−1.0,56−2N@0.4|−60,#0.0,?2
```

```
5103−2S              1.0,5103−3S|−60,−1.0,O5103−2|−60,#0.0,?3
5103−1S              1.0,5103−2S|60,1.0,D5103−2|60,#0.0,?3
```

Appendix C

Analysis Software User Guide

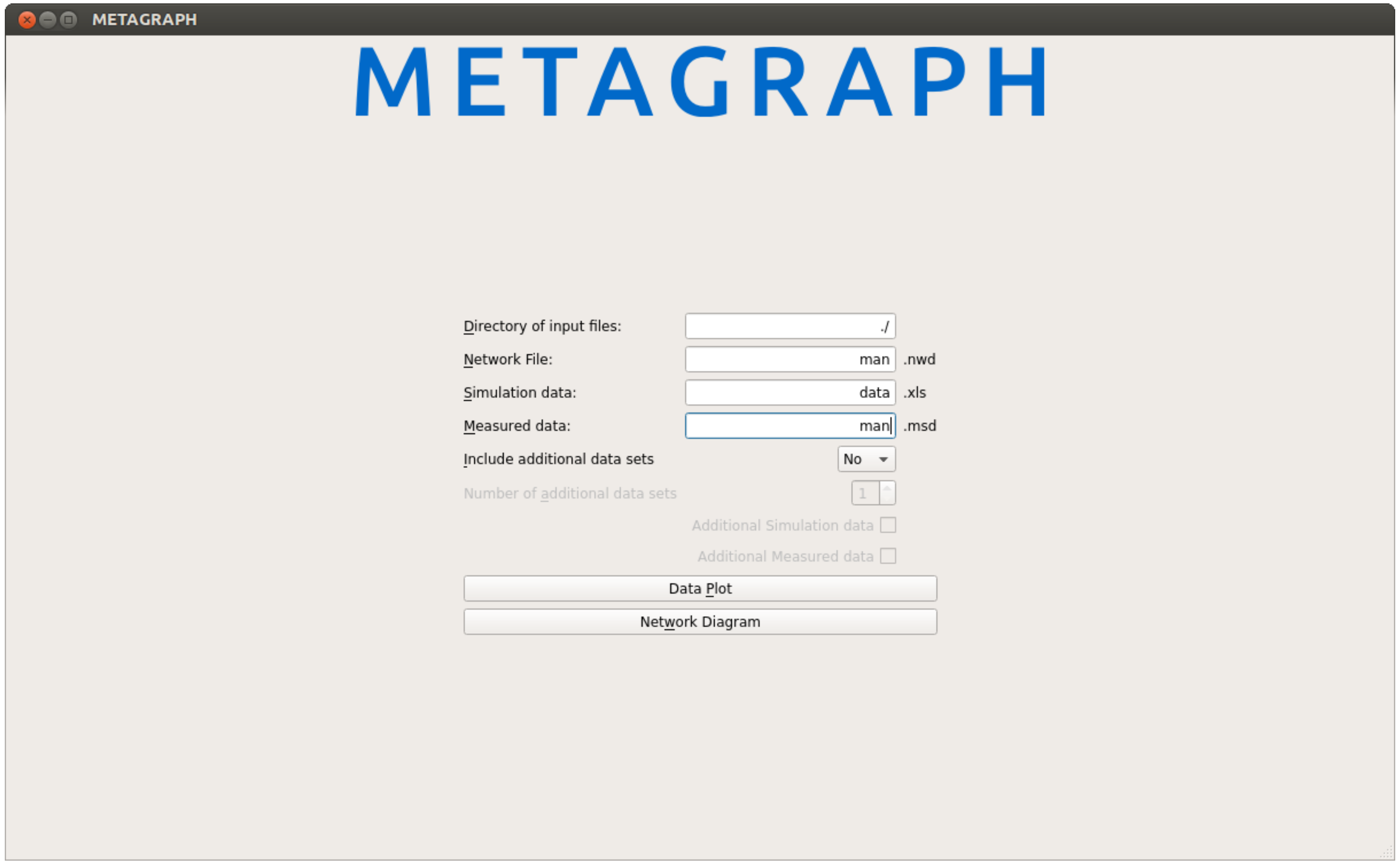
C.1 Starting

The analysis application has two main functions; looking at the network topology and comparing the output of a model to collected data. The program utilises the input and configuration files of the model and optimisation program. In network mode it is possible to see how the model is divided into sections. While in the comparison function it is also possible to calculate the objective function. The start-up screen is shown in Figure C.1, the user has the option to select additional data sets so a model can be compared to many sets of real world data or other models. The user must enter the filepath of the data files and their names in the top four boxes. The dropdown box 'Include additional data sets' defaults to 'No' by changing this to 'Yes' the next three rows will become enabled. If the user then declares the number of additional data sets and if they are including simulation data or measured data via the check-boxes. Once this data has been defined the user can go into the data analysis view by clicking 'Data Plot'. Or by selecting 'Network Diagram' the user can view the model topology. If extra data sets are included and the user selects 'Data Plot' then a pop-up appears to capture the filenames of the extra data sets.

C.2 Data Plot

The 'Data Plot' aspect of the program is shown in Figures C.2–C.5. The plots are generated in a Scalable Vector Graphics (SVG) format using Gnuplot [112]. This format means that the data exported can then be used to develop quality plots. By changing the settings tab it is possible to access and customise the Gnuplot settings used within the displayed plot. Settings available for modification include line styles, variable ranges, tic marks and grid lines.

In 'Data Plot' the main control is a tab-view with three pages the 'Plot' is shown



Directory of input files: ./

Network File: man .nwd

Simulation data: data .xls

Measured data: man .msd

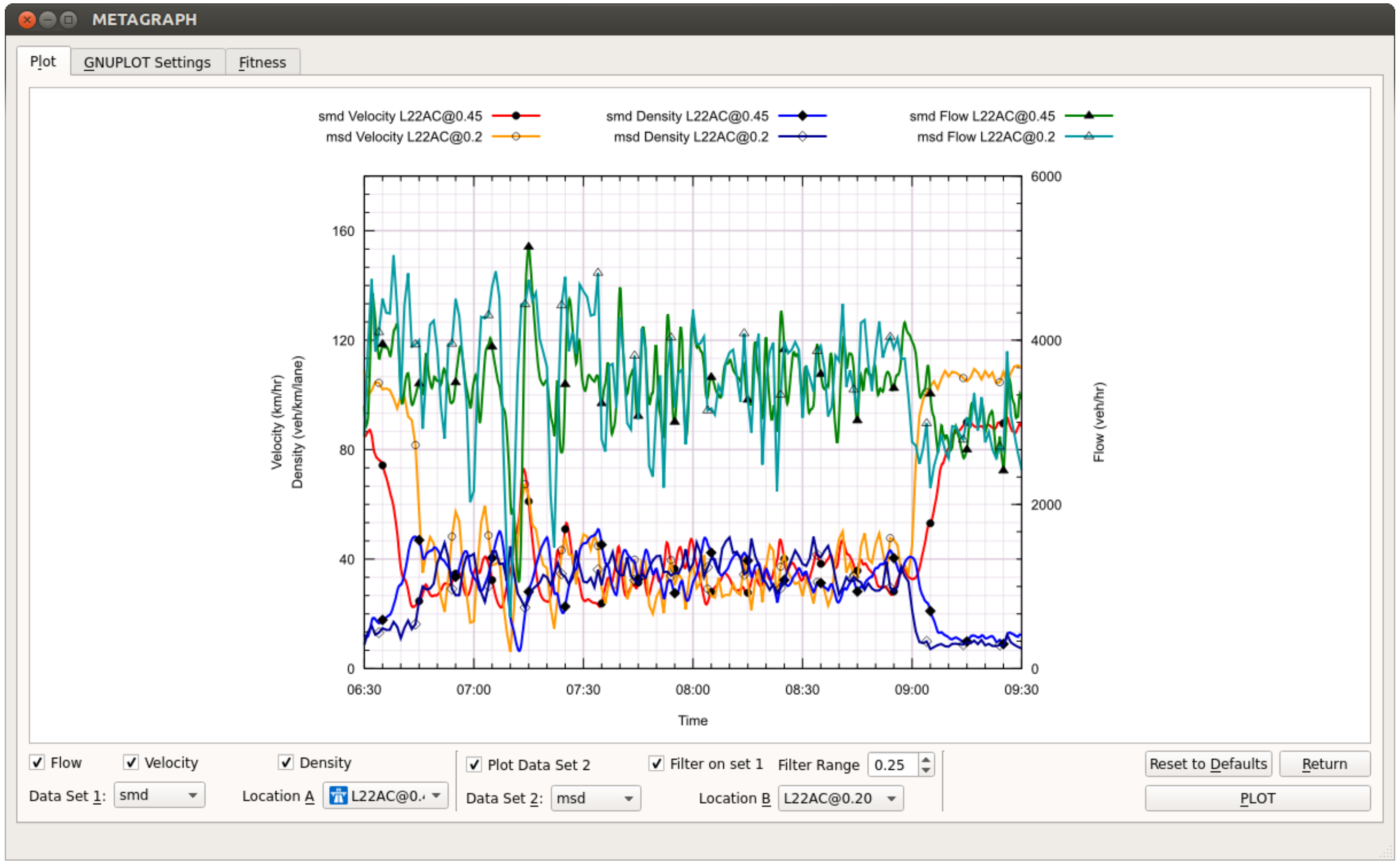
Include additional data sets No

Number of additional data sets 1

Additional Simulation data ☐

Additional Measured data ☐

Figure C.1: Start screen of METAGRAPH; a program with a GUI to aid analysis.



✓ Flow ✓ Velocity ✓ Density ✓ Plot Data Set 2 ✓ Filter on set 1 Filter Range 0.25

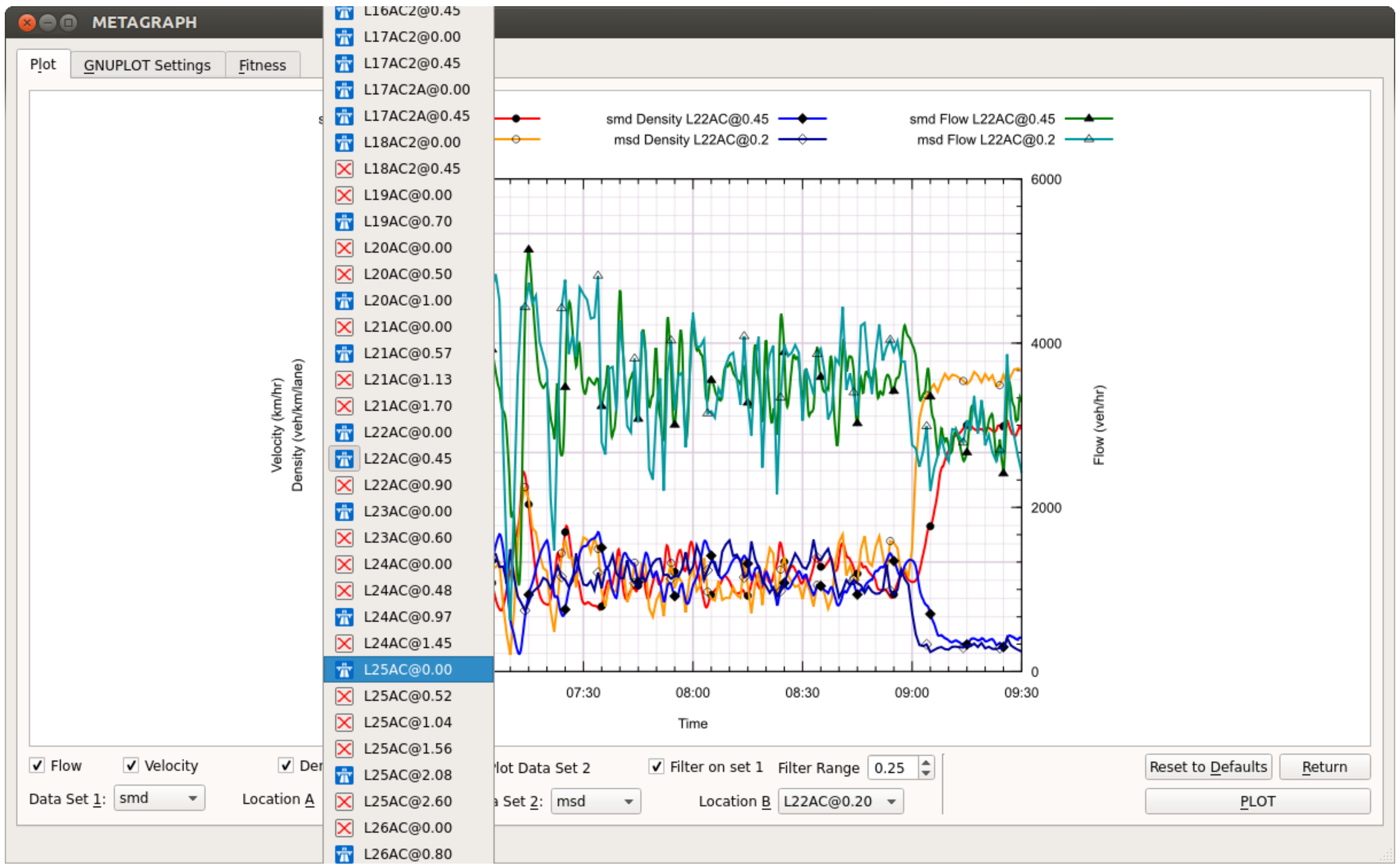
Data Set 1: smd Location A L22AC@0.4 Data Set 2: msd Location B L22AC@0.20

Figure C.2: The plot window.

in Figure C.2. The main part of this screen is the graph plotted and below are controls about what the plot should show. In the bottom right are three main controls. ‘Reset to Defaults’ reverts all the settings to their default values (this includes the ‘Gnuplot Settings’ tab), ‘Return’ which allows the user to return to the start screen, to select new data files or go to the network view and finally ‘Plot’. ‘Plot’ redraws the plot using all of the settings currently selected, the graph does no update on each change and this button has to be clicked to redraw the graph.

The settings and controls below the graph are divided into three sections. In the left most section there are three check-boxes ‘Flow’, ‘Velocity’ and ‘Density’ these allow for the user to toggle if these data series should be included within the plot. In Figure C.2 all the boxes are checked and therefore all three data series are plotted. Below the check boxes are two drop-downs; ‘Data Set 1’ allows the user to select which data set should be plotted. This data set is always plotted. The ‘Location A’ drop-down allows the user to choose which point in space that they wish to plot. This drop-down is shown in Figure C.3. With the Link name given and then an @ symbol with the number after being the distance along the link that the data is from. The icon on the left has two forms a motorway symbol and a red cross. If a motorway symbol is displayed this means there is direct comparison data available in the currently selected ‘Data Set 2’, a cross means that no matching location was found. In the second section of controls the ‘Data Set 2’ can be selected, unlike ‘Data Set 1’ this can be removed from the plot by removing the check from ‘Plot Data Set 2’. ‘Location B’ drop-down allows the selection of which point to plot from the second data set. Unlike the ‘Location A’ drop-down this can be filtered to only show locations that match the currently selected ‘Location A’. This can be turned off by removing the check from ‘Filter on set 1’. The final drop-down in section 2 is the ‘Filter Range’ this sets the limit on how close points need to be in space to consider comparing them. Only points within the same link will ever be considered for comparison.

The second tab ‘GNUPLOT Settings’ is shown in Figure C.4, here a wide variety of settings to change the style of the graph can be edited. The settings are grouped into three sections. On the left are line properties. If the top check box is unselected then the default Gnuplot line styles will be applied. This section allows the user to determine the main properties of the line such as its colour, width, point type and if the graph should be plotted using lines, points or lines and points. The second section on the top left of the screen allows for the ranges on the plot to be defined. If Automatic is selected then the range will be determined by Gnuplot and scaled to fit all data included. The final section contains some other attributes that can be edited such as font, axis marks and if the key should be included. Another feature

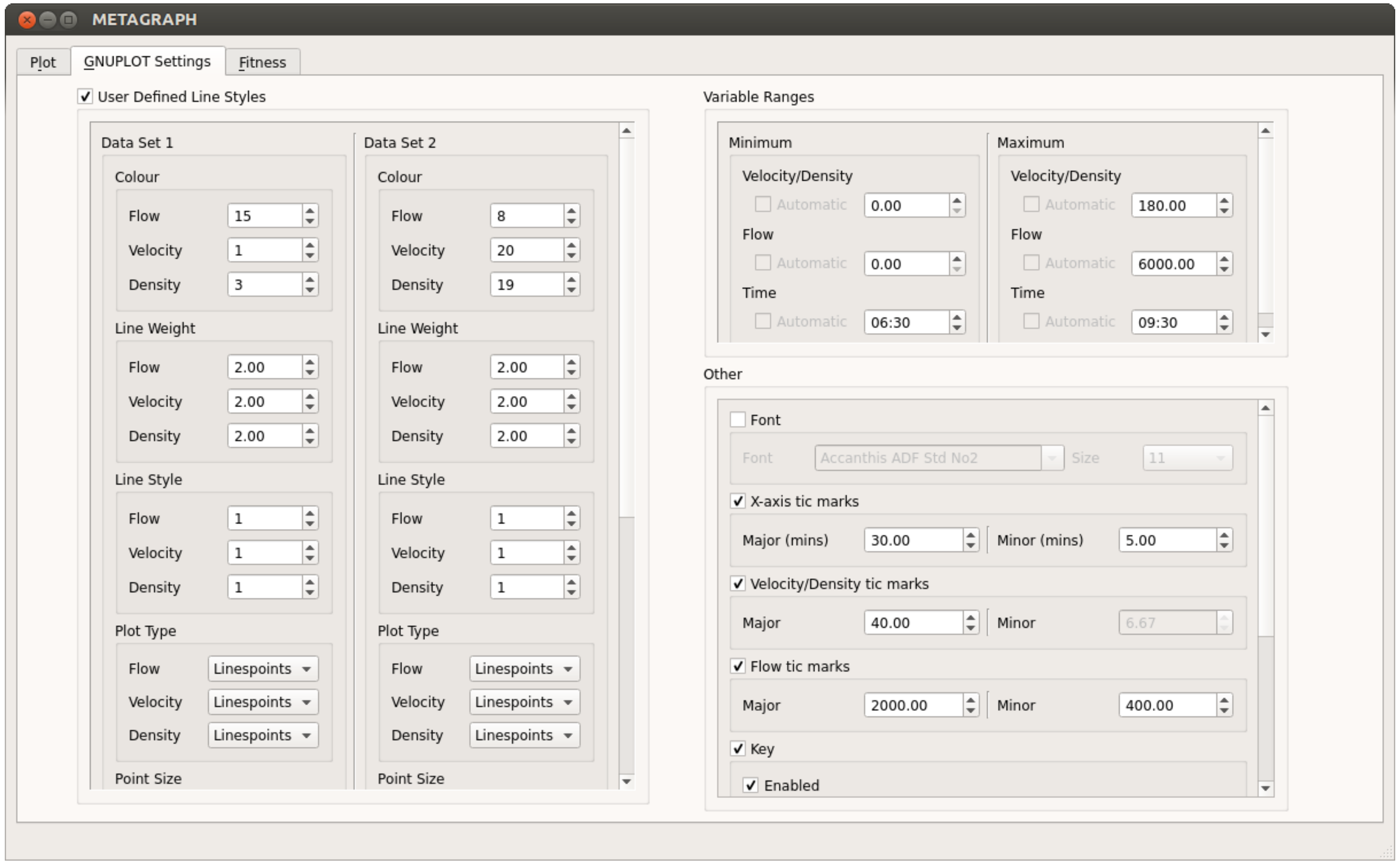


L16AC2@0.45
L17AC2@0.00
L17AC2@0.45
L17AC2A@0.00
L17AC2A@0.45
L18AC2@0.00
L18AC2@0.45
L19AC@0.00
L19AC@0.70
L20AC@0.00
L20AC@0.50
L20AC@1.00
L21AC@0.00
L21AC@0.57
L21AC@1.13
L21AC@1.70
L22AC@0.00
L22AC@0.45
L22AC@0.90
L23AC@0.00
L23AC@0.60
L24AC@0.00
L24AC@0.48
L24AC@0.97
L24AC@1.45
L25AC@0.00
L25AC@0.52
L25AC@1.04
L25AC@1.56
L25AC@2.08
L25AC@2.60
L26AC@0.00
L26AC@0.80

✓ Flow ✓ Velocity ✓ Der
Data Set 1: smd Location A

lot Data Set 2 ✓ Filter on set 1 Filter Range 0.25
Set 2: msd Location B L22AC@0.20

Figure C.3: Data points are selectable and icon shows if comparison data are available.


METAGRAPH

Plot | GNUPLOT Settings | Fitness

[✓] User Defined Line Styles

	Data Set 1	Data Set 2
Colour		
Flow	15	8
Velocity	1	20
Density	3	19
Line Weight		
Flow	2.00	2.00
Velocity	2.00	2.00
Density	2.00	2.00
Line Style		
Flow	1	1
Velocity	1	1
Density	1	1
Plot Type		
Flow	Linespoints	Linespoints
Velocity	Linespoints	Linespoints
Density	Linespoints	Linespoints
Point Size		

Variable Ranges

	Minimum	Maximum
Velocity/Density	[ ] Automatic 0.00	[ ] Automatic 180.00
Flow	[ ] Automatic 0.00	[ ] Automatic 6000.00
Time	[ ] Automatic 06:30	[ ] Automatic 09:30

Other

[ ] Font
Font: Accanthis ADF Std No2 Size: 11

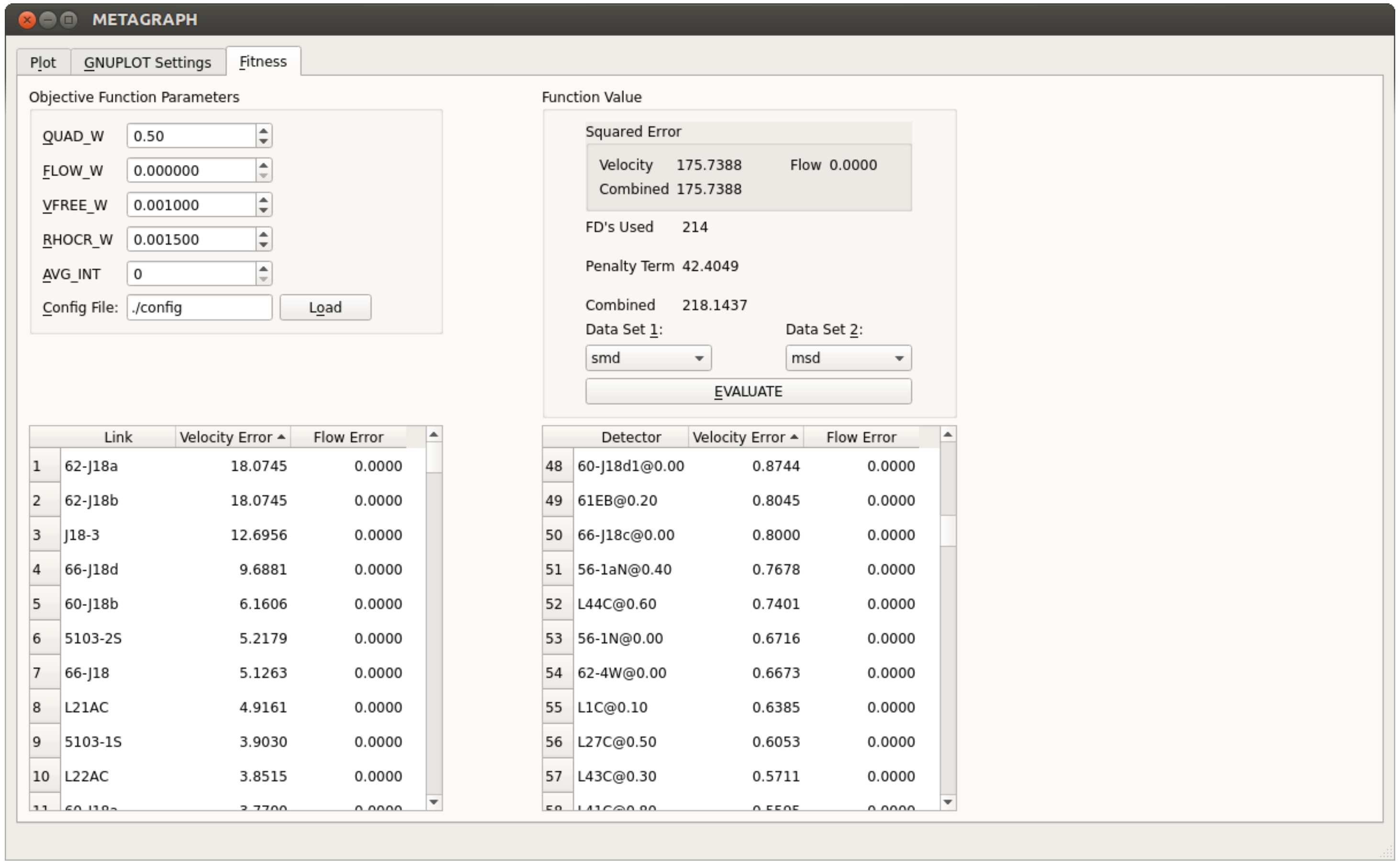
[✓] X-axis tic marks
Major (mins): 30.00 Minor (mins): 5.00

[✓] Velocity/Density tic marks
Major: 40.00 Minor: 6.67

[✓] Flow tic marks
Major: 2000.00 Minor: 400.00

[✓] Key
[✓] Enabled

Figure C.4: Graph settings are customisable.



Objective Function Parameters

QUAD_W	0.50
FLOW_W	0.000000
VFREE_W	0.001000
RHOCR_W	0.001500
AVG_INT	0
Config File:	./config

Function Value

Squared Error

Velocity 175.7388 Flow 0.0000
Combined 175.7388

FD's Used 214

Penalty Term 42.4049

Combined 218.1437

Data Set 1: smd
Data Set 2: msd

	Link	Velocity Error	Flow Error
1	62-J18a	18.0745	0.0000
2	62-J18b	18.0745	0.0000
3	J18-3	12.6956	0.0000
4	66-J18d	9.6881	0.0000
5	60-J18b	6.1606	0.0000
6	5103-2S	5.2179	0.0000
7	66-J18	5.1263	0.0000
8	L21AC	4.9161	0.0000
9	5103-1S	3.9030	0.0000
10	L22AC	3.8515	0.0000

	Detector	Velocity Error	Flow Error
48	60-J18d1@0.00	0.8744	0.0000
49	61EB@0.20	0.8045	0.0000
50	66-J18c@0.00	0.8000	0.0000
51	56-1aN@0.40	0.7678	0.0000
52	L44C@0.60	0.7401	0.0000
53	56-1N@0.00	0.6716	0.0000
54	62-4W@0.00	0.6673	0.0000
55	L1C@0.10	0.6385	0.0000
56	L27C@0.50	0.6053	0.0000
57	L43C@0.30	0.5711	0.0000

Figure C.5: Fitness tab of METAGRAPH.

in the other section is including grid lines, and an option to force the grid lines to become square. This is currently selected in the Figure (but is scrolled out of view) this overrides options in variable ranges to achieve the square grid as such it disables the automatic option on the variable ranges.

The final tab of the plot window is a breakdown of the objective function as shown in Figure C.5. This allows for the user to set the specific weightings for the objective function and also to load them from the configuration file used by the optimisation algorithm. The program then returns information about the objective function value and its breakdown. The left hand table in the figure shows the squared error components per link, the right hand one is per loop detector. These tables have sortable columns so the user can quickly identify the areas where the model and data correlate well and where they do not.

C.3 Network Diagram

The second part of the program creates a plot of the network as shown in Figure C.6. All the images are created during the loading of the screen from the main start screen, it can take a short time for this to happen. The created image is an SVG that is created based on optional node co-ordinates that can be defined in the network input file of METANET. The same routines as the calibration software are used so the the sectioning of the model (c.f Section 4.2.3) can easily be seen.

In this view the destinations are shown in red, origins in blue and the links and nodes in black. The arrows show the direction of travel. Each link, origin and destination is composed of a series of lines where an individual line represents one lane.

Along the bottom of the view are some simple controls to rotate and change the zoom level of the view port. By using the mouse and clicking and dragging the area of the model visible within the view can be changed. Hovering over Nodes, links, origins and destinations provides a tool-tip with the mouse-over objects name. ‘Reset’ returns the view to the default, a zoom level that will show the entire model in the original orientation and focus point.

In the bottom left there is a drop-down labelled ‘Section’ this can be used to make the only the selected section 100% opacity and with the rest at a reduced level. This makes it easy to locate sections and there ID assigned by the algorithm to assist in analysis. This utility is shown in Figure C.7, where section 8 of the Manchester model has been selected.

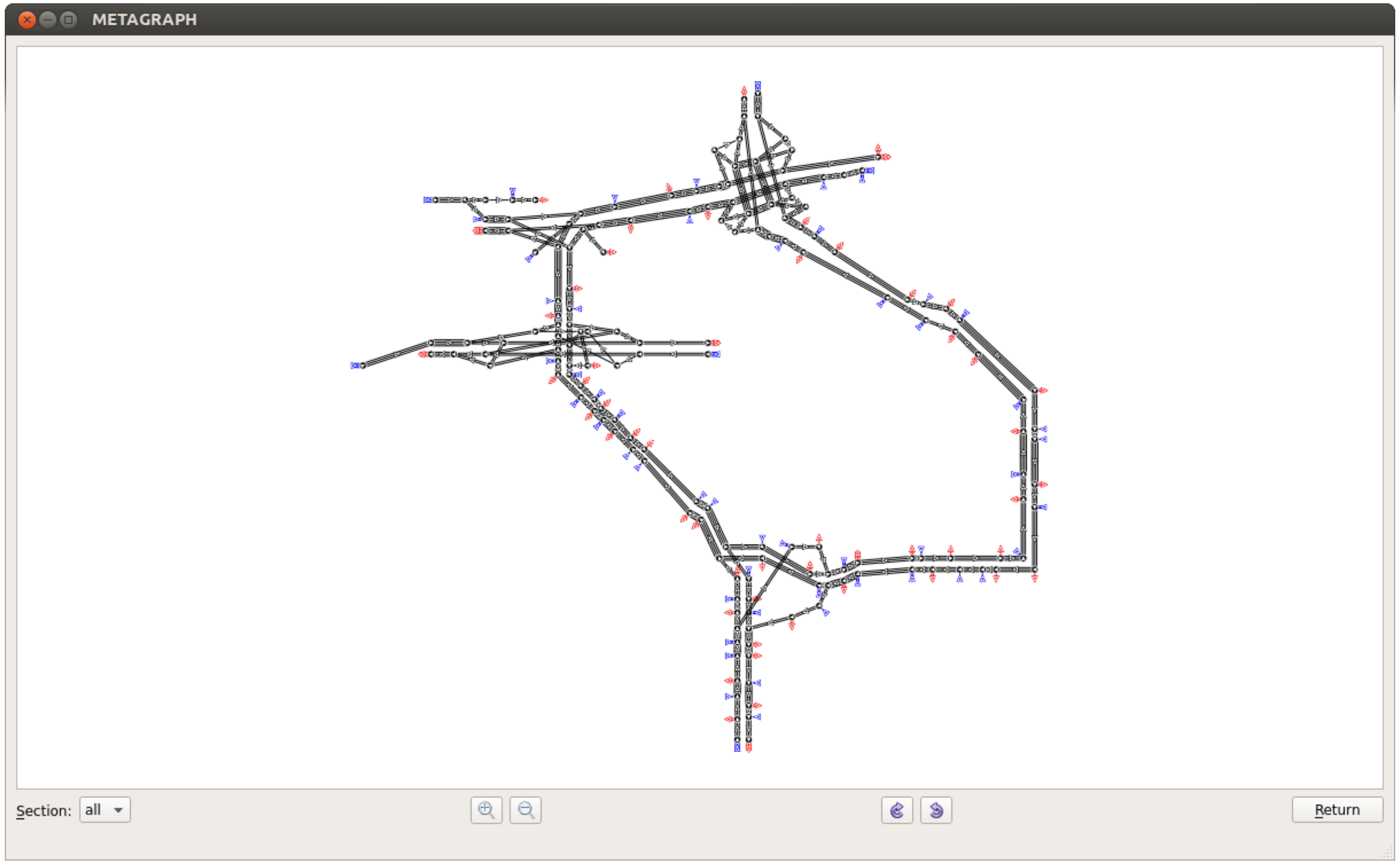


Figure C.6: Entire Network.

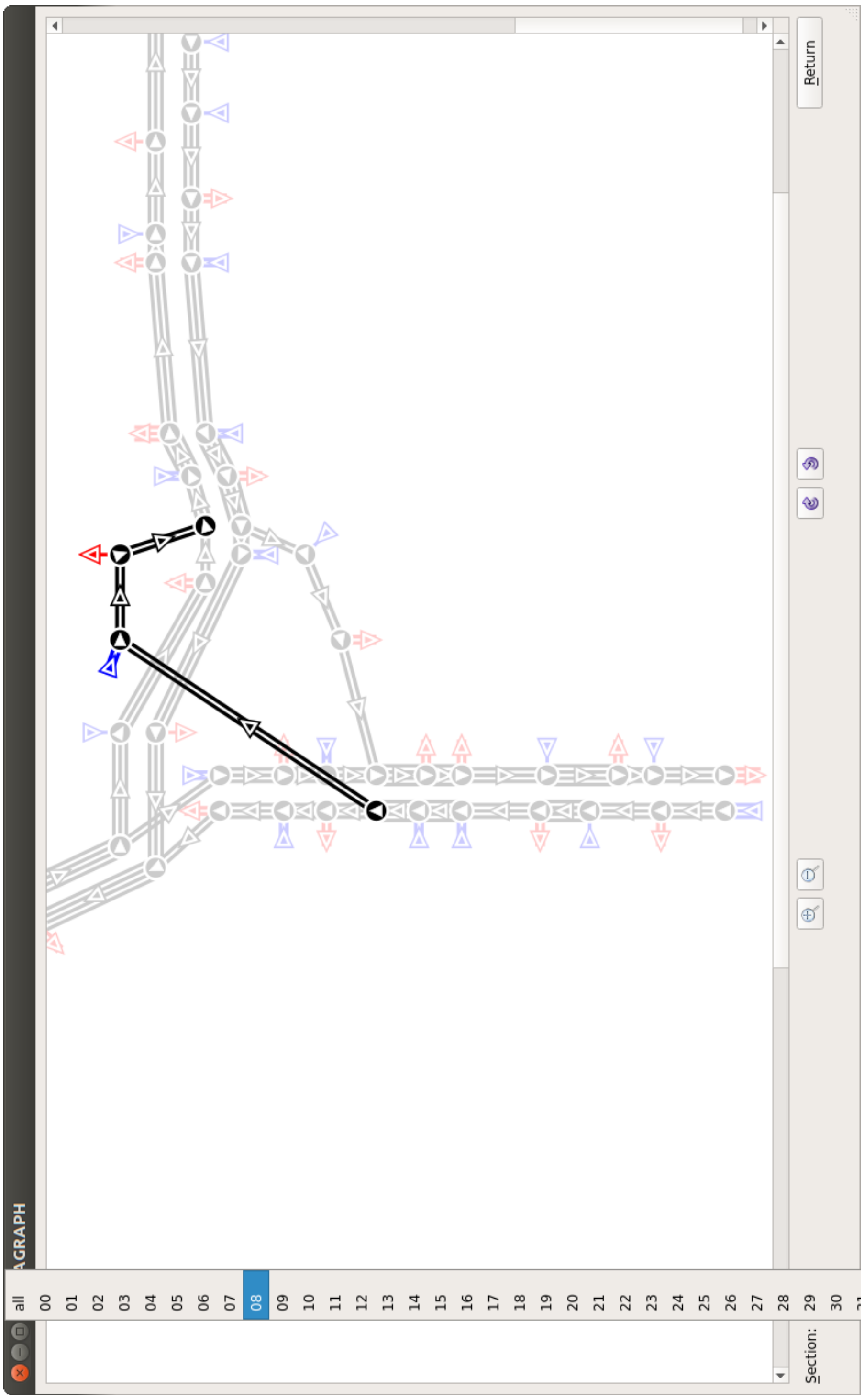


Figure C.7: Individual sections can be highlighted.

References

[1] A. Poole and A. Kotsialos, "Swarm intelligence algorithms for macroscopic traffic flow model validation with automatic assignment of fundamental diagrams," *Applied Soft Computing*, vol. 38, pp. 134–150, 2016.

[2] A. Poole and A. Kotsialos, "Second order macroscopic traffic flow model validation using automatic differentiation with resilient backpropagation and particle swarm optimisation algorithms," *Transportation Research Part C: Emerging Technologies*, vol. 71, pp. 356–381, 2016.

[3] A. Poole and A. Kotsialos, "METANET validation of the large scale Manchester ring-road network using gradient based and particle swarm optimisation," *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, no. 99, pp. 1–11, 2017.

[4] A. Poole and A. Kotsialos, "METANET model validation using a genetic algorithm," in *Proc. of the 13th IFAC Symp. on Control in Transportation Systems*, vol. 13, pp. 7–12, 2012.

[5] A. Poole and A. Kotsialos, "Deterministic model validation with a new approach to fundamental diagrams of the METANET model," in *Transportation Research Board 92nd Annual Meeting*, no. 13-4046 in 92, 2013.

[6] A. Kotsialos and A. Poole, "Autonomic systems design for ITS applications," in *Intelligent Transportation Systems (ITSC), 2013 16th International IEEE Conference on*, pp. 178–183, Oct 2013.

[7] A. Poole and A. Kotsialos, "Traffic Flow Model Validation Using METANET, ADOL-C and RPROP," in *Proc. of the 14th IFAC Symposium on Control in Transportation Systems*, (Istanbul, Turkey), pp. 291–296, 2016.

[8] A. Kotsialos and A. Poole, "Autonomic systems design for its applications: Modelling and route guidance," in *Autonomic Road Transport Support Systems* (T. L. McCluskey, A. Kotsialos, J. P. Müller, F. Klügl, O. Rana, and R. Schumann, eds.), pp. 131–145, Birkhäuser Basel, 2016.

[9] S. P. Hoogendoorn and P. H. L. Bovy, “State-of-the-art of vehicular traffic flow modelling,” *Proc. of the I MECH E Part I J. of Systems Control in Engineer*, vol. 215, no. 4, pp. 283–303, 2001.

[10] F. van Wageningen-Kessels, H. Van Lint, K. Vuik, and S. Hoogendoorn, “Genealogy of traffic flow models,” *EURO Journal on Transportation and Logistics*, vol. 4, no. 4, pp. 445–473, 2014.

[11] A. Kotsialos and M. Papageorgiou, “The importance of traffic flow modeling for motorway traffic control,” *Networks and Spatial Economics*, vol. 1, no. 1-2, pp. 179–203, 2001.

[12] M. Lighthill and G. Whitham, “On kinematic waves II: a traffic flow theory on long crowded roads,” *Proc. Roy. Soc. London Series A*, vol. 229, pp. 317–345, 1955.

[13] P. I. Richards, “Shock waves on the highway,” *Operations Research*, vol. 4, pp. 42–51, 1956.

[14] J.-P. Lebacque, “The godunov scheme and what it means for first order traffic flow models,” in *Internaional symposium on transportation and traffic theory*, pp. 647–677, 1996.

[15] J. Treiterer and J. Myers, “The hysteresis phenomenon in traffic flow,” *Transportation and traffic theory*, vol. 6, pp. 13–38, 1974.

[16] C. Daganzo, M. Cassidy, and R. Bertini, “Possible explanations of phase transitions in highway traffic,” *Transportation Research Part A: Policy and Practice*, vol. 33, no. 5, pp. 365 – 379, 1999.

[17] C. F. Daganzo, “The Cell Transmission Model, Part II: Network Traffic,” *Transportation Research - B*, vol. 29, pp. 79–93, 1995.

[18] C. Daganzo, “The Cell Transmission Model: A dynamic representation of highway traffic consistent with the hydrodynamic theory,” *Transportation Research B*, vol. 28B, pp. 269–287, 1994.

[19] H. J. Payne, “Models of freeway traffic and control,” *Proc. Simulation Council*, vol. 28, pp. 51–61, 1971.

[20] G. B. Whitham, *Linear and nonlinear waves.* New York: Wiley, 1974.

[21] C. F. Daganzo, "Requiem for second-order fluid approximations of traffic flow," *Transportation Research Part B: Methodological*, vol. 29, no. 4, pp. 277–286, 1995.

[22] M. Papageorgiou, "Some remarks on macroscopic traffic flow modelling," *Transportation Research Part A: Policy and Practice*, vol. 32, no. 5, pp. 323–329, 1998.

[23] H. Zhang, "Anisotropic property revisited—-does it hold in multi-lane traffic?," *Transportation Research Part B: Methodological*, vol. 37, no. 6, pp. 561–577, 2003.

[24] A. Messmer and M. Papageorgiou, "METANET: A macroscopic simulation program for motorway networks," *Traffic Engineering and Control*, vol. 31, pp. 466–470; 549, 1990.

[25] M. Cremer and A. D. May, "An extended traffic flow model for inner urban freeways," in *"IFAC/IFIP/IFORS International Conference on Control in Transportation Systems"*, (Vienna, Austria), pp. 383–388, 1986.

[26] M. Papageorgiou, J.-M. Blosseville, and H. Hadj-Salem, "Modelling and real-time control of traffic flow on the southern part of Boulevard Peripherique in Paris: Part I: Modelling," *Transportation Research Part A: General*, vol. 24, no. 5, pp. 345–359, 1990.

[27] A. Aw and M. Rascle, "Resurrection of "second order" models of traffic flow," *SIAM J. on Appl. Math.*, vol. 60, no. 3, pp. 916–938, 2000.

[28] H. M. Zhang, "A non-equilibrium traffic model devoid of gas-like behavior," *Transportation Research Part B: Methodological*, vol. 36, no. 3, pp. 275–290, 2002.

[29] J. M. Greenberg, "Extensions and amplifications of a traffic model of aw and rascle," *SIAM J. on Appl. Math.*, vol. 62, no. 3, pp. 729–745, 2002.

[30] J. Lebacque, S. Mammar, and H. Haj-Salem, "The Aw–Rascle and Zhangs model: Vacuum problems, existence and regularity of the solutions of the Riemann problem," *Transportation Research Part B: Methodological*, vol. 41, no. 7, pp. 710–721, 2007.

[31] M. Treiber, A. Hennecke, and D. Helbing, "Derivation, properties, and simulation of a gas-kinetic-based, nonlocal traffic model," *Physical Review E*, vol. 59, no. 1, p. 239, 1999.

[32] D. Helbing, A. Hennecke, V. Shvetsov, and M. Treiber, “Master: macroscopic traffic simulation based on a gas-kinetic, non-local traffic model,” *Transportation Research Part B: Methodological*, vol. 35, no. 2, pp. 183–211, 2001.

[33] M. Treiber and A. Kesting, “Traffic flow dynamics,” *Traffic Flow Dynamics: Data, Models and Simulation, Springer-Verlag Berlin Heidelberg*, 2013.

[34] R. Borges, M. Carmona, B. Costa, and W. S. Don, “An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws,” *Journal of Computational Physics*, vol. 227, no. 6, pp. 3191–3211, 2008.

[35] D. Helbing and M. Treiber, “Numerical simulation of macroscopic traffic equations,” *Computing in Science & Engineering*, vol. 1, no. 5, pp. 89–99, 1999.

[36] D. Ngoduy, S. Hoogendoorn, and H. Van Zuylen, “Comparison of numerical schemes for macroscopic traffic flow models,” *Transportation Research Record: J. of the Transportation Research Board*, vol. 1876, no. -1, pp. 52–61, 2004.

[37] L. Pareschi and G. Russo, “Implicit-explicit runge-kutta schemes and applications to hyperbolic systems with relaxation,” *Journal of Scientific computing*, vol. 25, no. 1-2, pp. 129–155, 2005.

[38] M. Zhang, C.-W. Shu, G. C. Wong, and S. Wong, “A weighted essentially non-oscillatory numerical scheme for a multi-class lighthill–whitham–richards traffic flow model,” *Journal of Computational Physics*, vol. 191, no. 2, pp. 639–659, 2003.

[39] A. Delis, I. Nikolos, and M. Papageorgiou, “High-resolution numerical relaxation approximations to second-order macroscopic traffic flow models,” *Transportation Research Part C: Emerging Technologies*, vol. 44, pp. 318–349, 2014.

[40] K. N. Porfyri, I. K. Nikolos, A. I. Delis, and M. Papageorgiou, “Calibration of a second-order traffic flow model using a metamodel-assisted differential evolution algorithm,” in *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*, pp. 366–371, IEEE, 2016.

[41] D. Ngoduy and M. Maher, “Calibration of second order traffic models using continuous cross entropy method,” *Transportation Research Part C: Emerging Technologies*, vol. 24, pp. 102–121, 2012.

[42] M. Cremer and M. Papageorgiou, “Parameter identification for a traffic flow model,” *Automatica*, vol. 17, no. 6, pp. 837–843, 1981.

[43] M. Papageorgiou, *Applications of automatic control concepts to traffic flow modeling and control*, vol. 50 of *Lecture Notes in Control and Information Sciences.* Springer, 1983.

[44] A. Kotsialos, M. Papageorgiou, C. Diakaki, Y. Pavlis, and F. Middelham, “Traffic flow modeling of large-scale motorway networks using the macroscopic modeling tool METANET,” *IEEE Trans. Intell. Transp. Syst.*, vol. 3, no. 4, pp. 282–292, 2002.

[45] K. K. Sanwal, K. Petty, J. Walrand, and Y. Fawaz, “An extended macroscopic model for traffic flow,” *Transportation Research Part B: Methodological*, vol. 30, no. 1, pp. 1–9, 1996.

[46] M. J. Box, “A new method of constrained optimization and a comparison with other method,” *Computer Journal*, vol. 8, no. 1, pp. 42–52, 1965.

[47] T. Monamy, H. Haj-Salem, and J.-P. Lebacque, “A macroscopic node model related to capacity drop,” *Procedia-Social and Behavioral Sciences*, vol. 54, pp. 1388–1396, 2012.

[48] A. Kotsialos, Y. Pavlis, F. Middelham, C. Diakaki, G. Vardaka, and M. Papageorgiou, “Modelling of the large scale motorway network around amsterdam,” *Preprints of the 8th IFAC Symposium on Large Scale Systems*, vol. 2, pp. 354–360, 1998.

[49] J. Nelder and R. Mead, “A simplex method for function minimization,” *The Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965.

[50] D. Ngoduy and S. Hoogendoorn, “An automated calibration procedure for macroscopic traffic flow models,” in *Proc. of the 10th IFAC Symp. on Control in Transportation Systems*, (Tokyo, Japan), pp. 52–61, 2003.

[51] A. Spiliopoulou, M. Kontorinaki, M. Papageorgiou, and P. Kopelias, “Macroscopic traffic flow model validation at congested freeway off-ramp areas,” *Transportation Research Part C: Emerging Technologies*, vol. 41, pp. 18–29, 2014.

[52] R. Y. Rubinstein and D. P. Kroese, *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning.* Springer Science & Business Media, 2013.

[53] A. Spiliopoulou, I. Papamichail, M. Papageorgiou, I. Tyrinopoulos, and J. Chrysoulakis, “Macroscopic traffic flow model calibration using different

optimization algorithms," *Transportation Research Procedia*, vol. 6, pp. 144–157, 2015.

[54] M. Kontorinaki, A. Spiliopoulou, I. Papamichail, M. Papageorgiou, Y. Tyrinopoulos, and J. Chrysoulakis, "Overview of nonlinear programming methods suitable for calibration of traffic flow models," *Operational Research Int. Jrn.*, vol. 15, no. 3, pp. 327–336, 2015.

[55] A. Ramezani, B. Moshiri, A. Rahimi Khan, and B. Abdulhai, "Design of an adaptive maximum likelihood estimator for key parameters in macroscopic traffic flow model based on expectation maximum algorithm," *Sci., Measurement & Technology, IET*, vol. 5, no. 5, pp. 189–197, 2011.

[56] T. Luspay, K. B., J. van Wingerden, M. Verhaegen, and J. Bokor, "Linear parameter varying identification of freeway traffic models," *IEEE Trans. Control Syst. Technol.*, vol. 19, pp. 31–45, Jan. 2011.

[57] T. Luspay, B. Kulcsár, J. van Wingerden, and M. Verhaegen, "On the identification of lpv traffic flow model," in *Control Conference (ECC), 2009 European*, pp. 1752–1757, IEEE, 2009.

[58] L. Rao, L. Owen, and D. Goldsman, "Development and application of a validation framework for traffic simulation models," in *Simulation Conf. Proc.*, vol. 2, pp. 1079–1086, IEEE, 1998.

[59] H. Liu, Q. Yu, W. Ding, D. Ni, H. Wang, and S. Shannon, "Feasibility study for automatic calibration of transportation simulation models," in *Proc. of the 44th Annu. Simulation Symp.*, (San Diego, CA, USA), pp. 87–94, 2011.

[60] J. R. D. Frejo, E. F. Camacho, and R. Horowitz, "A parameter identification algorithm for the METANET model with a limited number of loop detectors.," in *Proc. 51st IEEE Conference on Decision and Control*, pp. 6983–6988, 2012.

[61] M. Treiber and A. Kesting, "Validation of traffic flow models with respect to the spatiotemporal evolution of congested traffic patterns," *Transportation Research Part C: Emerging Technologies*, vol. 21, no. 1, pp. 31–41, 2012.

[62] A. Alessandri, R. Bolla, A. Grassia, and M. Repetto, "Identification of freeway macroscopic models using information from mobile phones," in *American Control Conf.*, pp. 6–pp, IEEE, 2006.

[63] L. Munoz, X. Sun, D. Sun, G. Gomes, and R. Horowitz, “Methodological calibration of the cell transmission model,” in *Proc. of the 2004 American Control Conf.*, (Boston, MA, USA), pp. 798–803, 2004.

[64] L. Munoz, X. Sun, R. Horowitz, and L. Alvarez, “A piecewise-linearized cell transmission model and parameter calibration methodology,” in *Proc. of the 85th Transportation Research Board Annu. Meeting*, (Washington D.C., USA), pp. 183–191, 2006.

[65] A.-C. Zăvoianu, G. Bramerdorfer, E. Lughofer, S. Silber, W. Amrhein, and E. P. Klement, “Hybridization of multi-objective evolutionary algorithms and artificial neural networks for optimizing the performance of electrical drives,” *Engineering Applications of Artificial Intelligence*, vol. 26, no. 8, pp. 1781–1794, 2013.

[66] R.-E. Precup, R.-C. David, E. M. Petriu, S. Preitl, and M.-B. Radac, “Novel adaptive gravitational search algorithm for fuzzy controlled servo systems,” *Industrial Informatics, IEEE Transactions on*, vol. 8, no. 4, pp. 791–800, 2012.

[67] Z. Zhang, Y. Jiang, S. Zhang, S. Geng, H. Wang, and G. Sang, “An adaptive particle swarm optimization algorithm for reservoir operation optimization,” *Applied Soft Computing*, vol. 18, pp. 167–177, 2014.

[68] Y. Wang, J. Zhou, C. Zhou, Y. Wang, H. Qin, and Y. Lu, “An improved self-adaptive pso technique for short-term hydrothermal scheduling,” *Expert Systems with Applications*, vol. 39, no. 3, pp. 2288–2295, 2012.

[69] B. Akay and D. Karaboga, “A modified artificial bee colony algorithm for real-parameter optimization,” *Information Sciences*, vol. 192, no. 0, pp. 120 – 142, 2012.

[70] T. Apostolopoulos and A. Vlachos, “Application of the firefly algorithm for solving the economic emissions load dispatch problem,” *International Journal of Combinatorics*, vol. 2011, 2010.

[71] X.-S. Yang, S. S. S. Hosseini, and A. H. Gandomi, “Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect,” *Applied Soft Computing*, vol. 12, no. 3, pp. 1180 – 1186, 2012.

[72] Y. Shi *et al.*, “Particle swarm optimization: developments, applications and resources,” in *Proc. IEEE Congr. Evol. Comput.*, vol. 1, pp. 81–86, IEEE, 2001.

[73] A. H. Gandomi and A. H. Alavi, "Krill herd: a new bio-inspired optimization algorithm," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 12, pp. 4831–4845, 2012.

[74] A. H. Gandomi, G. J. Yun, X.-S. Yang, and S. Talatahari, "Chaos-enhanced accelerated particle swarm optimization," *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 2, pp. 327–340, 2013.

[75] M. J. Mahmoodabadi, Z. Salahshoor Mottaghi, and A. Bagheri, "HEPSO: High exploration particle swarm optimization," *Information Sciences*, vol. 273, pp. 101–111, 2014.

[76] M. Randall, "Differential evolution for a constrained combinatorial optimisation problem," *International Journal of Metaheuristics*, vol. 1, no. 4, pp. 279–297, 2011.

[77] C.-T. Su and C.-L. Chiang, "An incorporated algorithm for combined heat and power economic dispatch," *Electric Power Systems Research*, vol. 69, no. 2, pp. 187–195, 2004.

[78] X.-S. Yang and S. Deb, "Engineering optimisation by cuckoo search," *Int. J. of Mathematical Modelling and Numerical Optimisation*, vol. 1, no. 4, pp. 330–343, 2010.

[79] S. Walton, O. Hassan, K. Morgan, and M. Brown, "Modified cuckoo search: A new gradient free optimisation algorithm," *Chaos, Solitons & Fractals*, vol. 44, no. 9, pp. 710–718, 2011.

[80] W. Wei, J. Wang, and M. Tao, "Constrained differential evolution with multiobjective sorting mutation operators for constrained optimization," *Applied Soft Computing*, vol. 33, no. 0, pp. 207 – 222, 2015.

[81] Z. Zhan, J. Zhang, Y. Li, and H. Chung, "Adaptive particle swarm optimization," *IEEE Trans. Syst. Man Cybern. B*, vol. 39, no. 6, pp. 1362–1381, 2009.

[82] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[83] Y. Shi and R. C. Eberhart, "Fuzzy adaptive particle swarm optimization," in *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, vol. 1, pp. 101–106, IEEE, 2001.

[84] A. V. Kononova, D. W. Corne, P. De Wilde, V. Shneer, and F. Caraffini, "Structural bias in population-based algorithms," *Information Sciences*, vol. 298, pp. 468–490, 2015.

[85] K. Sörensen, "Metaheuristics–the metaphor exposed," *International Transactions in Operational Research*, vol. 22, no. 1, pp. 3–18, 2015.

[86] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The rprop algorithm," in *Neural Networks, 1993., IEEE International Conference on*, pp. 586–591, IEEE, 1993.

[87] A. Kotsialos, "Non-smooth optimization based on resilient backpropagation search for unconstrained and simply bounded problems," *Optimization Methods and Software*, vol. 28, no. 6, pp. 1282–1301, 2013.

[88] A. Kotsialos, "Nonlinear optimisation using directional step lengths based on rprop," *Optimization Letters*, vol. 8, no. 4, pp. 1401–1415, 2014.

[89] M. Bartholomew-Biggs, S. Brown, B. Christianson, and L. Dixon, "Automatic differentiation of algorithms," *Journal of Computational and Applied Mathematics*, vol. 124, no. 12, pp. 171 – 190, 2000. Numerical Analysis 2000. Vol. IV: Optimization and Nonlinear Equations.

[90] S. K. Godunov, "A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics," *Matematicheskii Sbornik*, vol. 89, no. 3, pp. 271–306, 1959.

[91] M. Hadiuzzaman and T. Z. Qiu, "Cell transmission model based variable speed limit control for freeways," *Canadian Journal of Civil Engineering*, vol. 40, no. 1, pp. 46–56, 2013.

[92] J. Frejo, E. Camacho, and R. Horowitz, "A parameter identification algorithm for the metanet model with a limited number of loop detectors," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pp. 6983–6988, Dec 2012.

[93] Dynamic Systems and Simulation Laboratory, Technical University of Crete, Chania, Crete, Greece, *METANET Documentation.*

[94] D. Goldberg, *Genetic algorithms in search, optimization, and machine learning.* Addison-wesley, 1989.

[95] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. on Neural Netw.*, vol. 4, pp. 1942–1948, IEEE, 1995.

[96] Z. Michalewicz, *Genetic algorithms + data structures = evolution programs.* springer, 1998.

[97] M. Golub, "An implementation of binary and floating point chromosome representation in genetic algorithm," *Proc. of the 18th Int. Conf. on Inform. Technol. Interfaces*, pp. 417–422, 1996.

[98] C. Z. Janikow and Z. Michalewicz, "An experimental comparison of binary and floating point representations in genetic algorithms," in *Proc. of the fourth Int. Conf. on Genetic Algorithms*, vol. 31, p. 36, California, USA, 1991.

[99] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* MIT press, 1992.

[100] D. Goldberg, "Real-coded genetic algorithms, virtual alphabets, and blocking," *Complex Syst.*, vol. 5, pp. 129–167, 1991.

[101] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micromach. Human Sci.*, pp. 39–43, IEEE, 1995.

[102] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proc. IEEE World Congr. on Comput. Intell.*, pp. 69–73, IEEE, 1998.

[103] M. Zambrano-Bigiarini, M. Clerc, and R. Rojas, "Standard particle swarm optimisation 2011 at CEC-2013: A baseline for future PSO improvements," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pp. 2337–2344, IEEE, 2013.

[104] X. Hu and R. C. Eberhart, "Adaptive particle swarm optimization: detection and response to dynamic systems," in *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, vol. 2, pp. 1666–1670, IEEE, 2002.

[105] X.-F. Xie, W.-J. Zhang, and Z.-L. Yang, "Adaptive particle swarm optimization on individual level," in *Signal Processing, 2002 6th International Conference on*, vol. 2, pp. 1215–1218, IEEE, 2002.

[106] A. Chander, A. Chatterjee, and P. Siarry, "A new social and momentum component adaptive pso algorithm for image segmentation," *Expert Systems with Applications*, vol. 38, no. 5, pp. 4998–5004, 2011.

[107] W.-D. Chang, "A multi-crossover genetic approach to multivariable pid controllers tuning," *Expert Systems with Applications*, vol. 33, no. 3, pp. 620–626, 2007.

[108] X.-S. Yang and S. Deb, "Cuckoo search via lévy flights," in *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, pp. 210–214, IEEE, 2009.

[109] A. D. May, *Traffic flow fundamentals.* Prentice-Hall, 1990.

[110] J. Han, J. W. Polak, J. Barria, and R. Krishnan, "On the estimation of space-mean-speed from inductive loop detector data," *Transportation planning and technology*, vol. 33, no. 1, pp. 91–104, 2010.

[111] Qt, "Qt — Cross-platform software development for embedded & desktop." `https://www.qt.io/`, June 2015.

[112] T. Williams, C. Kelley, and many others, "Gnuplot 4.6: an interactive plotting program." `http://www.gnuplot.info/`, April 2013.

[113] A. Walther and A. Griewank, "Getting started with ADOL-C," in *Combinatorial Scientific Computing* (U. Naumann and O. Schenk, eds.), ch. 7, pp. 181–202, Chapman-Hall CRC Computational Science, 2012.

[114] A. Grama, A. Gupta, G. Karypis, and V. Kumar, *Introduction to parallel computing.* Pearson Education, 2003.

[115] W. R. Stevens and S. A. Rago, *Advanced programming in the UNIX environment.* Addison-Wesley, 2013.

[116] Message Passing Interface Forum, "MPI: A message-passing interface standard version 3.1." `https://www.mpi-forum.org`, June 2015.

[117] IEEE and The Open Group, "IEEE Std 1003.1 (POSIX.1-2008)." `http://pubs.opengroup.org/onlinepubs/9699919799/nframe.html`, 2013.

[118] J. Kephart and D. Chess, "The vision of autonomic computing," *Computer*, vol. 36, pp. 41–50, Jan 2003.

[119] M. Warnier, M. Van Sinderen, and M. Brazier, "Adaptive knowledge representation for a self-managing home energy usage system," in *Proceedings of the Fourth International Workshop on Enterprise Systems and Technology (I-WEST), Athens*, pp. 132–141, 2010.

[120] R. Sterritt, “Autonomic networks: engineering the self-healing property,” *Engineering Applications of Artificial Intelligence*, vol. 17, no. 7, pp. 727–739, 2004.

[121] G. Cheliotis and C. Kenyon, “Autonomic economics,” in *E-Commerce, 2003. CEC 2003. IEEE International Conference on*, pp. 120–127, IEEE, 2003.

[122] W. Truszkowski, H. Hallock, C. Rouff, J. Karlin, J. Rash, M. Hinchey, and R. Sterritt, *Autonomous and autonomic systems: With applications to NASA intelligent spacecraft operations and exploration systems.* Springer Science & Business Media, 2009.

[123] U. Richter, M. Mnif, J. Branke, C. Müller-Schloer, and H. Schmeck, “Towards a generic observer/controller architecture for organic computing.,” *GI Jahrestagung (1)*, vol. 93, pp. 112–119, 2006.

[124] H. Etemadnia, K. Abdelghany, and S. Hariri, “Toward an autonomic architecture for real-time traffic network management,” *Journal of Intelligent Transportation Systems*, vol. 16, no. 2, pp. 45–59, 2012.

[125] I. Dusparic and V. Cahill, “Multi-policy optimization in decentralized autonomic systems,” in *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pp. 1203–1204, International Foundation for Autonomous Agents and Multiagent Systems, 2009.

[126] C. Guzmán, V. Alcázar, D. Prior, E. Onaindia, D. Borrajo, J. Fdez-Olivares, and E. Quintero, “Pelea: a domain-independent architecture for planning, execution and learning,” in *Proceedings of ICAPS*, vol. 12, pp. 38–45, 2012.

[127] A. Walther and A. Griewank, “Getting started with ADOL-C,” *Combinatorial Scientific Computing*, pp. 181–202, 2012.

[128] J. Kennedy and R. Mendes, “Population structure and particle swarm performance,” in *Computational Intelligence, Proceedings of the World on Congress on*, vol. 2, pp. 1671–1676, IEEE, 2002.

[129] A. Engelbrecht, “Particle Swarm Optimization: Global Best or Local Best?,” in *Computational Intelligence and 11th Brazilian Congress on Computational Intelligence (BRICS-CCI CBIC), 2013 BRICS Congress on*, pp. 124–135, Sept 2013.

[130] E. Burke, G. Kendall, J. Newall, E. Hart, P. Ross, and S. Schulenburg, "Hyper-heuristics: An emerging direction in modern search technology," in *Handbook of metaheuristics*, pp. 457–474, Springer, 2003.

[131] M. Papageorgiou and A. Kotsialos, "Freeway ramp metering: An overview," in *Intelligent Transportation Systems, 2000. Proceedings. 2000 IEEE*, pp. 228–239, IEEE, 2000.

[132] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang, "Review of road traffic control strategies," *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2043–2067, 2003.