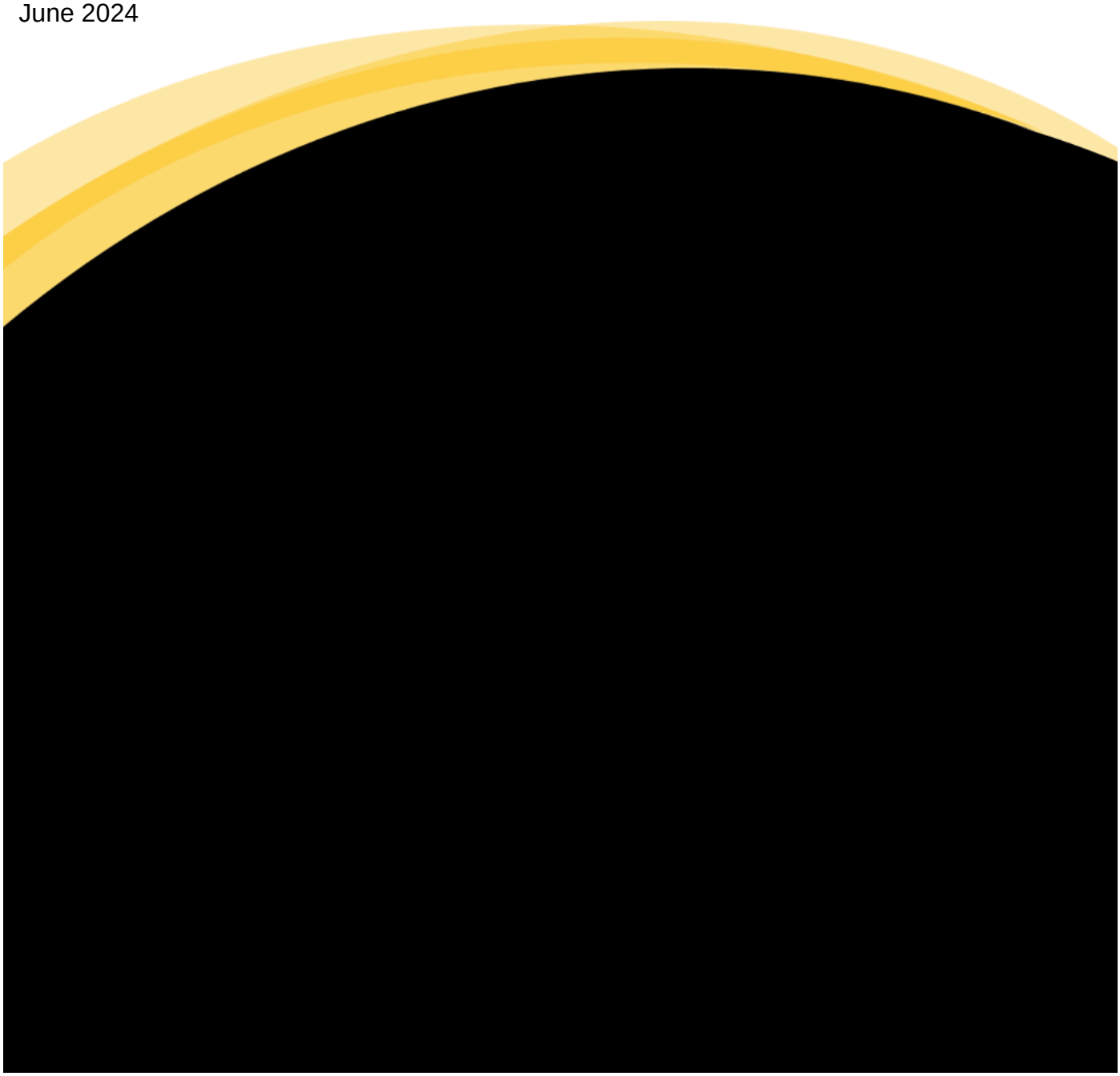
Transport Predictive Solution – Stage 2 – R&D – WA Node: AI-assisted Model Calibration for Real-time Traffic Simulation

June 2024



Transport Predictive Solution – Stage 2 – R&D – WA Node: AI-assisted Model Calibration for Real-time Traffic Simulation

Prepared by

Tom Lymburn, Liam Cummins, Thomas Stemler, and Chao Sun

Version control

Final

Project No

iMOVE Project 1-025

Acknowledgment

This research is funded by iMOVE CRC and supported by the Cooperative Research Centres program, an Australian Government initiative.

About PATREC

The Planning and Transport Research Centre (PATREC) is a collaboration between the Government of Western Australia and local universities, constituted to conduct collaborative, applied research and teaching in support of policy in the connected spaces of transport and land use planning. The collaborating parties are: The University of Western Australia, Curtin University, Edith Cowan University, Department of Transport, Main Roads Western Australia, Western Australian Planning Commission and the Western Australian Local Government Association.

Publisher

Planning and Transport Research Centre
The University of Western Australia (M087)
35 Stirling Highway, Crawley, WA 6009
+61 8 6488 3385
patrec@uwa.edu.au
https://patrec.org/

Executive Summary

Real time prediction of traffic is an important tool in the arsenal of transport authorities, enabling a proactive approach to network operations, such as stopping gridlocks before they appear and providing rapid assessment of different incident response measures prior to taking the action. This can be achieved by large-scale simulation models, such as the Perth Live Aimsun model. The aim of this project is to improve the accuracy of this model by developing methods for the automatic calibration of **the driver beheaviour and supply-side parameters**, e.g., reaction time and jam density, excluding the demand side inputs and parameters related to OD metrics and route choice.

The key outputs of the project are as follows:

1. **The development of methods for the offline calibration of the model.**
   The key challenge lies in the model's characteristics: it handles noisy, high-dimensional data, and requires significant computational resources. To address this, sensitivity analysis is performed to identify the most consequential parameters. This allows us to focus the calibration problem only on these key parameters, effectively reducing its complexity.

   Two cases are considered: estimating best parameters for typical pattern days (average weekdays, Saturdays, and Sundays), and estimating best parameters for individual days throughout the year.

   - **Typical day calibration:** we employ Bayesian optimisation, which is a technique well-suited for noisy simulation outputs. It efficiently utlises computational resources by reducing the number of simulation evaluations required.

   - **Individual day calibration:** even Bayesian optimisation is not efficient enough, so we develop a method to reuse pre-computed simulations for the concurrent optimisation of multiple days in the dataset. The resulting parameter values reveal patterns in driver behaviours across hourly, daily, and weekly timescales.

2. **Identification of patterns in both traffic volume and optimal model parameters.**
   Hierarchical clustering techniques reveal meaningful patterns, both from detector volume data and optimised parameters. From the detector data we group days into distinct categories of weekdays, Fridays, Saturdays, Sundays, and holiday adjacent days. At a more fine-grained level, we compare each 15-minute period throughout the year of available data. Clustering at this level reveals traffic patterns that are common across the different categories of days, though they occur at different times. Specifically, these clusters are overnight, early morning and evening, morning peak, middle of day, and afternoon peak.

   Applying the same clustering techniques to optimised parameter values reveals patterns in driver behaviour. In contrast to the detector data clusters, these results are largely independent of the dominant effects of varying demand, which is highlighted by the same clusters being observed in the morning and afternoon peaks. The patterns identified from the parameters are overnight and holiday, pre-peak/early peak, peak/late peak, post-peak, and Saturday. These patterns are similar to the 15-minute detector data clusters, though they are not an exact match.

   In order to validate that the optimised parameters lead to improved performance we run the model with the average parameter values for each cluster and compare the results with the default parameter values. There is a small improvement in average GEH in general, with notable improvements for the morning peak, holiday periods, and June-July. There are also some times where the performance is marginally worse, particularly

in the afternoons, though on average the performance improves. This improvement is not large, though this is not unexpected, as we expect the driver beheaviour and supply-side parameters to play a secondary role to the correct demand in getting accurate results.

The key contribution of this project is a systematic methodology for calibrating driver behavior and supply-side parameters, although the specific patterns found are limited to the Perth Aimsun Live model and 2019 data.

3. **Compare metrics for evaluating the quality of the model prediction.** Evaluation metrics play a crucial role in model calibration, as they quantify the model's goodness-of-fit. Several metrics are investigated, including traffic-specific metrics such as GEH and its derivatives, as well as statistics more common in machine learning such as root-mean-square error (RMSE), and mean-absolute-scaled error (MASE). Objective comparison of error metrics is difficult without a ground-truth notion of goodness. Measuring the ability of an optimisation algorithm to accurately recover known parameters with synthetic data is one approach, though it is not feasible for this project. Instead, we use a criterion that evaluates the degree to which the quality metrics separate simulations with different parameters, analogous to the concept of a signal-to-noise ratio. Based on this criterion RMSE is determined to be the the best quality metric, though the percentage of detector stations with GEH greater than 5 ($\%GEH > 5$) also performes well. Both of these metrics are scale-dependent, so it is important to consider the errors reported within the context of typical values throughout the weekly cycle.

Looking forward, there is the potential to extend these results to real-time optimisation via emulation of the theory-driven model by a simpler machine learning model, though further investigations are required to achieve this.

In summary, this project has successfully revealed patterns in driver beheaviour and supply-side parameters and how they can be used to improve the predictive performance of the Perth Live model.

Table of Contents

1. Introduction

1.1. Background

The iMOVE project 1-025, *Transport Predictive Solution – Stage 2 – R&D – WA Node: AI-assisted Model Calibration for Real-time Traffic Simulation*, was co-funded by the following organisations:

- iMOVE CRC and supported by the Cooperative Research Centres program, an Australian Government initiative.
- Aimsun Pty Ltd
- Main Roads Western Australia
- The University of Western Australia

The authors would like to thank all people involved, especially Mohammad Saifuzzaman and Ferran Torrent for their valuable inputs, and Alexandre Torday, Hannah Saunders, Scott Aitken, Emilie Alexandre, and Steve Atkinson for initiating and setting up the project.

1.2. Project Brief

In order to best implement controls and interventions in a traffic system, it is not sufficient to only monitor the current state of the network. Prediction of the traffic state is also required in order to better understand when interventions are necessary and what their consequences may be. This prediction can be done by large-scale simulation models, such as the Perth Live Aimsun model, which runs in real-time and predicts the traffic conditions over the coming hour. The aim of this project is to improve the accuracy of this model by improving the methods used to calibrate the driver beheaviour and supply-side parameters such as reaction time and jam density.

1.3. The Perth Live Model

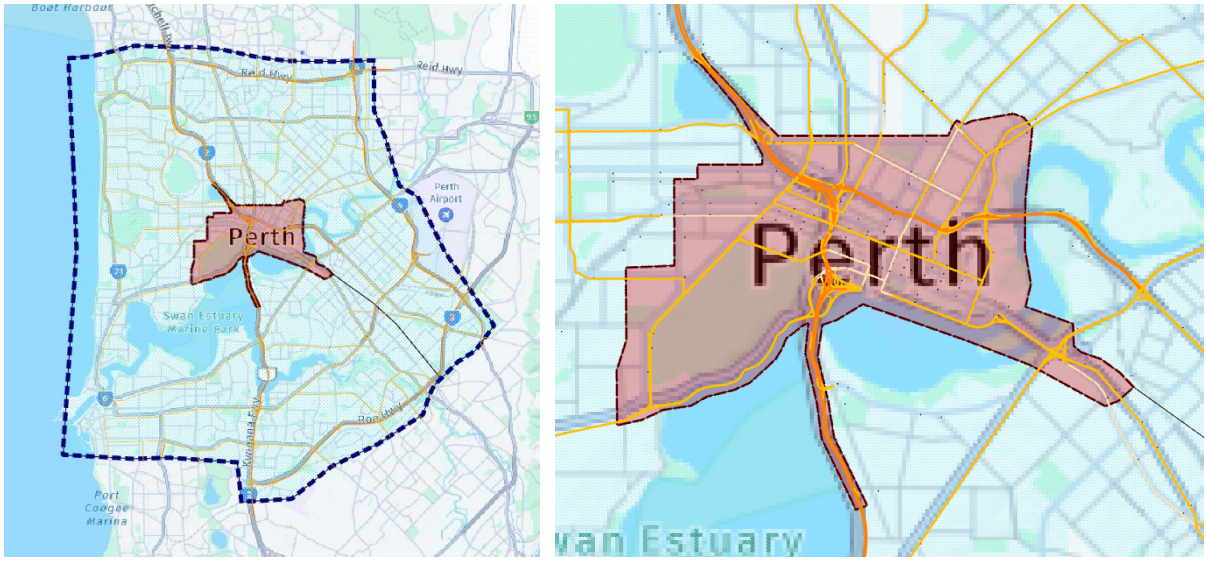


Figure 1: Macroscopic (left, blue) and mesoscopic (right, red) model areas.

The Perth Live model consists of a hybrid macroscopic-mesoscopic model, wherein the larger Perth region is modelled by a static macroscopic model with a smaller region around the CBD modelled by a mesoscopic discrete-event based simulation (see regions in Figure 1). Traffic in the macroscopic region is modelled at the aggregate level of vehicle flow according to the principle of user equilibrium, origin-destination (OD) demand, and travel times associated with each segment and intersection.

In the mesoscopic region the model is an agent-based simulation. Each vehicle is an agent and the movements of each agent is specified. However, due to the size of the region being modelled and the number of vehicles present, it would be computationally prohibitive to perform a microscopic simulation, in which the position, velocity, and acceleration of each vehicle is modelled in detail. Instead, Aimsun's mesoscopic simulation leverages discrete-event simulation to efficiently model individual vehicles. This approach focuses on key events, such as vehicles being generated at an origin centroid, entering and exiting road sections, signal phases changing, rather than simulating every minor movement. It reduces computational demands while still capturing the essential behaviour of traffic flow. These events are scheduled to occur at specific times based on the parameters of the model and are stored in a list. They are triggered in sequence and the list is dynamically updated in response to the changing traffic state. For example, if vehicle B follows vehicle A along a road, the event of vehicle B moving to the next road segment can only occur after vehicle A does so first, and therefore must come later in the list. This mechanism is an example of how updating the event list can result in dynamically occurring traffic phenomena, such as queuing and congestion. This is how the mesoscopic model is able to capture details of real traffic dynamics without performing the more computationally expensive kinematic calculations of microsimulation.

A high-level schematic of the mesoscopic simulation is shown in Figure 2, where the model inputs, processes, and outputs are shown with their respective relationships. The OD matrix contains unrealised demand for trips between two nodes in the network. Candidate routes exist from precomputed user equilibrium calculations. Vehicles are generated throughout the simulation period in order to realise the aggregate demand for each route. The movement of vehicles along their trips is governed by the mesoscopic simulation via the model parameters, resulting in a set of vehicle trajectories, which are summarised in aggregate measures such as volume and speed.

The Aimsun Live model operates online, continuously adapting its predictions based on incoming data from the real world.The final state of the previous simulation is carried forward as the initial state for the next. The OD matrix and parameters are adjusted based on comparisons of the model output and the observed traffic state. However, for this project, we run the model in an offline mode without this feedback, in order to learn the optimal parameters of the model in different conditions.

The model parameters considered in this project are listed in Table 1 along with the scope at which they are broadcasted to objects in the model. For example, global parameters such as reactionTime apply to all vehicles. Road-type parameters are applied separately for different road classes defined by their size, speed-limit, and road quality, e.g. Suburban-60.0-Avg and Motorway-80.0-Good. Vehicle-type parameters are applied separately to cars and trucks, as a simple system of vehicle classification. We also introduce a global version of each parameter where one does not already exist. These global parameters scale the corresponding parameter values of all scopes and are useful for simplifying the parameterisation of the model in cases where the distinction between parameter scopes is unimportant.

Some parameters, such as maxDesiredSpeed, are defined as a distribution rather than point values, in order to capture the heterogeneity of drivers. In these cases we consider the mean of the distribution in the optimisation algorithms, shift the entire distribution, and keep the range between the minimum and maximum values constant.

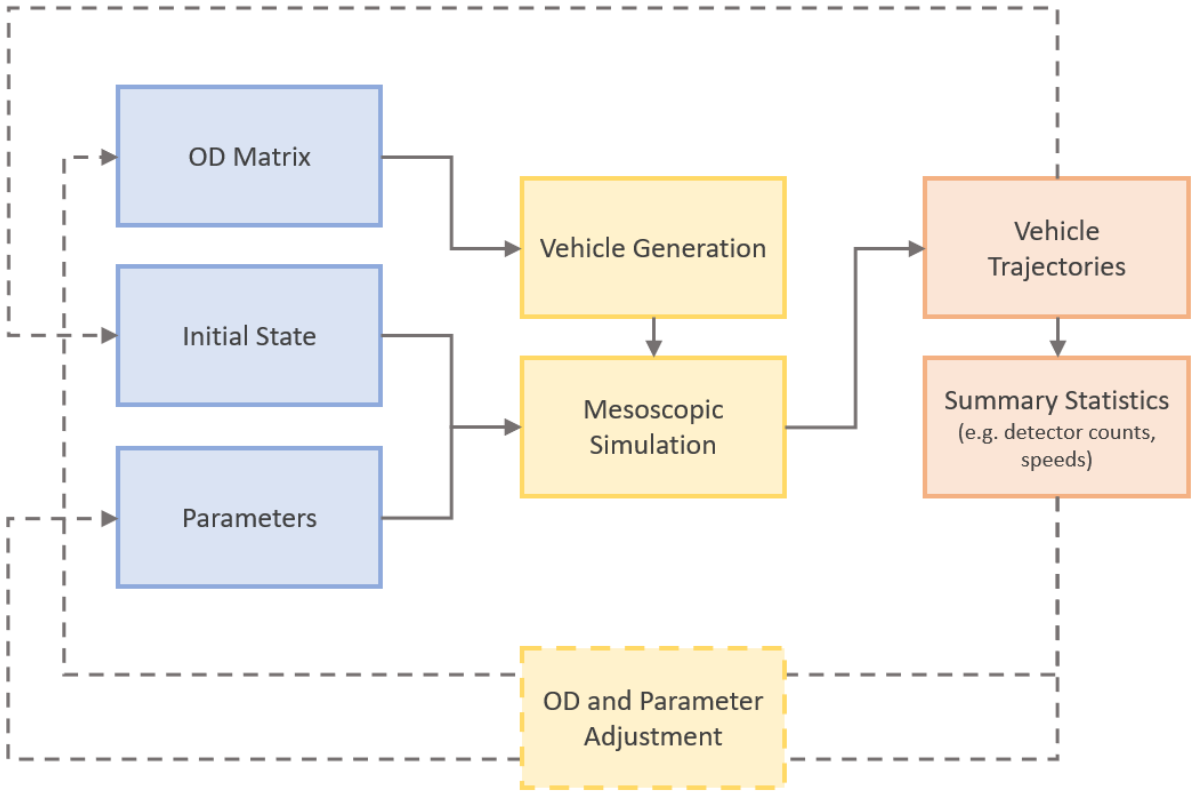


Figure 2: Aimsun model schematic. Model inputs are highlighted in blue, processes in yellow, and outputs in red. Dashed connections show the feedback when the model is run in real-time.

Table 1: Parameters of the mesoscopic model.

Parameter name	**Parameter scope**
reactionTime	Global
reactionTimeTrafficLight	Global
lookaheadDistanceVariability	Global
jamDensity	Road Type - Section
reactionTimeFactor	Road Type - Section
cooperationGap	Road Type - Section
mergingGap	Road Type - Section
sharedLanePenalised	Road Type - Section
slowLanePenalised	Road Type - Section
lookaheadDistance	Road Type - Turn
initialSafetyMargin	Road Type - Turn
finalSafetyMargin	Road Type - Turn
giveWayTimeFactor	Road Type - Turn
visibilityDistanceMainStream	Road Type - Turn
speedLimitAcceptance	Vehicle Type

Parameter name	Parameter scope
maxDesiredSpeed	Vehicle Type
clearance	Vehicle Type
maximumGiveWayTime	Vehicle Type

1.4. Project Tasks

This project consists of three main tasks.

1. **AI-assisted calibration module** – Develop and implement statistical learning techniques to estimate optimal driver behaviour and supply parameters in different conditions.

2. **Pattern refinement module** – Use clustering techniques to identify patterns in driver behaviour and supply via the optimised parameter values.

3. **Prediction confidence module** – Identify the best statistical measures for model confidence, in both training and evaluation contexts.

2. Data

This project uses loop detector data as observations of the real world traffic state, covering the period from 01/01/2019 to 29/11/2019. The data consists of volume observations at a 5-minute resolution. Because of the focus on mesoscopic model parameters, we only consider detectors within this region.

2.1. Data preprocessing

We aggregate the 5-minute detector counts into 15-minute intervals in order to match the control timescale of Aimsun Live. We also aggregate the individual detectors into detector stations, which correspond to turn movements. These steps reduce the size of the dataset with negligible reductions in fidelity.

Erroneous data anomalies may be caused by detector failure or unexpected events. We manually identify cases where there are significant errors in detector stations and remove these stations from the dataset (see examples in Figure 3). Automated methods exist for identifying anomalies, but they add unnecessary complication in a one-off case such as this, where manual inspection is sufficient.

There are many examples of missing data points throughout the dataset. Imputation is used to replace these missing values, rather than removing the detector station entirely and significantly reducing the amount of available data. The imputation is performed by replacing the missing entry with the corresponding value from similar traffic states throughout the rest of the dataset. Specifically, the average of the 5 most similar time-points is used, with similarity determined by the Euclidean distance with all of the available detector stations. If a detector station is missing a large amount of data (more than a day) it is completely removed. Because the number of remaining detector stations is significantly larger than the intrinsic dimension of the traffic state, there are large redundancies and there is little risk of removing important information.

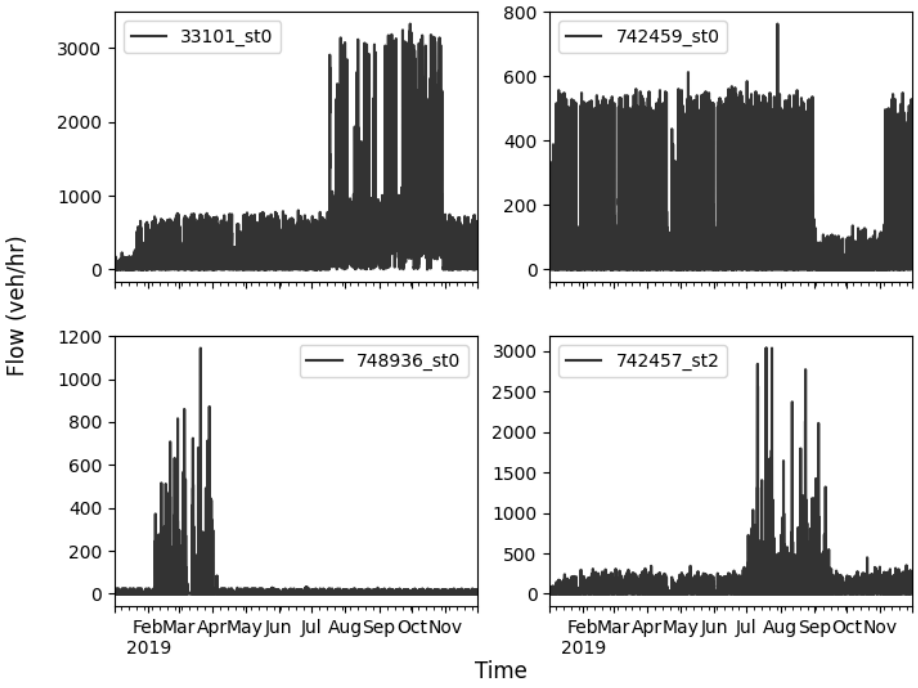


Figure 3: Examples of erroneous data anomalies.

3. AI-assisted calibration

The main aim of this project is to develop methods to automatically calibrate the Perth Live model using techniques from machine learning and AI. Specifically, we seek to optimise the parameters of the model by minimising an objective function (also known as a cost function) which measures the similarity between the model output and a target, which will typically be the observed traffic state. Due to high complexity, it is difficult to use knowledge of the model structure to inform the calibration, so this problem can be treated as black-box optimisation. The optimisation of black-box models can be broadly simplified to a trial-and-error approach, where a sequence of parameter values are tested and the resulting cost function value is recorded. Optimisation algorithms are used to direct this sequence of parameter values towards the optimum, which best matches the target according to the cost function.

Although it is difficult to use specific details of the model to inform calibration, the high-level characteristics are known and should to be considered when designing the calibration framework. Some of these characteristics for the Perth Live model are listed below:

1. **The model has a high-dimensional parameter space.** There are many parameters in the model, each of which can be segregated by road-type or vehicle-class. The size of the parameter space which needs to be searched through increases exponentially with the number of parameters. For this model there are approximately 250 possible parameters to consider.

2. **The output of the model is noisy.** Randomness arises in the model at various stages, notably including the vehicle generation process. This can be removed by fixing the random seed, but it reflects the stochastic nature of traffic in reality.

3. **Each evaluation of the model has a significant computational cost.** Although the mesoscopic simulation is relatively efficient, due to the size of the modelled area it still takes between 1 and 5 minutes to simulate 1 hour, depending on hardware.

These characteristics inform which optimisation algorithm is most suitable for this project. The high-dimensional nature of the model is mitigated by only considering the most influential parameters (see Section 3.2). We also discuss whether to reduce the effects of randomness by fixing the random seed (Section 3.3).

3.1. Optimisation algorithms

We considered several optimisation algorithms, including gradient descent, genetic algorithms, and Bayesian optimisation, which are summarised in Table 2. We also benchmark the algorithms by applying them to a test function

$$y = \left(A + \sum_{i=1}^{N} \left(x_i^2 - \frac{A\cos(2\pi x_i)}{N} \right) \right) \times \xi,$$

where N represents the number of parameters being optimised and ξ is a factor drawn from a lognormal distribution to introduce noise. If no noise is present, then $\xi = 1$ always. The trigonometric portion of the function introduces local optima, with the parameter $A = 1$ controlling their depth. The global optimum is $(\boldsymbol{x}, y) = (\mathbf{0}, 0)$. The progress of each optimisation algorithm for different cases is shown in Figure 4.

Table 2: Summary of optimisation algorithms.

Algorithm	Pros	Cons
Gradient descent – The direction in which the cost function is decreasing most rapidly is used to choose the next parameter set to evaluate, analogous to taking the steepest route down a hill.	• Uses knowledge of the shape of the cost function to inform next evaluation	• Calculating the gradient requires a simulation for each dimension of the parameter space • Gradient calculations are prone to error in the presence of noise
Genetic algorithms – A "population" of solutions is generated with the cost function associated to a particular parameter configuration representing the individual's "fitness". Simplified evolutionary pressure removes poor performing individuals from the population and high performing individuals combine to generate new solutions. Random mutation allows the population to explore unseen regions of parameter space.	• Capable of finding solutions to highly complex problems	• Many hyper-parameters control the evolution of the population, each of which must be chosen somehow. • Convergence to an optimal solution can be very slow.
Bayesian optimisation – A surrogate model estimates the relationship between the parameter values and the cost function, including uncertainty where the parameter space has not been sampled. Based on the surrogate cost function and estimated uncertainty, the parameter value with the largest expected improvement over the current best value is selected to be tested next. This is done to avoid wasting simulations where it is very unlikely to improve. Because of this, this method is used to optimise processes which are expensive to run. Examples include neural network hyperparameter optimisation, which is computationally intensive, and nuclear fusion experiments, which use expensive equipment.	• Assumptions about the presence of noise in the cost function can be built into the surrogate model. • Designed to efficiently explore parameter space with minimal evaluations of the cost function.	• Estimating the surrogate model imposes a computational overhead. For high-dimensional cases this can take more time than it takes to evaluate the cost function itself. • Relies on a surrogate model which may not be accurate.

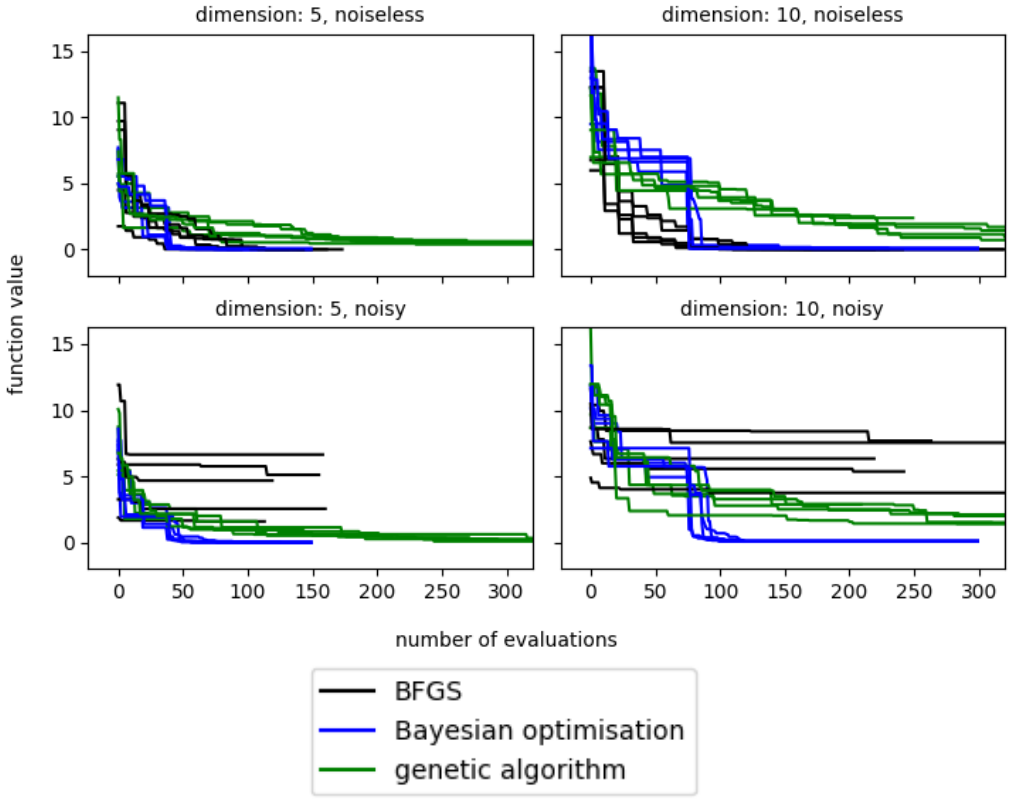


Figure 4: Convergence of optimisation algorithms on a test function showing the current minimum value. The true minimum is $y = 0$. Five examples of each algorithm are shown. BFGS is a gradient descent algorithm. Note that the computational overhead associated with each method is ignored in this plot, as we expect the evaluation of the Aimsun model to be significantly more time consuming.

In the noiseless, $N = 5$ case gradient descent and Bayesian optimisation are comparable, but gradient descent is more efficient in the higher dimension $N = 10$ case. However, gradient descent fails to converge to the global optima in all examples once noise is introduced. In all cases the genetic algorithm converges significantly slower than Bayesian optimisation, which is relatively unaffected by the presence of noise.

Based on the qualitative and quantitative comparisons of optimisation algorithms, gradient descent is the most suitable for the noiseless case, while Bayesian optimisation should be used in the presence of noise. However, the computational overhead associated with estimating the surrogate model for Bayesian optimisation is significant when the dimension of the input space is large (For this project it becomes significant for $N > 10$). It is not sensible to attempt to optimise all 250 possible parameter values, regardless of optimisation algorithm (see Section 3.2 for parameter selection). However, using $N \approx 10$-30 parameters could be reasonable and would slow Bayesian optimisation significantly. In order to mitigate this issue, the optimisation is split into multiple groups, which are optimised one-by-one, with the optimal parameter from one group carrying forward to the next.

A demonstration of Bayesian optimisation is shown in Figure 5.

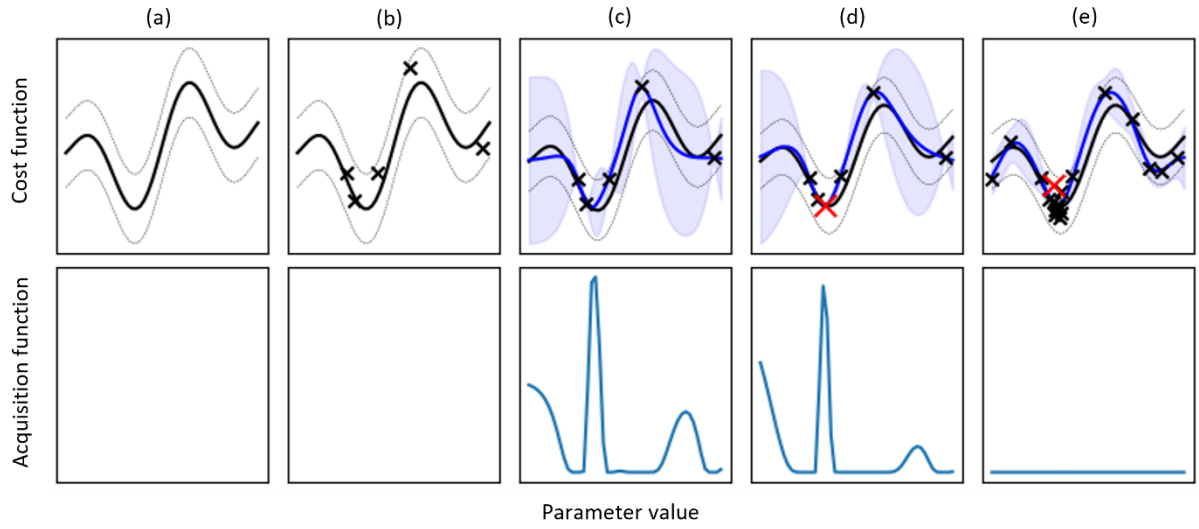


Figure 5: Bayesian optimisation demonstration. (a) The true noisy cost function, with the mean shown by the solid line and ± 2 standard deviations shown by the faint line. (b) The cost function is initially evaluated at random parameter values (crosses). (c) A Gaussian process surrogate model estimates the cost function distribution based on current observations (blue). The shaded region shows the uncertainty in the surrogate estimate of the cost function. The acquisition function plotted in the bottom panel is the expected improvement over the current best observation based on the surrogate model. The parameter value at the peak of the acquisition function is used for the next evaluation of the cost function (red cross in (d)). (e) Results after 20 function evaluations. Note that the function evaluations are concentrated near the minima while less are needed at larger function values.

3.2. Sensitivity analysis for efficient parameter selection

Due to the number of parameters available for calibration and the computational resources required to run the model, it is not feasible to include all parameters in the optimisation process. Instead, we focus on a manageable subset of the available calibration parameters that accounts for the majority of the variability in the model output. This subset can be optimised while the remaining parameters are set to a default value.

In order to identify which parameters are most influential on the outcome of the model we perform a systematic sensitivity analysis. One-by-one each parameter is varied in 5% increments from 50% below to 50% above the default value, while all other parameters are fixed at their default value. The default value for the parameters “cooperationGap” and “mergingGap” is 0 seconds, so the range of 0 – 10 seconds is used instead of the ±50% bounds. The check-box parameters “sharedLanePenalised” and “slowLanePenalised” are tested for both cases.

The sensitivity analysis is performed for both peak (07:30-08:30am) and off-peak (10:00-11:00am) conditions. In each case the average GEH value of detector stations in the mesoscopic simulation area is used to quantify the effect of each parameter on the model output. The variability is measured as the range of cost function values observed across the adjusted parameter values. The percentage of GEH values above 5 and above 10 were also tested with similar results. The random seed is fixed for this analysis, as we aim to isolate the model’s dependence on each parameter from random variation. All parameters in Table 1 are considered in this analysis, including global and disaggregated values.

The results for global parameters are shown in Figure 6 and Figure 7. There is greater sensitivity in the peak period, which is expected as the system is closer to capacity. In both cases a handful of parameters are responsible for the majority of variability in the output of the model, particularly jamDensity, reactionTime, and speedLimitAcceptance.

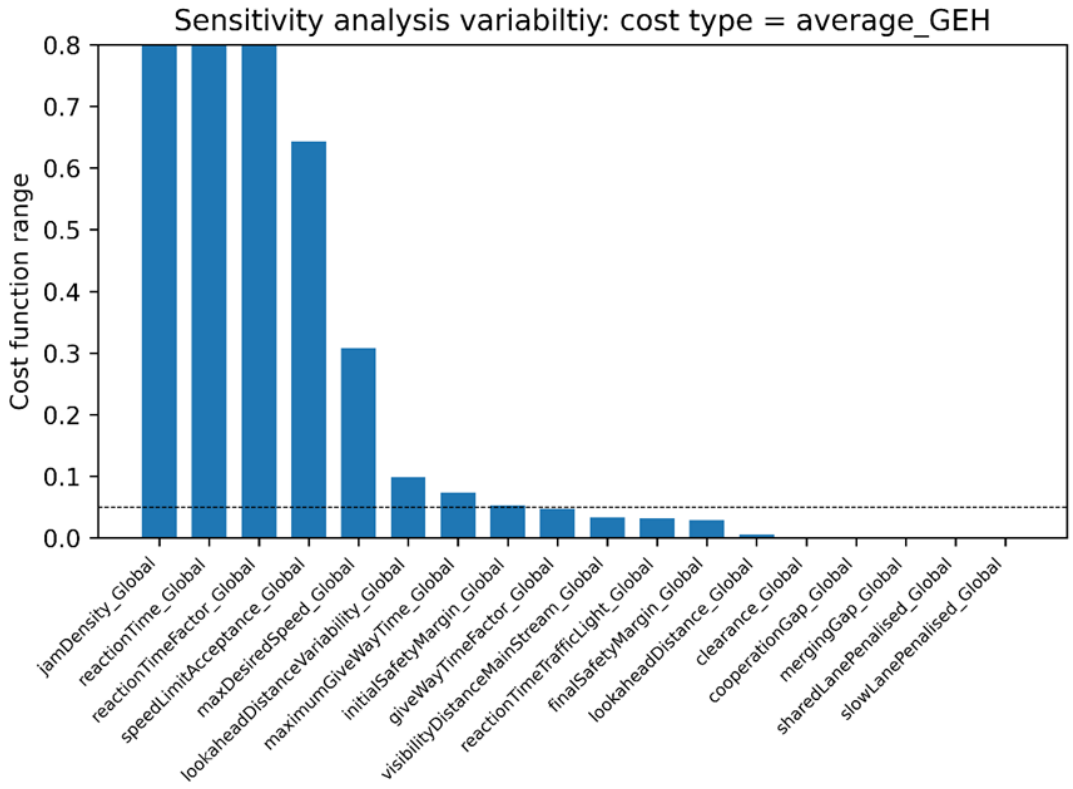


Figure 6: Sensitivity of global parameters during peak conditions (07:30-08:30am).

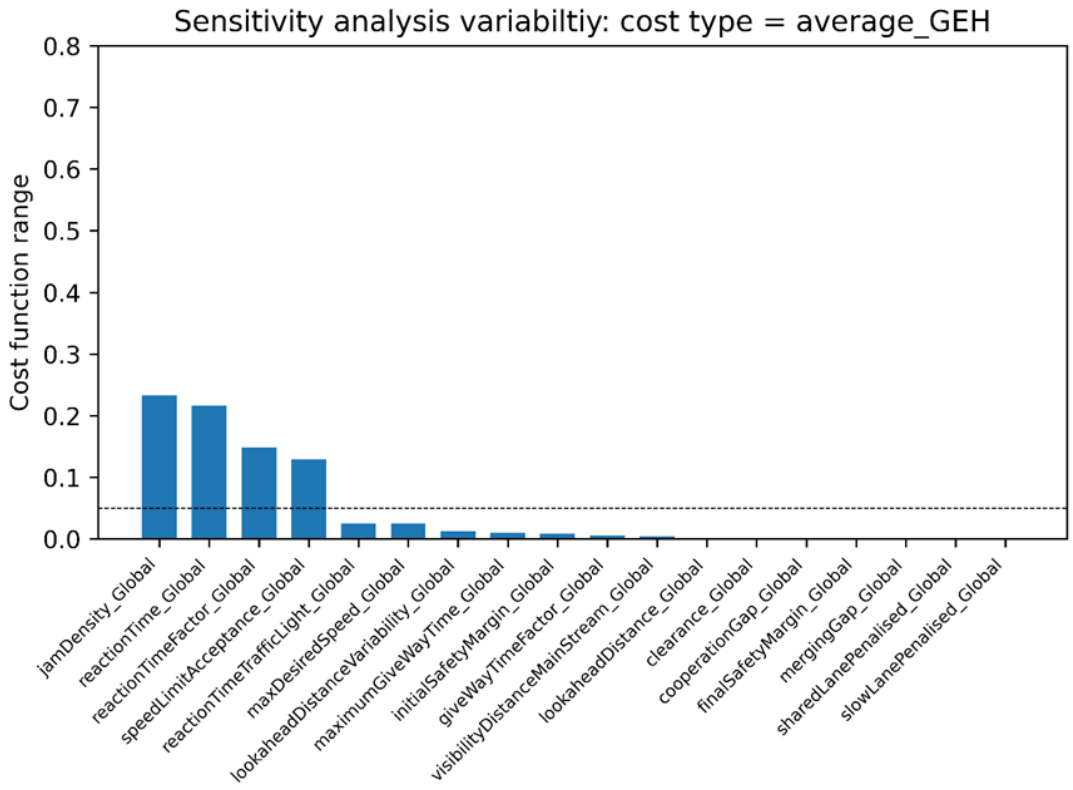


Figure 7: Sensitivity of global parameters during off-peak conditions (10:00-11:00am).

The full analysis including disaggregated parameters is shown in Figure 8 and Figure 9. Although the labels are difficult to read due to the large number of parameters tested, we see again that a small proportion of parameters are responsible for the majority of variation in the output of the model and that the model is more sensitive in the peak period.

Archetypical exmples of the sensitivity to individual parameters are shown in Figure 10. Some parameters have no effect (Figure 10(a)) and can be ignored. This may be because the parameter governs behaviours that do not occur in the model, but they have been included here for certainty and completeness. Other parameters have minor fluctuations with no clear trend or pattern (Figure 10(b)). The fluctuations may be the result of the specific random seed used during this analysis. If there is no pattern above these random fluctuations the parameter will not be useful for optimisation and should be ignored. Some parameters have an approximately flat dependence, except for a sharp spike at an extreme parameter value (Figure 10(c)). For example, extremely low jamDensity values for some road-types can cause extreme congestion, but above a critical value have little impact on the model. There is no need to include parameters like this because they have no influence around reasonable parameters. Finally, some parameters have a significant impact on the function output at reasonable parameter values (Figure 10(d)) and should be included in the optimisation.

Parameter selection is performed by first selecting all parameters with a cost function range above an arbitrary threshold of $\Delta GEH = 0.05$. We then manually inspect each parameter and remove those that do not have significant dependence at reasonable parameter values. Following the sensitivity analysis and manual inspection, the parameters in Table 3 were chosen for optimisation.

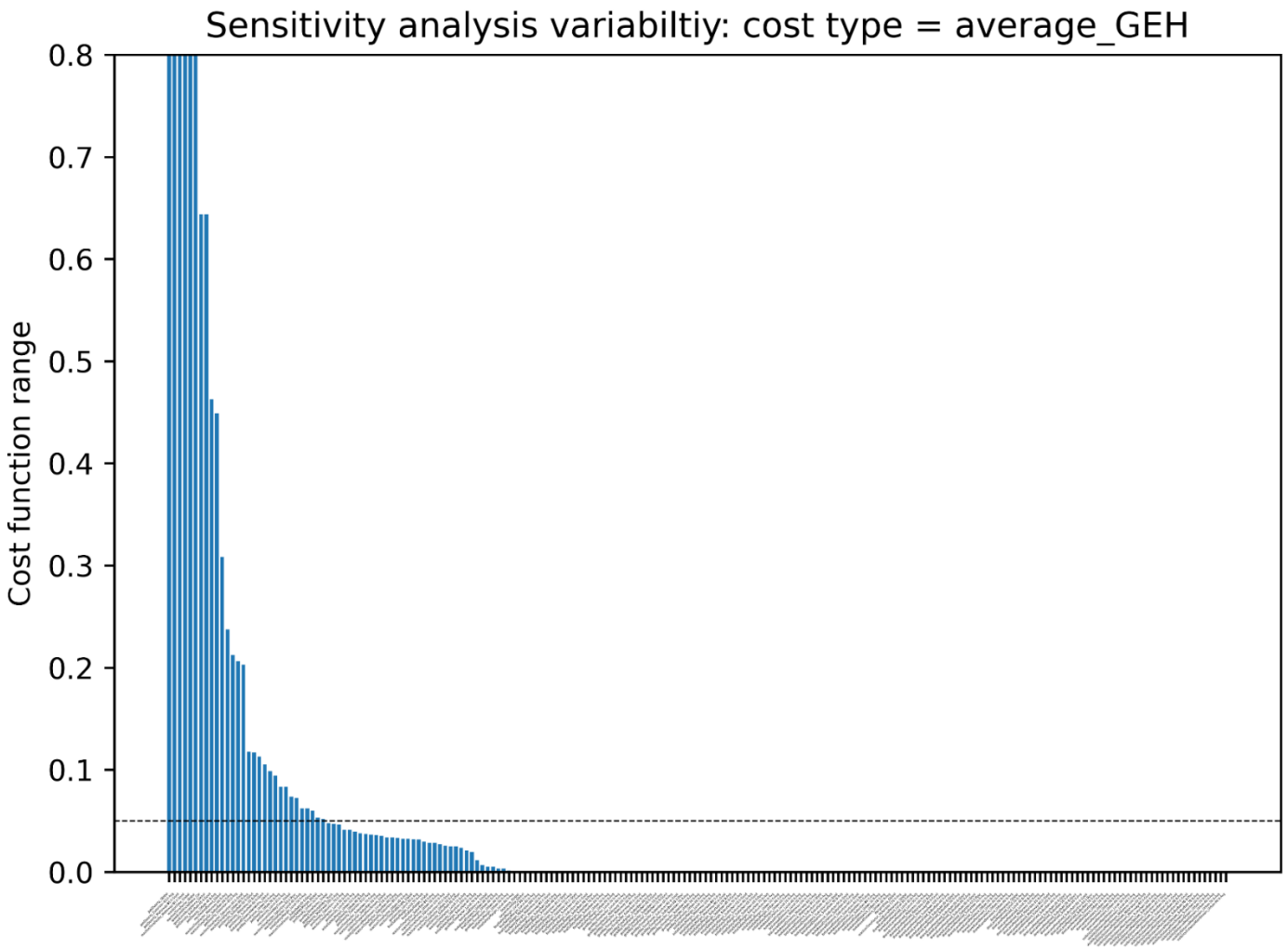


Figure 8: Sensitivity of all parameters during peak conditions (07:30-08:30am).

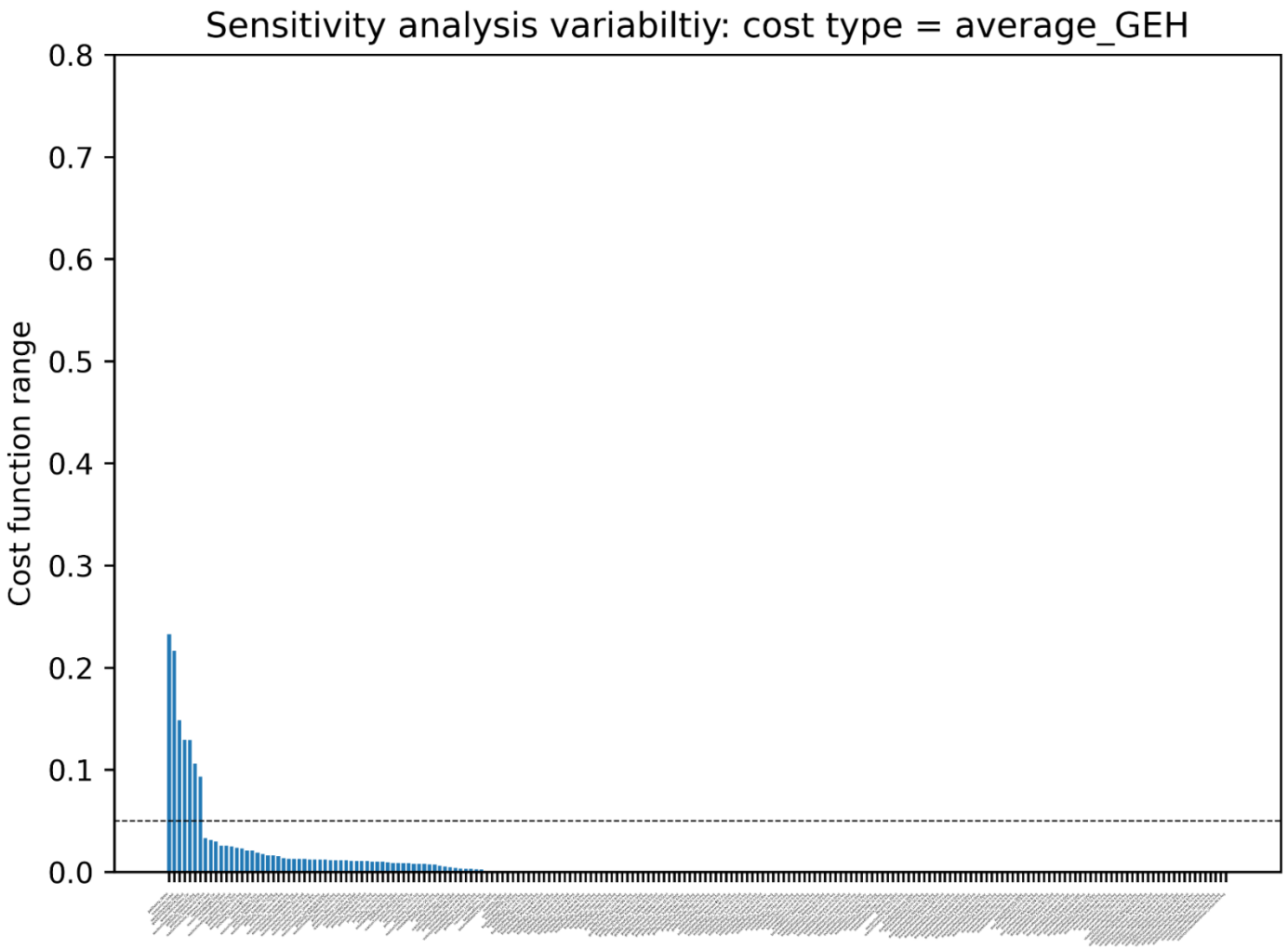


Figure 9: Sensitivity of all parameters during off-peak conditions (10:00-11:00am).

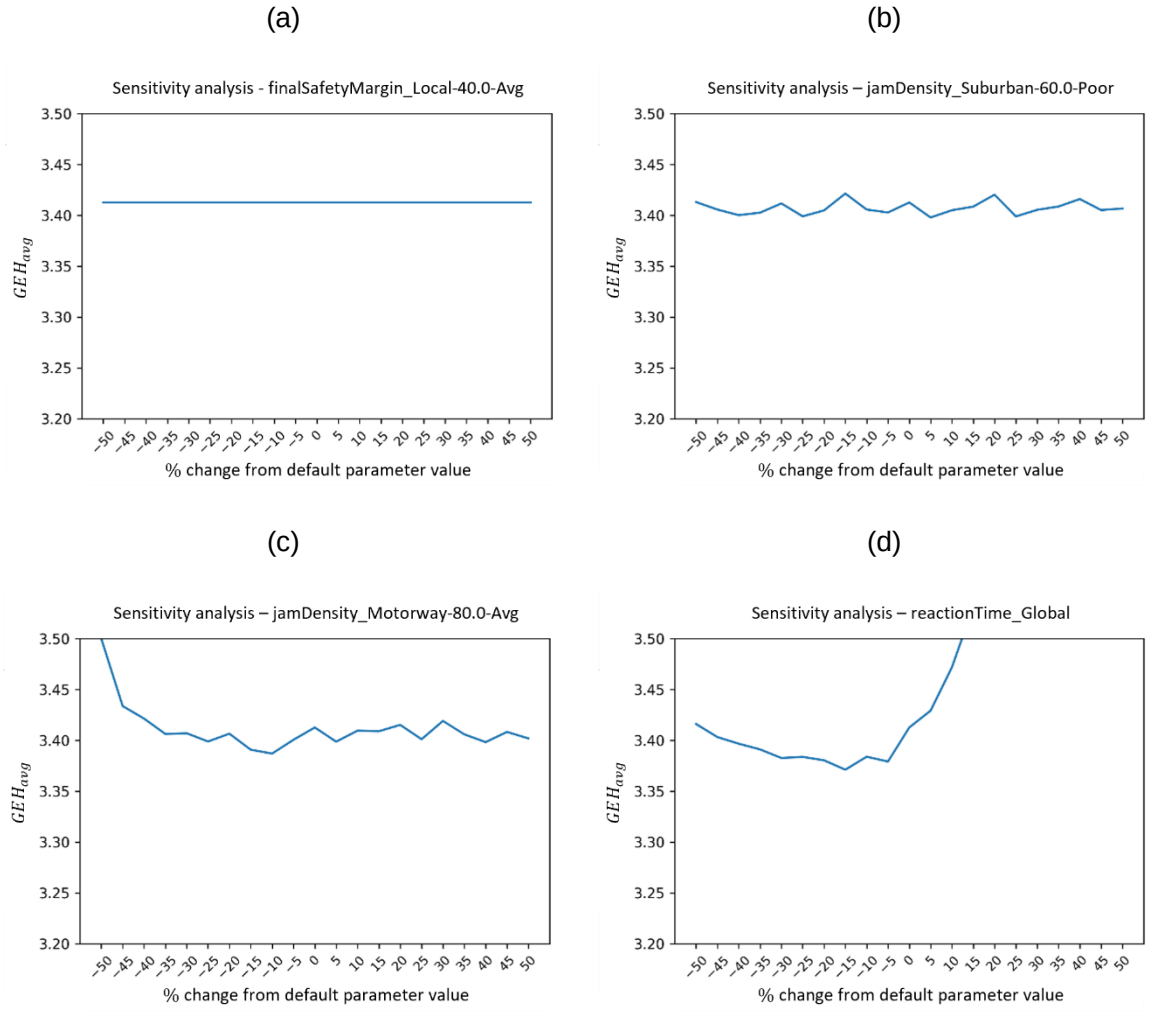


Figure 10: Examples of cost function dependence on parameters with (a) no effect on the cost function, (b) an approximately flat dependence except for small fluctuations, (c) an approximately flat dependence except for a sharp increase at an extreme value, and (d) a significant impact on the cost function at reasonable parameter values.

Table 3: Parameters selected for optimisation.

Parameter Label	Default value
reactionTime_Global	1.2 s
reactionTimeTrafficLight_Global	1.6 s
reactionTimeFactor_Local-50.0-Avg	1.0
reactionTimeFactor_Suburban-60.0-Avg	1.0
reactionTimeFactor_Suburban-60.0-Good	1.0
reactionTimeFactor_Motorway-80.0-Avg	1.0
reactionTimeFactor_Motorway-80.0-Good	1.0
jamDensity_Global	150 veh/km
jamDensity_Local-50.0-Avg	150 veh/km
jamDensity_Suburban-60.0-Avg	150 veh/km
jamDensity_Motorway-80.0-Good	150 veh/km
speedLimitAcceptance_Car	1.1
speedLimitAcceptance_Truck	1.05
maxDesiredSpeed_Car	110 km/hr
maxDesiredSpeed_Truck	100 km/hr

3.3. Stochastic effects

Randomness arises in Aimsun due to several aspects of the model that are governed by stochastic processes, such as the generation of individual vehicles from aggregate demand in the OD matrix, path assignment, individual realisations of supply parameters governed by distributions, etc. This reflects the random nature of the reality being modelled. The effect of this randomness can be seen in the output of the model, as well as the cost function values. Randomness can be reduced by fixing the random seed, which is commonly done by engineers when testing the impact of changing design variables.

In the context of this project, fixing the random seed makes optimisation easier by allowing gradient descent to be used. However, the optimisation result is conditioned on the specific realisations of the stochastic processes resulting from the fixed random seed, which are not knowable in real-time. If the optimal parameters found across multiple fixed random seeds are consistent, then it is sufficient to use the fixed random seed. We test this with a single variable optimisation across 5 separate fixed random seeds (Figure 11). The results range between 1.0 and 1.4, indicating that the parameter reactionTime_Global is exploited to compensate for the specific realisations of the stochastic processes in the model, such that the model output best matches the target observation. This is a form of overfitting, and should be avoided. Therefore we allow the random seed to vary across simulations and deal with noise in the optimisation process. Another option involves performing multiple optimisation runs with distinct, fixed random seeds. The optimal parameters from each run are then averaged. However, because Bayesian optimisation intrinsically accounts for noise in the cost function, this will not have any advantage.

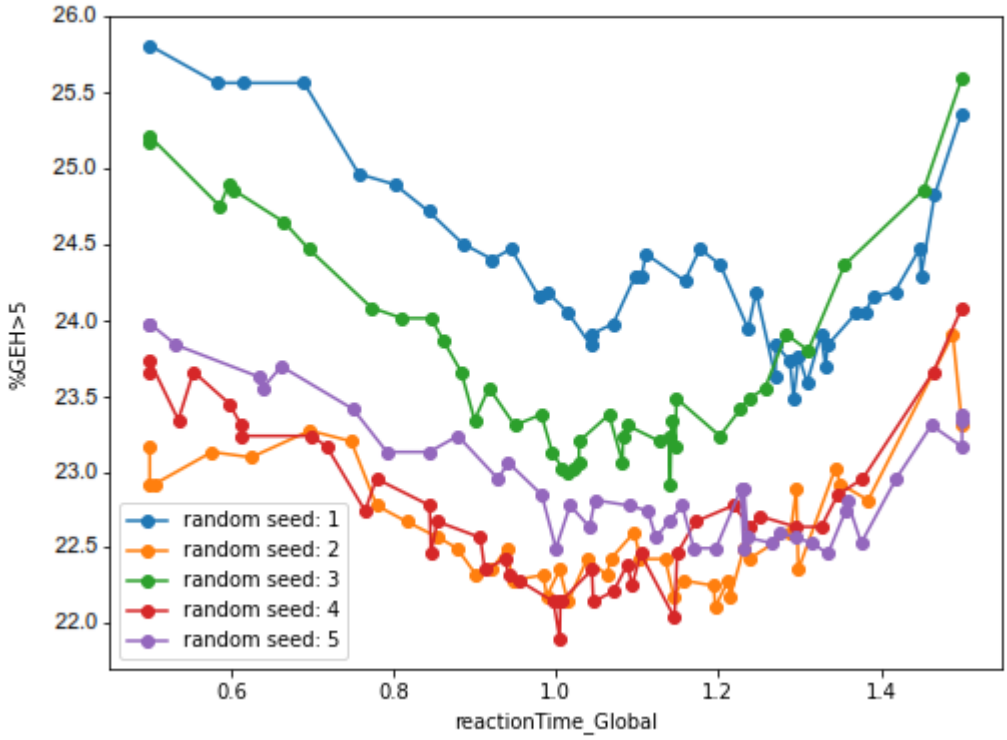


Figure 11: Single variable optimisation with fixed random seeds.

3.4. Optimisation approaches

We consider two different cases for performing optimisation. One in which the target for calibration is a hypothetical typical day, formed by taking the average detector volumes across all days belonging to each pattern (weekdays, Saturdays, and Sundays). For the second case the individual days are considered separately, since we wish to learn the optimal parameters of the model in different conditions.

In both cases we run the simulation for periods of 1 hour covering the times between 5am and 9pm, totalling 16 periods. The overnight conditions were ignored due in order to maximise the use of computational resources.

For these experiments we use average GEH as the cost function for optimisation, though we also tested the percentage of detector stations with $GEH > 5$, which led to similar results. We will discuss cost functions in greater detail in Section 5. The GEH statistic is commonly used in traffic applications and aims to balance absolute and relative errors,

$$GEH = \sqrt{\frac{(m-o)^2}{\frac{1}{2}(m+o)}}.$$

3.4.1. Optimisation with respect to typical days

When optimising with respect to typical days, we run the model with the unadjusted pattern OD matrices as the demand and use the average detector counts across all days corresponding to that pattern as the target.

We apply Bayesian optimisation with a non-fixed random seed for 1000 iterations per parameter group (Table 4), yielding a set of parameter vectors and their respective cost function values. However, due to the presence of noise in the model output (and therefore in the cost function), simply selecting the parameter vector associated with the lowest cost function value is insufficient to determine the optima, as a lower value may only be the result of random fluctuations. Instead, we need to find the parameter vector which results in the lowest expected cost, a process referred to as *identification* in optimisation. The Gaussian process surrogate model calculated during Bayesian optimisation estimates the mean cost value as a function of the parameters, so can be used for this purpose. The optimum of the surrogate model corresponds to the lowest expected cost, so is returned as the optimal parameter vector.

In addition to using the surrogate model for identification, we also add a regularising penalty for deviations from the default parameters. The penalty term is

$$\lambda\left(\frac{\gamma-\gamma_{\text{default}}}{\gamma_{\max}-\gamma_{\min}}\right)^2$$

for each parameter γ, with regularisation parameter $\lambda = 0.5$ controlling the severity of the penalty. This term is added to the average GEH cost function at the point of identification.

Table 4: Parameter groups for Bayesian optimisation. The results from group 1 are carried forward when optimising group 2.

Parameter group 1	Parameter group 2
reactionTime_Global (s)	jamDensity_Motorway-80.0-Good (veh/km)
reactionTimeTrafficLight_Global (s)	jamDensity_Local-50.0-Avg (veh/km)
jamDensity_Global	jamDensity_Suburban-60.0-Avg (veh/km)
speedLimitAcceptance_Car	reactionTimeFactor_Motorway-80.0-Avg
speedLimitAcceptance_Truck	reactionTimeFactor_Motorway-80.0-Good
maxDesiredSpeed_Car (km/hr)	reactionTimeFactor_Local-50.0-Avg
maxDesiredSpeed_Truck (km/hr)	reactionTimeFactor_Suburban-60.0-Avg
	reactionTimeFactor_Suburban-60.0-Good

3.4.2. Optimisation with respect to individual days

In order to capture the impact of different conditions on driver behaviour, as described by the parameter values, we also aim to optimise the model with respect to individual days in the dataset. The most natural approach to this problem is to perform demand adjustment as if the model were live, using the historical volume observations. These volume observations would also be used as the target for the optimisation algorithm, applying the same procedure as in the typical day case. However, there is a significant increase in the computational cost associated with this approach.

Suppose 2000 simulations are required to optimise the parameters for each 1-hour period, and that each simulation takes 2 minutes on average. This equates to 192,000 computer-minutes for all 3 patterns in the previous approach. Spread across 10 concurrent simulations this would take approximately 2 weeks, which is a significant but manageable amount of time. However, if this is

scaled from 3 typical days to 330 individual days, the time required increases to over 4 years, not including the time associated with performing the demand adjustment. This is obviously not feasible, so a different approach is needed.

In order to reduce the time required to optimise with respect to each day in the dataset, we use the unadjusted base OD matrices matrices corresponding to precomputed demand patterns (weekday, Saturday, Sunday), instead of performing demand adjustment. This means that the same demand inputs are used for all days belonging to the same demand pattern, so a single simulation can be reused for multiple days. For example, a simulation with weekday demand could be used for both Wednesday 13/06/2019 and Thursday 14/06/2019. Because simulation results are being compared with multiple observations, it no longer makes sense to use an optimisation algorithm to choose which parameters to sample, since the location of the optima will be different for different days.[1] Instead, we randomly sample the space of possible parameter values, with the loss of efficiency due to not using an optimisation algorithm more than compensated by reusing simulation outputs for multiple days.

Sampling the parameter space results in a set of simulation outputs corresponding to each parameter vector. Each simulation output is compared separately against the real world detector counts for each day matching the demand pattern, resulting in a sample of the cost function for each day. These values are then used to identify the optimal parameters. One way to perform this identification is to construct a surrogate model, as done in Bayesian optimisation. Alternatively, we can avoid introducing unnecessary models by averaging the parameter vectors of the k lowest cost functions.

This approach to identifying the optima not only has the advantage of being quick, but also has a regularising effect with respect to the distribution from which the parameter values are sampled, which is analogous to the prior in Bayesian statistics. Consider a parameter that is influential on the simulation outcome. Because the average of the cost function depends more significantly on this parameter value than the scale of the random fluctuations, the k lowest cost values will be concentrated in a small region. On the other hand, a parameter that is not influential on the simulation outcome will have its k lowest cost values spread across the range of parameter values, since random fluctuations dominate the functional dependence. Taking the average will converge towards the centre of the distribution of parameter values as k increases, meaning that overoptimising to the random fluctuations in the cost function is prevented.

Since the centre of the distribution is approached when the dependence on a parameter is insignificant, it is important to align the distribution with the default value that we wish to regularise towards. The parameter distributions used for each parameter are shown in Table 5. For parameters that behave like multiplicative factors (e.g. jamDensity_Global multiplies all other jamDensity parameters) a log-uniform distribution is used. In these cases the geometric mean is used when calculating averages. In all other cases a uniform distribution and the arithmetic mean is used.

A schematic of the optimisation procedure is shown in Figure 12.

[1] It is possible to construct an aggregate cost function which accounts for the errors across all days being compared, which could be used to direct the Bayesian optimisation algorithm, but this would require testing and probably only result in minor improvements.

Table 5: Parameter distributions.

Parameter label	Min value	Max value	Default value	Distribution
reactionTime_Global (s)	0.2	2.2	1.2	uniform
reactionTimeTrafficLight_Global (s)	0.6	2.6	1.6	uniform
jamDensity_Global	0.5	2	1	log-uniform
jamDensity_Motorway-80.0-Good (veh/km)	50	250	150	uniform
jamDensity_Local-50.0-Avg (veh/km)	50	250	150	uniform
jamDensity_Suburban-60.0-Avg (veh/km)	50	250	150	uniform
reactionTimeFactor_Motorway-80.0-Avg	0.5	2	1	log-uniform
reactionTimeFactor_Motorway-80.0-Good	0.5	2	1	log-uniform
reactionTimeFactor_Local-50.0-Avg	0.5	2	1	log-uniform
reactionTimeFactor_Suburban-60.0-Avg	0.5	2	1	log-uniform
reactionTimeFactor_Suburban-60.0-Good	0.5	2	1	log-uniform
speedLimitAcceptance_Car	0.6	1.6	1.1	uniform
speedLimitAcceptance_Truck	0.55	1.55	1.05	uniform
maxDesiredSpeed_Car (km/hr)	60	160	110	uniform
maxDesiredSpeed_Truck (km/hr)	50	150	100	uniform

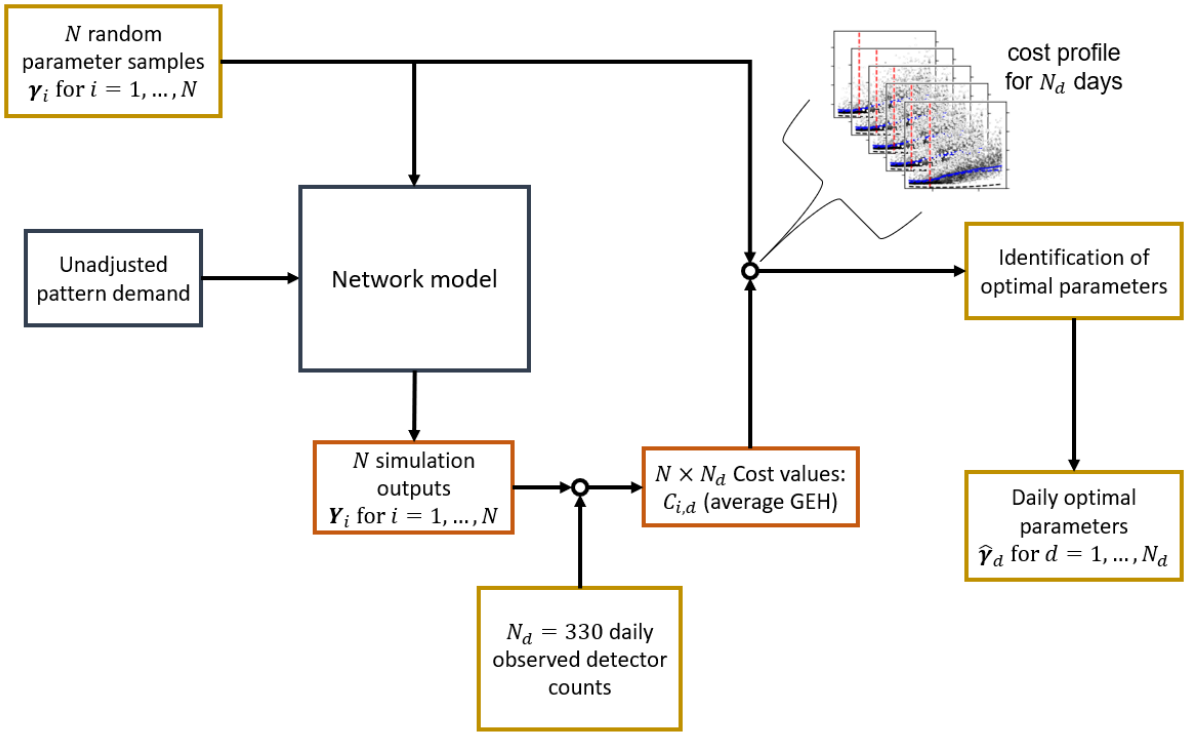


Figure 12: Schematic of optimisation procedure for individual days with unadjusted demand. The model is run with N random parameter vectors and undadjusted demand, resulting in N simulation outputs. Each output with the corresponding demand is compared against each real-world observation in the dataset, resulting in $N \times N_d$ cost values. The combination of these values with their respective parameter vectors results in a sample of the cost function, which is used for identifying the daily optimal parameters.

3.5. Optimisation results

3.5.1. Optimisation with respect to typical days

We perform optimisation with respect to the typical day for the weekday pattern for the hours of 5am-9pm. The results are shown in Figure 13. Clear patterns emerge for some parameters, including reactionTime_Global and jamDensity_Global. Smaller trends are also present for the local versions of these parameters.

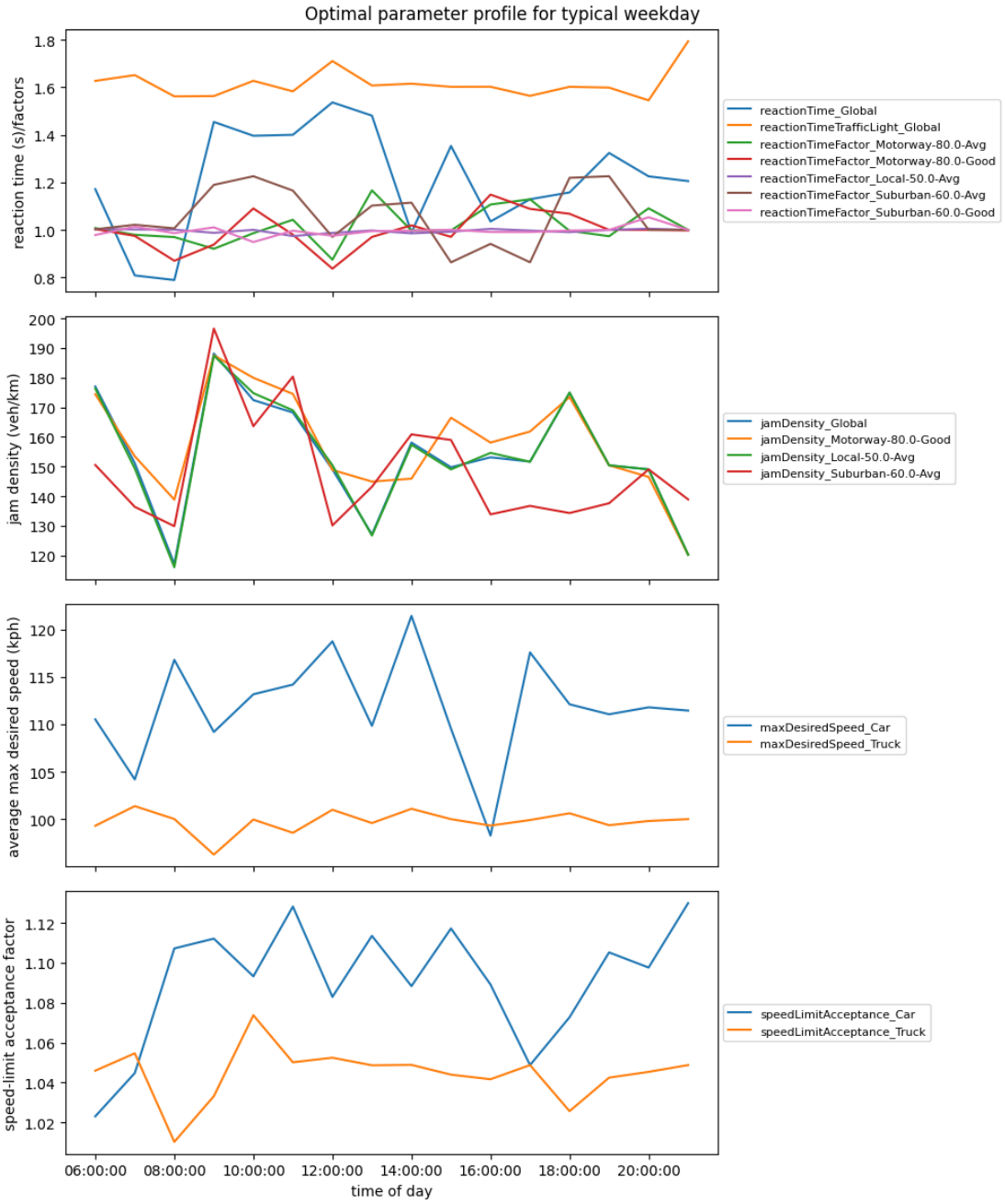


Figure 13: Optimal parameters for the typical weekday. Parameters are separated according to their typical scale for viewing purposes. Local jamDensity parameters are shown scaled by the global parameter. Similarly, jamDensity_Global is scaled by 150veh/km for comparison. The results for each hour long period are plotted at it's beginning (e.g., 8am-9am is plotted at 8am).

3.5.2. Optimisation with respect to individual days

We perform the optimisation with respect to individual days for the hours of 5am-9pm for all demand patterns. Public holidays use the Sunday demand pattern. Due to errors performing simulations, results for 5am-8am on Saturdays are missing. For each hour 1500 simulations are performed. In the identification step the 25 simulations with the lowest cost function values are used. This number was chosen such that the average had converged to a seemingly stable value and was tested with multiple random samples. The random seed is fixed for these simulations but tests with an unfixed random seed also resulted in sensible parameter patterns.

The results for a subset of parameters are shown in Figure 14. There are clear structures in the parameter values, reflecting changes in driver behaviour throughout the day and year.

At this stage the performance of the optimised parameters is not validated, as these results would be subject to overfitting and are unlikely to be informative. Instead, we delay this step until after we have identified clusters in the parameter values, as this will remove the issue of overfitting and be more reflective of the parameters used in real operation. The identification of parameter clusters is discussed in Section 4.2 and the validation of the performance with those results is shown in Section 4.4.

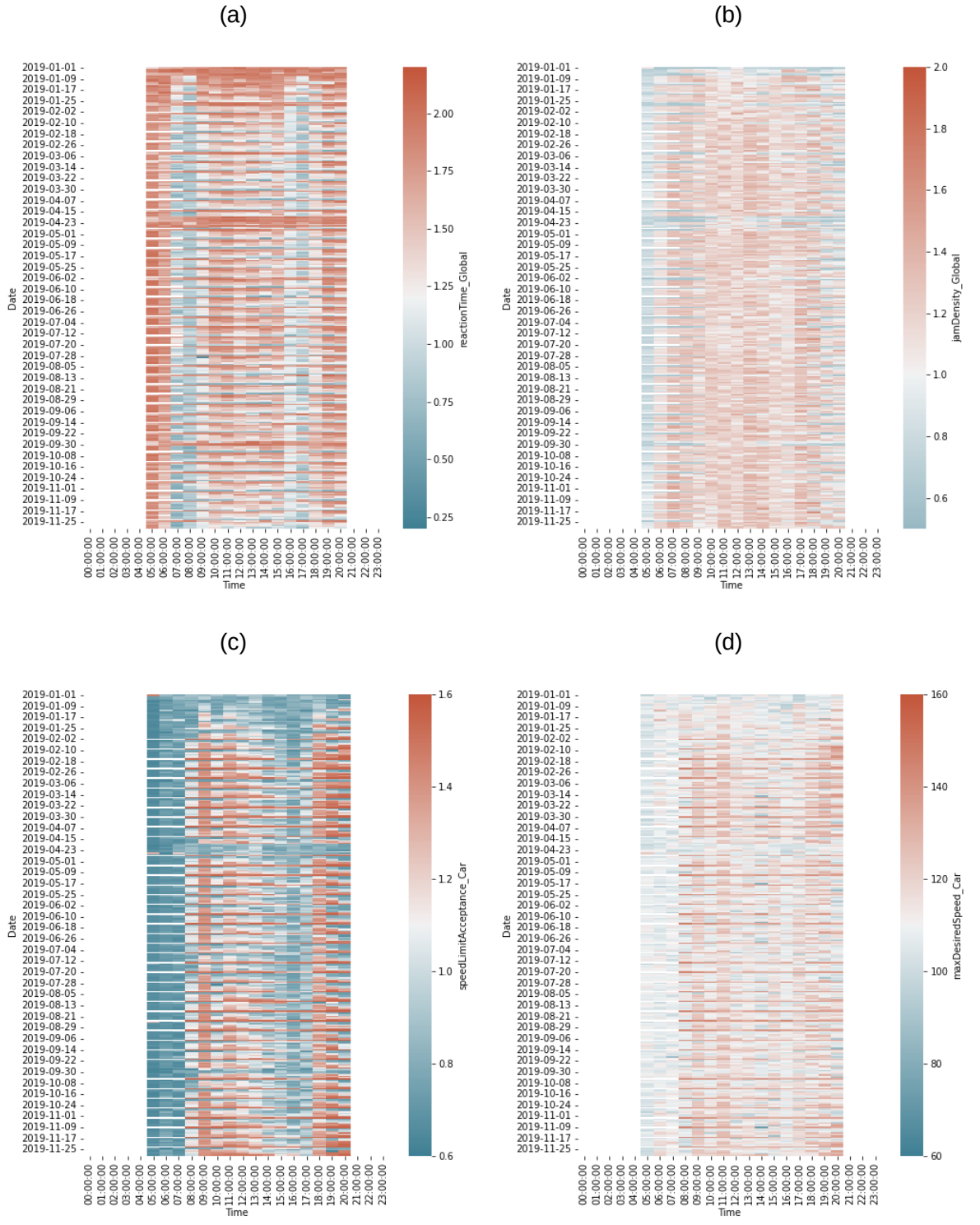


Figure 14: Optimised parameter values for (a) reactionTime_Global, (b) jamDensity_Global, (c) speedLimitAcceptance_Car, and (d) maxDesiredSpeed_Car. Horizontal rows on each plot correspond to each day, with time-of-day on the horizontal axis. Each block corresponds to an hour of simulation, with the time-of-day label denoting the beginning of each period.

4. Pattern refinement

Traffic dynamics shift cyclically between different modes that can be grouped together into different patterns. For the Perth Live model, the demand is split into weekday, Saturday, and Sunday patterns, which are used as the base demand for OD adjustment when the model is deployed.

As part of this project we aim to improve these patterns, which we do from two separate approaches. For one approach we look directly at the observable data, such as the historical volume. In the other we look for patterns in the optimised supply and driver behaviour parameters.

Clustering algorithms are commonly used to identify groups in data. For this project we use hierarchical agglomerative clustering, in which each data is initialised in individual clusters that are linked together with an increasing distance threshold. Eventually all points belong to the same cluster. Some intermediate distance results in a sensible clustering of the data. This method produces a nested hierarchy of clusters, which explain the structure of the data being analysed.

4.1. Patterns from traffic observations

We first identify traffic patterns directly from observations of the traffic state. The choice of what is defined as a data-point for the purposes of clustering has a significant effect on the meaning of the resulting clusters. For example, if we choose to define each detector station as a data-point, then the resulting clusters would reflect detectors which experience similar patterns of traffic. By applying a constraint that these clusters must be spatially contiguous, we may find regions of the network which experience similar traffic patterns. Alternatively, we could perform temporal clustering based on the state of the entire network during different time periods, which results in patterns of traffic that appear at different points in time, e.g. morning peak, Saturdays, Friday afternoons, etc.

Here we perform two forms of temporal clustering, one in which every 15-minute period is defined as a data-point and another where each day is defined as a data-point. For the daily clustering approach each data-point is a $(96 \times N_d)$-dimension vector for the 96 15-minute measurements at each of the N_d detector stations. For the 15-minute clusters each data point is a N_d-dimension vector.

The clustering results for each approach are shown in Figure 15 and Figure 16. For the daily patterns we can identify a distinct separation between weekdays and weekend-like days, as expected. At a lower level there is further separation into classes which can be described as:

- Fridays (orange)
- Weekdays near holidays (green)
- Typical weekday (red)
- Saturdays (purple)
- Sundays and public holidays (brown).

For the 15-minute clusters we see groups that appear throughout the period of analysis. The times at which these patterns begin or end is related to the different classes of days identified in the daily clustering approach, but notably the clusters that appear on weekends, for example, are the same as those appearing on other days. Roughly these can be described as:

- Overnight (orange)
- Early morning and evening (green)

- Morning peak (red)
- Middle of day (brown)
- Afternoon peak (purple).

(a) (b)

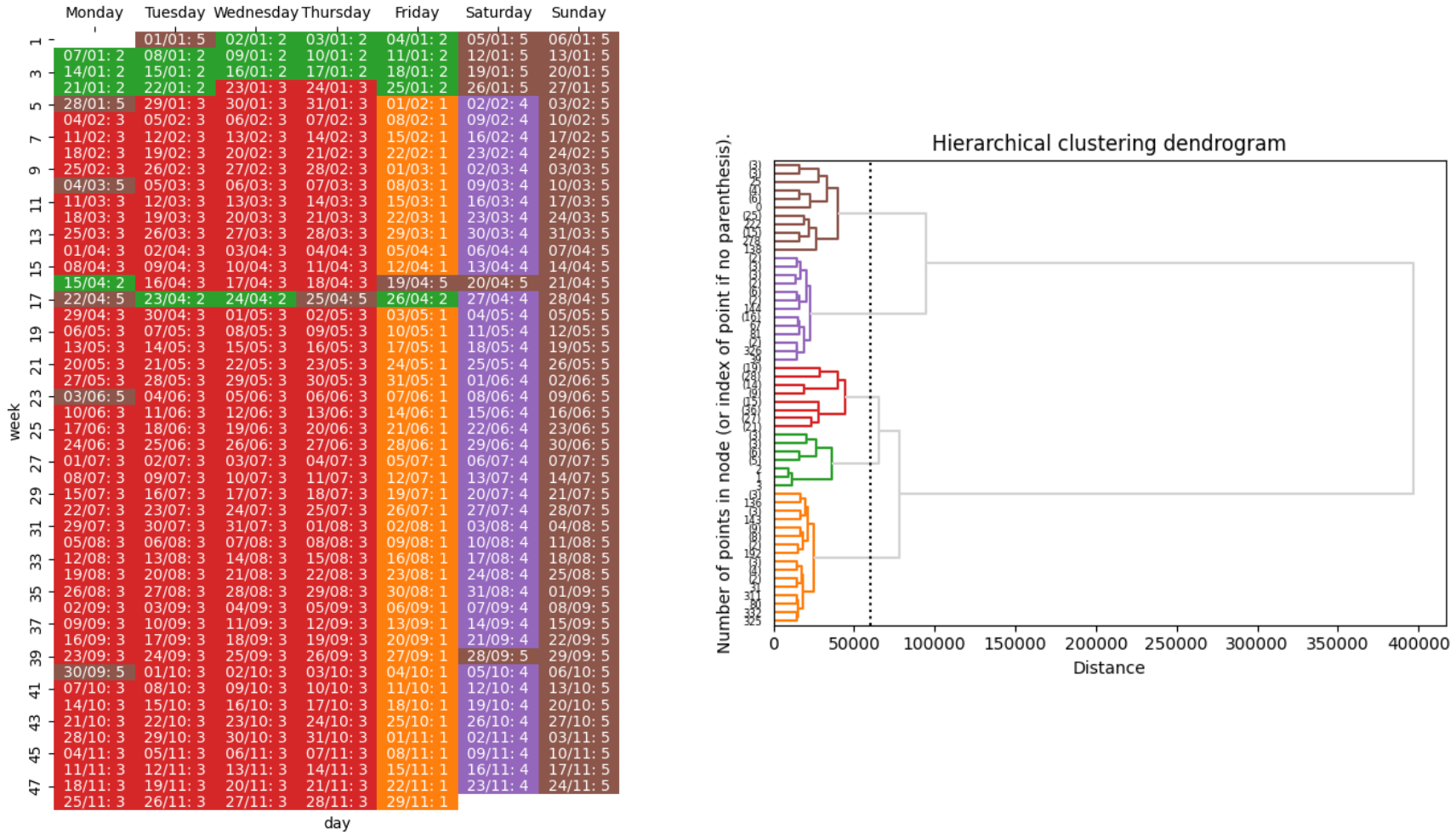


Figure 15: (a) Daily clusters from detector counts and (b) corresponding dendrogram.

(a) (b)

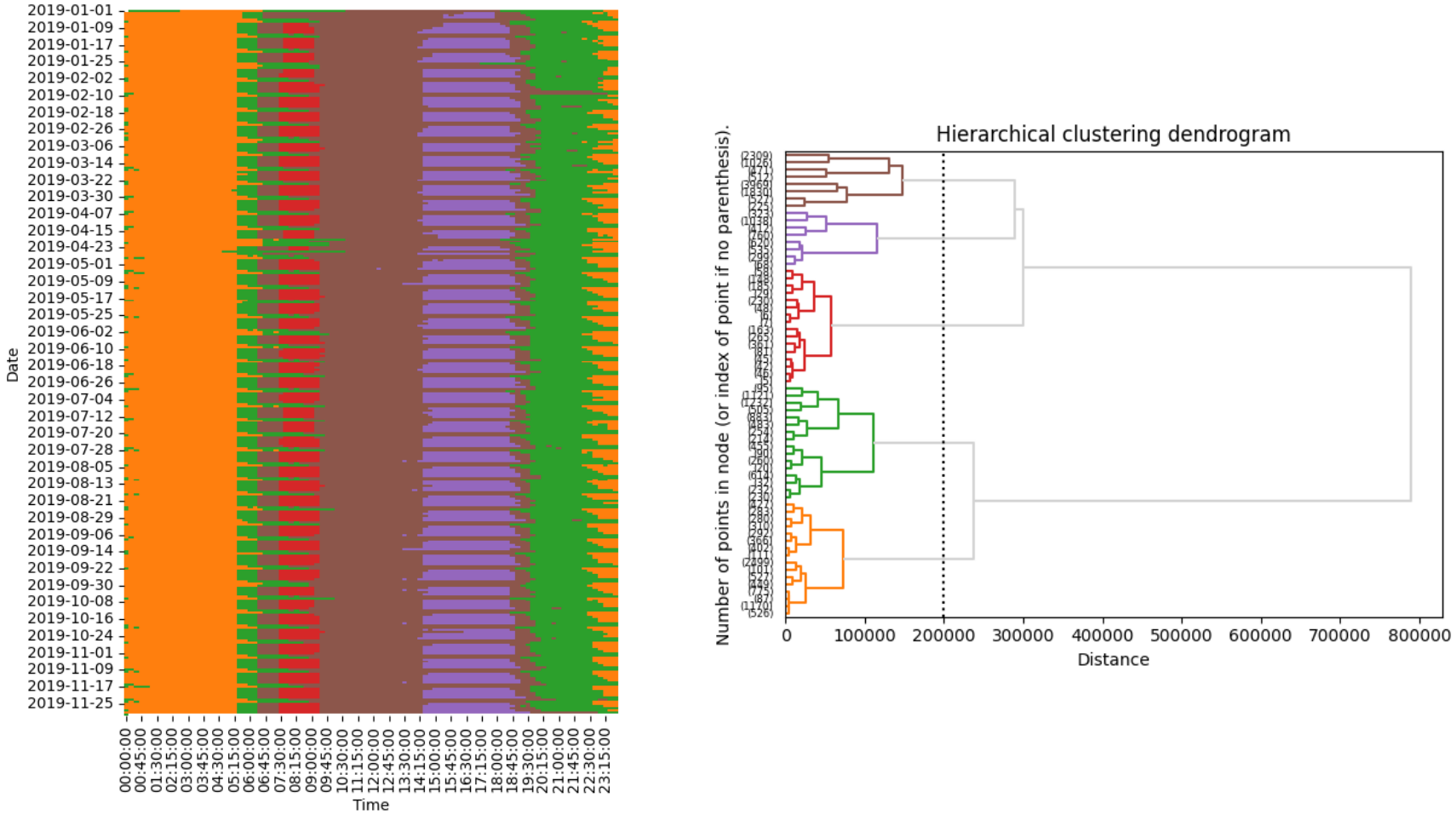


Figure 16: (a) 15-minute clusters from detector counts and (b) corresponding dendrogram.

4.2. Patterns from optimised parameter values

As well as identifying patterns from direct observation of the traffic state, we also seek to identify clusters in the optimised parameter values. These will reflect patterns of driver behaviour, separated from the influence of demand.

Again, we apply hierarchical clustering techniques to identify the patterns, with the optimal parameter vectors for each hour being used as the data-points for the analysis. However, because different parameters represent different quantities (e.g. veh/km vs s for jamDensity and reactionTime), we cannot naïvely use Euclidean distance as a measure of similarity between different parameter vectors. Instead, it is important to take the typical scale of each parameter into account. This could be done based on the bounds used when performing optimisation, by scaling the values to a range between 0 and 1. However, this places undue importance on the range of allowable parameters, which was selected arbitrarily.

Another approach is to use the scale of the parameter values in the identification step of optimisation (see Section 3.4.2). This can be done by taking the standard deviation of the values selected for each parameter and each hour being optimised. Let γ_i refer to the ith parameter value and $\gamma_{i,t}^*$ be an optimised value, with t indexing the different time periods which are optimised separately. We also label the standard deviation of the set of parameter values which are selected during identification as $\sigma_{i,t}$, with the same indexing as the optimised value. The average standard deviation over time is denoted as $\sigma_i = \langle \sigma_{i,t} \rangle_t$. In order to implement the distance metric that accounts for the sensitivity of each parameter, we first apply a coordinate transformation to the parameter space which normalises by the standard deviation of each parameter:

$$\gamma_i' = \frac{\gamma_i}{\sigma_i}.$$

Euclidean distance is used in the transformed coordinates as the measure of dissimilarity between parameter values for the clustering algorithm.

The advantages of this approach are twofold. First, the parameters are transformed into dimensionless quantities, which removes the problem of comparing variables of significantly different scales. Second, differences between parameter vectors are weighted more heavily by those parameters that are more influential on the outcome of the model. These parameters will have a relatively smaller distribution of points which contribute to the identification of the optimal parameter vector when compared to less influential parameters and thus have a larger range in the transformed coordinate system. This is illustrated in Figure 17 where toy data is generated by

$$Cost = (\gamma_1 - 0.5)^2 + 0.1(\gamma_2 - 0.2)^2 + 0.01\xi,$$

where $\xi \sim \mathcal{N}(0,1)$ is a noise term. Parameters γ_1 and γ_2 are normalised to the unit interval, but γ_1 is more influential on the outcome of the simulation. The difference between changing x units of γ_1 has a larger impact on the cost function than changing x units of γ_2, so should be weighted more in the distance metric. This is shown in Figure 18.

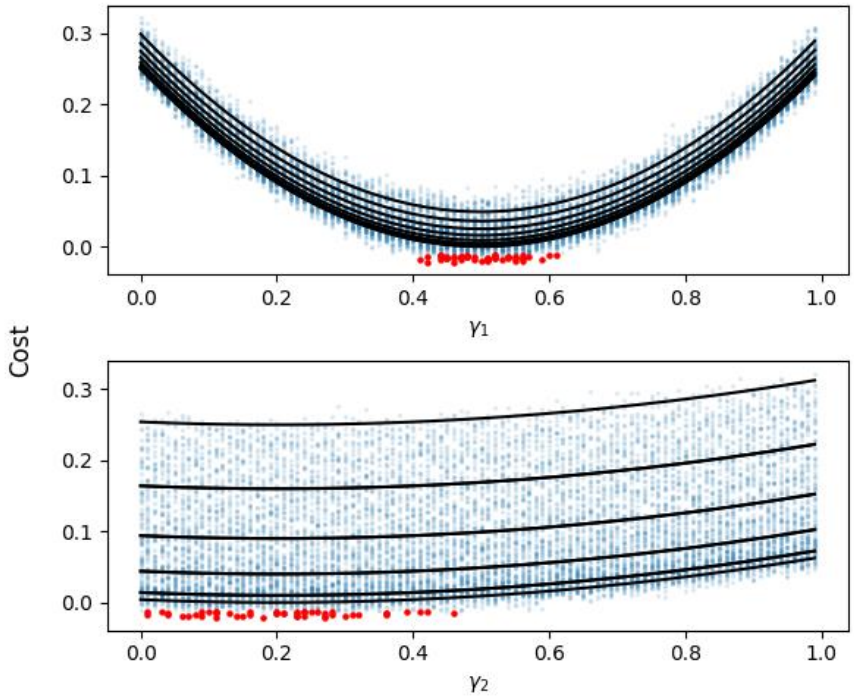


Figure 17: Demonstration of sensitivity in optimal parameter identification. The cost function values are the same in both panels. The black lines follow the noiseless cost function and the blue dots are the noisy observations. Red dots are the 50 smallest observations which are used for identifying the optimal parameters. The true optima are $\gamma_1 = 0.5$ *and* $\gamma_2 = 0.2$.

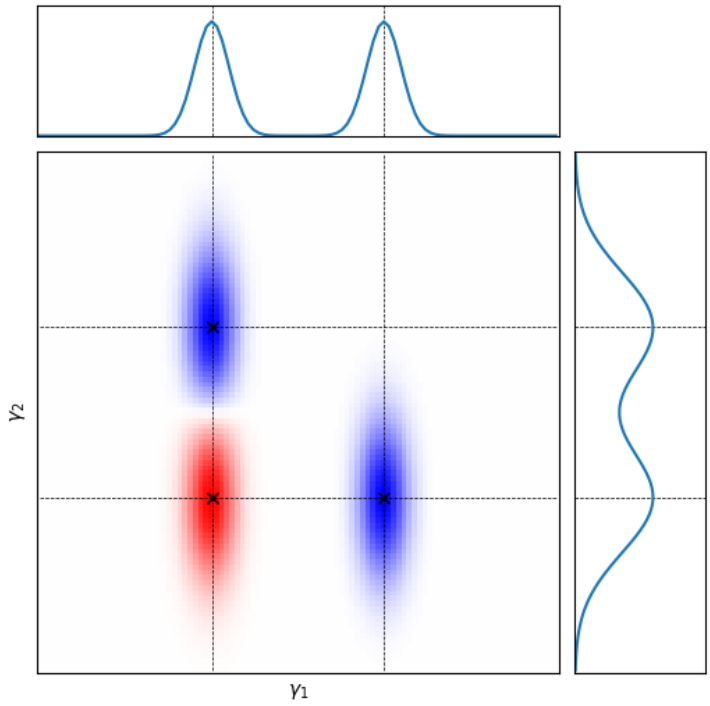


Figure 18: Effect of nonuniform uncertainty on distance estimation. The centres of the blue distributions are equidistant from the centre of the red distribution. The marginal distributions show that the separation in the γ_1 *direction is more significant than in the* γ_2 *direction.*

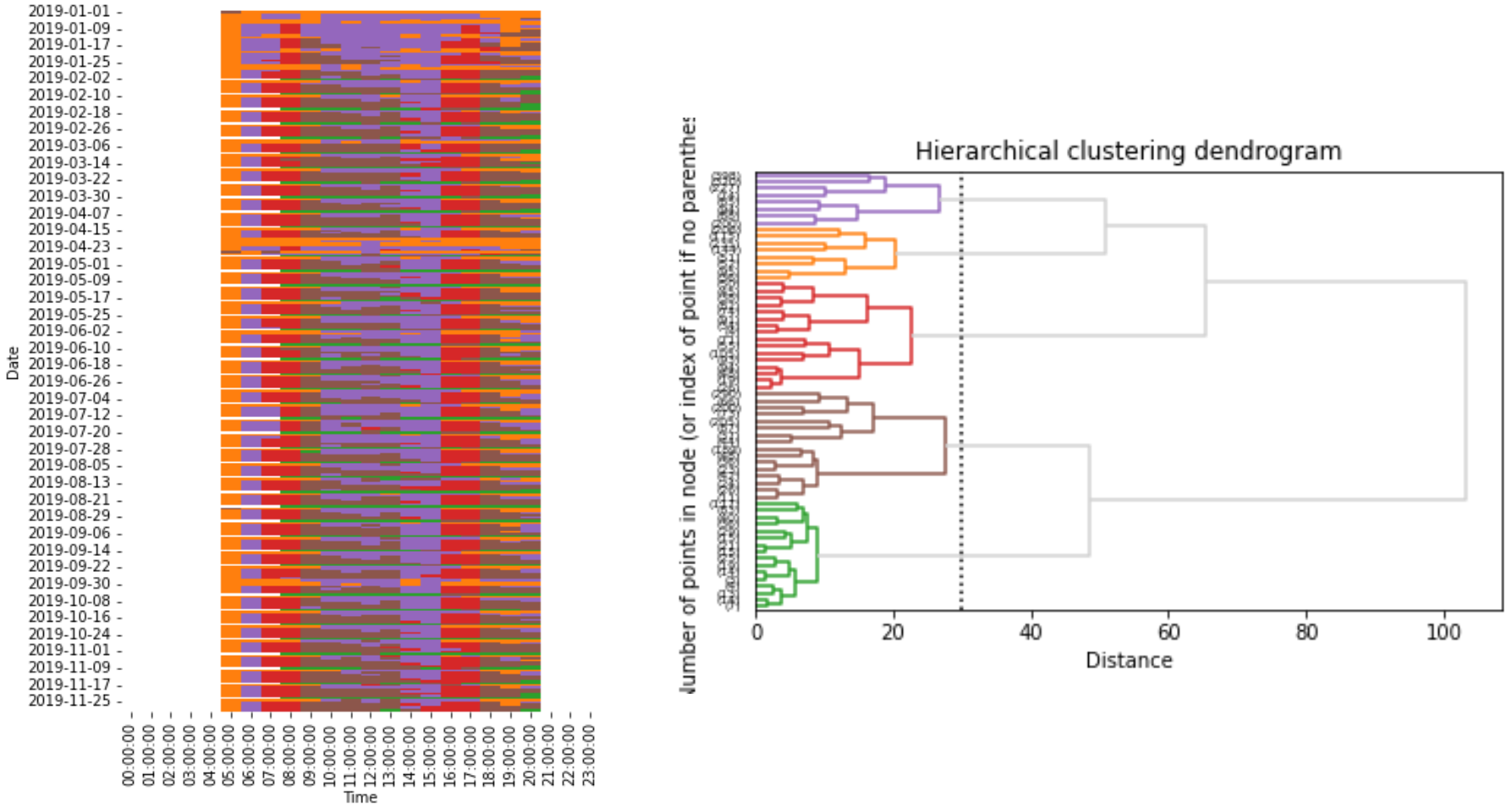


Figure 19: (a) Hourly supply parameter patterns for 5am-9pm and (b) corresponding dendrogram with matching colours. (a) Each block corresponds to an hour of simulation, with the time-of-day label denoting the beginning of each period.

Using this similarity metric we perform clustering of all optimised parameter values concurrently with the results shown in Figure 19. We see meaningful clusters emerge, though they are less clear than the detector data counterparts. They can be roughly described as:

- Overnight and holiday (orange)
- Pre-peak/early peak (purple)
- Peak/late peak (red)
- Post-peak (brown)
- Saturday (Green).

Notably, both morning and afternoon peaks follow the same pattern, since although the mobility patterns are significantly different, the driver behaviour will be similar. The orange “quiet” cluster also extends beyond the overnight period around public holidays, reflecting changes in driver behaviour which are not obvious from the detector data alone.

The cluster centroids corresponding to Figure 19 are shown in Table 6. Some results agree with prior expectations. For example, the red cluster which corresponds to the peak periods has lower reactionTime and higher jamDensity than normal. On the other hand, The orange cluster which is associated with early mornings, evenings, and holiday periods has higher reactionTime and lower jamDensity. This agrees with how we expect business to impact these parameters. We also see lower reactionTimeFactor values for motorways in general. The green cluster, which is almost exclusively present on Saturdays is unusual, with a lower reactionTime_Global value than the peak cluster.

Table 6: Hourly parameter cluster centroid values.

Parameter Label	Orange	Green	Red	Purple	Brown
reactionTime_Global	1.82	0.76	0.95	1.67	1.37
reactionTimeTrafficLight_Global	1.74	1.44	1.53	1.63	1.50
jamDensity_Global	0.78	1.23	1.25	1.15	1.19
jamDensity_Motorway-80.0-Good	141.46	153.42	167.08	165.48	152.39
jamDensity_Local-50.0-Avg	146.49	152.41	145.05	147.57	150.09
jamDensity_Suburban-60.0-Avg	142.81	164.80	166.80	172.13	163.40
reactionTimeFactor_Motorway-80.0-Avg	0.99	1.01	0.89	0.91	0.91
reactionTimeFactor_Motorway-80.0-Good	1.03	0.98	0.95	0.83	0.87
reactionTimeFactor_Local-50.0-Avg	1.02	1.03	0.99	1.01	0.99
reactionTimeFactor_Suburban-60.0-Avg	1.21	0.92	0.95	1.03	1.19
reactionTimeFactor_Suburban-60.0-Good	1.06	0.97	1.02	0.97	0.94
speedLimitAcceptance_Car	0.74	1.51	0.87	0.89	1.29
speedLimitAcceptance_Truck	1.04	1.17	1.03	1.04	1.08
maxDesiredSpeed_Car	106.90	134.87	110.69	110.68	117.01
maxDesiredSpeed_Truck	98.26	107.14	100.80	101.03	101.46

4.3. Cluster comparison

We compare clusters in the observed data and clusters in the optimised parameter values by matching each 15-minute period in Figure 16 to the corresponding hour in Figure 19.[2] The comparison is shown in Figure 20. There are similarities between the two clusterings, such as the "overnight", and "morning and evening" clusters in the detector data mostly mapping to the orange cluster in the parameter space. The "morning peak" detector cluster almost completely maps to the red peak parameter cluster. However there are some discrepancies. Some cross-links are the result of the lower resolution of the optimised parameter values (e.g. 07:00-07:15am detector clusters are compared with 07:00-08:00am parameter clusters). These cross-links occur for neighbouring clusters, such as the "morning and evening" detector data cluster mapping to the brown, purple and green parameter clusters. The "middle of day" detector data cluster splits across the brown, purple and green clusters, with some cross-links to neighbouring red and orange clusters. This suggests that we can't directly assign parameter values based on only observing which cluster the current detector data falls into. However, we observe that the purple parameter cluster precedes the peak, the brown cluster follows the peak, and the green cluster only exists on Saturdays, so this extra information can be used to determine which parameter cluster to use.

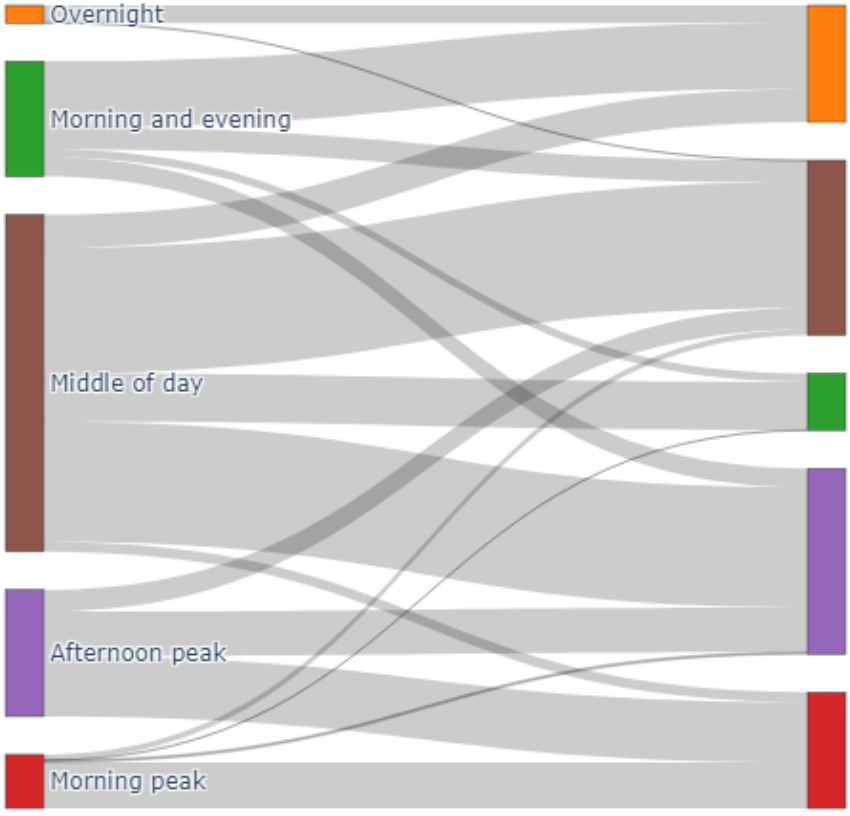


Figure 20: Mapping from detector count clusters (Figure 16, left) to parameter clusters (Figure 19, right). Colours in the previous figures were heuristically remapped in order to best illustrate this comparison.

[2] Note that the cluster colours in Figure 16 and Figure 19 were relabelled in order to match according to a heuristic assessment of their meaning. For example, the peak parameter cluster was labelled red in order to match with the morning peak detector count cluster.

4.4. Model performance with parameter cluster centroids

In order to validate the optimised parameters, we run the model with the cluster centroids and compare the results to simulations with default parameters. This is performed for all demand patterns and for all hours from 5am to 9pm. In general the performance improves when the optimised parameters are used, with notable improvements for the morning peak and holiday times. The scale of this improvement is small, on the order of $\Delta GEH_{ave} = -0.5$. There are also some cases where the performance is marginally worse, particularly in the afternoons. The period from June-July seems to have larger improvements than the rest of the year, which may be due to seasonal effects. We investigated whether this could be related to environmental conditions, such as weather, but found no conclusive results.

In summary, these results show that the performance of the model can be improved by using optimised parameters, though this improvement is small.

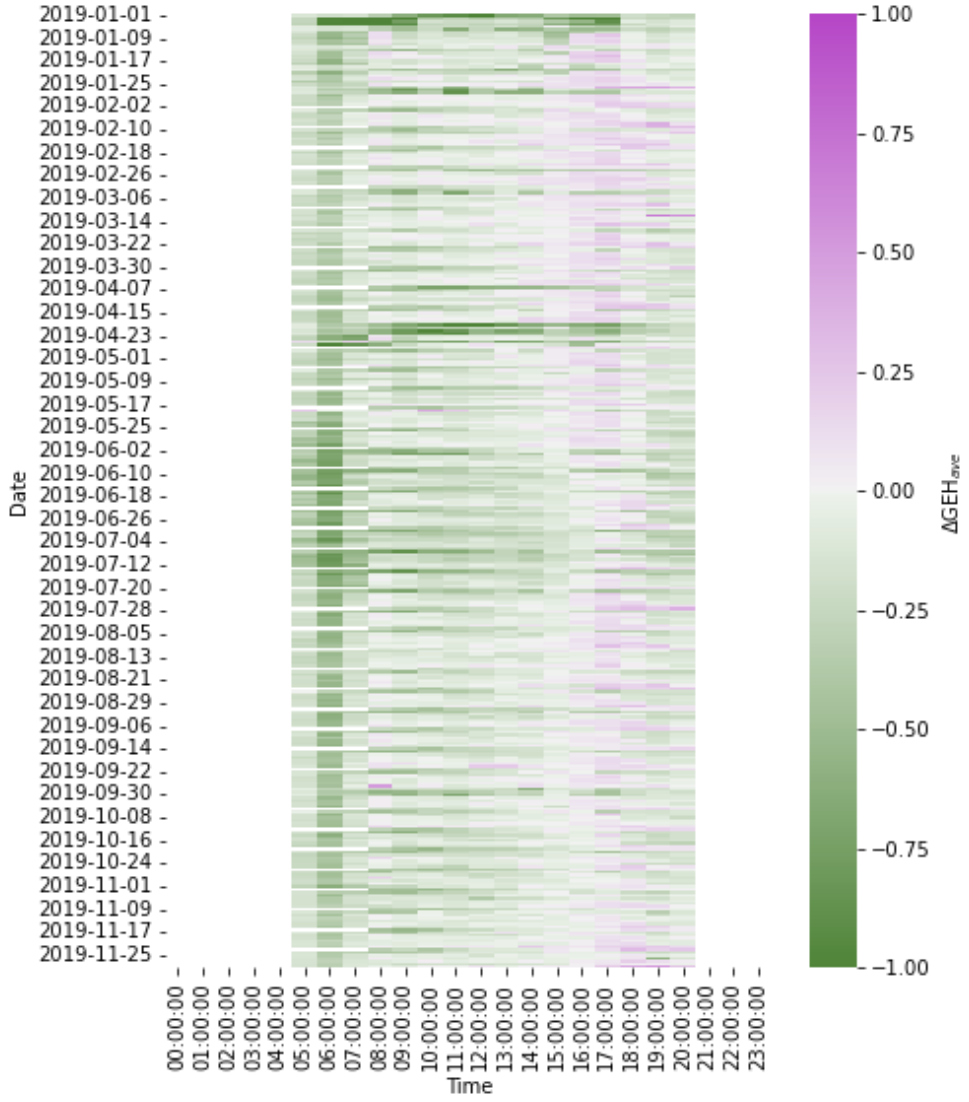


Figure 21: Change in average GEH between default parameters and optimised parameter centroids.

5. Prediction confidence

In this section we investigate different metrics for measuring the quality of a simulation. First we catalogue some candidate metrics for measuring prediction accuracy, including some metrics which include historical information about the traffic states that are being modelled. We then quantify the performance of the different metrics with a measure based on clustering analysis.

5.1. Prediction quality metrics

Let m and o be modelled and observed values respectively. For now, we assume that these are scalar values, such as the volume at a single detector station. Methods of aggregating multivariate errors will be discussed later. Potential quality metrics are listed below.

Absolute error

$$\mathcal{E}_{abs} = |m - o|$$

Relative error

$$\mathcal{E}_{rel} = \frac{|m - o|}{|o|}$$

Root-mean-square error (RMSE)

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1,\dots,N} (m - o)^2}$$

GEH

$$GEH = \sqrt{\frac{(m - o)^2}{\frac{1}{2}(m + o)}}$$

The GEH statistic is a metric commonly used in traffic applications that is designed to balance the overestimation of errors by $\mathcal{E}_{abs}$ at large scales with the overestimation of errors by $\mathcal{E}_{rel}$ at low scales. For example, a 50% relative error is 50veh/hr at a scale of 100veh/hr and 500veh/hr at a scale of 1000veh/hr. The latter is clearly a more significant error. On the other hand, an absolute error of 50veh/hr at a scale of 100veh/hr is more significant than an absolute error of 50veh/hr at a scale of 1000veh/hr, according to the rationale behind GEH.

From a theoretical perspective, GEH achieves this balance by taking the geometric mean of the absolute and relative errors,

$$GEH = \sqrt{\frac{(m - o)^2}{\frac{1}{2}(m + o)}} = \left(\frac{|m - o|}{\frac{1}{2}(m + o)} \times |m - o| \right)^{\frac{1}{2}} = (\mathcal{E}_{rel} \times \mathcal{E}_{abs})^{\frac{1}{2}}.$$

Although GEH is widely used, it is also criticised for a number of reasons, including:

1. Asymmetry with respect to overestimation and underestimation of the observed count o by the modelled count m.
2. GEH values are scale dependent, and therefore cannot be compared across different scales (e.g. hourly flows vs daily volume, high volume detectors vs low volume detectors).
3. GEH has units of $\sqrt{\text{veh/hr}}$ (assuming the quantities are flows).

Asymmetry is a result of the presence of the modelled count m in the denominator $\frac{1}{2}(m+o)$, which has that form because of the more ubiquitous use of GEH outside of a modelling context. For example, GEH could also be used to compare the volumes on a highway from year-to-year. In this case neither value is privileged as the "ground truth", so their average should be included in the denominator. However, in a modelling context the observed value o is known to be an accurate (or at least much more accurate) measurement of the quantity that is being modelled by m. Therefore the denominator $\frac{1}{2}(m+o)$ can be replaced by o in order to deal with the first criticism.

The second and third criticisms cover the same point, and are a consequence of the presence of the absolute error in the geometric mean. Furthermore, this is actually a restatement of the goal of GEH, whereby errors have different meanings depending on the scale of the data. Also, once the asymmetry has been removed by replacing the denominator, the scale dependence of GEH is irrelevant to optimisation.

Weighted GEH

The aim of GEH is to balance the overemphasis of relative errors at low flows against the against the underemphasis of absolute errors at high flows, by taking the geometric mean of the two quantities, $GEH = (\mathcal{E}_{rel} \times \mathcal{E}_{abs})^{\frac{1}{2}}$. However, there is no reason why the two metrics should be given equal weighting in the calculation of the mean. Instead, we propose a weighted GEH which performs the trade-off with different proportions for each metric:

$$\begin{aligned} wGEH_\alpha &= \left(\mathcal{E}_{rel}^{\alpha} \times \mathcal{E}_{abs}^{\beta}\right)^{\frac{1}{\alpha+\beta}} \\ &= \mathcal{E}_{rel}^{\alpha} \times \mathcal{E}_{abs}^{1-\alpha}\,. \end{aligned}$$

Scalable quality value (SQV)

The scalable quality value (SQV) [1] was introduced to address the criticisms of GEH listed previously, while also suggesting that the range of possible values should be between 0 and 1. As previously discussed, SQV also replaces the denominator $\frac{1}{2}(m+o)$ with o:

$$\sqrt{\frac{(m-o)^2}{\frac{1}{2}(m+o)}} \to \sqrt{\frac{(m-o)^2}{o}}.$$

The scale dependence of GEH is addressed by normalising both m and o by some factor f representing the typical scale of the quantity being measured (e.g. $f = 10$ for the mean trip distance in kilometres, $f = 1000$ for traffic flow in veh/hr):

$$\sqrt{\frac{(m-o)^2}{o}} \to \sqrt{\frac{\left(\frac{m}{f}-\frac{o}{f}\right)^2}{\frac{o}{f}}} = \sqrt{\frac{(m-o)^2}{f \cdot o}} = SQV_1 \,.$$

If a fixed value of f is used then SQV_1 is equivalent to GEH up to a scalar multiple, so there is no meaningful difference. However, we can set f as the typical scale (e.g. yearly average flow) of each detector station, which makes SQV_1 more akin to the relative error, as we expect m and o to be correlated with f.

In order to achieve values ranging from 0 to 1, the authors transform SQV_1:

$$SQV_1 \to \frac{1}{1+SQV_1} = SQV_2 \,,$$

where a value of 0 corresponds to an infinite difference between o and m, wile a value of 1 corresponds to perfect agreement. We distinguish between SQV_1 and SQV_2 for the purposes of quantitative comparison in Section 5.2.

Mean absolute scaled error (MASE)

The mean absolute scaled error (MASE) measures the mean absolute error relative to the equivalent error achieved by a naïve estimate $\hat{m}$:

$$MASE = \frac{\frac{1}{N}\sum_{i=1,\dots,N}|m_i - o_i|}{\frac{1}{N}\sum_{i=1,\dots,N}|\hat{m}_i - o_i|} \,.$$

For traffic data which has a dominant seasonal component, the naïve estimate is the historical average at the corresponding point in the cycle. For example, the naïve estimate at 8am on a Wednesday is the average count across all Wednesday 8ams in the dataset. Public holidays are grouped with Sundays for this purpose.

Mahalanobis distance

The Mahalanobis distance is a metric which gives the distance between two points relative to the range over which those values typically vary. This is done by normalising the absolute difference between two measurements by the standard deviation σ of the distribution from which they are drawn:

$$\mathcal{E}_{\text{mahalanobis}} = \frac{|o-m|}{\sigma} \,.$$

Typically the points being compared are exchangeable, so they belong to the same distribution. Though this is not strictly true in this case, because o and m are generated by different processes, we will assume that the value of σ can be calculated from historical observations of traffic counts.

The premise of this metric is that the errors at different detectors and times should be correlated to the corresponding spread of the data at those detectors and times. Therefore, in order to know whether an error is large or small, we should normalise by a scale that is correlated with the typical size of errors, given by the standard deviation. As for MASE, we use the time-dependent standard deviation for each point in the weekly cycle.

Under ideal assumptions of independent, exponentially distributed vehicle arrivals, the traffic counts have a Poission distribution and the standard deviation is equal to the square root of the mean $\sigma_{ideal} = \sqrt{\bar{o}}$, where $\bar{o}$ is the historical mean. This means that the Mahalanobis distance would be equivalent to GEH, with the current observed count o replaced by a historical average $\bar{o}$ in the denominator. Of course, these ideal assumptions do not reflect reality, but indicates that the Mahalanobis distance is similar to GEH, but with more rigorous statistical background.

Because of the multivariate nature of data for this project, we consider two versions of the Mahalanobis distance. First, we calculate the distance for each detector station and take the average. The second version considers correlations between detector stations by measuring distance in the high-dimensional space relative to the multivariate distribution of the data. This is done by performing a whitening transformation to normalise the distribution and remove correlation between state variables, then calculate the Euclidean distance in the transformed state. Note that because we normalise by measures of the spread of the data from finite samples, both versions of this metric are capable of massively overstating errors. We avoid this issue by regularising the estimates of the spread of the data (standard deviation and covariance matrix respectively).

5.2. Quantitative evaluation of quality metrics

Qualitative discussions of error metrics and their theoretical properties can only go so far. We also want a quantitative method of evaluating each metric. For the purpose of calibration the most desirable quality of an error metric is that it recovers the “true” parameter values when included in an optimisation framework. This could be tested by generating synthetic data with known parameters, then attempting to reconstruct those parameters by optimising each quality metric. In order to do this properly several iterations of the optimisation would need to be performed in different conditions for each quality metric, so this is not feasible with the full model. Instead, we could use a simpler substitute for the model with similar characteristics (see Section 6.1.1 for a brief discussion), but we do not consider that for this project.

Another desirable property of the quality metric is that is should distinguish “good” simulations from “bad” simulations, though without measuring parameter reproduction the concept of “goodness” is nebulous. Instead, a prerequisite for this property is that the metric should separate different simulations as much as possible, where “different” in this context refers to simulations with different parameters, rather than simulations with different outputs. When running simulations with noise, randomness will produce variation in the outcome of the simulation even if identical parameters are used. This reflects the random variations in the real-world traffic system which we can never hope to model, even if we have the optimal parameters and demand. If the parameters are changed, then the output of the model averaged over realisations of the noise will also change, which is the dependence that we aim to observe. Therefore, we want a metric that maximises the distance between two simulations caused by changes in the parameter values, relative to the variance due to noise.

This concept is illustrated with toy data in Figure 22, where multiple simulations are performed for each of a set of parameter values. Each parameter set produces a group of simulation outputs, with their spread caused by noise. Drawing inspiration from the evaluation of clustering methods, we define a ratio R of the average intergroup and intragroup distance to quantify the degree to which the clusters are separated:

$$R = \frac{\sum_i \sum_{j \notin g_i} \mathcal{E}(\boldsymbol{c}_i, \boldsymbol{c}_j)}{\sum_i \sum_{j \in g_i} \mathcal{E}(\boldsymbol{c}_i, \boldsymbol{c}_j)},$$

where $\mathcal{E}(\cdot,\cdot)$ is the metric being tested, $\boldsymbol{c}_i$ and $\boldsymbol{c}_j$ are simulated counts for simulations i and j, and g_i is the group of simulations to which i belongs. Larger values of R correspond to tighter clusters and greater separation of different parameter values.

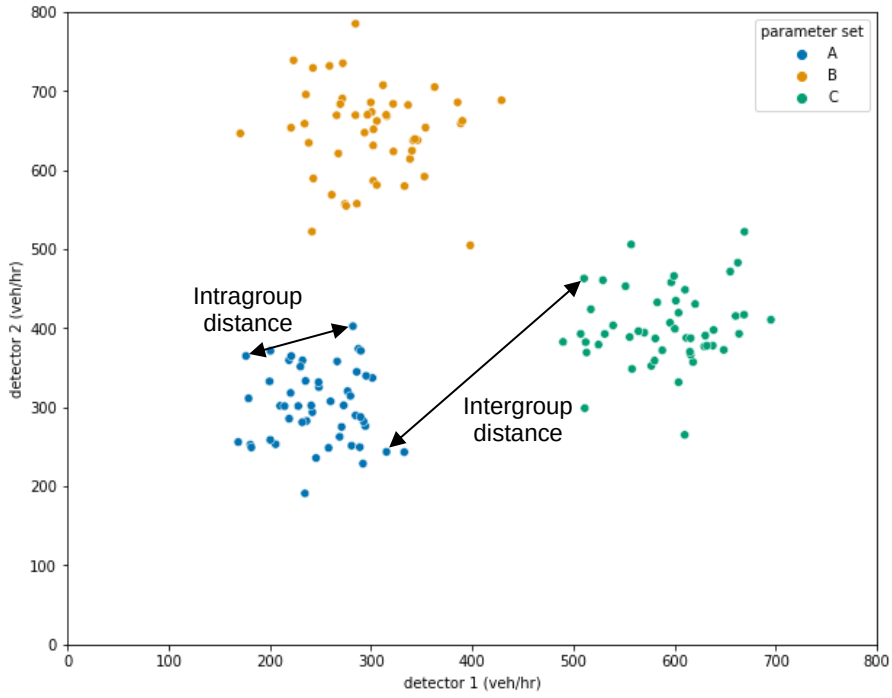


Figure 22: Illustration of noisy simulation outputs with different parameter values using toy data. Each point represents a single simulation with the colours indicating different parameter sets. Within group differences are the result of randomness, while between group differences are also caused by differences in parameter values.

This concept is relevant to optimisation, as shown in Figure 23. Both examples have equal levels of noise, but due to the larger macroscopic scale of the blue function, it has higher R and is easier to optimise. When the macroscopic scales are matched, the noise on the red function is significantly larger and confounds optimisation algorithms to a greater degree.

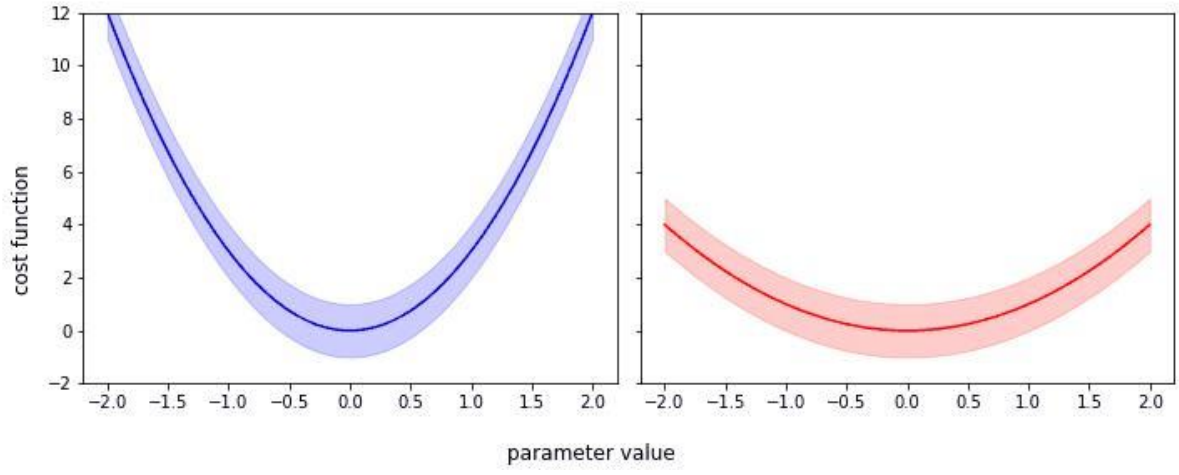


Figure 23: Importance of the ratio R in the context of optimisation. The blue cost function will have a higher ratio R than the red function.

In order to calculate R for the model, 10 parameter groups are randomly selected using the bounds from previous optimisations (See Table 5). For each group 10 simulations are performed resulting in 100 total simulation outputs. We calculate the distance between simulation outputs using the metrics discussed in Section 5.1. For cases where the distances are calculated separately for each detector station, the individual values are averaged for aggregation. We also consider the percentage of GEH values greater than 5 ($\%GEH > 5$), because of its common use in traffic engineering.

Note that the results of these experiments are not transferable to other contexts, as the ratio is a property of not only the quality metric, but also the model used to generate the data and the specific parameter values used.

The results for all quality metrics are shown in Figure 24. Root-mean-square error (RMSE) is the best metric according to the ratio R, with $\%GEH > 5$ a close second.

Metrics which range from 0 to 1 (correlation coefficient and SQV_2) are notably poor. Though this may be a desirable property from a reporting perspective due to their interpretable scale, they should be avoided for optimisation. Even in a reporting context, it is important to distinguish good simulations from bad simulations, so they should be avoided. Note that aggregations based on a threshold value, such as $\%GEH > 5$, also fall into this category since the values range between 0 and 100%. The threshold value is very important and will be discussed later.

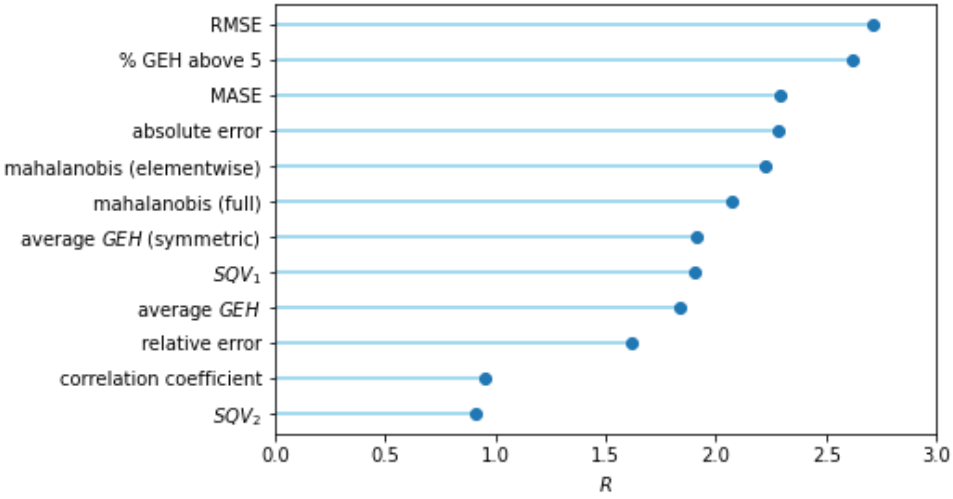


Figure 24: Separation ratio R of candidate quality metrics.

We also evaluate the weighted GEH for all $\alpha \in [0,1]$, with $\alpha = 0$ corresponding to the absolute error, $\alpha = 0.5$ corresponding to the usual GEH, and $\alpha = 1$ corresponding to relative error. There is a monotonic decrease in R from the absolute error, through GEH, to the relative error as α increases, meaning that there is no trade-off with relative error that can improve the absolute error with respect to R.

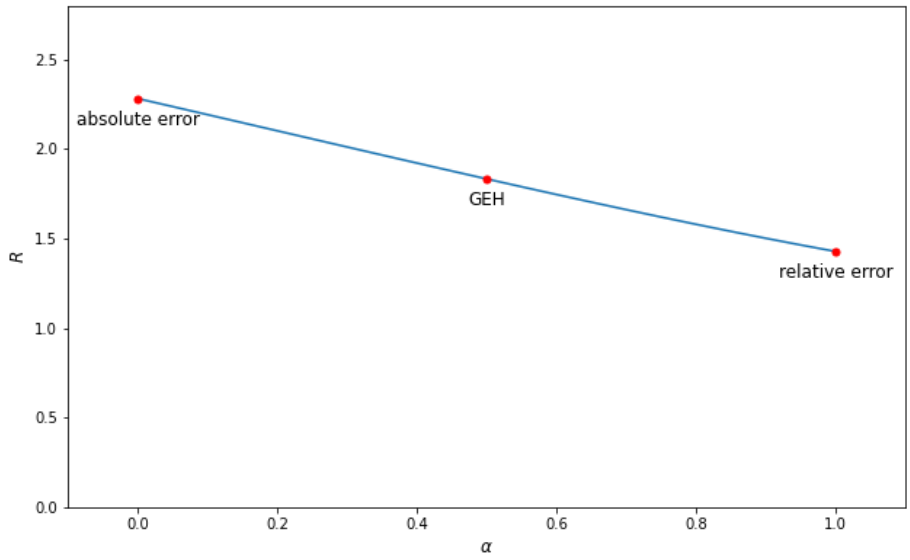


Figure 25: Separation ratio R for weighted GEH ($wGEH_\alpha$) metric.

The results in Figure 24 suggest that $\%GEH > 5$ is one of the best quality metrics, performing significantly better than the average GEH. We also test this metric with other thresholds, with the results in Figure 26. There is a trend for larger values of R with increasing thresholds, due to the fact that random fluctuations in cost values are not large enough to cross over the threshold. Thus we would have very low intra-group distances. On the other hand, low thresholds are crossed by random fluctuations so frequently that it is difficult to distinguish whether a difference in simulation output is due to a meaningful change of the parameters.

Clearly, $\%GEH > 20$ is a poor metric for evaluating the quality of a simulation, since the difference between a detector station with $GEH = 3$ and $GEH = 17$ is not accounted for. However, it is not so clear whether setting $GEH = 10$ as the threshold is a good or bad choice. As mentioned previously, this could be done by measuring the ability of an optimisation algorithm to find the true parameters, but we cannot run those experiments for this project. This highlights the difficulty of using the R metric in isolation without greater context. Since $\%GEH > 5$ has significant previous use by traffic experts, it may still be used for reporting purposes, but we would avoid other thresholds without further investigation.

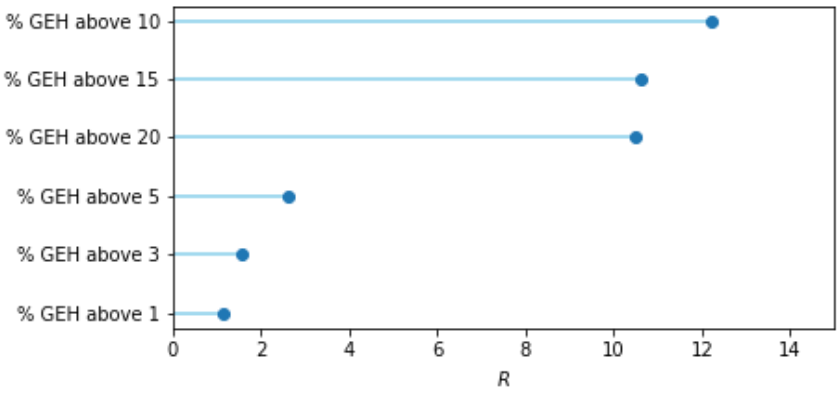


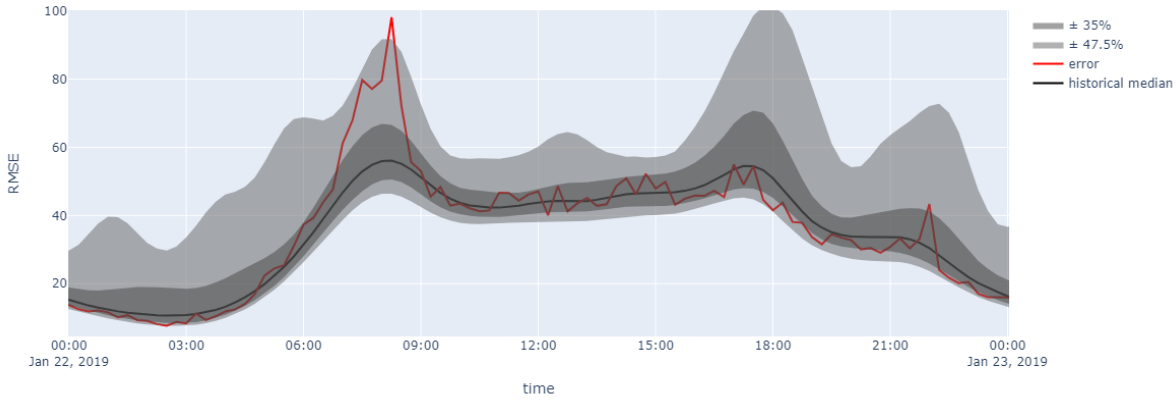
Figure 26: Separation ratio R *for the percentage of* GEH *values above a threshold.*

Since $\%GEH > 5$ has significant previous use by traffic experts, it may still be used for reporting purposes, but we would avoid other thresholds without further investigation. For optimisation,

RMSE has the largest value of R, does not have a discontinuity like $\%GEH > 5$, and is ubiquitously used in other contexts, so it is the recommended metric.

Note that both $\%GEH > 5$ and RMSE are scale dependent, so the meaning of a particular value changes throughout the traffic cycle. This is not an issue for optimisation, but it is important to consider individual cost values within context for monitoring and reporting (see Figure 27).

(a)



(b)

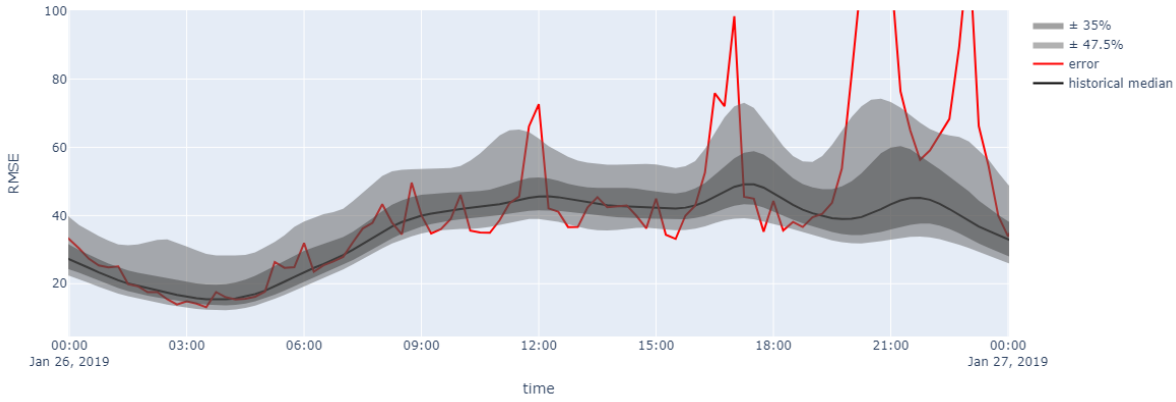


Figure 27: Prediction errors within historical context. A simple machine learning model is used to predict traffic flows. The RMSE is shown in red, with historical error distributions at the corresponding point in the weekly cycle shown in grey. Panel (a) shows a day (22/01/2019) where the errors are within expectations, while panel (b) shows a day with significant deviations due to exceptional traffic on Australia day (26/01/2019).

6. Future work

6.1. Approaches to real-time optimisation

The methods presented in this report use optimisation with respect to historical data to recommend parameter values for running the model in practise. This approach is limited in the degree to which it can respond to unexpected changes in traffic dynamics, only allowing flexibility via matching with patterns in the historical data. True real-time optimisation involves selecting the parameter values for the model based on current information about the state of the network, alongside the historical data. This advantage of the real-time approach is that the model is more capable of responding to unusual and unseen behaviours in the system. However, due to the computational requirements of the simulation model, basic attempts at real-time optimisation are not feasible and more advanced methods are required. We will briefly discuss some approaches to this problem that may be implemented in future work.

6.1.1. Model emulation

The real-time optimisation of the Perth Live model is difficult because of the computational complexity of a theory-driven traffic simulator at large scale. More light-weight machine learning techniques, such as neural networks or nearest-neighbour forecasting, can also be used for traffic prediction with significantly less computational requirements, thereby enabling real-time calibration. However, the extensibility of theory-driven models is a significant advantage for the model operators. For example, if there is a change in the structure of the traffic network or an incident forces lane closures, then the simulation model can be altered with little effort, and without the need for recalibration. If the simulation model is calibrated on the prediction of detector volumes, it is straightforward to extract speeds, CO_2 emissions, or other summary statistics from the realistic physics of the simulation. In order to get the same functionality out of a data-driven machine learning model, expensive retraining is required.

One approach to getting the best of both worlds is to build an emulation of the simulation model, or a model of the model. The emulation models the input-output relationship of the simulation model and is significantly less computationally intensive, without the high level of detail and generative capabilities of the full model. Optimal parameters can be learned on the emulation in real-time, then transferred across to the simulation model for prediction. In this sense, the digital twin of reality and simulation is replaced by a digital triplet of reality, simulation, and emulation.

The emulation can also be used for other purposes, such as determining the best prediction quality metric, as discussed in Section 5.2. The ultimate measure of a quality metric for the purposes of optimisation is the degree to which it accurately returns the true parameters of the model. This can be measured generating fake data by running the model, then tasking an optimisation algorithm to minimise the error between the simulation output and the fake data. This is not feasible for the full model, due to computational requirements, but is relatively easy to do with an emulation.

In order to build an emulation, we first need a large sample of the input-output relationship of the model. In particular, we are interested in only the inputs and outputs that are relevant to performing real-time optimisation. For the Perth Live model, the adjusted OD matrix, initial state, and parameters are the inputs, and the detector station counts are the outputs. The initial state is a highly complex object, consisting of many vehicle locations, routes, etc. Because of this, it is not feasible to sample the space of possible initial states, and we just use the precomputed states that the Perth Live model reverts back to in cases where errors are large. At the surface level the OD matrix is also a complex object, consisting of millions of entries. However, because only certain patterns of these entries occur in reality, meaning that the intrinsic dimension of the demand is much lower, it is possible to sample the demand space. This can be verified by simple dimensionality-reduction techniques, such as principle component analysis (PCA), which shows

that the pattern OD matrices exist in a latent space that is lower than 50 dimensions. In other words, each OD matrix OD_t can be constructed to a reasonable degree of accuracy from only 50 basis matrices OD_i, which can be thought of as possible demand modes,

$$OD_t \approx \sum_{i=1}^{50} \alpha_{i,t} OD_i \, .$$

The time-dependent coefficient vectors $\boldsymbol{\alpha}_t$ fully describe the OD matrices that are allowable in the Perth Live model. In order to sample this space randomly, we first select a point in the weekly cycle and its corresponding vector $\boldsymbol{\alpha}_t$, then perturb this vector with a multiplicative ξ_i factor that is randomly selected from a small distribution around 1 (e.g. $\xi \sim \mathcal{N}(1, 0.01)$). The result of this is a randomly generated OD matrix that is a reasonable demand for the Perth Live model,

$$OD^* = \sum_{i=1}^{50} \xi_i \alpha_{i,t} OD_i \, .$$

The parameters can be sampled by the same procedure used in Section 3.4.2. Following this procedure it is possible to generate a large sample of inputs with a distribution which corresponds to what is expected when the model is run online. Running the model with these inputs we get a set of corresponding outputs which we can combine in a database to describe the input-output relationship of the model in typical use. From this database we can then use some machine learning technique to build an emulation.

The potential of this approach is significant, possibly allowing for true real-time optimisation of the model parameters and demand adjustment via the emulation model, while maintaining the full functionality of the theory driven Aimsun model. However, due to the limitations of the project timeline and the work required for this approach, we do not have concrete results and can only discuss these ideas for future study.

7. Conclusions

Using machine learning and AI techniques, this project successfully developed methods for calibrating the driver beheaviour and supply-side parameters for Aimsun Live. One of the key challenges for the task of calibrating this type of large-scale, noisy simulation model is the computational requirements of running many simulations. This is addressed by reducing the dimensionality of the problem with sensitivity analysis, and using Bayesian optimisation, which is designed to be efficient with function evaluations. This approach is possible when working with only a few typical days, but is not extendable to optimising a year worth of data. In order to address this problem without using massive computational resources, a new optimisation framework is developed to efficiently reuse simulation results across multiple days in the dataset. The resulting optimal parameter values capture variations in driver behaviour throught the day and year.

The patterns in driver behaviour, alongside raw traffic volume data, are analysed via hierarchical clustering methods. In both cases fine-grained meaningful patterns are observed, including overnight, early morning and evening, morning peak, middle of day, and afternoon peak from the raw data. More complicated patterns are present in the optimal parameter values, distinguishing between overnight and holidays, pre-peak/early-peak, peak/late-peak, and Saturdays. Notably, the patterns do not match exactly, indicating that there are changes in driver behaviour that are not easily observable from only the detector counts.

The parameter values corresponding to each pattern are validated against the default values of the model, resulting in small improvements generally, with particularly good results for the morning peak and holiday periods, although the latter result is most likely a function of the unusual demand patterns during these periods. There were also some cases where the performance is marginally worse, particularly in the afternoon, though on average there is small improvement. The scale of this improvement is not entirely unexpected, since it is expected that the correct demand is a more important factor in the model outcome.

It is important to note that the patterns presented in this report are specific to the Perth Aimsun Live model and the 2019 historical data. The use of pattern demands without OD adjustment may have also influenced the results. Nevertheless, **the key contribution of this project is the establishment of a methodology for the systematic calibration of driver behavior and supply-side parameters**, which can be applied to other models and datasets in the future.

Several metrics are tested for measuring the prediction quality and they were evaluated on their ability to separate distinct parameter values with high resolution. Based on this analysis the root-mean-square error (RMSE) is the best metric, which is in line with conventional wisdom regarding optimisation. The percentage of detector stations with $GEH > 5$ is also reasonable, though there are issues with metrics involving threshold values like this.

The results presented in this report show two main approaches. The first approach involves clustering days into several groups and learning a time-dependent parameter schedule for each class of day using Bayesian optimisation, which is similar to the current methodology. The second approach involves learning fine-grained clusters of optimal parameters and applying these to the model. This has the advantage of using the commonality between different points in the weekly cycle, though it is more difficult to know which parameter cluster to use at any given moment.

Further improvements could be made to this work in future studies by the following recommendations.

- Apply similar techniques to calibrate offline models.

- Based on the results in Section 5, root-mean-squared error (RMSE) should be adopted as the cost function for performing optimisation.
- The source of unexpected optimised parameter values (Table 6, green) could be investigated.
- Model emulation could be explored to develop a simplified surrogate model that approximates the full model. This could enable:
    - True real-time parameter optimisation via accelerated model evaluations.
    - Better testing of cost functions. The ability of optimisation to recover the the known ground truth parameters of the emulation indicates the most suitable cost function.
- Clustering results could be improved by considering an adaptive distance threshold across different branches of the hierarchical tree structure. For example, a finer clustering resolution may exist at lower levels for weekdays, while a coarser resolution is sufficient for Sundays.
- More data sources could be included for calibration (e.g., vehicle speeds) if available.

In summary, this project demonstrated frameworks for the automated calibration of the Perth Live model. The methods are able to identify variations in driver behavior via optimised parameters, and use these to improve model performance.

References

[1] M. Friedrich, E. Pestel, C. Schiller and R. Simon, “Scalable GEH: A quality measure for comparing observed and modeled single values in a travel demand model validation,” *Transportation Research Record,* pp. 722-732, 2019.