Traffic congestion monitoring by using data-driven simulation and Incident impact analysis

Zheyuan (David) Liu

Summer Vacation Student 2017/2018

Table of Contents

1. Introduction

Traffic congestion in major cities is a critical issue for traffic management that can cause serious influence on the overall traffic behaviour. Moreover, random incidents could induce further impact on major corridors. Therefore, it is important to investigate traffic congestion and its contributory factors so that one could understand the nature of the traffic behaviour, foresee the congestion, predict its characteristics, and even reduce its impact by pre-planning. To achieve the goal, transport modelling can be used to simulate traffic behaviours under different scenarios, and data-driven models are helpful in investigating the outcome.

In this project, the main objective is to use data-driven simulations in Aimsun to investigate the traffic behaviour under congestions, as well as the impact of the incident. In addition, steps have been taken to predict the incident duration using data-driven modelling. The project contributes to the ultimate objective of the ADAIT team at Data61 to analyse and predict traffic congestions in real-time based on external factors.

The main challenges for this project can be summarised as:

- analysing the impact of the incident in a congested network with a large number of sections and nodes;
- accurately simulating the traffic conditions using latest available data;
- correlating external events (e.g. public school) with traffic condition;
- modelling the long-term impact of a new incident.

This report contains the following sections. *Project description* describes the main objectives of the project as well as the transport model used for the simulations. It also details the subnetwork was targeted at. *Preliminary data collection* is regarding the data collected in preparation for the simulation and the data-driven modelling process. *Aimsun model calibration* documents the steps that taken in calibrating the Aimsun transport model for the specific date 07/06/2017. It involves multi-layer static traffic adjustments as the author was mainly targeting at the Victoria Corridor subnetwork. *Dynamic traffic simulation* shows the settings and the results of the mesoscopic and microscopic simulations, which involve comparisons between scenarios with and without incidents. *Automatic incident simulation* elaborates on the details of the automatic incident import, simulation and results exporting Python scripts composed by the author. *Correlation analysis* details the data-driven modelling for investigating in the correlation between contributory factors and the incident duration, as well as the incident duration predicting process. Finally, *Future work* and *Conclusion* summarise the limitations of the current study and outlines the future perspectives of the project, and conclude the work.

2. Project Description

2.1. Objective

This project has three main objectives:

- Traffic congestion monitoring by using data-driven simulation
- Incident impact analysis using scenario testing & performance evaluation of the traffic condition
- Data-driven prediction modelling of incident duration

2.2. Simulation model

The simulation model used for this project is labelled *Aimsun Model STM_Victoria_Rd_Dec2017*, it contains the traffic network for the entire Sydney. The focus in this project, however, is the Victoria Corridor subnetwork (Figure 1). The reasons for targeting at the subarea are as follows. First, it is both time-consuming (up to several hours) and computationally intensive to perform simulations for the entire Sydney network. Moreover, to investigate the behaviour of traffic congestions and the impact of the incidents, we need to perform dynamic traffic simulation, which is virtually impossible to accomplish for such large areas. Also, checking the correctness of the details of the model can be very difficult for large networks. This work represents a continuation of previous studies which are mixing the traffic simulation modelling together with data-driven aproaches, via various simulation environments, traffic signal control, or multi-agent approaches (see previous works in [1]-[2]..[19]). The current approach will be focusing more on microscopic traffic simulation modelling, powered by data driven streams.

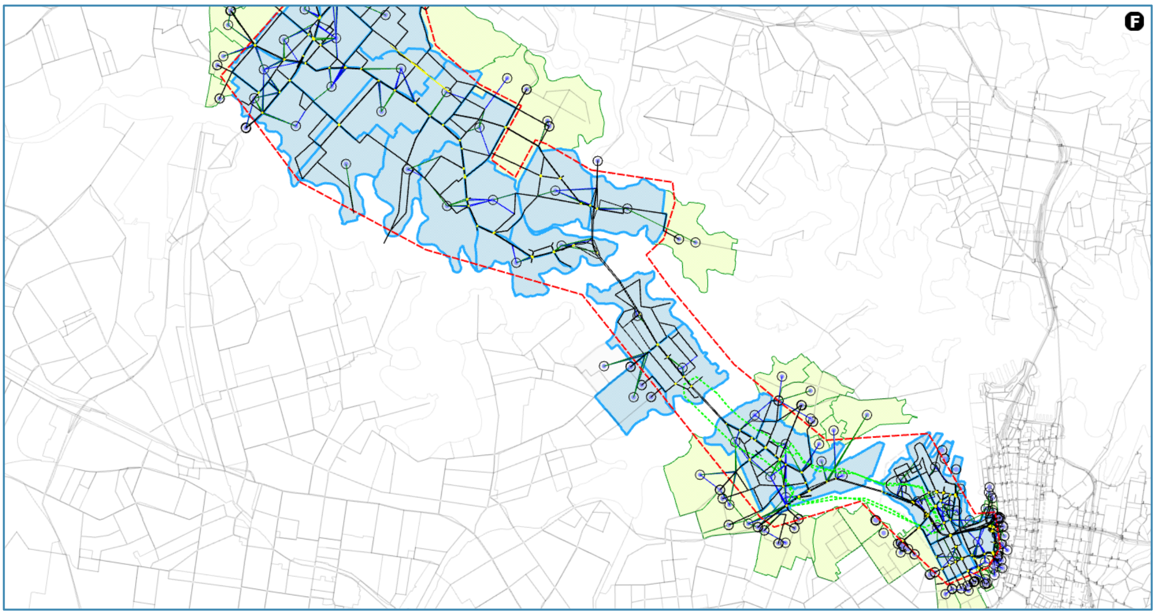


Figure 1 The Victoria Corridor in Aimsun

3. Part I – Preliminary data collection

Data collection is the first step in incident impact analysis. Details of the collected data are as follows.

3.1. Traffic dataset

Collecting traffic records is necessary for the simulation of incident events. It is also critical for the data-driven analysis on incident durations (as elaborated in section 0). Specifically, the author aimed at collecting data on events that have a direct impact on traffic behaviour, including traffic incidents, roadworks and major events.

Initially, the author was expecting to obtain the historical data from the official sources, such as the NSW Transport or the Roads and Maritime Services (RMS). However, after contacting the authority, the author was informed that the official sources do not store or publish the historical data. However, the real-life traffic reports can be accessed through the *RMS Developer API* and the *LiveTraffic NSW* website (https://www.livetraffic.com/). Therefore, the author composed a Python script that could record the real-time data stream. The script ran 24-hour nonstop and refreshed every 15-minute to store the updated events into local CSV files, as shown in Figure 2.

As mentioned hereinbefore, the stored event data is categorised into three types- incidents, roadworks and major events. The original data stream is in the form of JSON files; an auto-process script will read through the events and organise the columns for better readability. Finally, the processed events will be appended to the local CSV files, as shown in Figure 3.

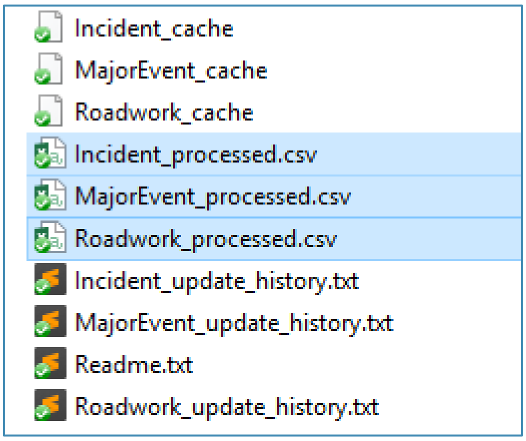
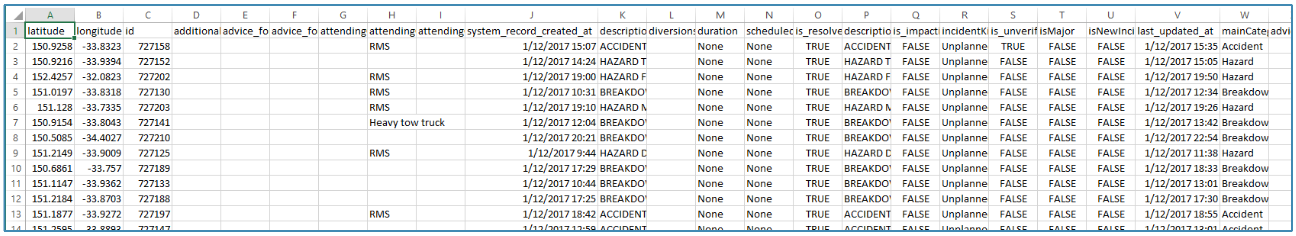


Figure 2 The recorded traffic dataset



	latitude	longitude	id	additional	advice_fo	advice_fo	attending	attending	attending	system_record_created_at	descriptio
2	150.9258	-33.8323	727158					RMS		1/12/2017 15:07	ACCIDENT
3	150.9216	-33.9394	727152							1/12/2017 14:24	HAZARD T
4	152.4257	-32.0823	727202					RMS		1/12/2017 19:00	HAZARD F
5	151.0197	-33.8318	727130					RMS		1/12/2017 10:31	BREAKDO
6	151.128	-33.7335	727203					RMS		1/12/2017 19:10	HAZARD N
7	150.9154	-33.8043	727141					Heavy tow truck		1/12/2017 12:04	BREAKDO
8	150.5085	-34.4027	727210							1/12/2017 20:21	BREAKDO
9	151.2149	-33.9009	727125					RMS		1/12/2017 9:44	HAZARD D
10	150.6861	-33.757	727189							1/12/2017 17:29	BREAKDO
11	151.1147	-33.9362	727133							1/12/2017 10:44	BREAKDO
12	151.2184	-33.8703	727188							1/12/2017 17:25	BREAKDO
13	151.1877	-33.9272	727197					RMS		1/12/2017 18:42	ACCIDENT

	diversions	duration	schedulec	is_resolve	descriptio	is_impacti	incidentKi	is_unverif	isMajor	isNewInci	last_updated_at	mainCate	advi
2		None	None	TRUE	ACCIDENT	FALSE	Unplanne	TRUE	FALSE	FALSE	1/12/2017 15:35	Accident	
3		None	None	TRUE	HAZARD T	FALSE	Unplanne	FALSE	FALSE	FALSE	1/12/2017 15:05	Hazard	
4		None	None	TRUE	HAZARD F	FALSE	Unplanne	FALSE	FALSE	FALSE	1/12/2017 19:50	Hazard	
5		None	None	TRUE	BREAKDO	FALSE	Unplanne	FALSE	FALSE	FALSE	1/12/2017 12:34	Breakdow	
6		None	None	TRUE	HAZARD N	FALSE	Unplanne	FALSE	FALSE	FALSE	1/12/2017 19:26	Hazard	
7		None	None	TRUE	BREAKDO	FALSE	Unplanne	FALSE	FALSE	FALSE	1/12/2017 13:42	Breakdow	
8		None	None	TRUE	BREAKDO	FALSE	Unplanne	FALSE	FALSE	FALSE	1/12/2017 22:54	Breakdow	
9		None	None	TRUE	HAZARD D	FALSE	Unplanne	FALSE	FALSE	FALSE	1/12/2017 11:38	Hazard	
10		None	None	TRUE	BREAKDO	FALSE	Unplanne	FALSE	FALSE	FALSE	1/12/2017 18:33	Breakdow	
11		None	None	TRUE	BREAKDO	FALSE	Unplanne	FALSE	FALSE	FALSE	1/12/2017 13:01	Breakdow	
12		None	None	TRUE	BREAKDO	FALSE	Unplanne	FALSE	FALSE	FALSE	1/12/2017 17:30	Breakdow	
13		None	None	TRUE	ACCIDENT	FALSE	Unplanne	FALSE	FALSE	FALSE	1/12/2017 18:55	Accident	

Figure 3 Partial snapshot of the incident dataset

During the three-month period, the author recorded 4833 incidents,97 major events and 467 roadworks data for the entire Sydney area. The CSV files are stored in *Dropbox\David Liu Internship\LiveTrafficData* along with the 15-minute interval update history logs. For details of the Python scripts, please refer to Dropbox\David Liu Internship_scripts\LiveTraffic\ or the corresponding Git source at:

https://github.com/Cuberick-Orion/RMSTrafficHazard/tree/master/Auto_All_in_one

3.2. Data on potential factors

In addition to the traffic dataset, the author also manually collected dataset on any potential factors that could affect the traffic behaviours. This includes weather, public holiday and school events from 2016 to 2018 (where applicable).

The weather data was obtained from the Australian Government Bureau of Meteorology at:

http://www.bom.gov.au/climate/data-services/station-data.shtml

The observation station chosen is: 066062 Sydney (Observatory Hill). The station is the largest observation station near the Victoria Corridor. For the year 2017 and the year 2016 from October to December, weather data includes the following items (daily), as shown in Figure 4:

temperature/Celsius (min),

temperature/Celsius (max),

temperature/Celsius (average),

temperature/Celsius (at 9 AM),

temperature/Celsius (at 3 PM);

rainfall/mm;

wind/(km/h) (max),

wind/(km/h) (at 9 AM),

wind/(km/h) (at 3 PM);

cloud/oktas (at 9 AM),

cloud/oktas (at 3 PM);

humidity/% (at 9 AM),

humidity/% (at 3 PM).

For the year 2016 from January to September, only the max and the average temperature are available online.

The CSV data file is stored at *Dropbox\David Liu Internship\WeatherData*.

	A	B	E	F	G	H	I	J	K	L	M	N	O	P	Q
1		Date	Temperature (°C)				Rainfall (mm)	Wind (km/h)				Cloud (oktas)		Humidity (%)	
2	Month	Day	Temp_max	Temp_avg	Temp_9am	Temp_3pm	Rainfall	Wind_max	Wind_max_time	Wind_9am	Wind_3pm	Cloud_9am	Cloud_3pm	Hum_9am	Hum_3pm
290	10	14	21.3	15.15	15.1	19.4	0.2	30	2:20	17	19	1	1	50	37
291	10	15	23.9	17.5	15.9	23	0	50	17:30	11	26	0	0	57	36
292	10	16	29	22.45	23.1	25.4	0	56	14:49	22	28	0	1	31	37
293	10	17	22.4	22.05	22.4	14.5	0			28	15	7	7	43	78
294	10	18	25.8	18.65	18.3	25.5	3	48	13:09	19	26	1	3	39	23
295	10	19	24.5	20.35	19.1	22.9	0.2	50	9:58	24	6	3	2	30	20
296	10	20	21.6	19	19	20.4	0	44	16:24	19	26	6	5	47	47
297	10	21	25.2	20.25	21.5	23.3	0	46	14:45	20	26	5	7	55	55
298	10	22	18	17.05	17	16.4	11.8	57	17:49	15	22	7	7	84	69
299	10	23	19.8	15	13.1	17.5	0.6	59	9:36	22	28	7	2	53	42

Figure 4 Partial snapshot of the weather data

The public holiday data includes the scheduled starting and end dates of the events and the affected area and population (where applicable). Since public holidays are mostly affecting the entire population, no further location information is required, as shown in Figure 5. The CSV data file is stored at *Dropbox\David Liu Internship\HolidayData*.

	A	B	C	D	E	F	G	H
1	id	holiday_name	start_date	end_date	holidy_type	isRegional	area	details
2	PH000001	New Year's Day	1/01/2018	1/01/2018	Public	FALSE	NSW Wide	
3	PH000002	Australia Day	26/01/2018	26/01/2018	Public	FALSE	NSW Wide	
4	PH000003	Good Friday	30/03/2018	30/03/2018	Public	FALSE	NSW Wide	
5	PH000004	Easter Saturday	31/03/2018	31/03/2018	Public	FALSE	NSW Wide	
6	PH000005	Easter Sunday	1/04/2018	1/04/2018	Public	FALSE	NSW Wide	
7	PH000006	Easter Monday	2/04/2018	2/04/2018	Public	FALSE	NSW Wide	
8	PH000007	Anzac Day	25/04/2018	25/04/2018	Public	FALSE	NSW Wide	
9	PH000008	Queen's Birthday	11/06/2018	11/06/2018	Public	FALSE	NSW Wide	
10	PH000009	Bank Holiday	6/08/2018	6/08/2018	Public	FALSE	NSW Wide	Only banks and certain financial institutions receive the Bank Holiday
11	PH000010	Labour Day	1/10/2018	1/10/2018	Public	FALSE	NSW Wide	
12	PH000011	Christmas Day	25/12/2018	25/12/2018	Public	FALSE	NSW Wide	

Figure 5 Partial snapshot of the public holiday data

However, school events and school holidays data are more complicated, as different school- especially private schools- can have different calendars. So far, the author has not found comprehensive location data for public and private schools; the available NSW school list is only detailed to City/Suburb. Although websites with school addresses are available (e.g. https://www.australianschoolsdirectory.com.au/sydney-schools.php), there is no clear method to extract the dataset automatically.

Another issue regarding school holiday data is that, since the schedules of independent and Catholic schools are maintained by themselves, it is difficult to obtain the detailed data for each institution. Moreover, the websites of such individual schools usually only show the schedule for the current and future year. The author managed to use the web-archive service to access the previous record for certain web pages, but such method is not guaranteed to succeed.

Compared to school data, university data is easier to access, mainly because there are only six universities in the city of Sydney. However, the author found that the schedules of universities can be complicated, with multiple arrangements for different divisions/colleges throughout the year. Also, different students may go to different sessions (e.g. winter/summer sessions) and may have different examination periods. Right now, the author has collected data for all the sessions/terms available, but due to time limitations, the author only focused on the main calendar, which applies to most of the students.

A snapshot of the school events data is shown in Figure 6. The CSV data file is stored at *Dropbox\David Liu Internship\SchoolData*.

	A	B	C	D	E	F	G	H	I	J
1	id	type	start_date	end_date	School_type	school_subtype	duration	name	data_source	
2	SC000001	term_break	17/12/2015	27/01/2016	public	eastern_division		Summer Holidays	http://www.nswschool	
3	SC000002	term_break	17/12/2015	2/03/2016	public	western_division		Summer Holidays	http://www.nswschool	
4	SC000003	term_break	9/04/2016	26/04/2016	public	all		Autumn Holidays	http://www.nswschool	
5	SC000004	term_break	2/07/2016	18/07/2016	public	all		Winter Holidays	http://www.nswschool	
6	SC000005	term_break	24/09/2016	9/10/2016	public	all		Spring Holidays	http://www.nswschool	
7	SC000006	term_break	21/12/2016	29/01/2017	public	eastern_division		Summer Holidays	https://publicholidays.	
8	SC000007	term_break	22/12/2016	5/02/2017	public	western_division		Summer Holidays	https://publicholidays.	
9	SC000008	term_break	8/04/2017	25/04/2017	public	all		Autumn Holidays	https://publicholidays.	
10	SC000009	term_break	1/07/2017	17/07/2017	public	all		Winter Holidays	https://publicholidays.	
11	SC000010	term_break	23/09/2017	8/10/2017	public	all		Spring Holidays	https://publicholidays.	
12	SC000011	term_break	16/12/2017	29/01/2018	public	eastern_division		Summer Holidays	https://publicholidays.	
13	SC000012	term_break	16/12/2017	5/02/2018	public	western_division		Summer Holidays	https://publicholidays.	
14	SC000013	term_break	14/04/2018	30/04/2018	public	all		Autumn Holidays	https://publicholidays.	
15	SC000014	term_break	7/07/2018	23/07/2018	public	all		Winter Holidays	https://publicholidays.	
16	SC000015	term_break	29/09/2018	14/10/2018	public	all		Spring Holidays	https://publicholidays.	
17	SC000016	term_break	20/12/2018	29/01/2019	public	eastern_division		Summer Holidays	https://publicholidays.	
18	SC000017	term_break	20/12/2018	5/02/2019	public	western_division		Summer Holidays	https://publicholidays.	
19	SC000018	term_break	19/12/2015	27/01/2016	catholic	all		term4 break	https://web.archive.org	
20	SC000019	term_break	9/04/2016	26/04/2016	catholic	all		term1 break	https://web.archive.org	
21	SC000020	term_break	2/07/2016	18/07/2016	catholic	all		term2 break	https://web.archive.org	
22	SC000021	term_break	24/09/2016	10/10/2016	catholic	all		term3 break	https://web.archive.org	
23	SC000022	term_break	17/12/2016	30/01/2017	catholic	all		term4 break	https://cg.catholic.edu.	
24	SC000023	term_break	8/04/2017	25/04/2017	catholic	all		term1 break	https://cg.catholic.edu.	

Figure 6 Partial snapshot of the school event data

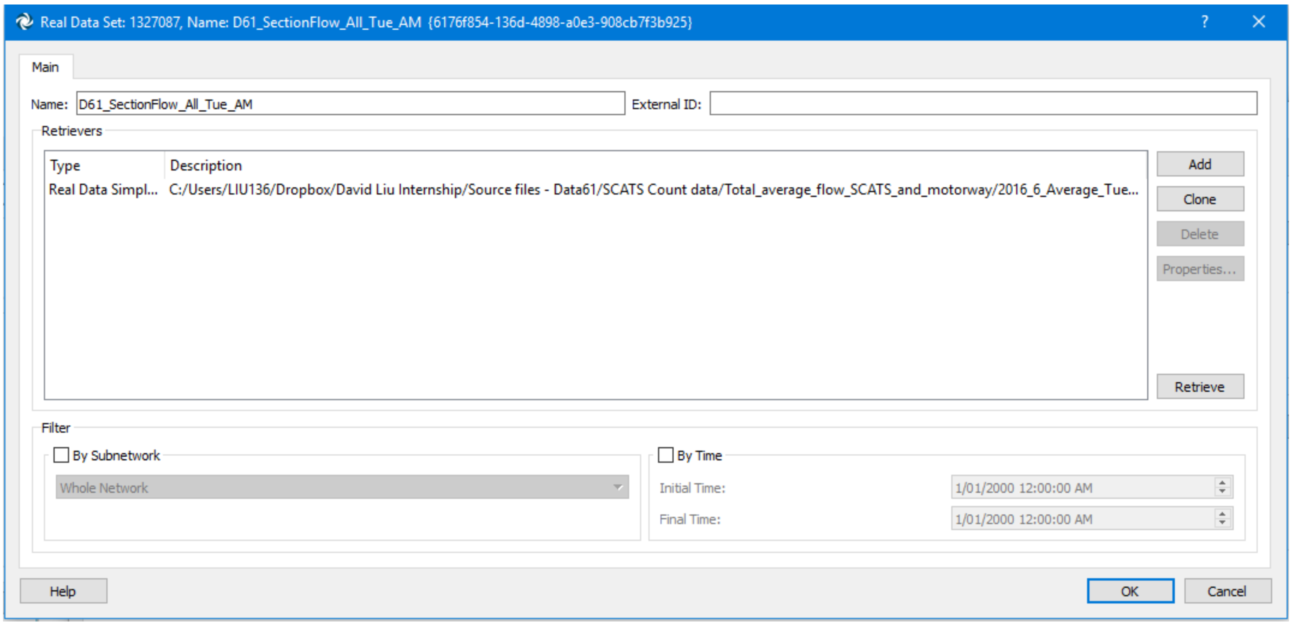
4. Part II – Aimsun model calibration

The aim of the model calibration is to match the simulation with real traffic conditions provided by traffic detectors, in this case, the traffic flow data of the SCATS sections. Such step is the prerequisite for monitoring of the traffic congestions. Detailed steps performed are elaborated on as follows.

4.1. Aimsun objects to be imported

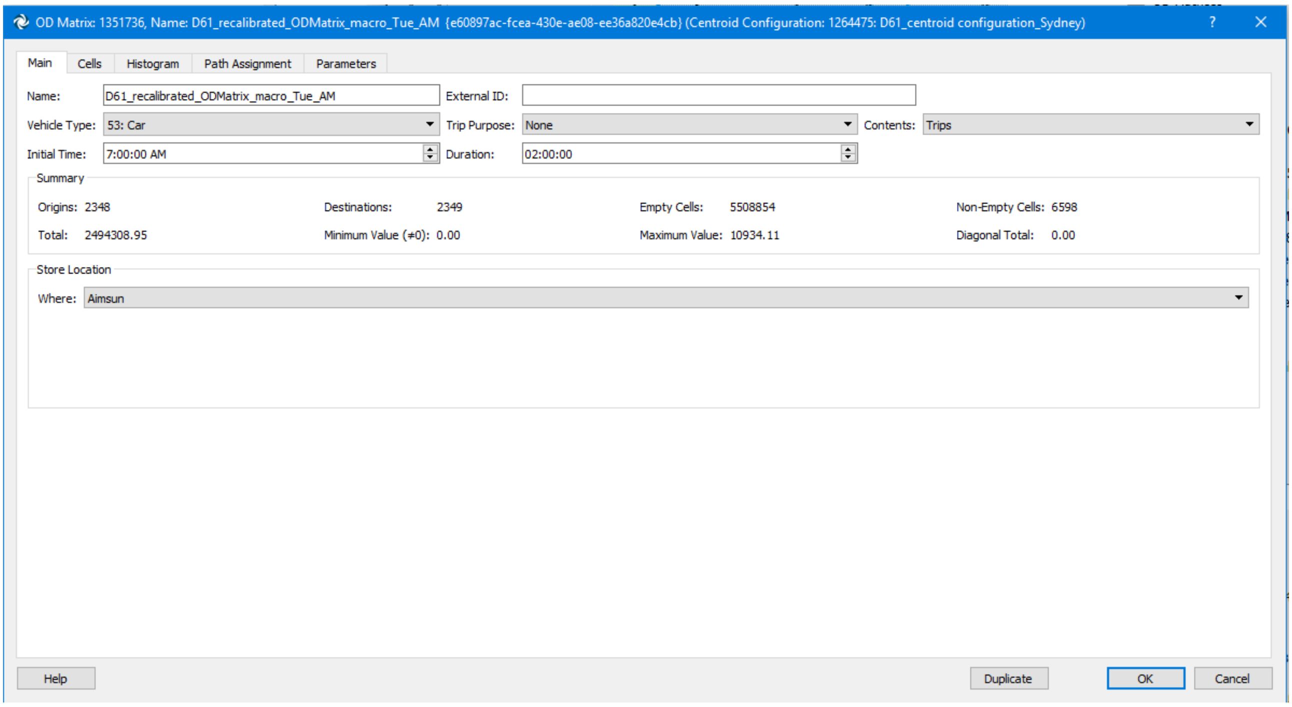
RDS: D61_SectionFlow_All_Tue_AM

Retrieved by GUI.

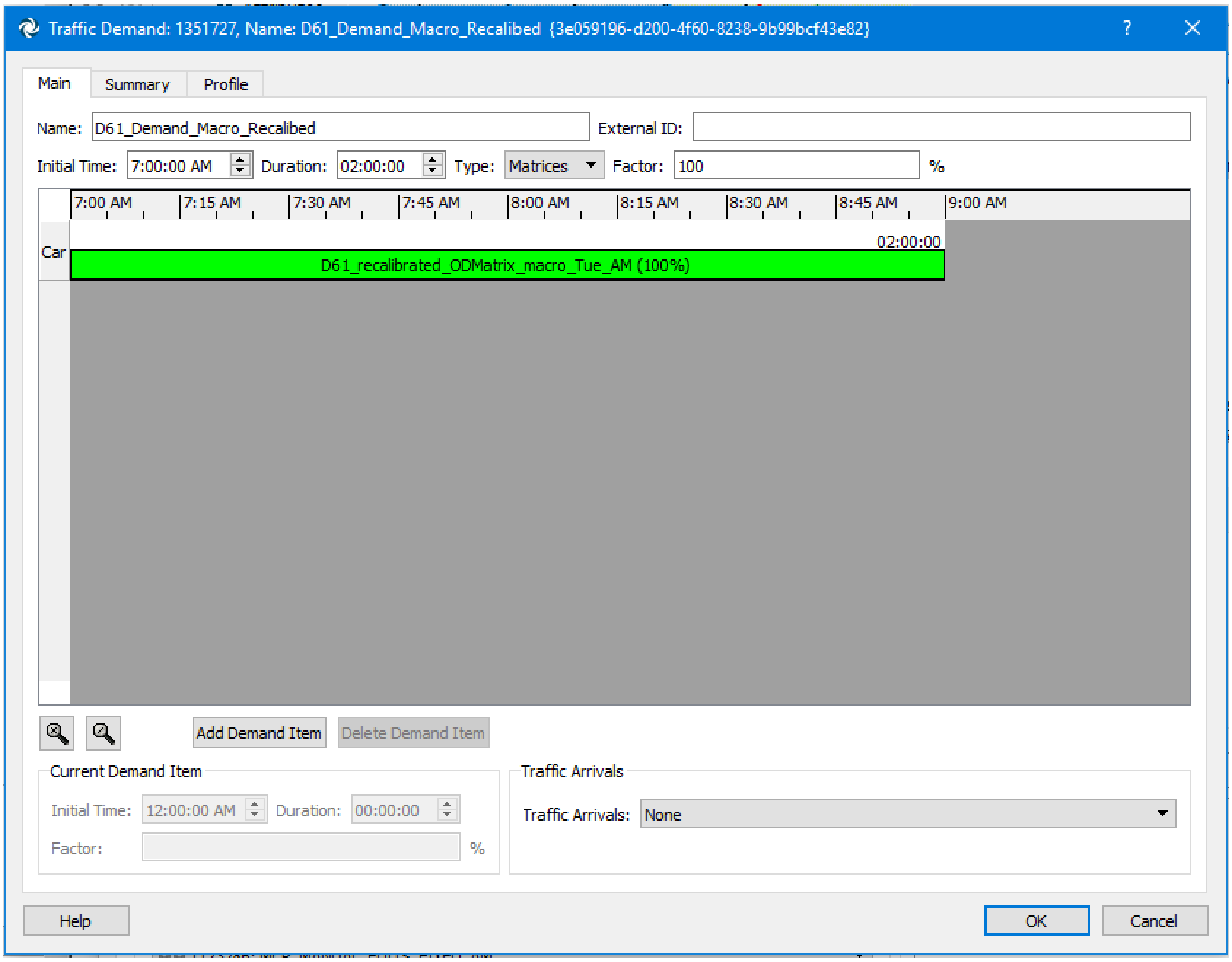


Demand Data → OD Matrices: D61_recalibrated_ODMatrix_macro_Tue_AM

Loaded using script (modified): 0_D61_Import O-D matrix

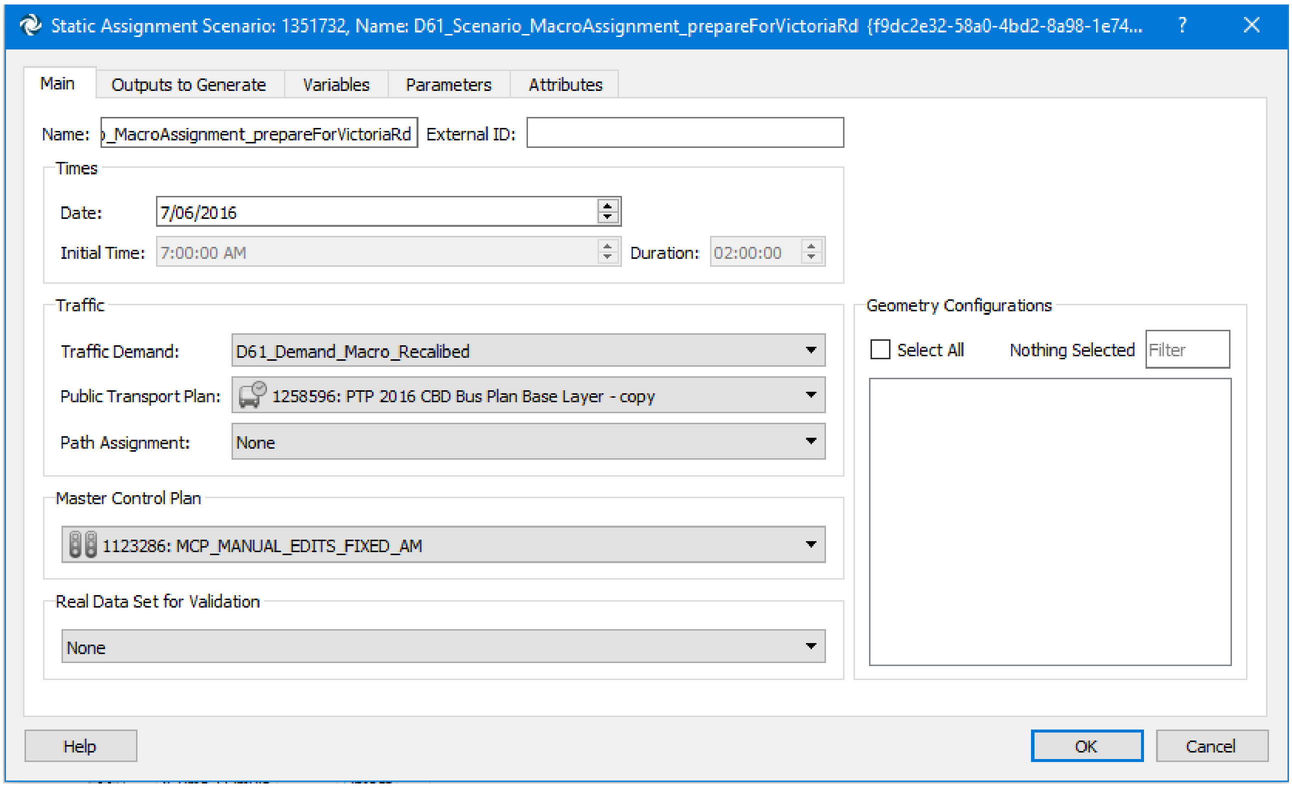


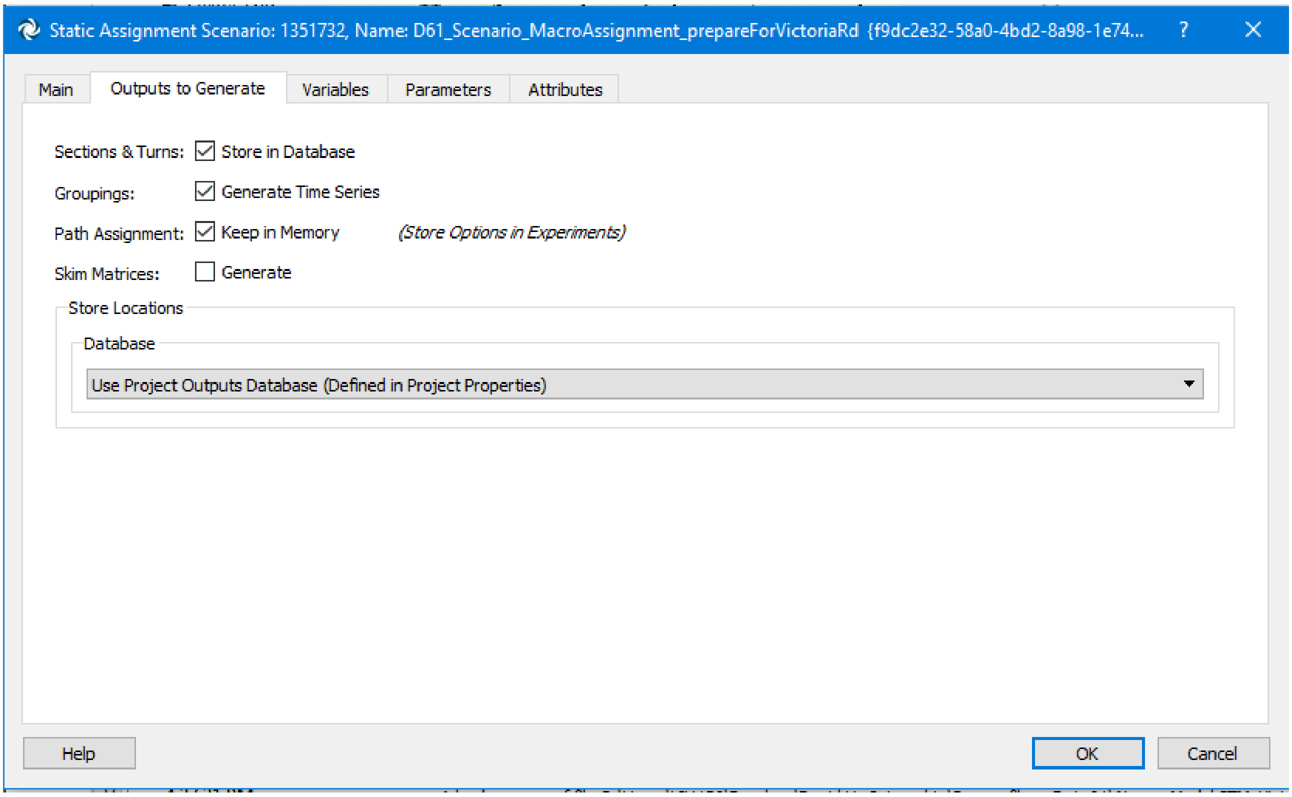
Use the OD matrix to generate Traffic Demand: D61_Demand_Macro_Recalibed



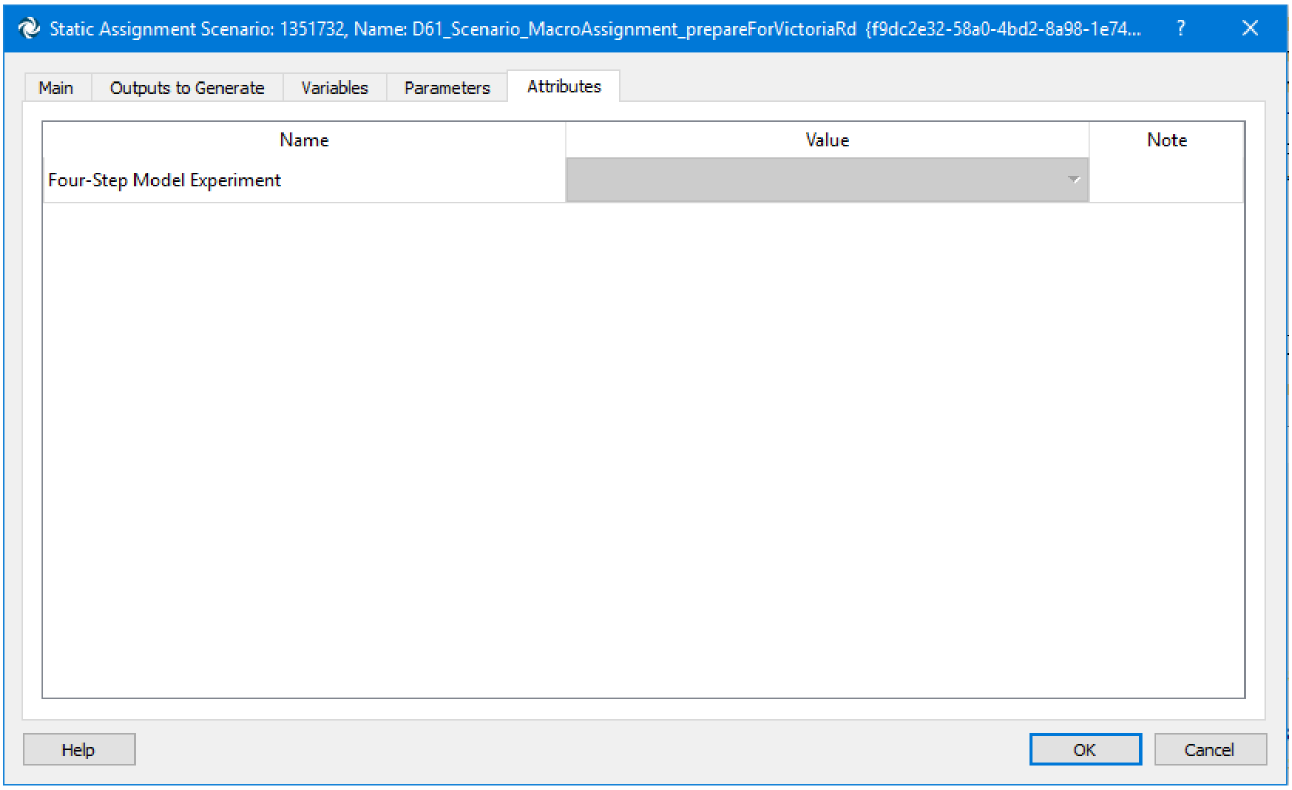
4.2. Run the Macro Static Traffic Assignment

Scenario: D61_Scenario_MacroAssignment_prepareForVictoriaRd

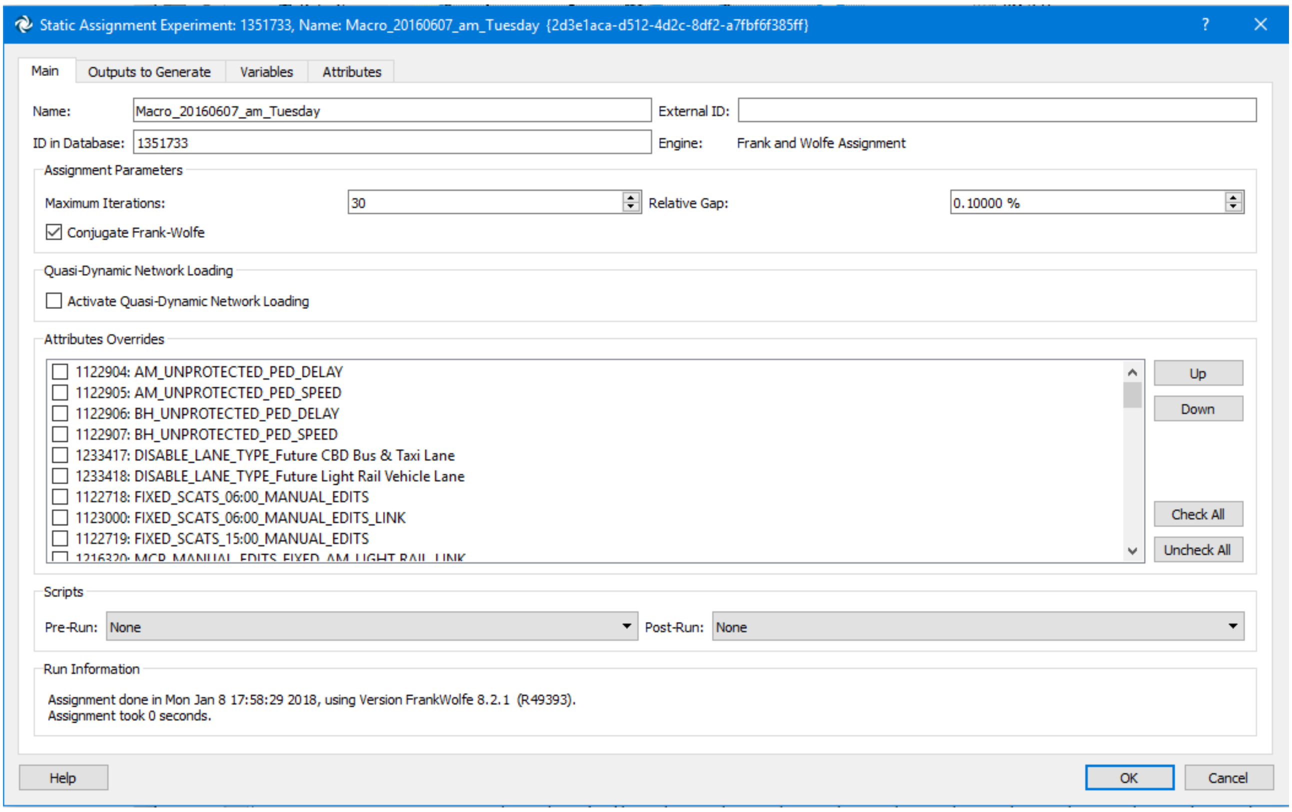

Static Assignment Scenario: 1351732, Name: D61_Scenario_MacroAssignment_prepareForVictoriaRd {f9dc2e32-58a0-4bd2-8a98-1e74...
Main
Outputs to Generate
Variables
Parameters
Attributes
Name: _MacroAssignment_prepareForVictoriaRd
External ID:
Times
Date: 7/06/2016
Initial Time: 7:00:00 AM
Duration: 02:00:00
Traffic
Traffic Demand: D61_Demand_Macro_Recalibed
Public Transport Plan: 1258596: PTP 2016 CBD Bus Plan Base Layer - copy
Path Assignment: None
Master Control Plan
1123286: MCP_MANUAL_EDITS_FIXED_AM
Real Data Set for Validation
None
Geometry Configurations
Select All
Nothing Selected
Filter
Help
OK
Cancel

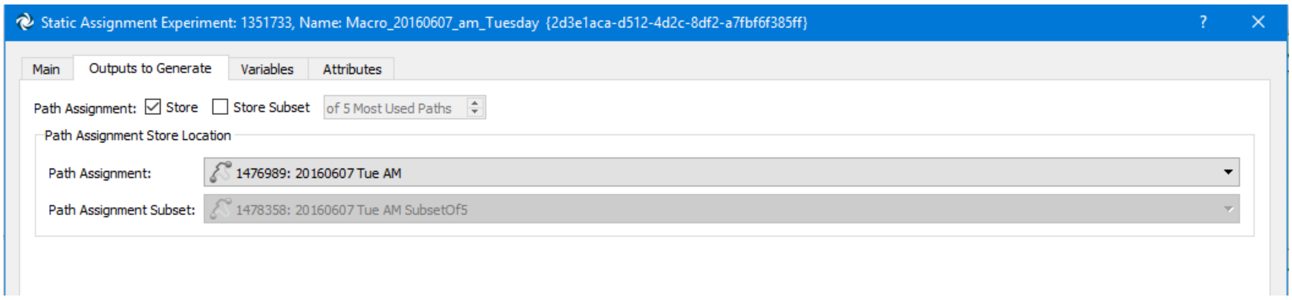

Static Assignment Scenario: 1351732, Name: D61_Scenario_MacroAssignment_prepareForVictoriaRd {f9dc2e32-58a0-4bd2-8a98-1e74...
Main
Outputs to Generate
Variables
Parameters
Attributes
Sections & Turns: Store in Database
Groupings: Generate Time Series
Path Assignment: Keep in Memory
(Store Options in Experiments)
Skim Matrices: Generate
Store Locations
Database
Use Project Outputs Database (Defined in Project Properties)
Help
OK
Cancel

Part II – Aimsun model calibration

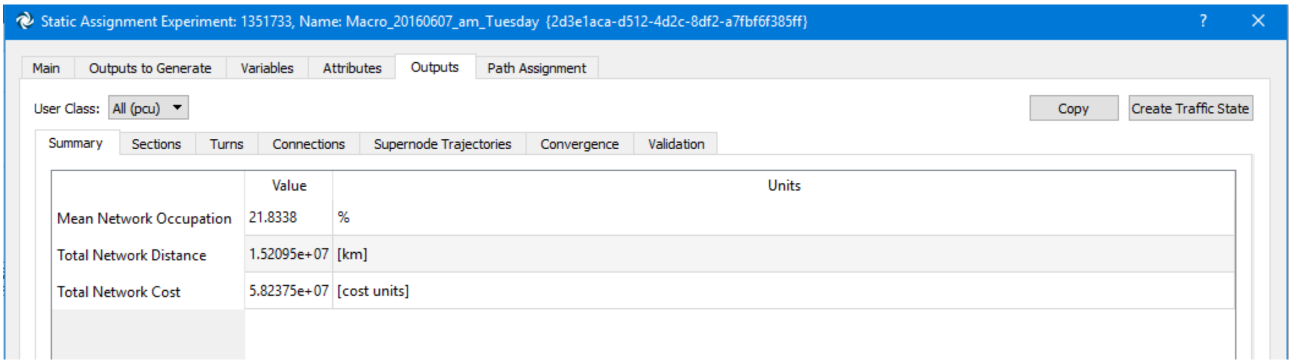


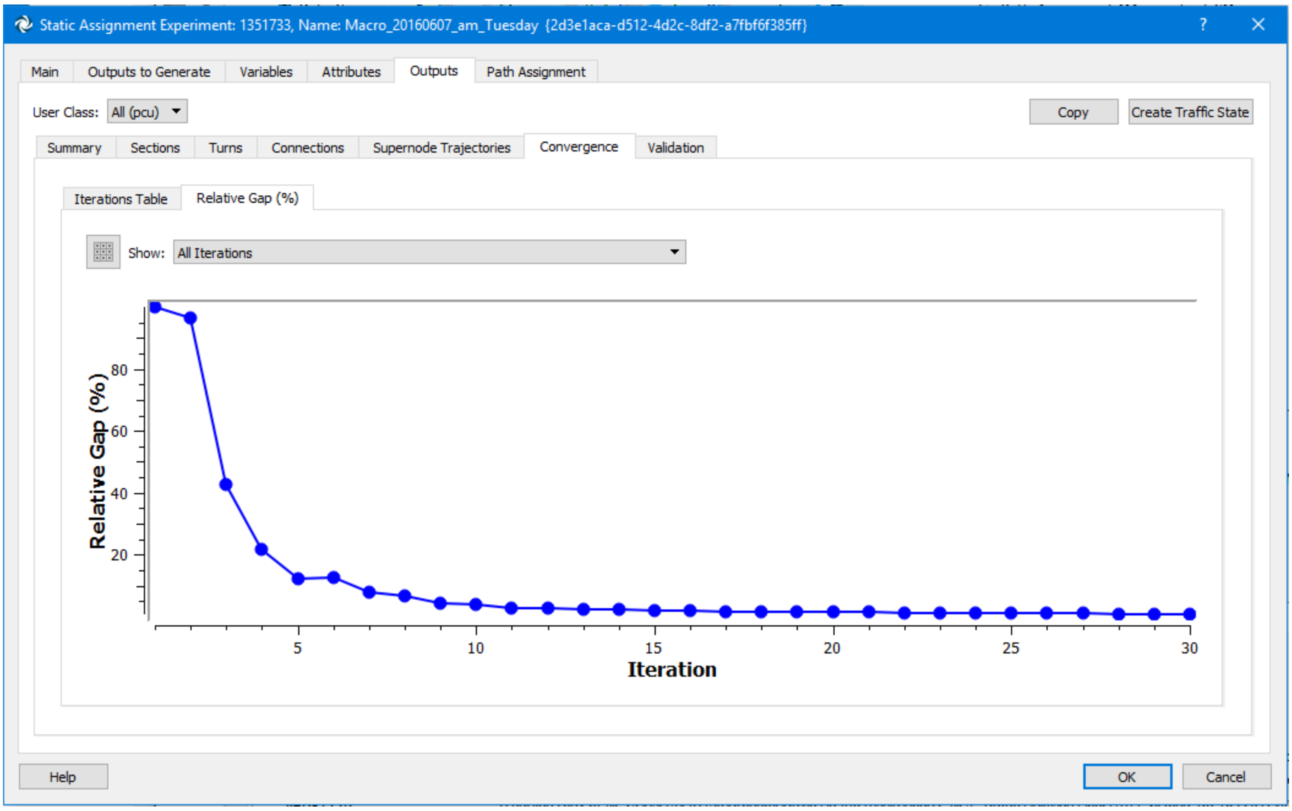
Static Assignment Experiment: Macro_20160607_am_Tuesday





Result for Macro simulation





Path Assignment generated from the Macro simulation:

- Path Assignment: 1476989
    - Name: **20160607 Tue AM**
    - Saved to file **PathAssignment_1476989.apa**
    - File Size: **1.96 GB**
    - File Size: **365 MB**

The output of the MACRO Assignment scenario is stored in the file:

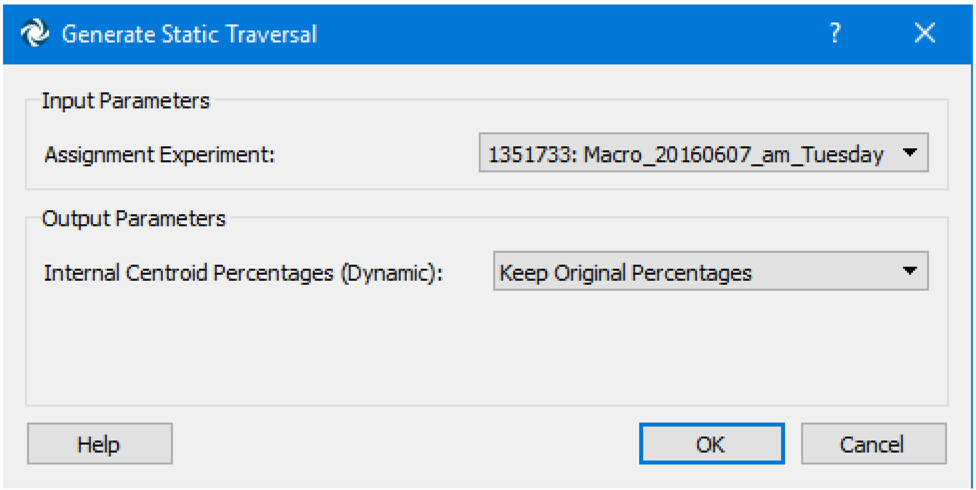
$Simulation Model Folder$\\Result\20160607\\MACRO_STA_Output.xlsx

The spreadsheet contains statistics for **51361** sections.

4.3. Prepare for the simulations in the subnetwork

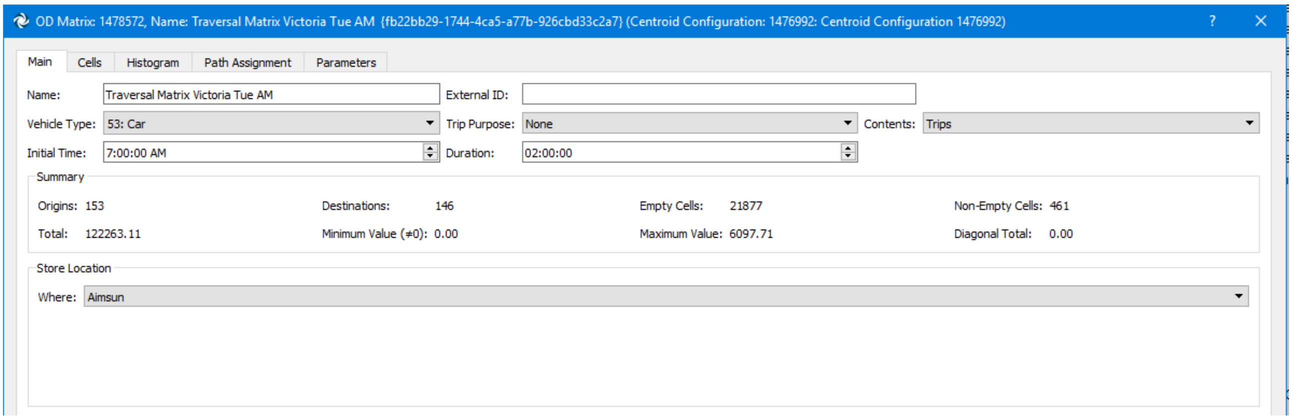
Generate Static traversal for the subnetwork

Right click on subnetwork- generate traversal –



Will generate an OD matrix for the subnetwork

Listed under the folder SUBNETWORK → Victoria Road Corridor → Centroid Configuration → Centroid Configuration 1376992 → OD Matrices

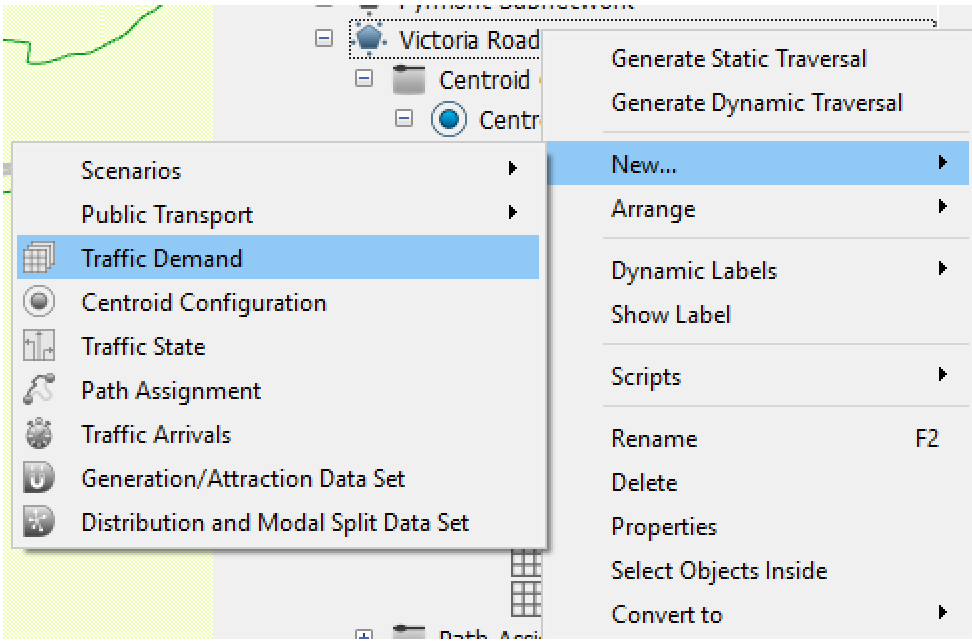


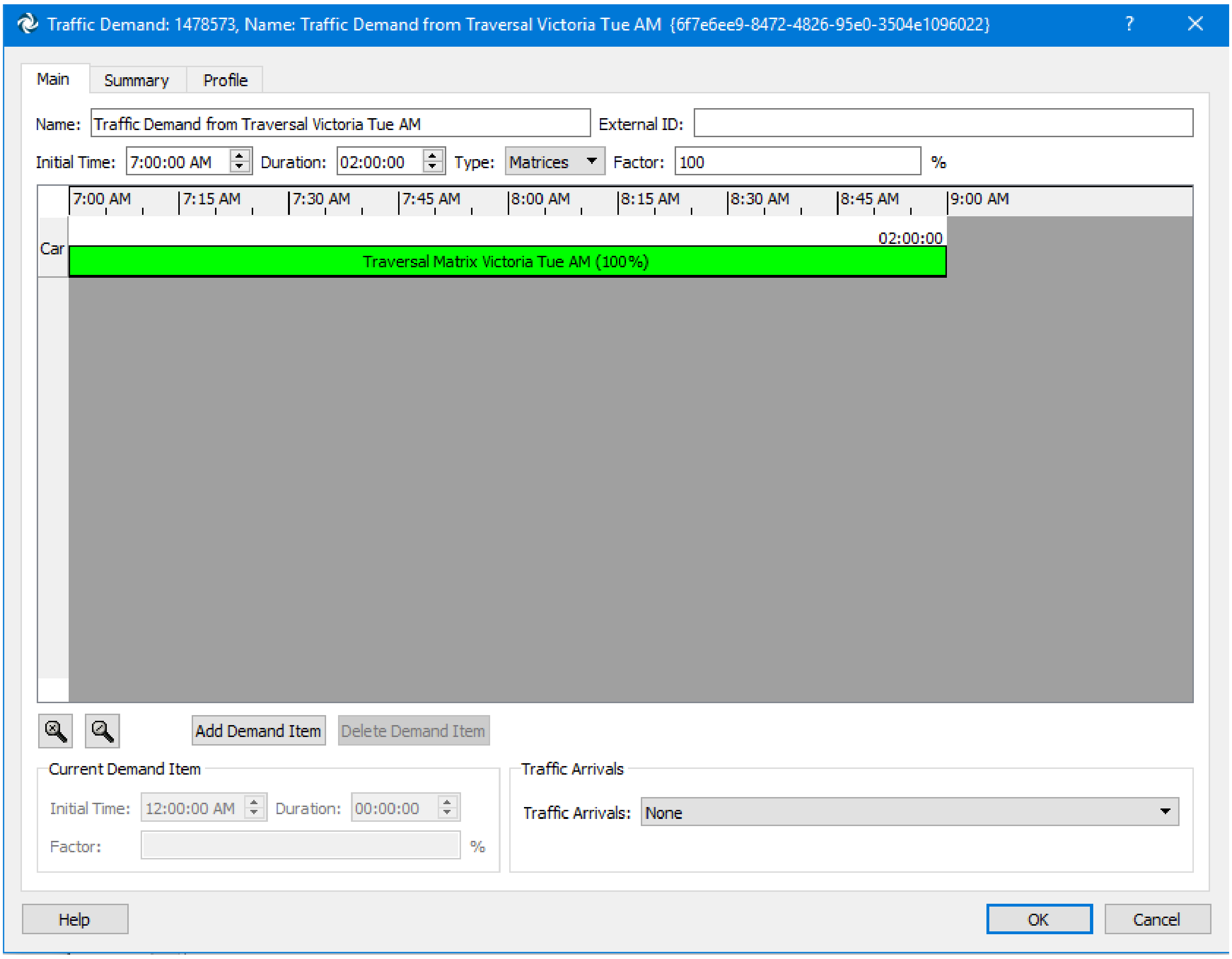
Manually change the Contents to Trips.

Create new traffic demand in the subnetwork

Add the Traversal Matrix to it.

To be used for Subnetwork Static OD Adjustment (the first level of adjustment).

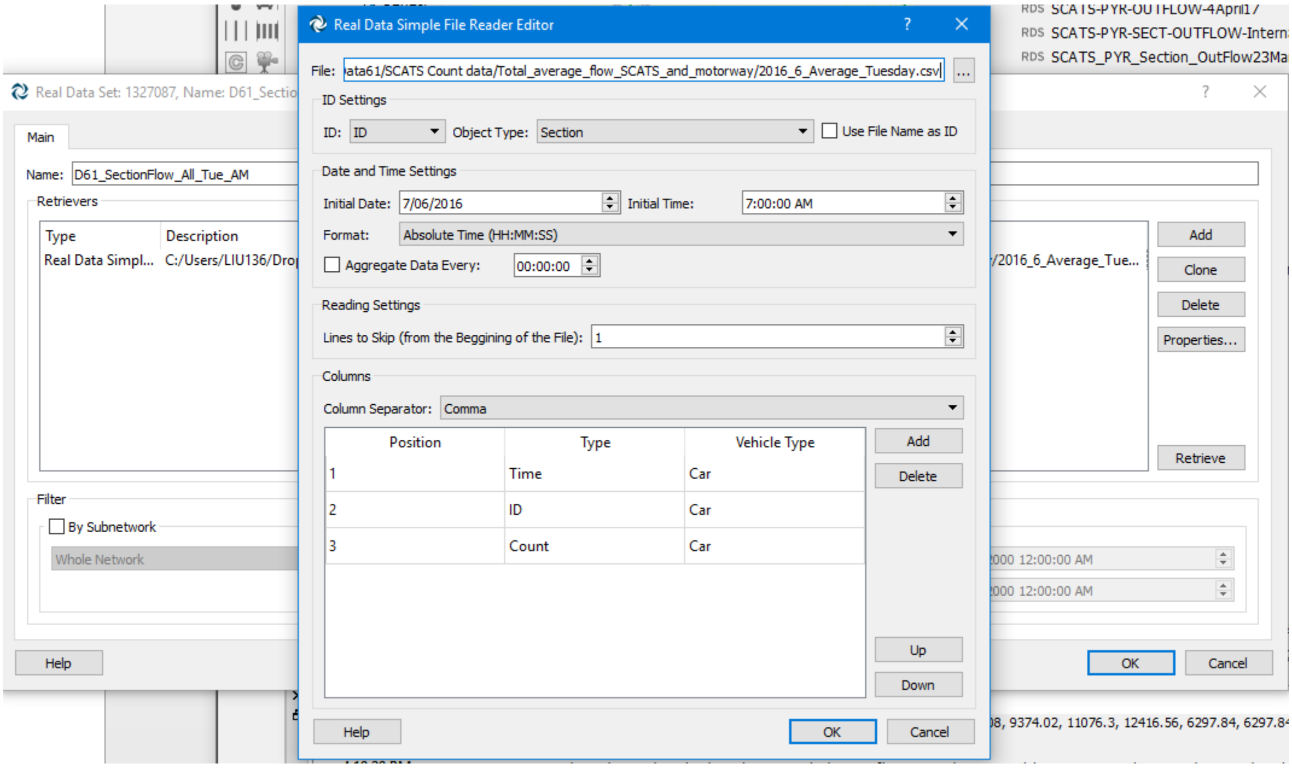




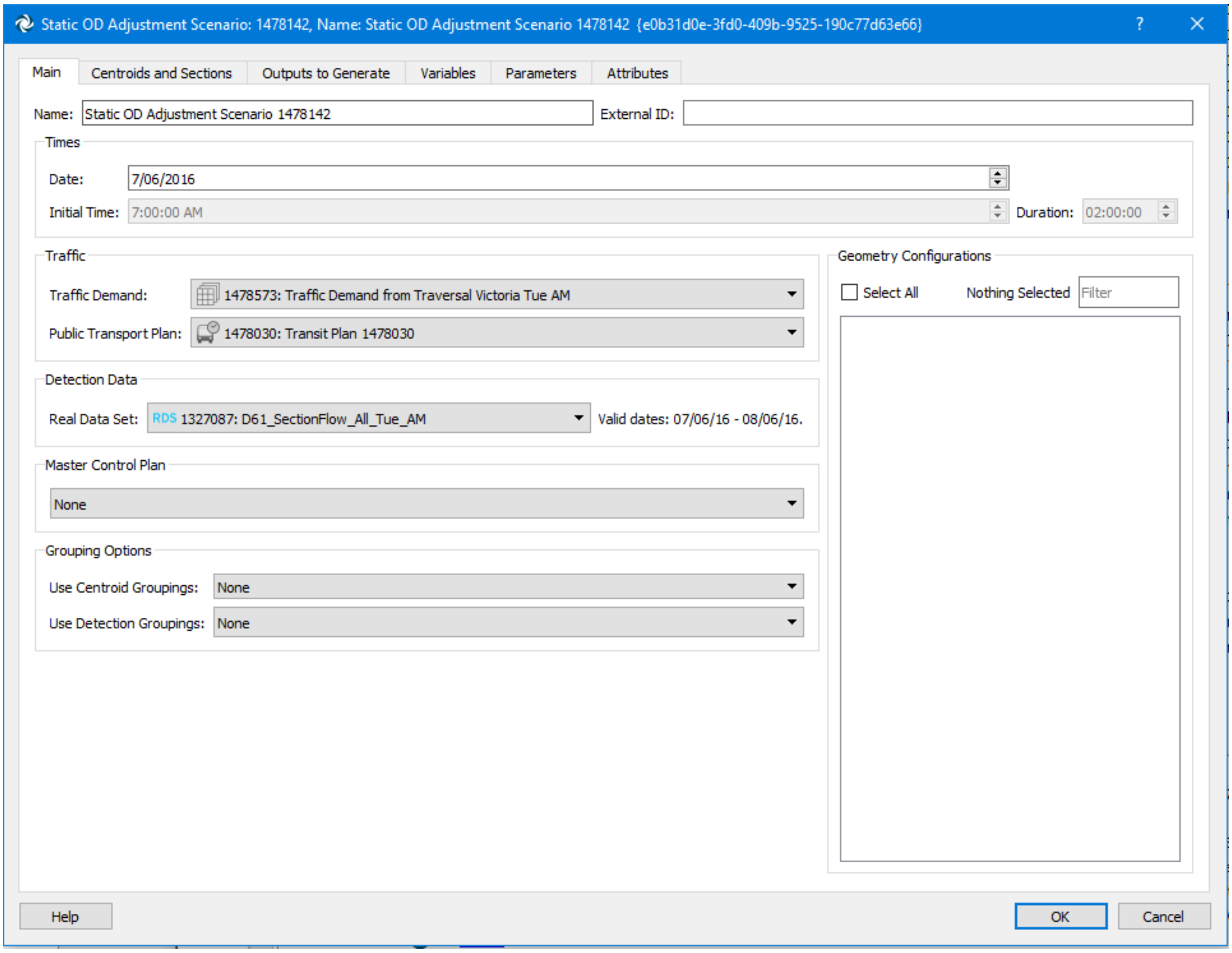
Note (instruction from the original document): Some might suggest splitting the demand separately for the warm-up and the rest of the simulation period, but in the training we did not do it.

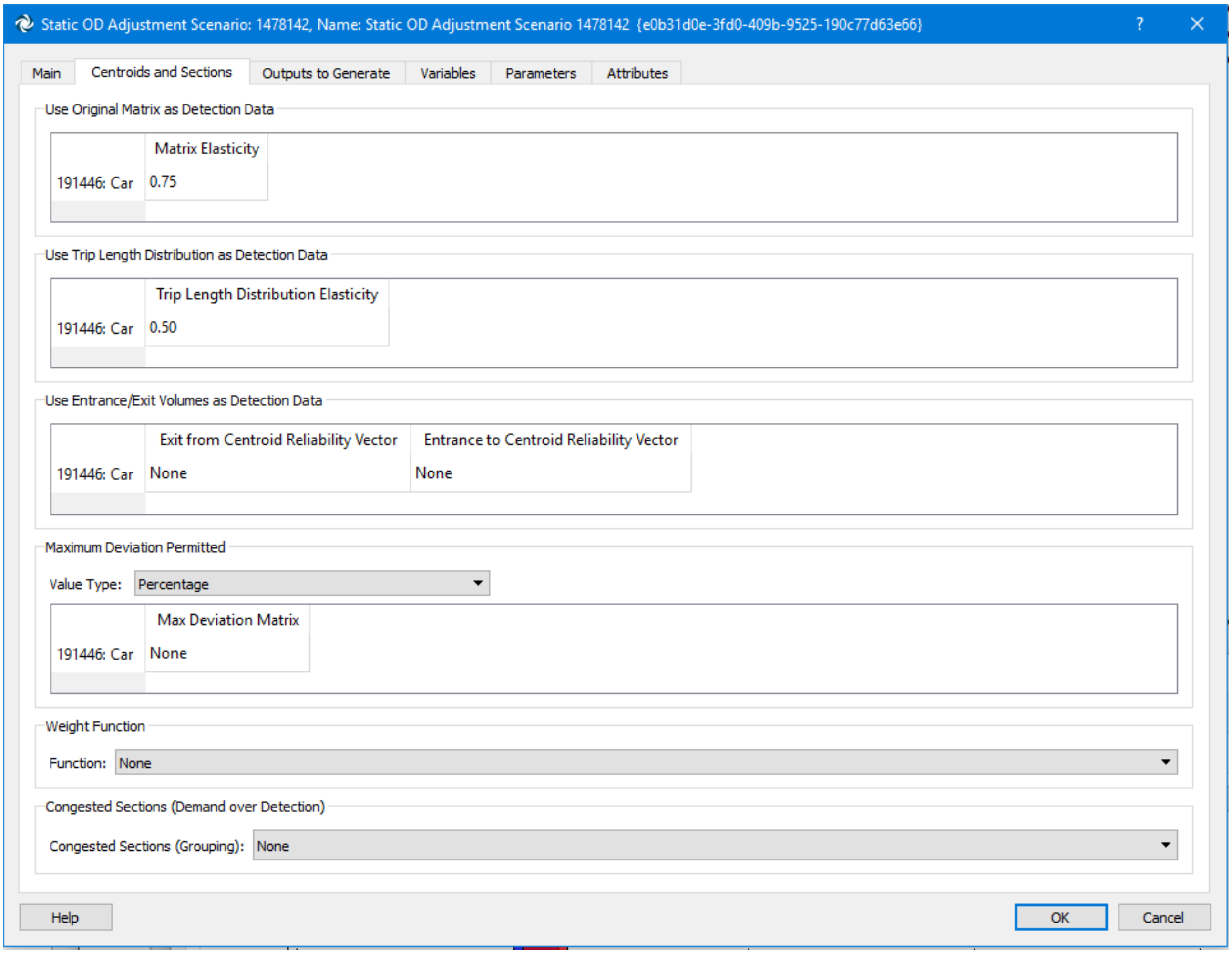
4.4. Static OD Adjustment

Retrieve the Real Data Set values if it has not been loaded

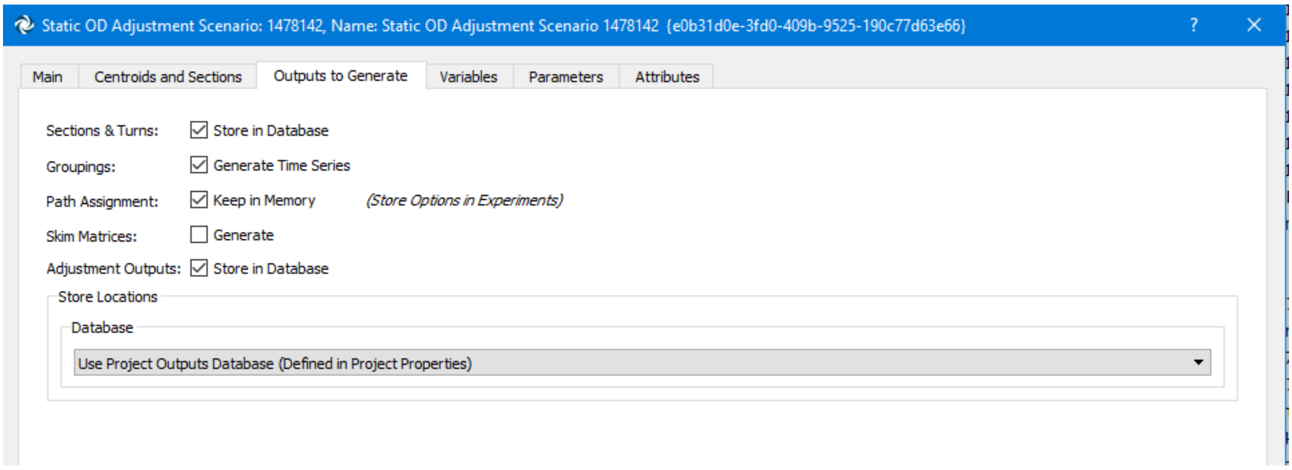


Create new STATIC OD Adjustment Scenario

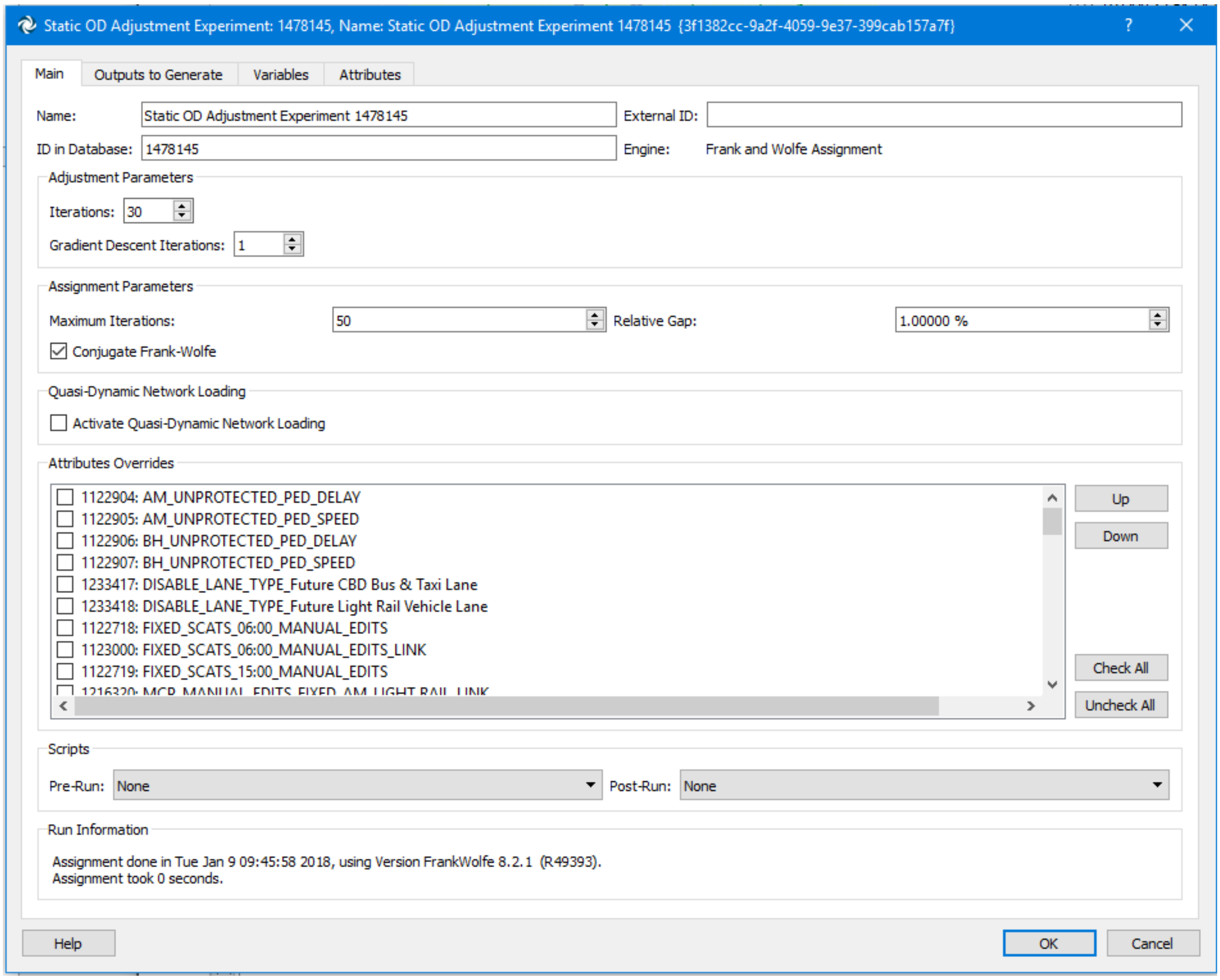




Simulations show that setting the Matrix Elasticity to a higher value can increase the validation R^2 (although the effect is limited).

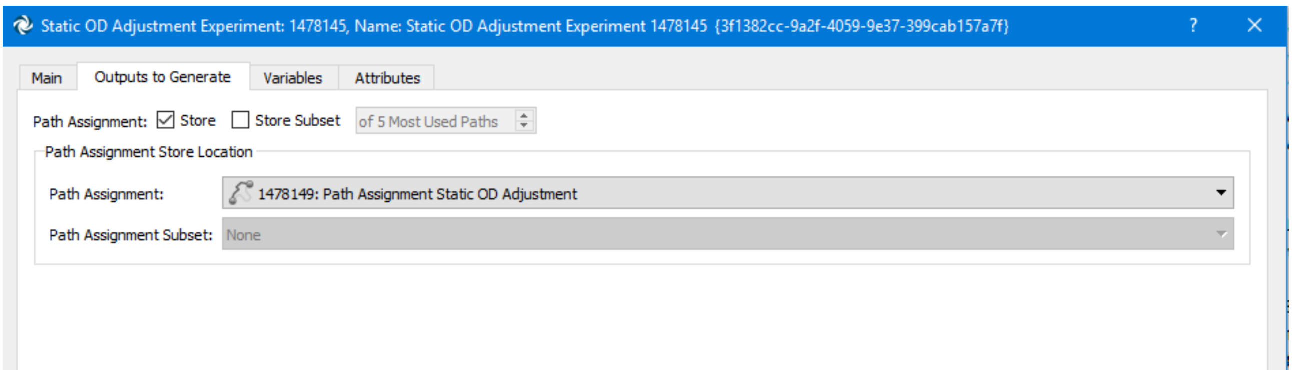


Create Static OD experiment



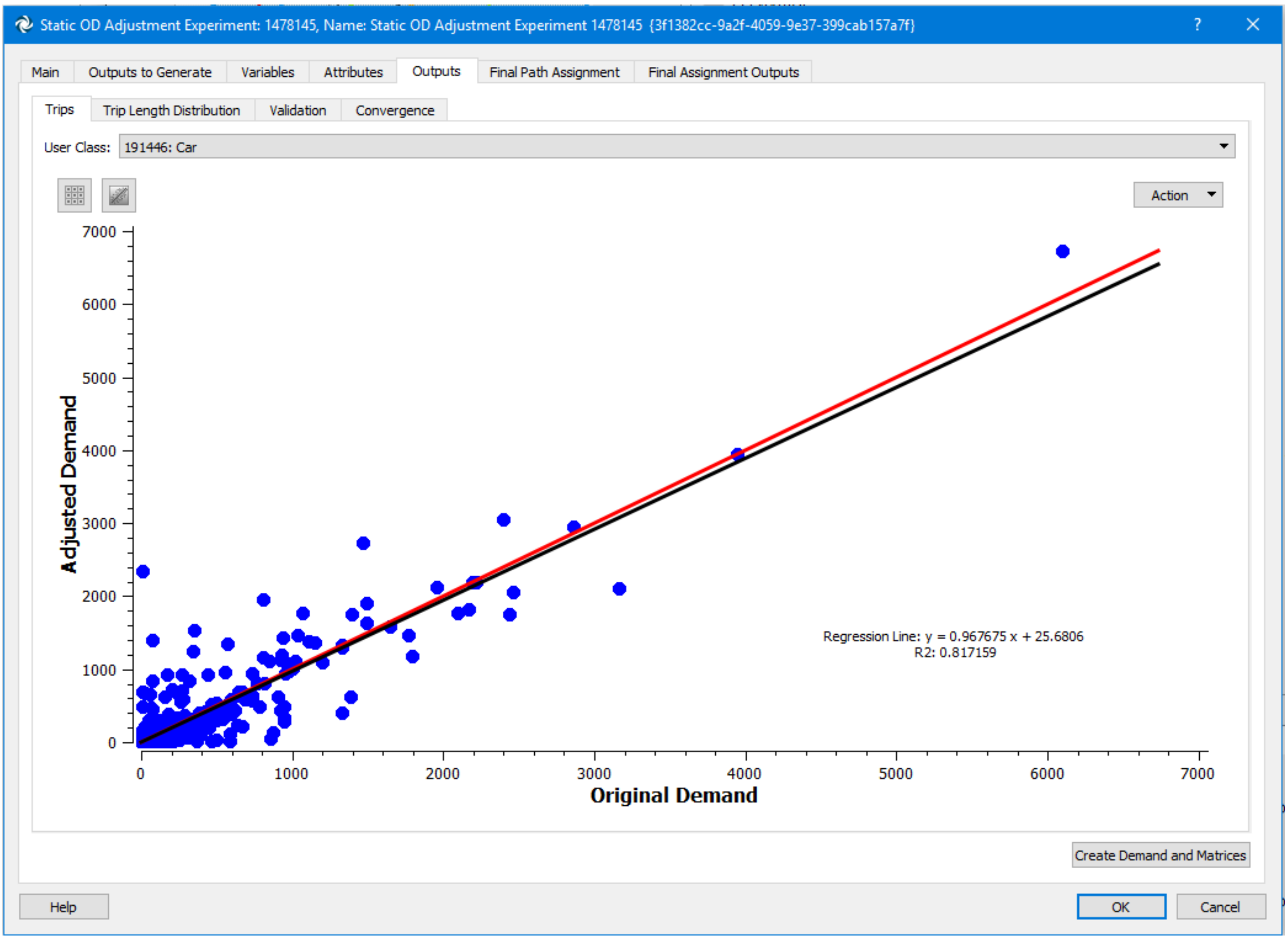
The iterations setting was changed from 20 to 30 for a better R^2 result, although the result is still below 85%. However, test showed that further increasing the number of iterations has limited effect on the convergence.

<u>Further note: changing the iterations to 50 cannot smoothen the convergence plot. On the contrary, more fluctuation will occur, and the validation R^2 will slightly drop.</u>

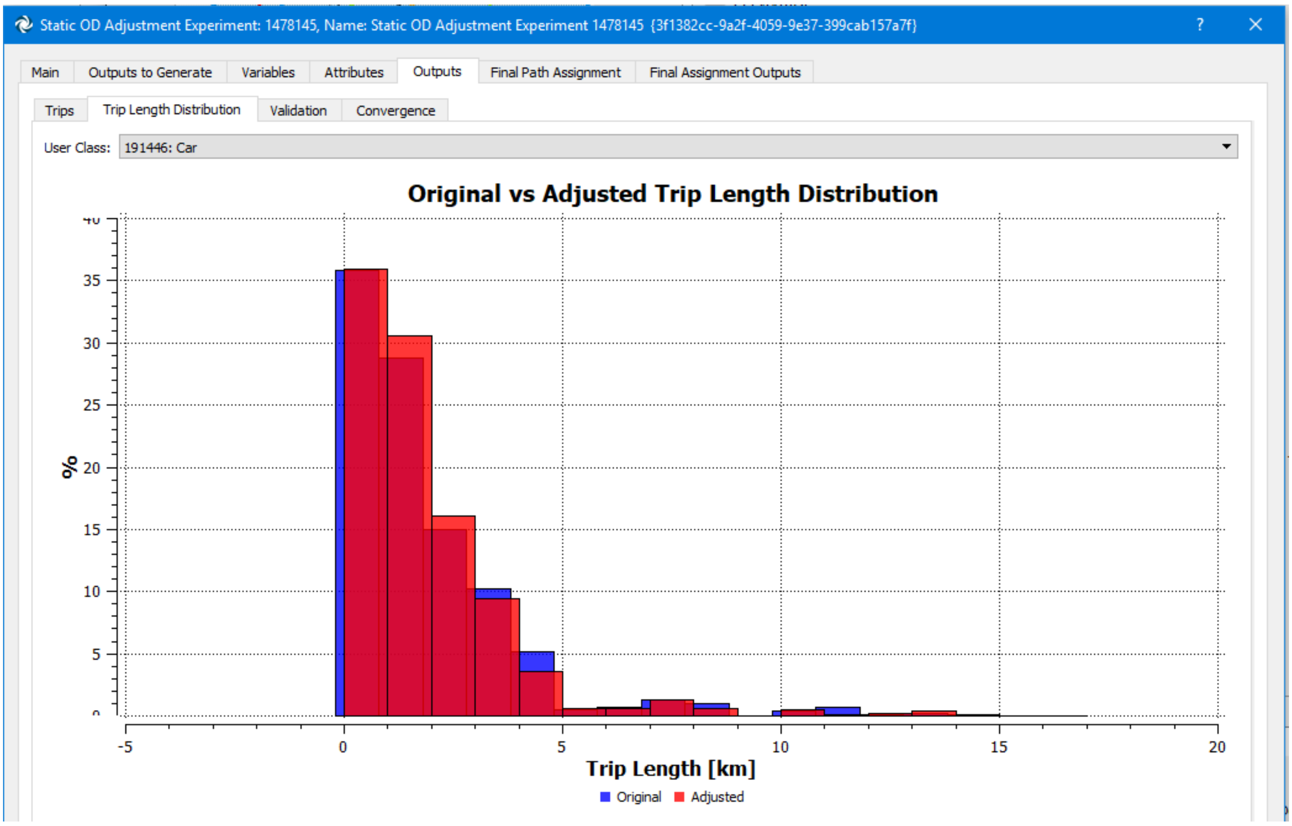


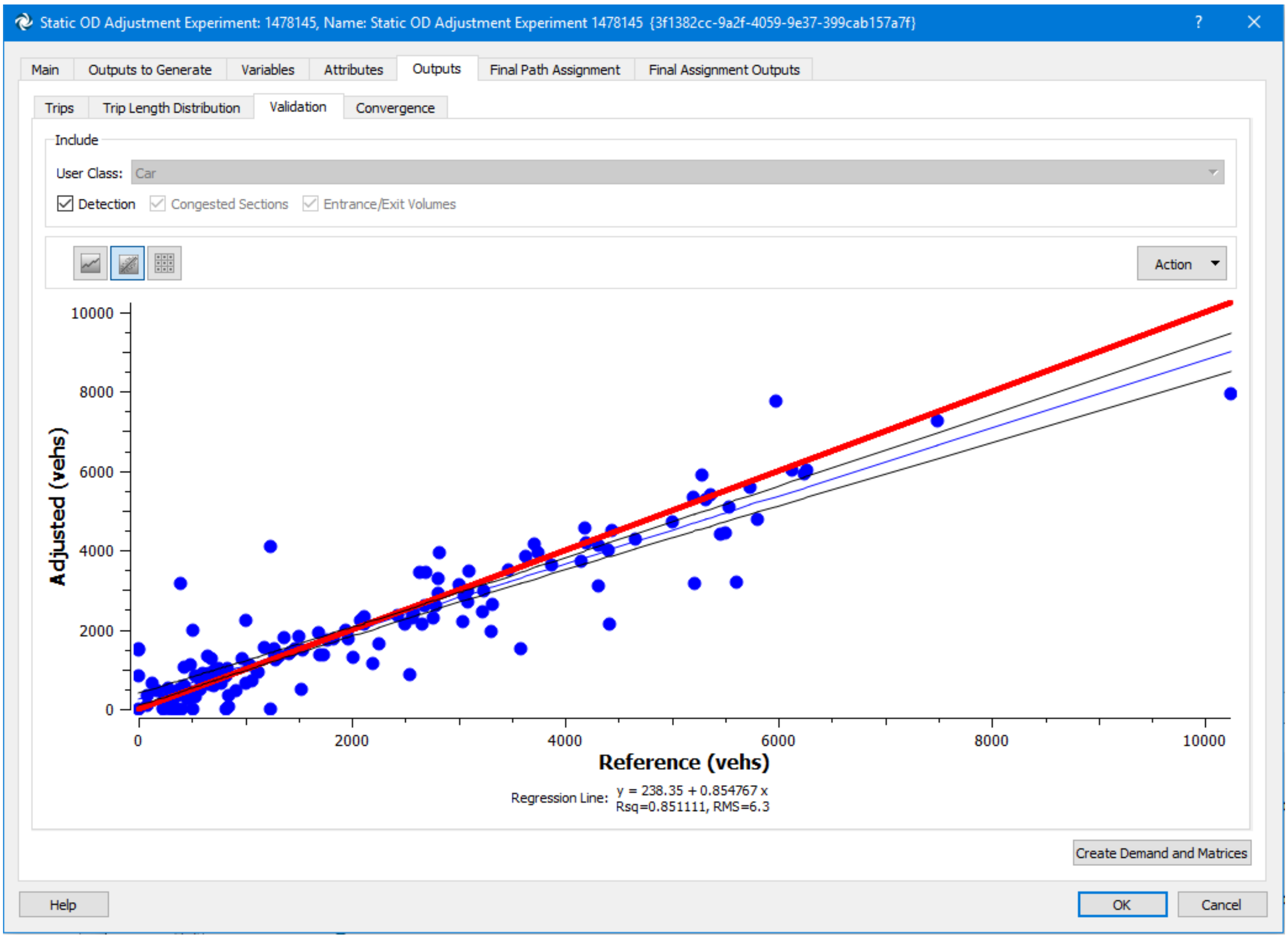
The result of Static OD Adjustment (Elasticity=0.75, iterations=30)

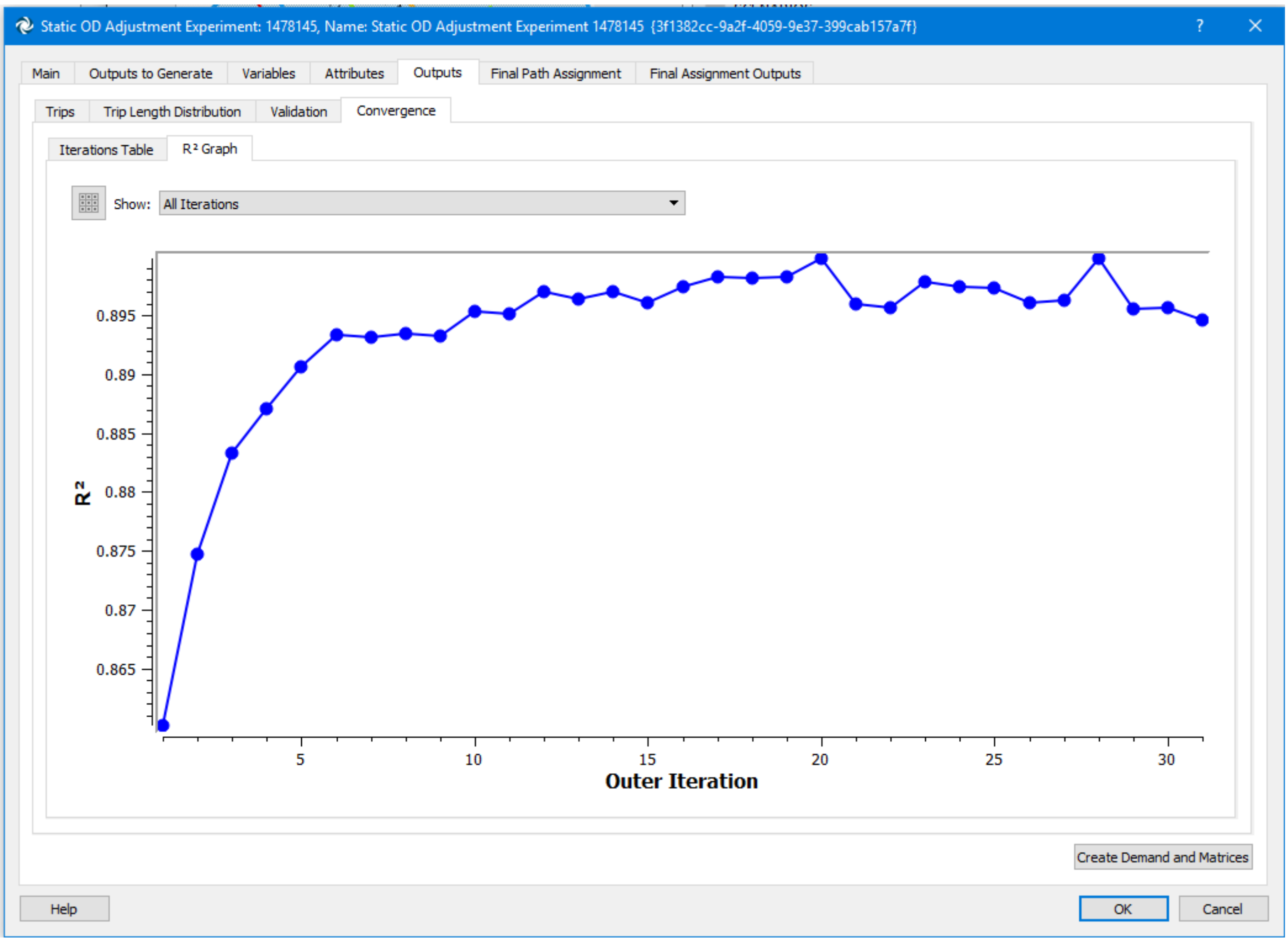
Runtime = 0 h 2 m 17 s

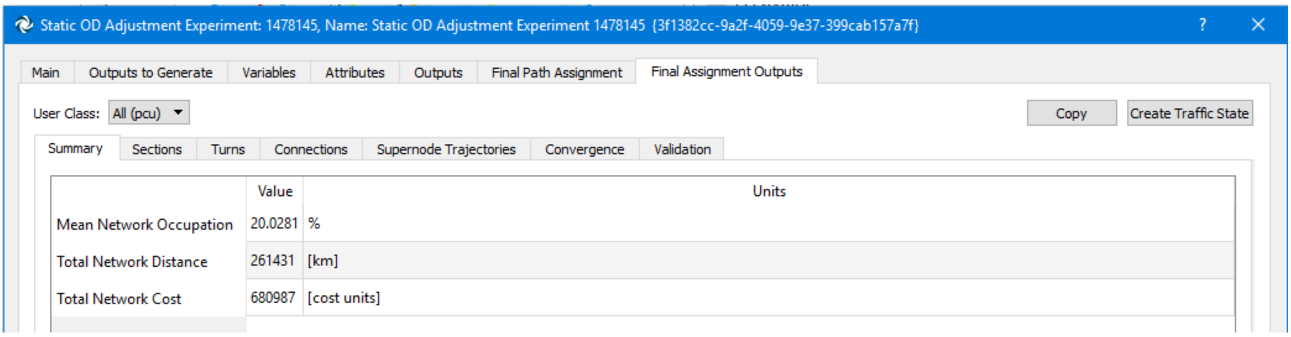


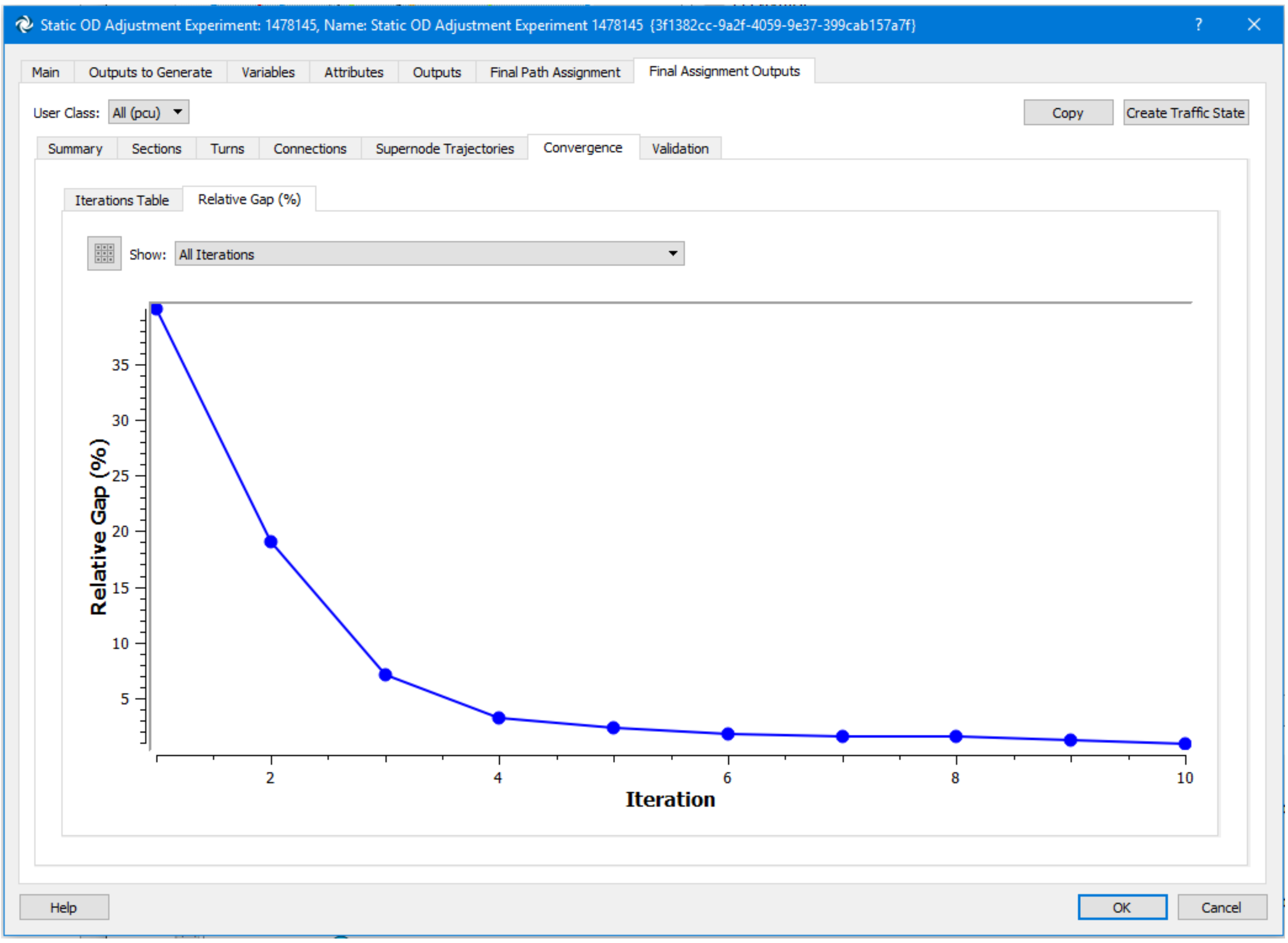
Compared to previous version (when the Elasticity was set to 0.5), the R2 for Trip Demand drops from 0.9 to 0.82/




Static OD Adjustment Experiment: 1478145, Name: Static OD Adjustment Experiment 1478145 {3f1382cc-9a2f-4059-9e37-399cab157a7f}
Main
Outputs to Generate
Variables
Attributes
Outputs
Final Path Assignment
Final Assignment Outputs
Trips
Trip Length Distribution
Validation
Convergence
Include
User Class:
Car
Detection
Congested Sections
Entrance/Exit Volumes
Action
Adjusted (vehs)
10000
8000
6000
4000
2000
0
Reference (vehs)
Regression Line:
y = 238.35 + 0.854767 x
Rsq=0.851111, RMS=6.3
Create Demand and Matrices
Help
OK
Cancel

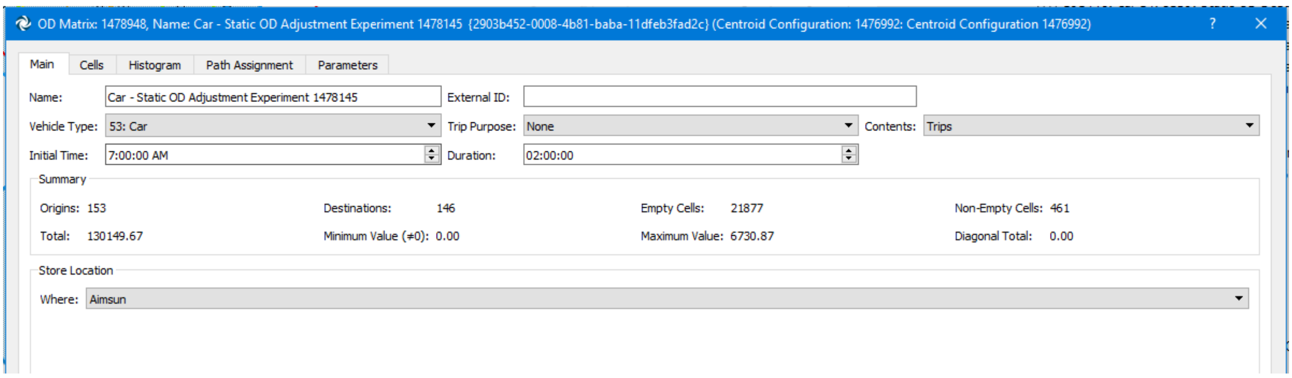

Static OD Adjustment Experiment: 1478145, Name: Static OD Adjustment Experiment 1478145 {3f1382cc-9a2f-4059-9e37-399cab157a7f}
Main
Outputs to Generate
Variables
Attributes
Outputs
Final Path Assignment
Final Assignment Outputs
Trips
Trip Length Distribution
Validation
Convergence
Iterations Table
R² Graph
Show:
All Iterations
R²
0.895
0.89
0.885
0.88
0.875
0.87
0.865
5
10
15
20
25
30
Outer Iteration
Create Demand and Matrices
Help
OK
Cancel





Press create Demand and matrices to save the adjusted Demand

New Car - Static OD Adjustment Experiment 1478145

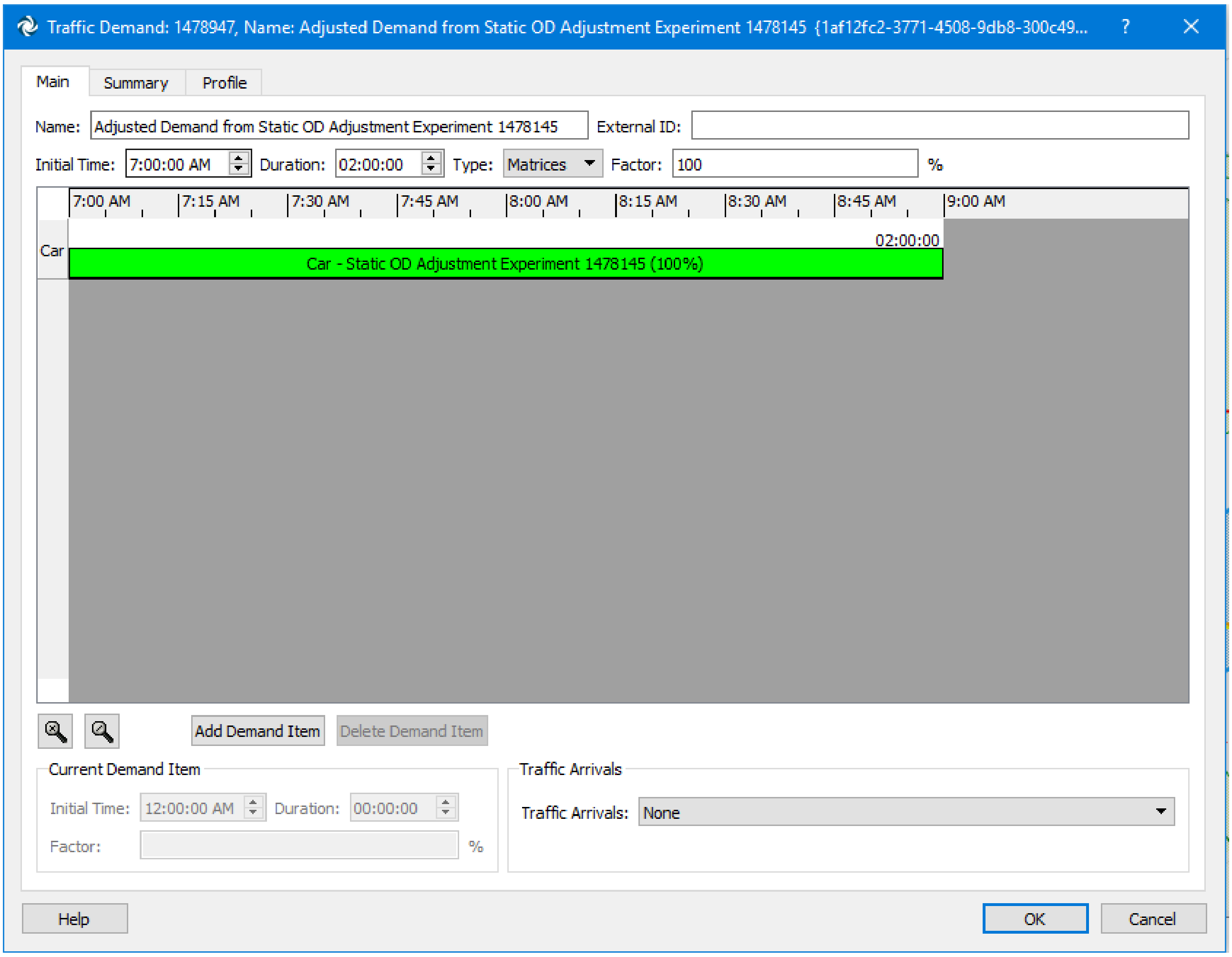


Create new traffic demand using this matrix

New Adjusted Demand from Static OD Adjustment Experiment 1478145

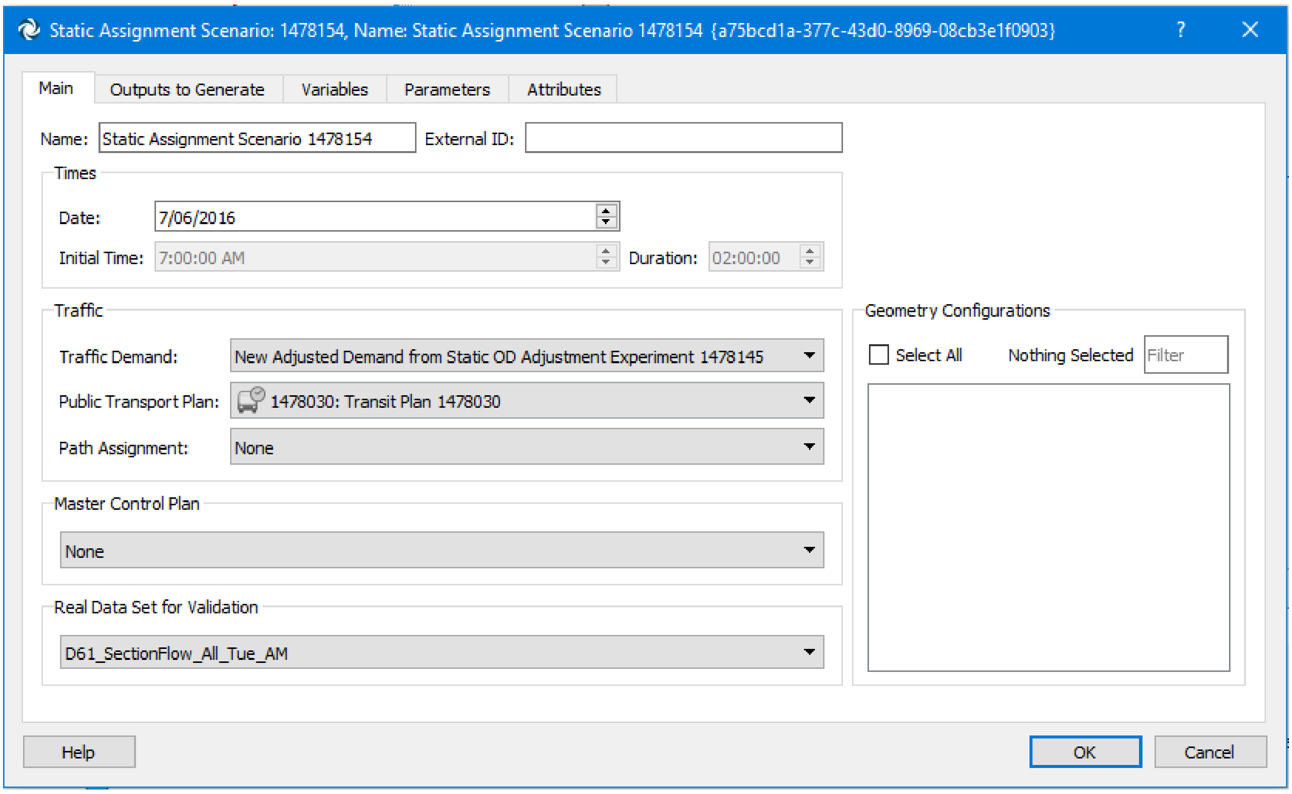
Part II – Aimsun model calibration

To be used for the Static Assignment Scenario that prepares for the OD Departure Adjustment.

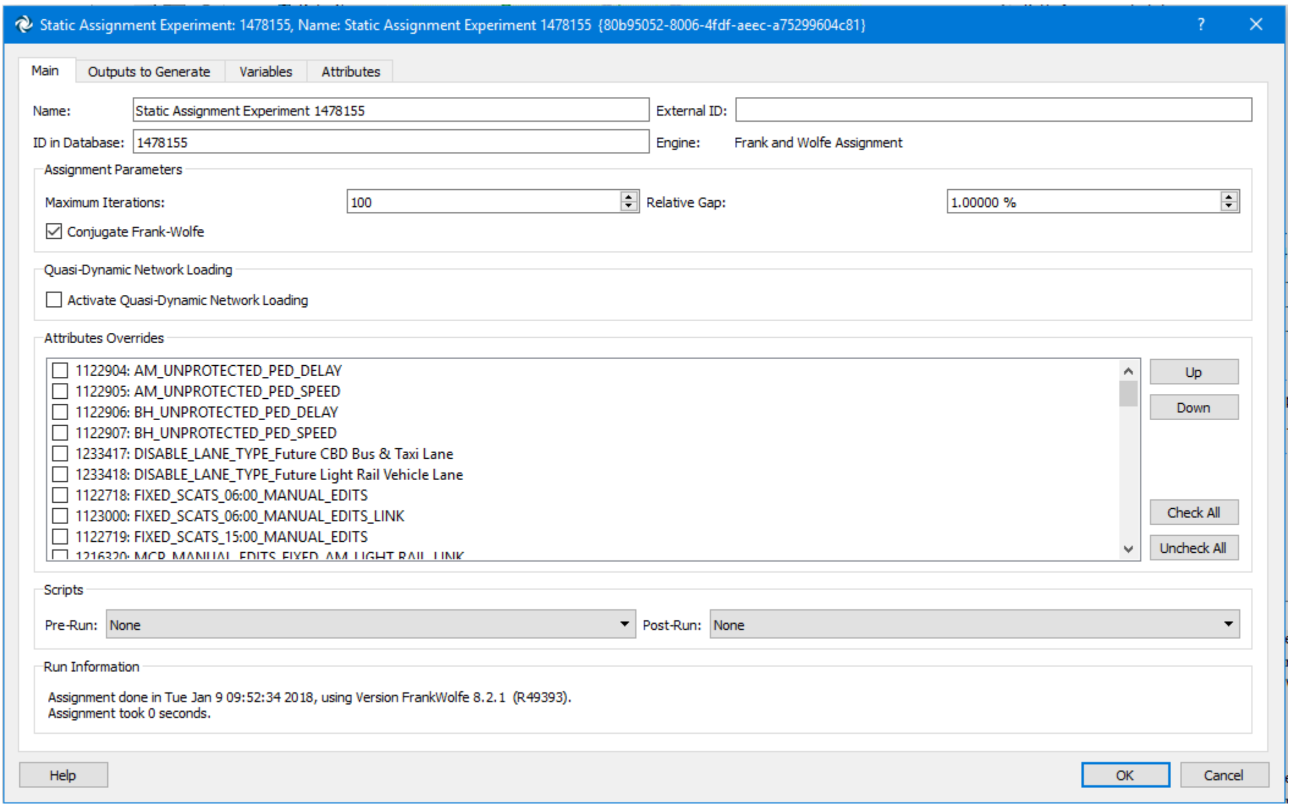


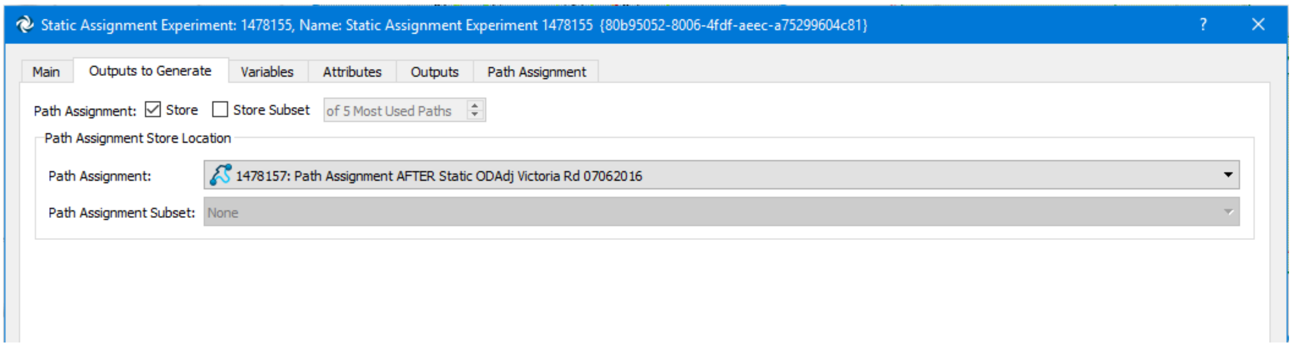
4.5. Prepare for the Static OD Departure Adjustment

Create new Static Assignment Scenario for subnetwork



Create Static Assignment experiment

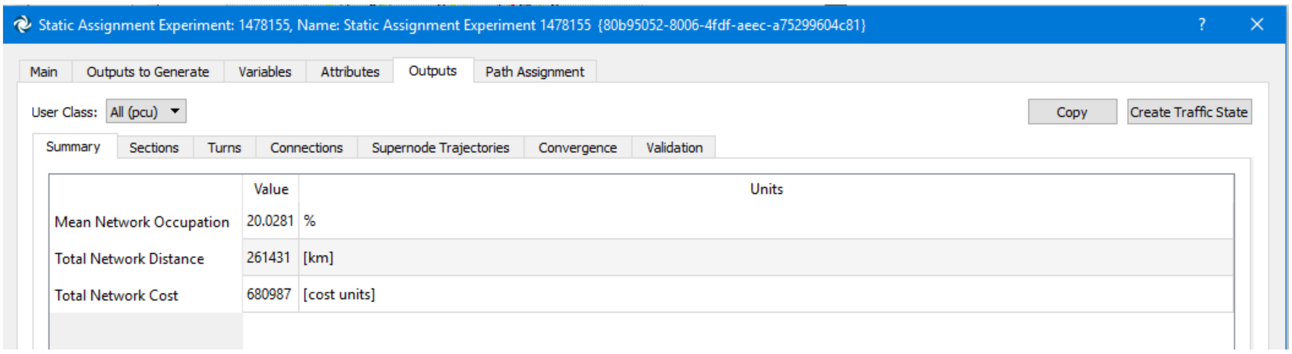




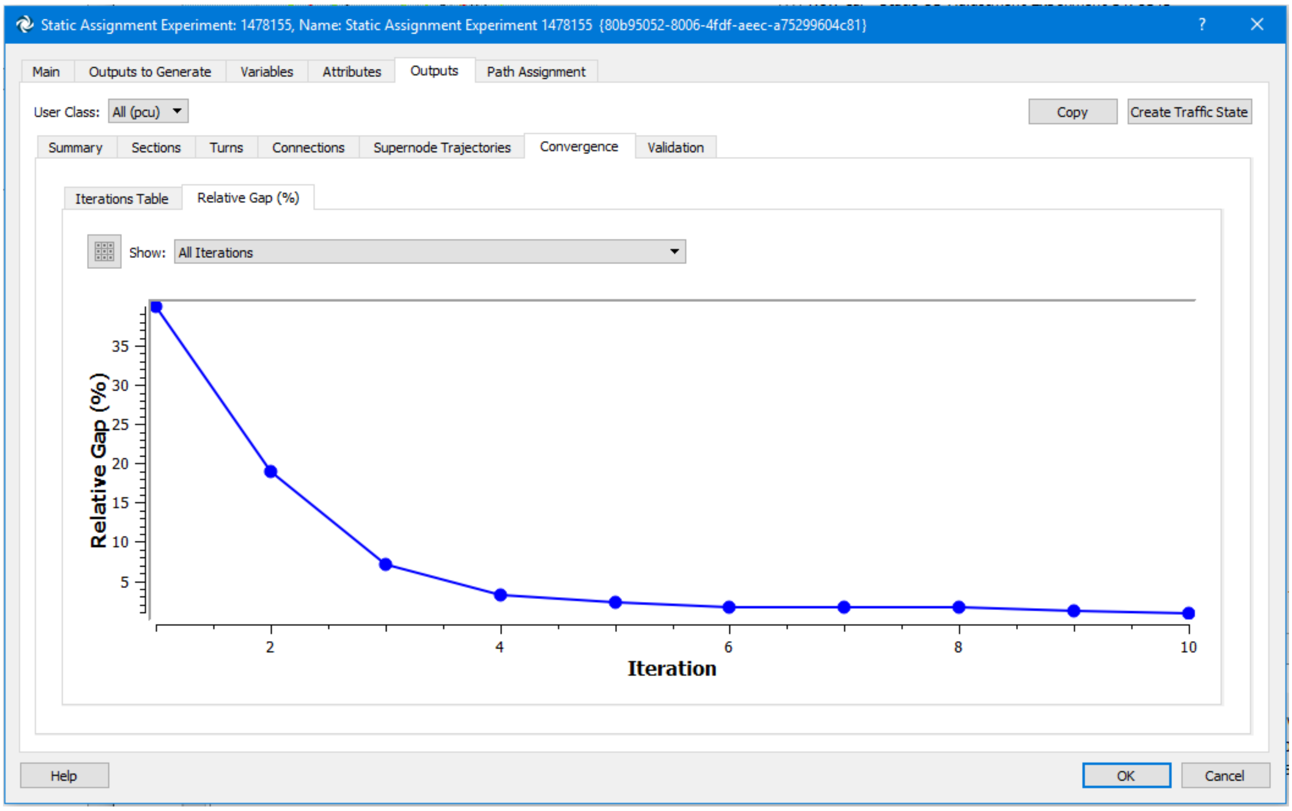
The resultant Path Assignment will be saved to: **1478157 Path Assignment AFTER Static ODAdj Victoria Rd**.

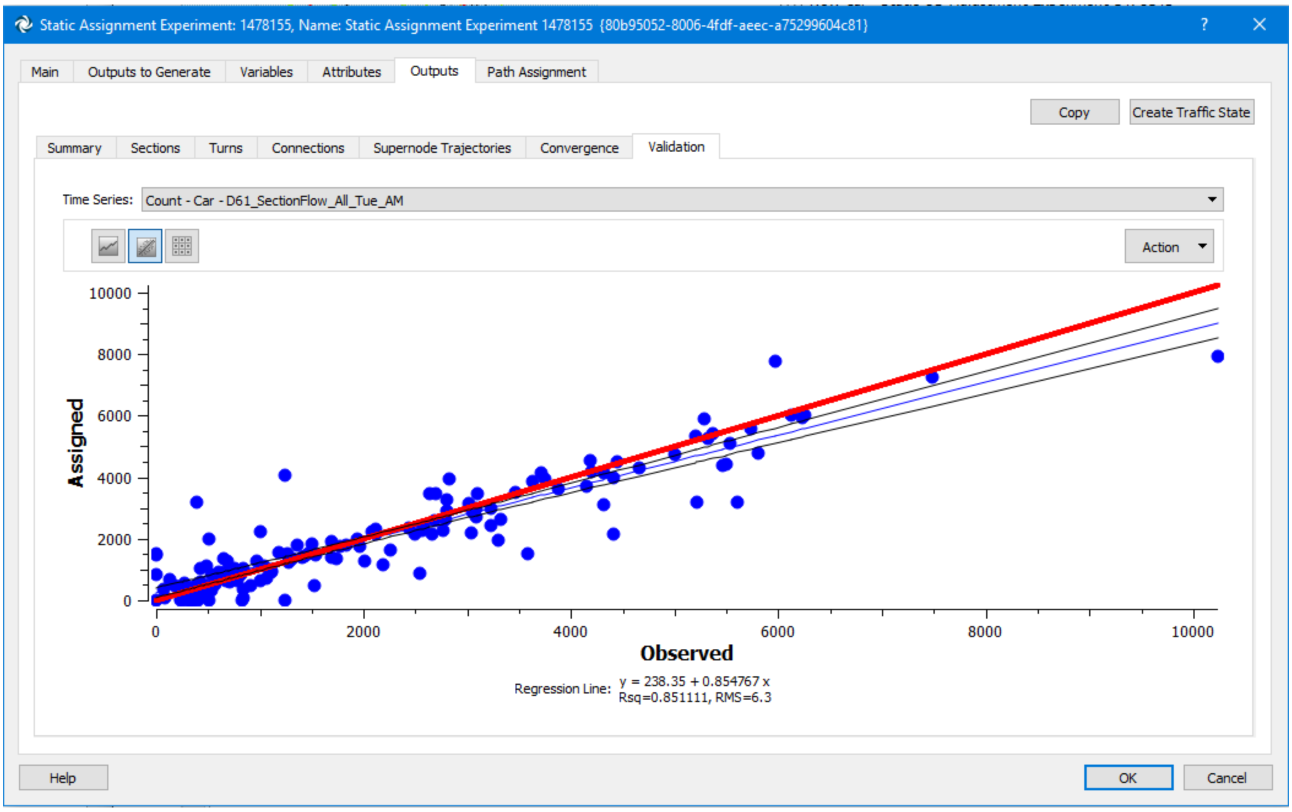
Save results of the Experiment

Runtime 0 h 0 m 7 s



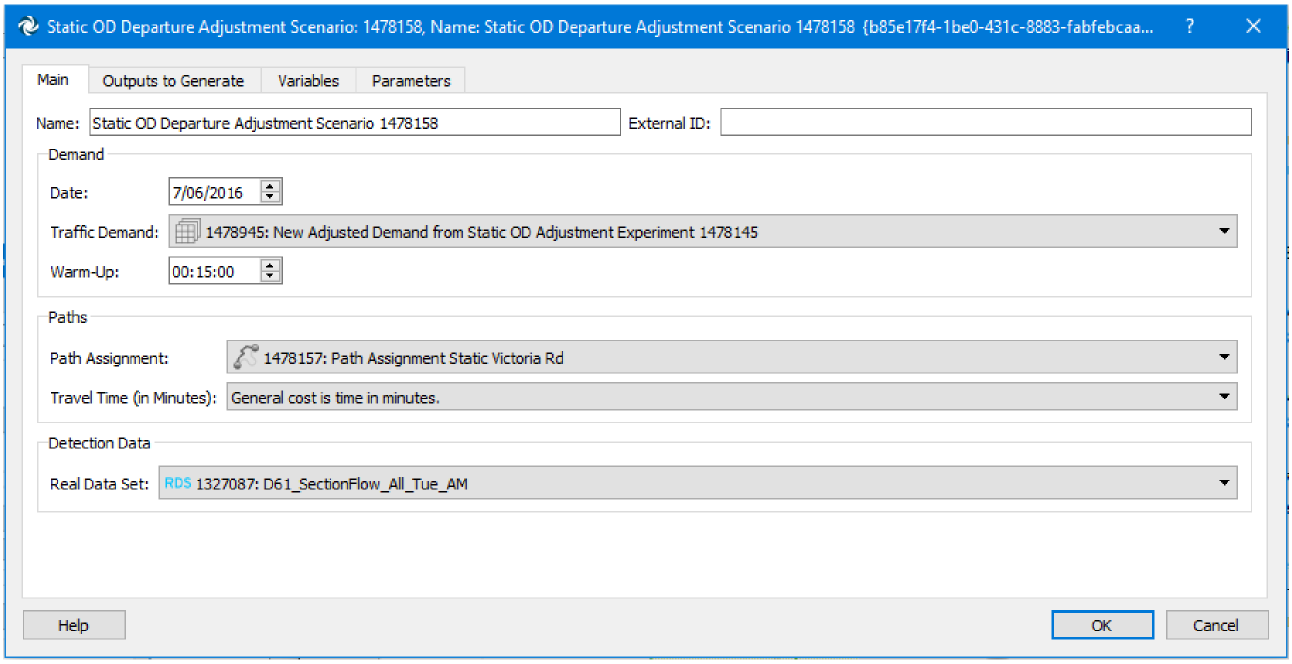
The Experiment did not stop after one iteration (as documented in the original file). Instead it ran for ten iterations.

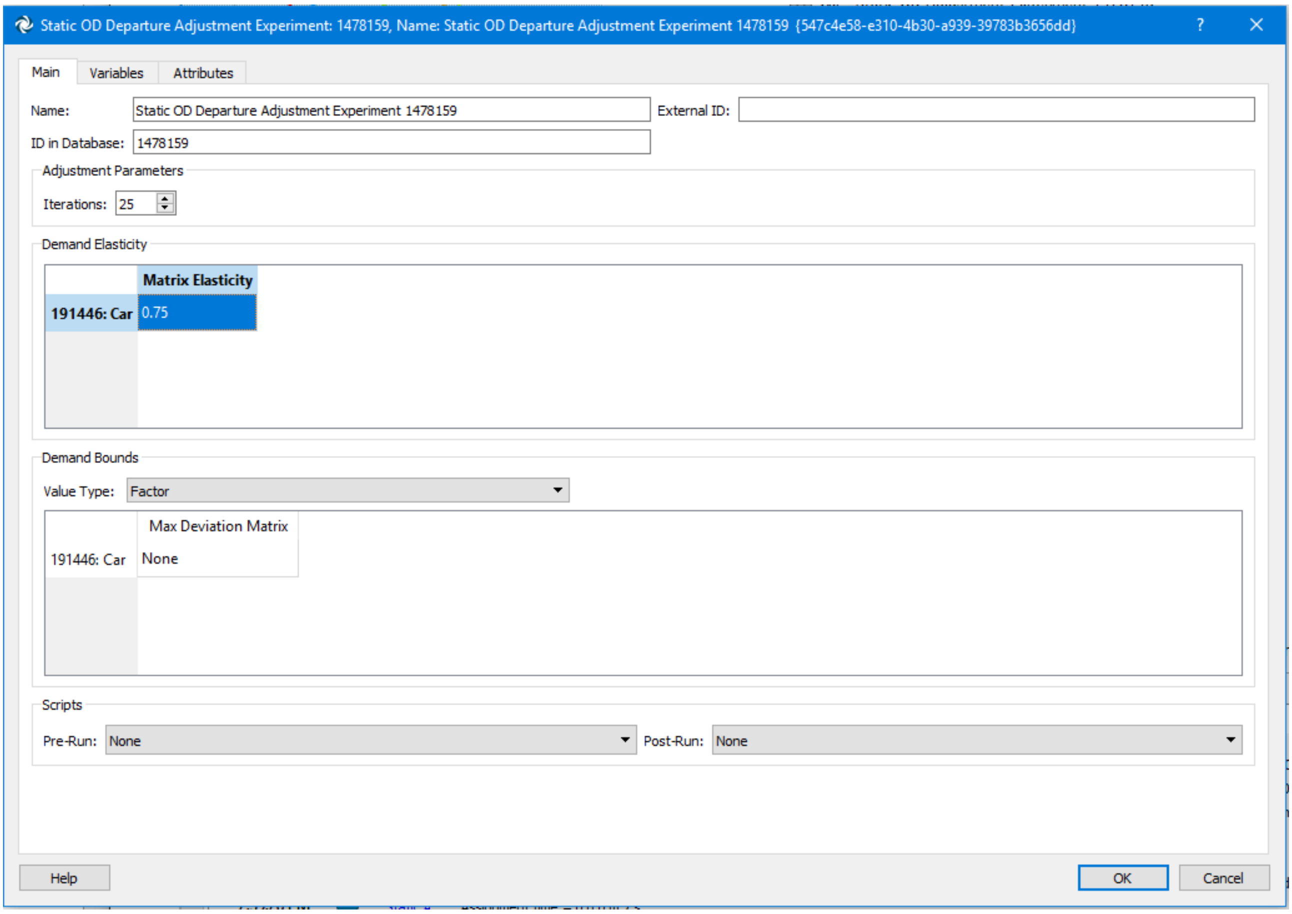




4.6. Static OD Departure Adjustment

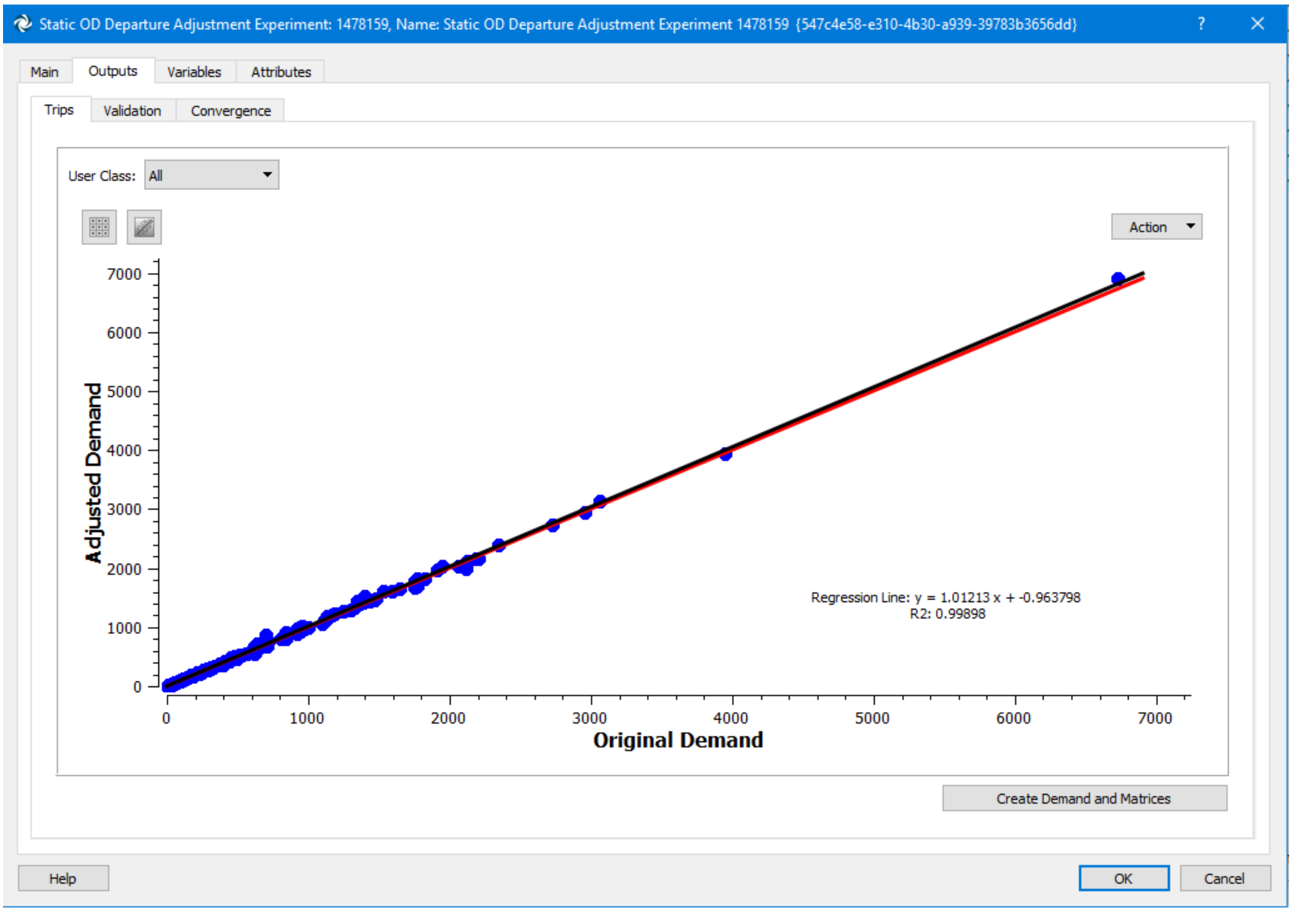
Create Static OD Departure Adjustment

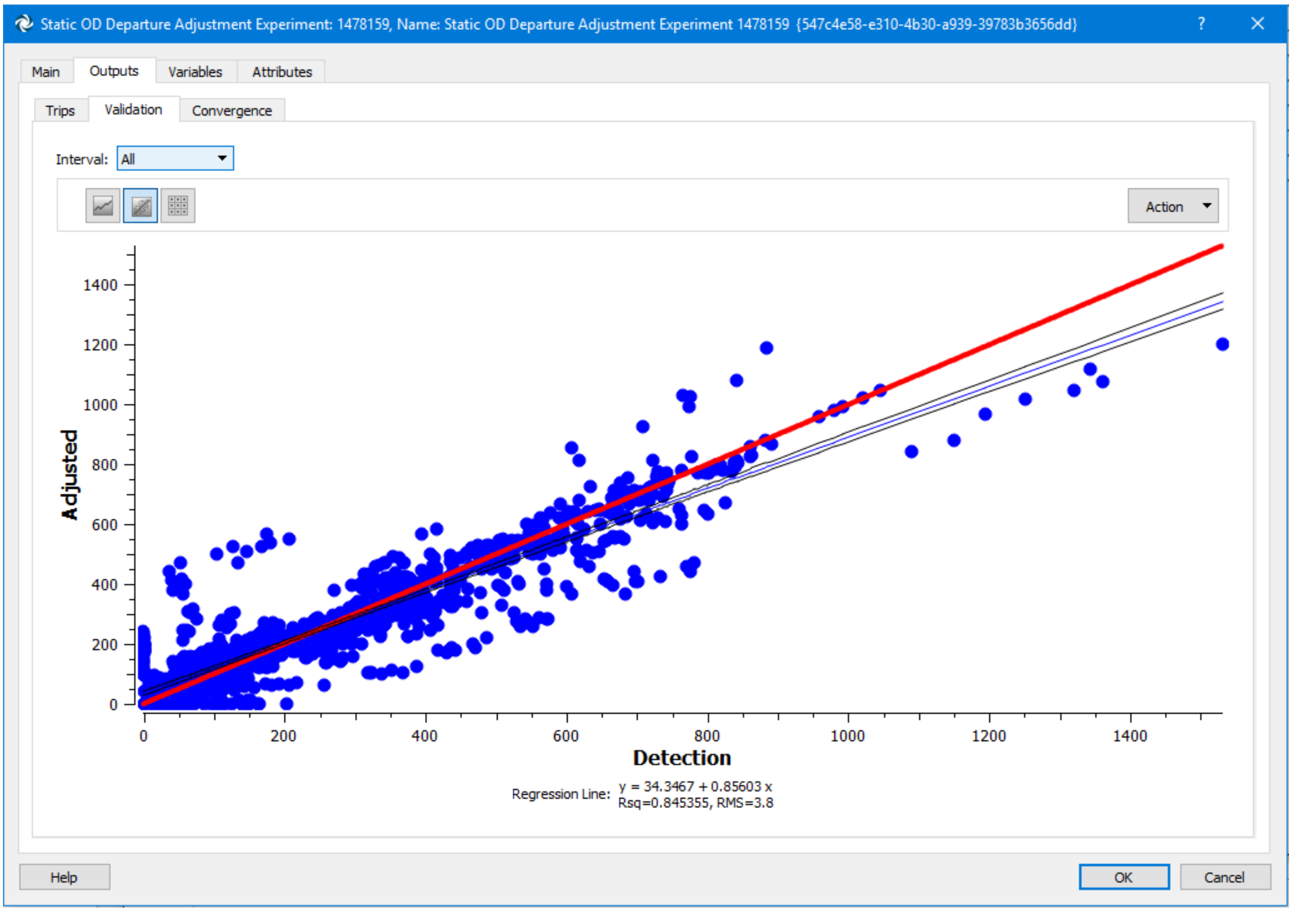


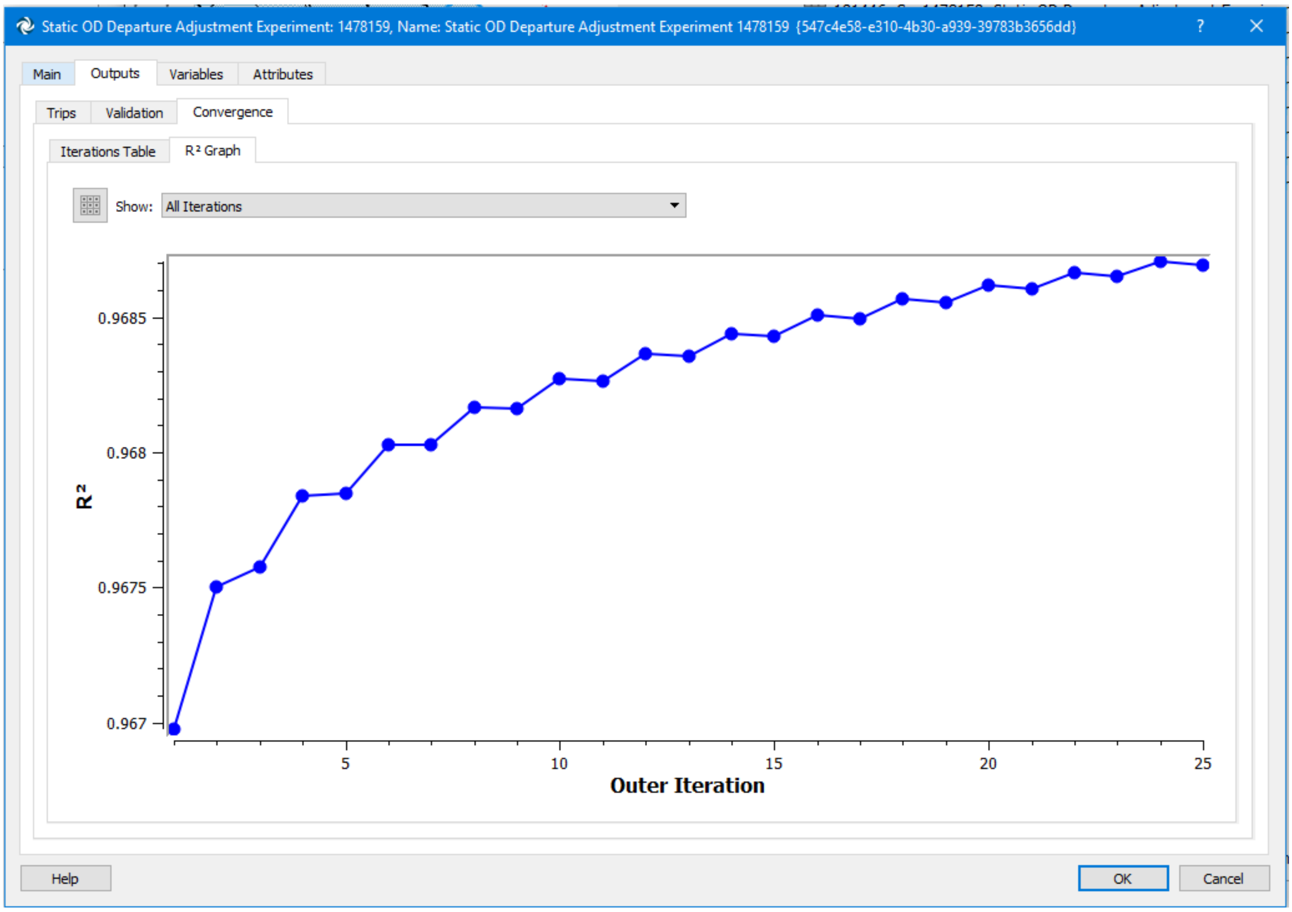


The result of Static OD Departure Adjustment

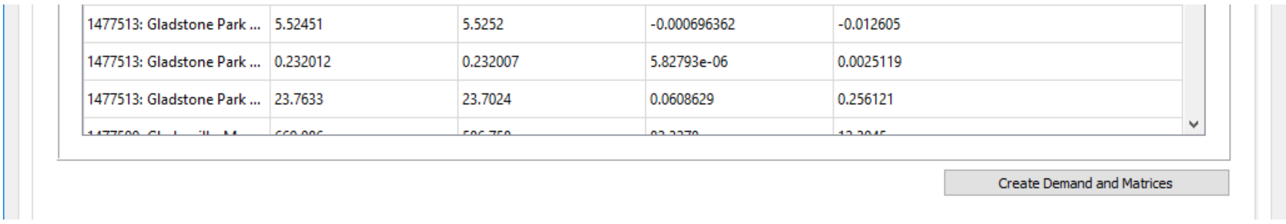
Runtime = 0 h 0 m 17 s


Static OD Departure Adjustment Experiment: 1478159, Name: Static OD Departure Adjustment Experiment 1478159 {547c4e58-e310-4b30-a939-39783b3656dd}
Main
Outputs
Variables
Attributes
Trips
Validation
Convergence
User Class: All
Action
Adjusted Demand
Original Demand
Regression Line: y = 1.01213 x + -0.963798
R2: 0.99898
Create Demand and Matrices
Help
OK
Cancel


Static OD Departure Adjustment Experiment: 1478159, Name: Static OD Departure Adjustment Experiment 1478159 {547c4e58-e310-4b30-a939-39783b3656dd}
Main
Outputs
Variables
Attributes
Trips
Validation
Convergence
Interval: All
Action
Adjusted
Detection
Regression Line: y = 34.3467 + 0.85603 x
Rsq=0.845355, RMS=3.8
Help
OK
Cancel



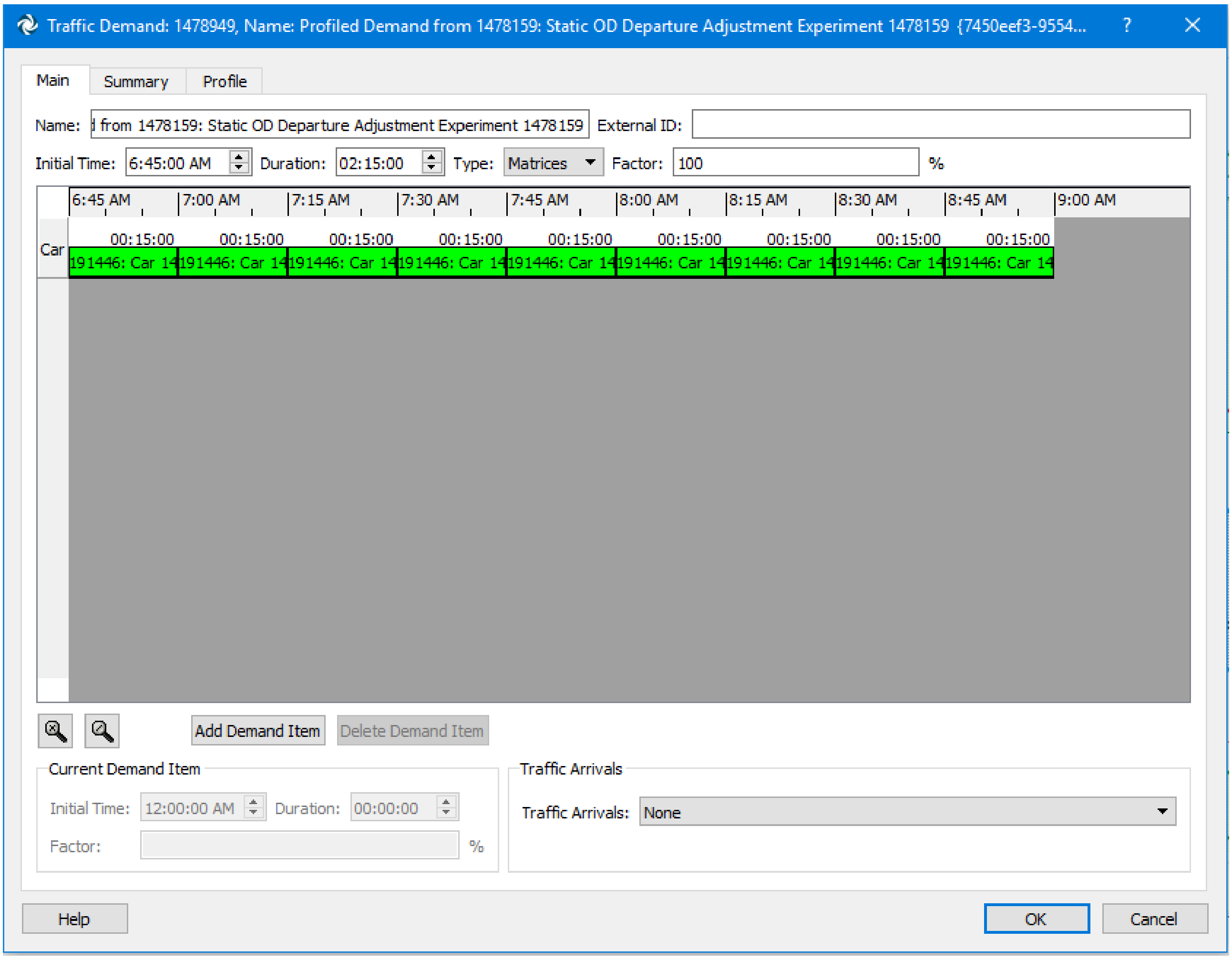
Create profiled traffic demand from the results

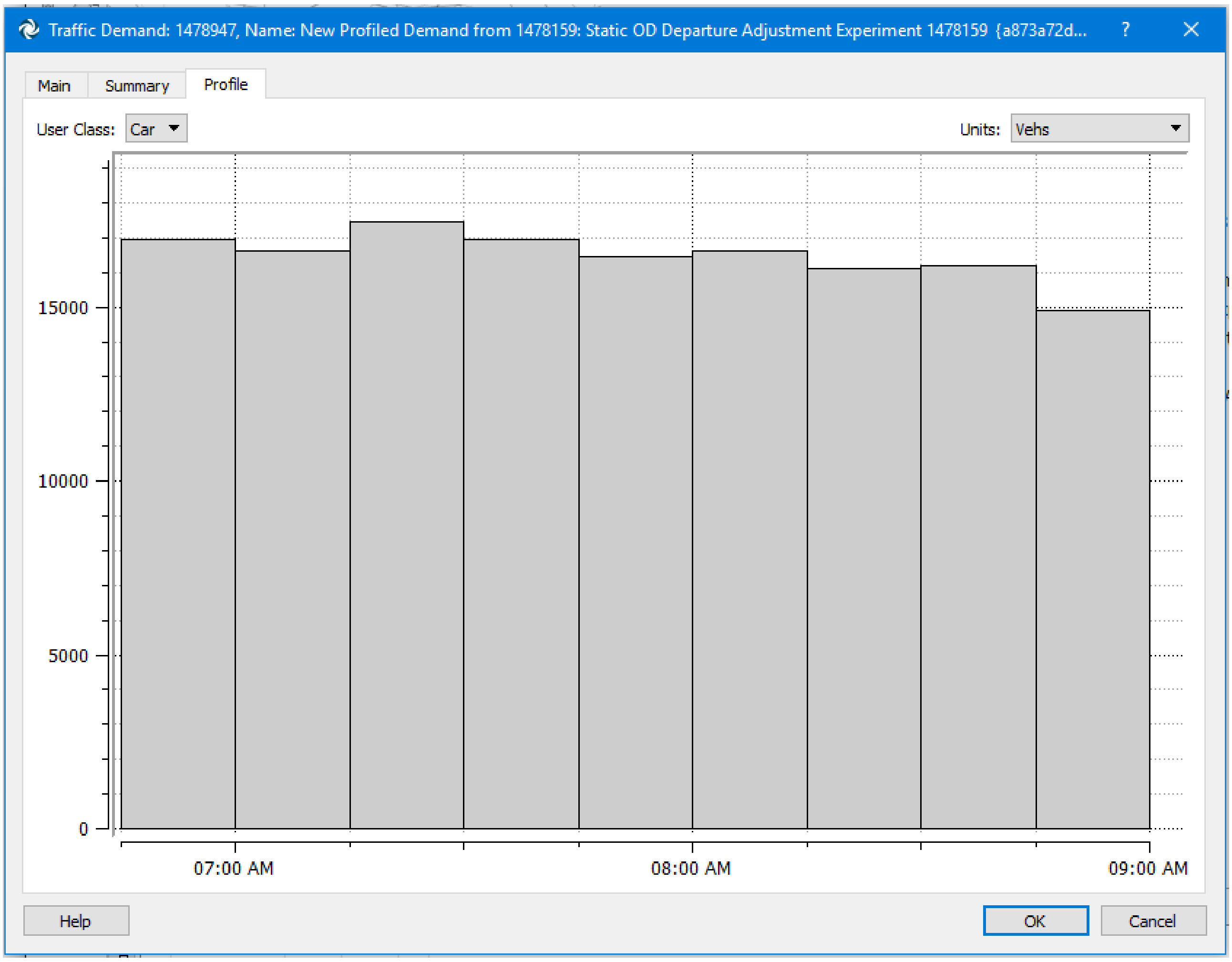


1477513: Gladstone Park ...	5.52451	5.5252	-0.000696362	-0.012605
1477513: Gladstone Park ...	0.232012	0.232007	5.82793e-06	0.0025119
1477513: Gladstone Park ...	23.7633	23.7024	0.0608629	0.256121

Aimsun will automatically generate the 15-min sliced OD Matrices and the Profiled Demand:

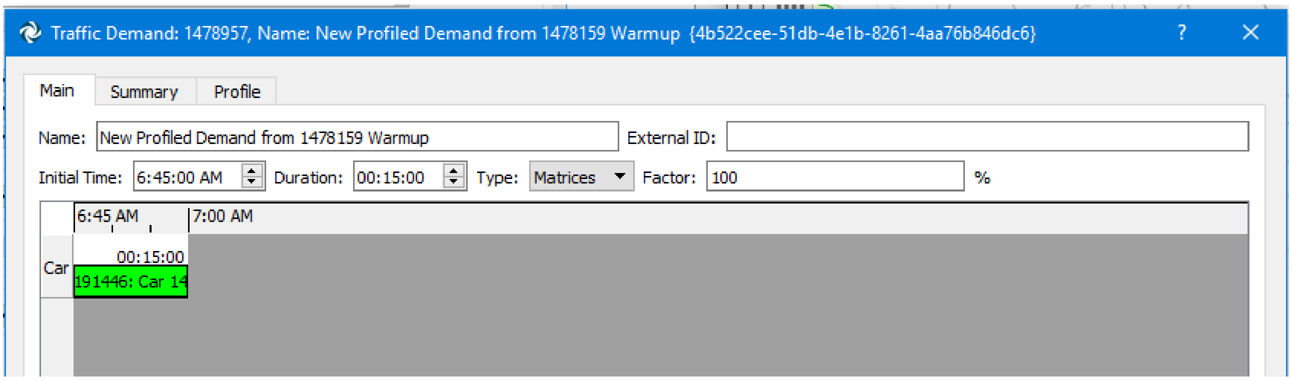
Profiled Demand from 1478159: Static OD Departure Adjustment Experiment 1478159


Traffic Demand: 1478949, Name: Profiled Demand from 1478159: Static OD Departure Adjustment Experiment 1478159 {7450eef3-9554...
Main
Summary
Profile
Name:
from 1478159: Static OD Departure Adjustment Experiment 1478159
External ID:
Initial Time:
6:45:00 AM
Duration:
02:15:00
Type:
Matrices
Factor:
100
%
6:45 AM
7:00 AM
7:15 AM
7:30 AM
7:45 AM
8:00 AM
8:15 AM
8:30 AM
8:45 AM
9:00 AM
Car
00:15:00
191446: Car 14
Add Demand Item
Delete Demand Item
Current Demand Item
Initial Time:
12:00:00 AM
Duration:
00:00:00
Factor:
Traffic Arrivals
Traffic Arrivals:
None
Help
OK
Cancel



Compared to the previous version, unchanged.

Create the 15-min Warmup Traffic Demand:



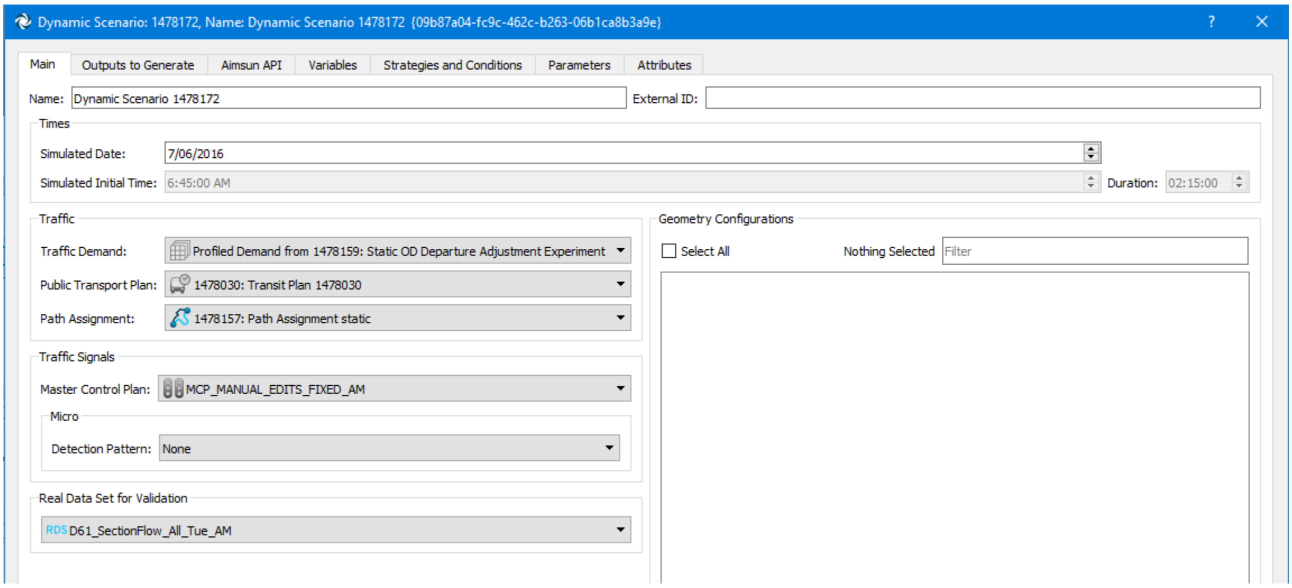
When the Static OD Departure Adjustment is completed, the model is ready for the dynamic traffic simulation.

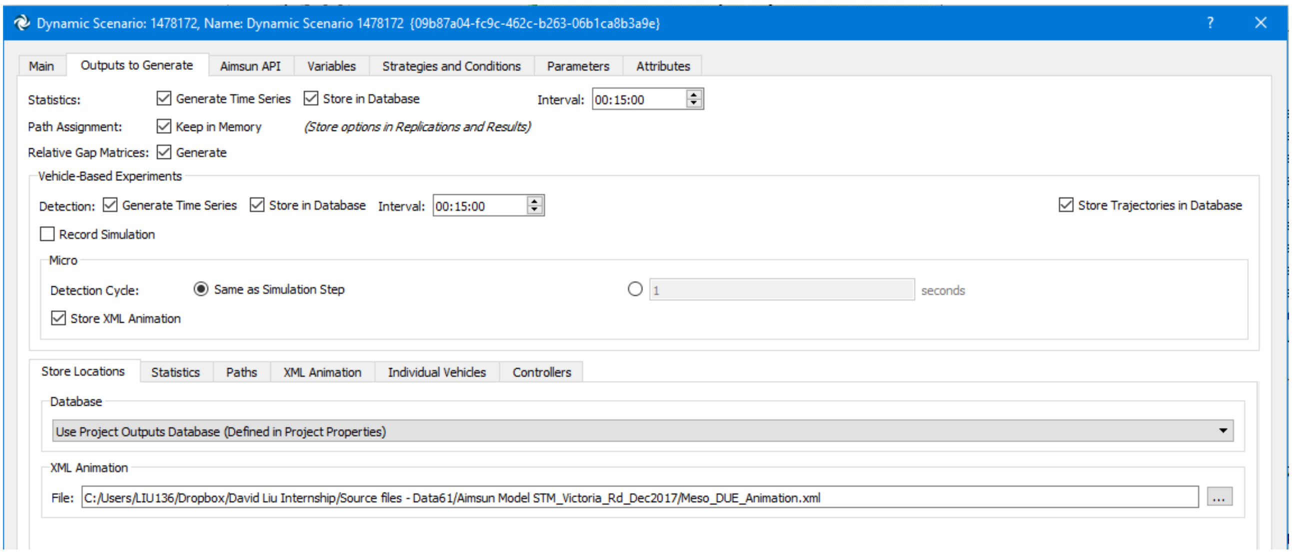
5. Part III - Dynamic Traffic Simulation

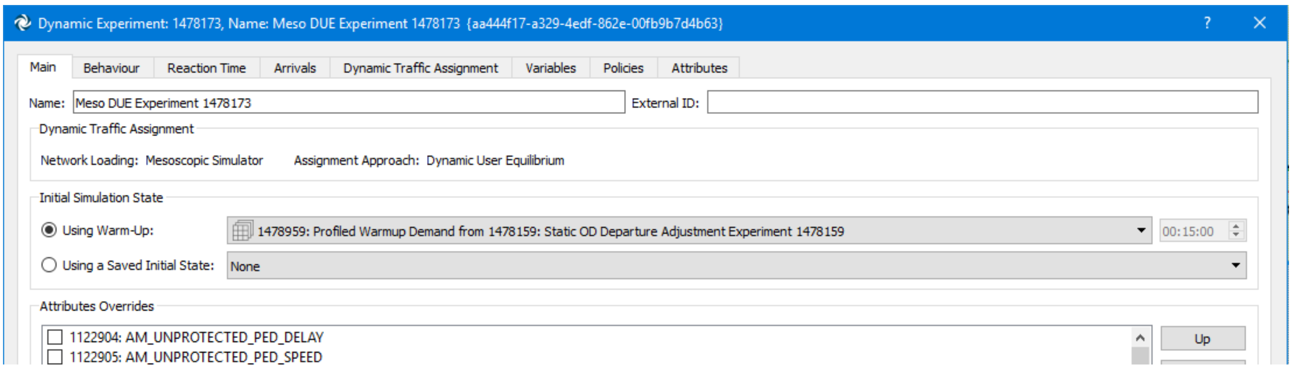
For the investigation of the traffic congestion, the author performed dynamic traffic simulation both on the mesoscopic and microscopic level. As a comparison, the author also simulated the traffic behaviour when the incident(s) happened. The incident dataset (for the year 2017) adopted in this step was provided by Dr Huong, which is stored at *\Dropbox\David Liu Internship\Source files - Data61\Data_new\ corridor2_xlsx.xlsx*.

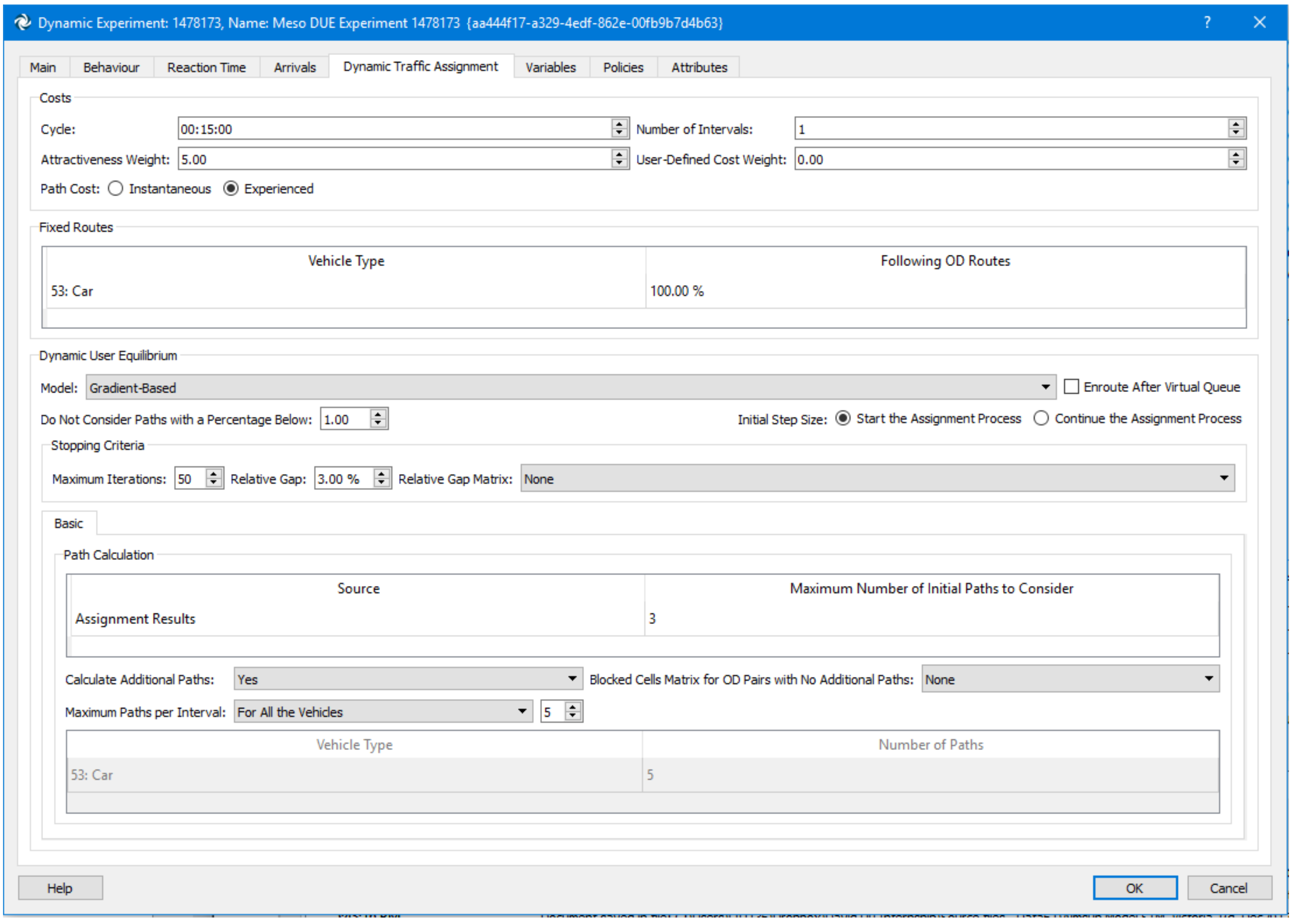
For the creation of the incident object in Aimsun, please refer to section 0.

5.1. Meso DUE Experiment

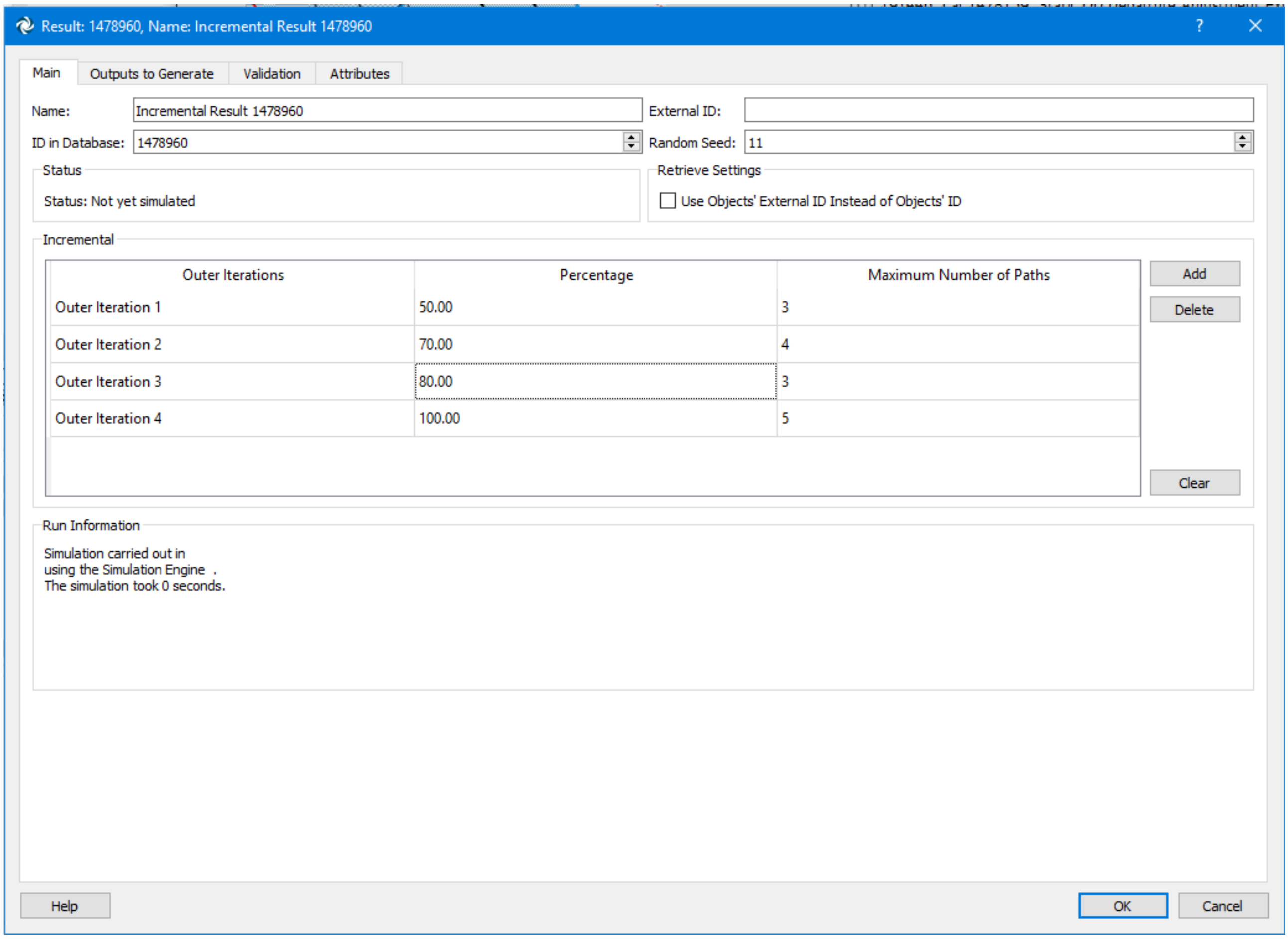






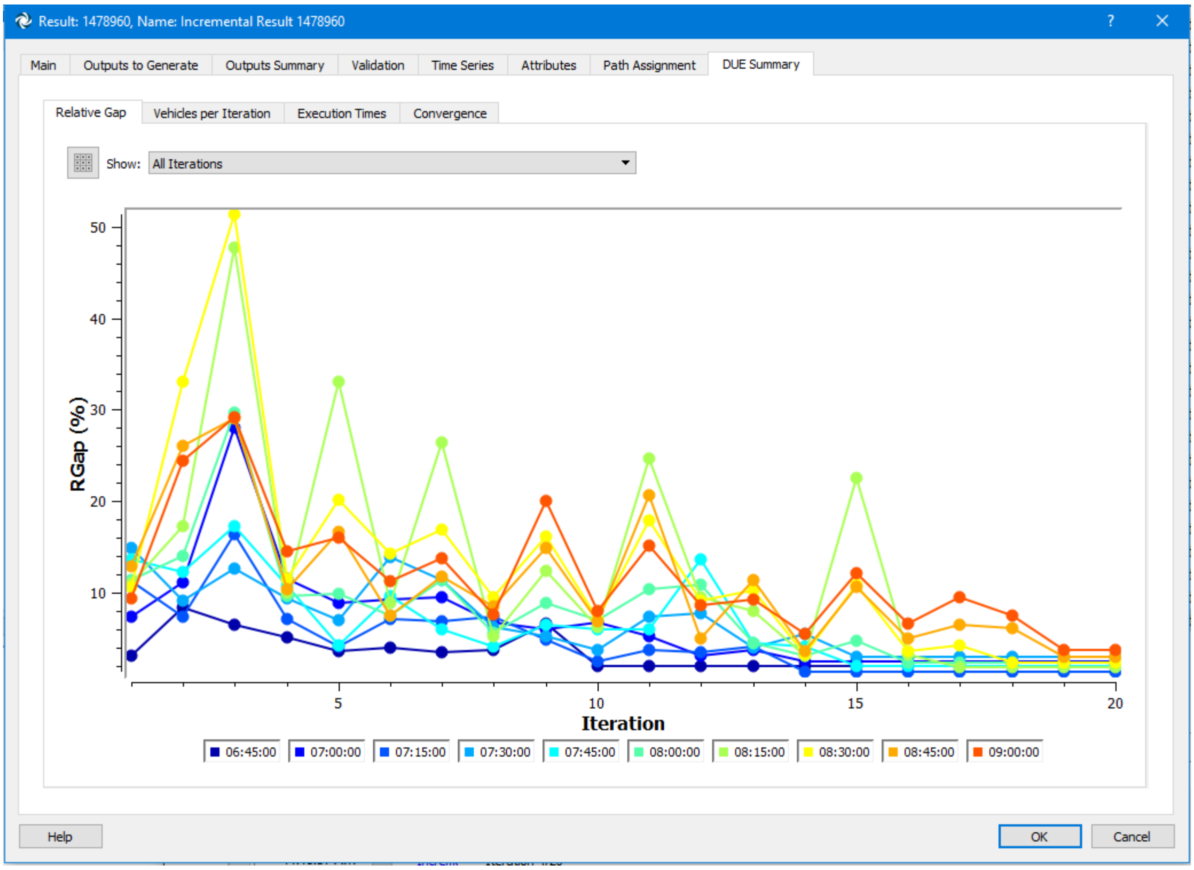


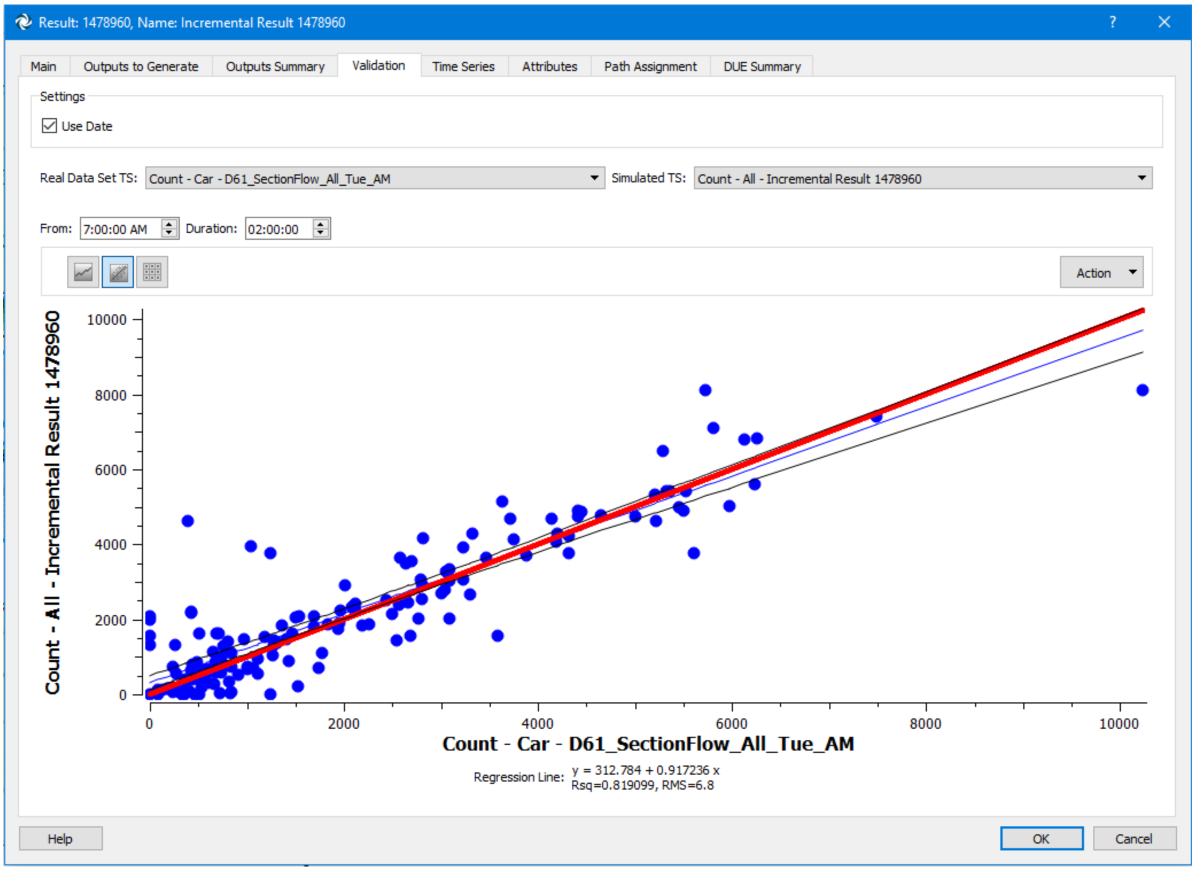
Create Incremental Result in Meso DUE Experiment, use Random Seed 11:



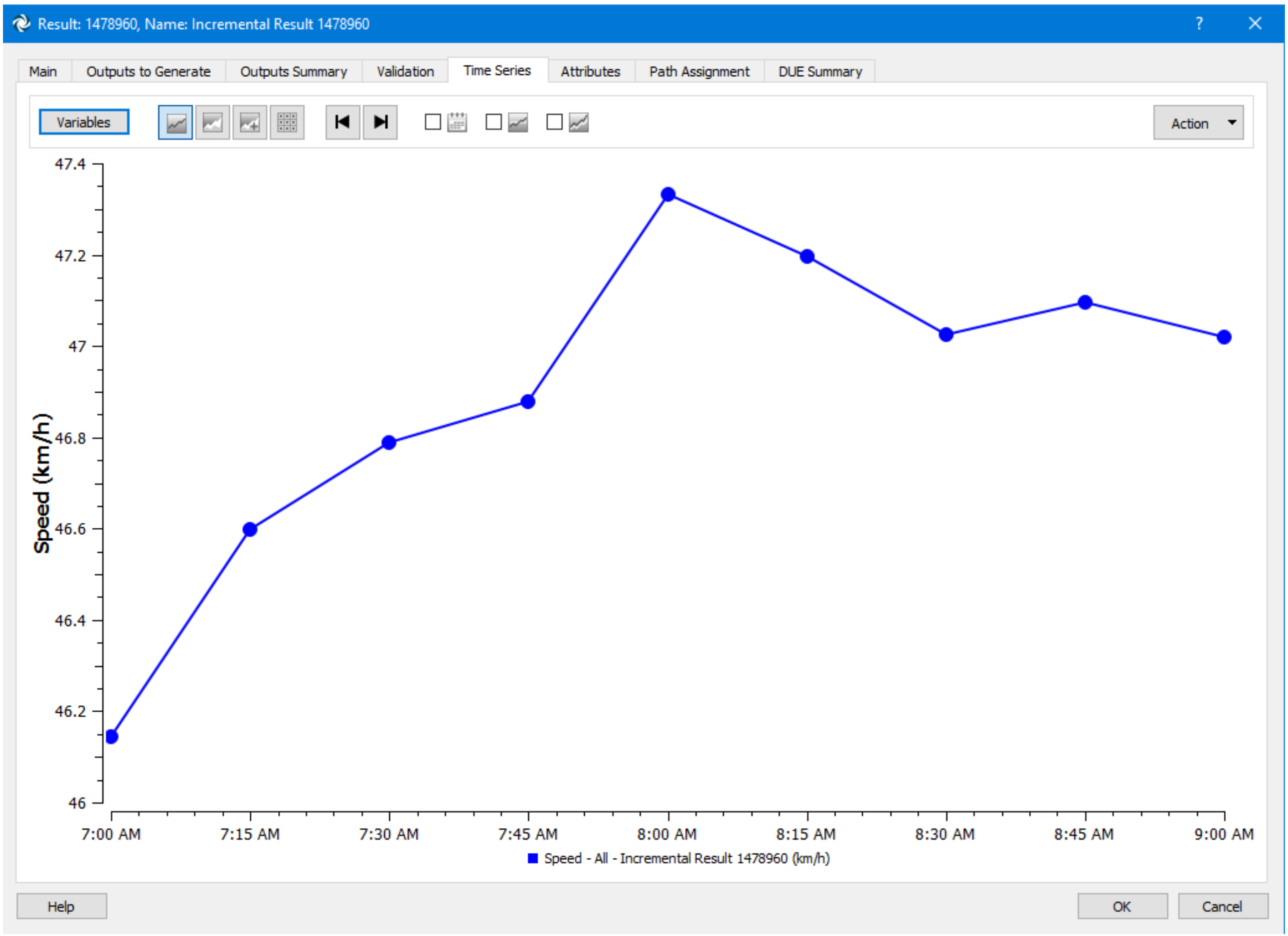
Default case, setting in DTA: Gradient-based

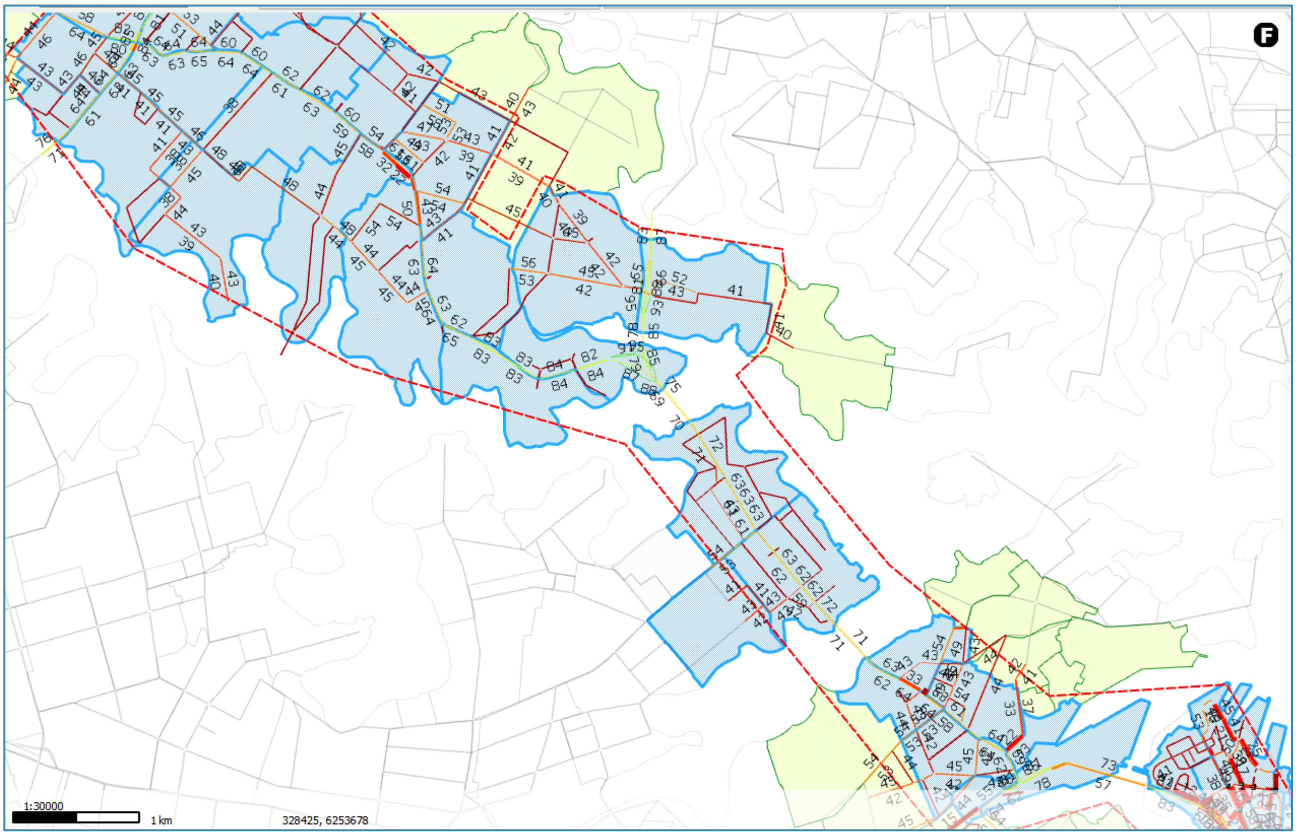
Convergence and Validation



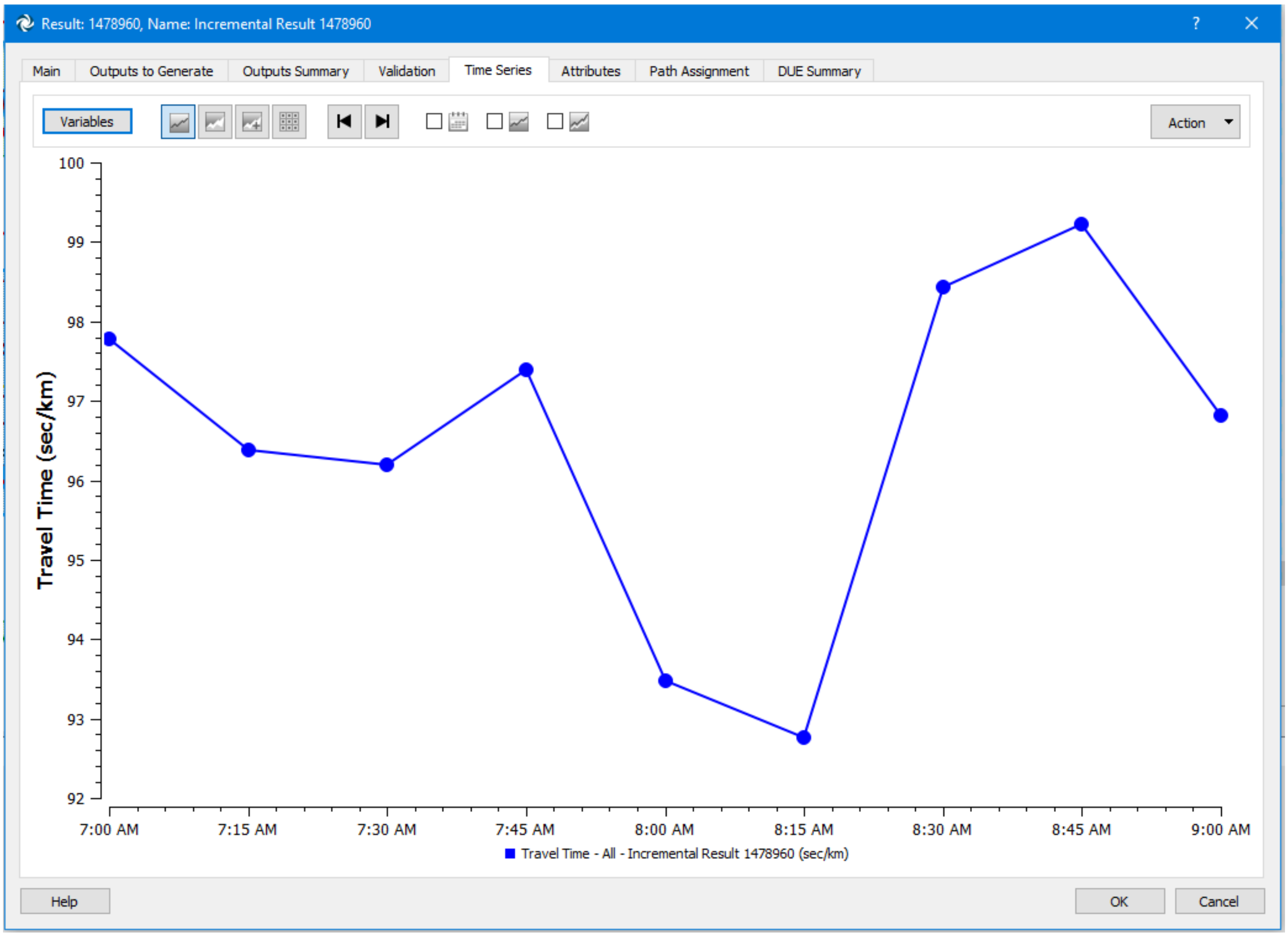


Speed (two Incremental Results with the same Random Seed)

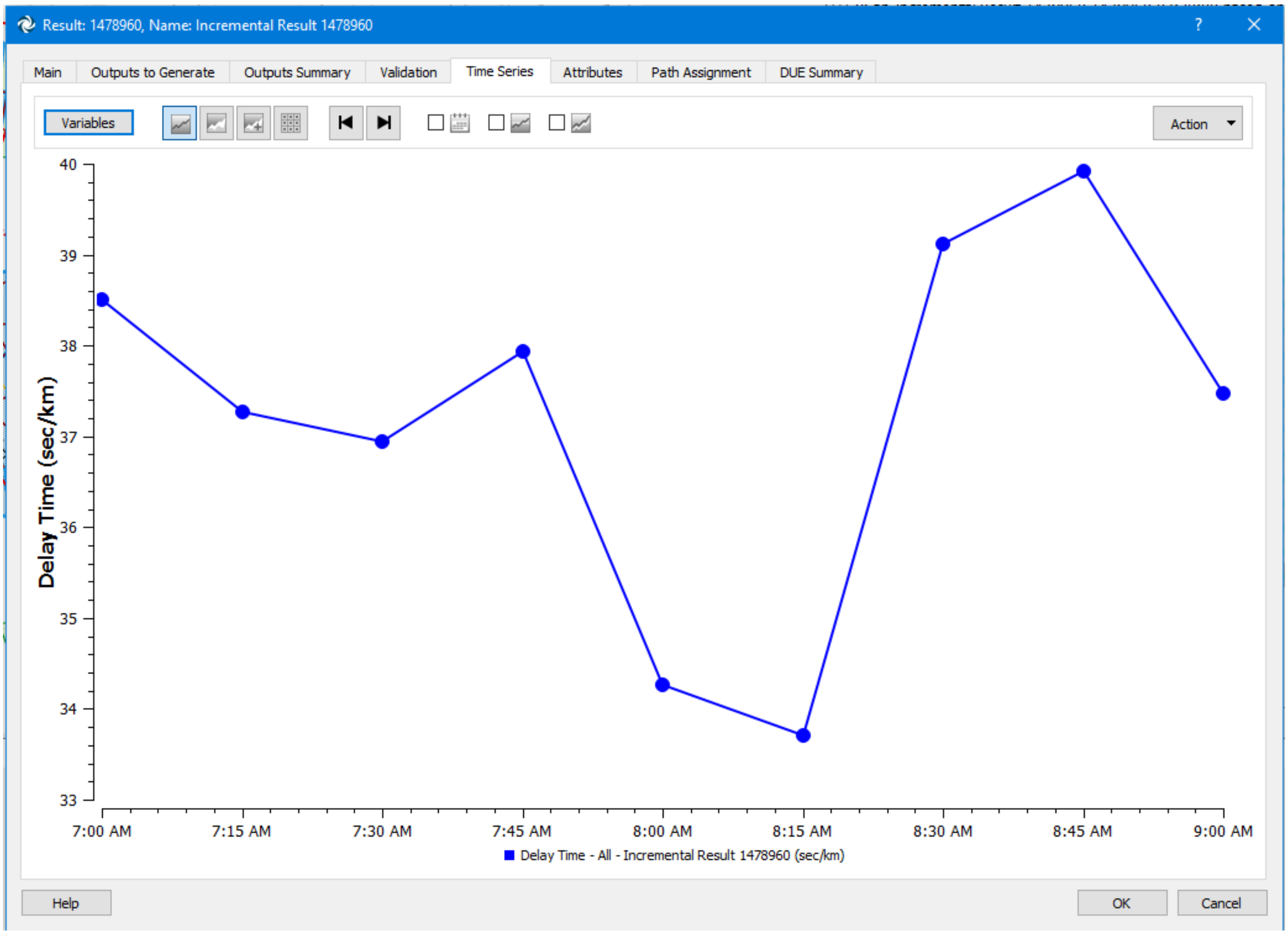


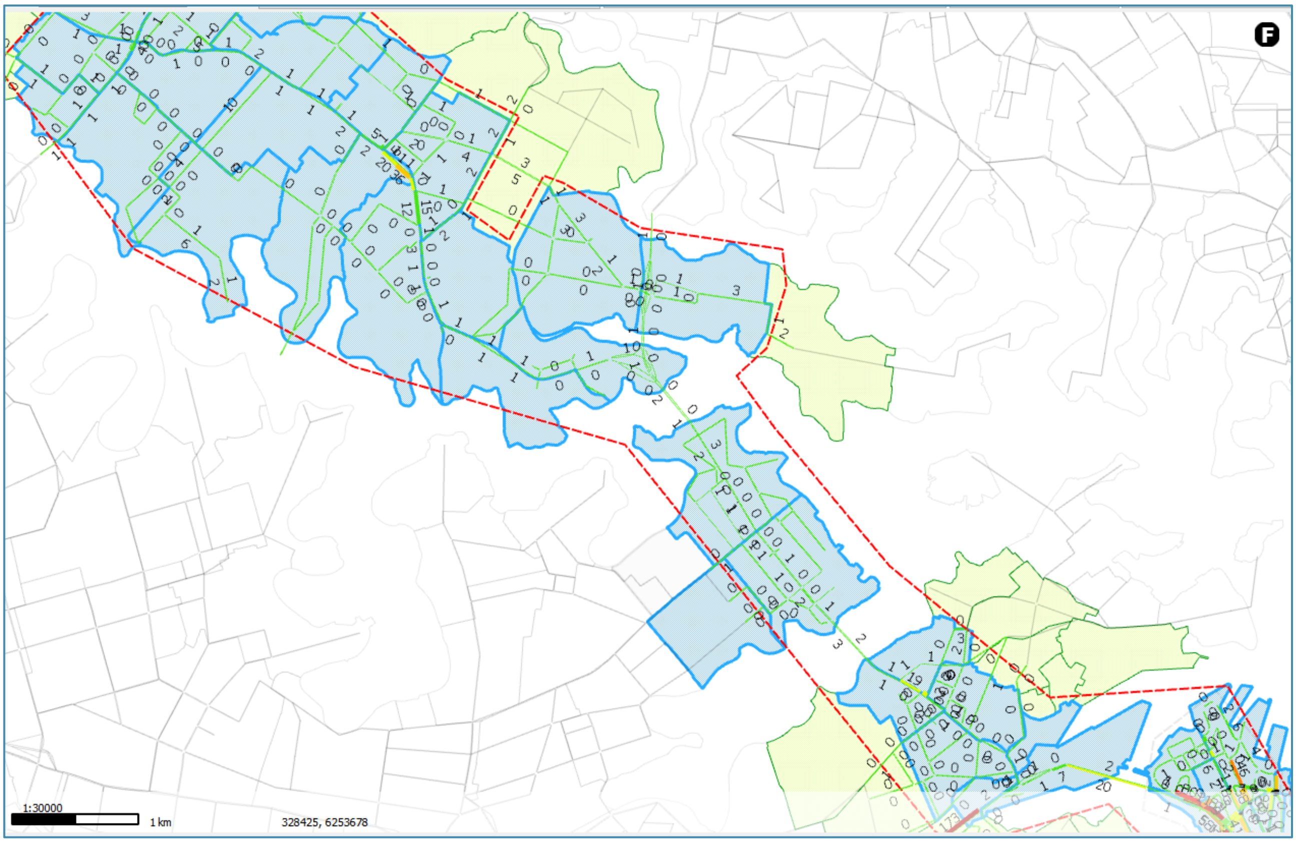


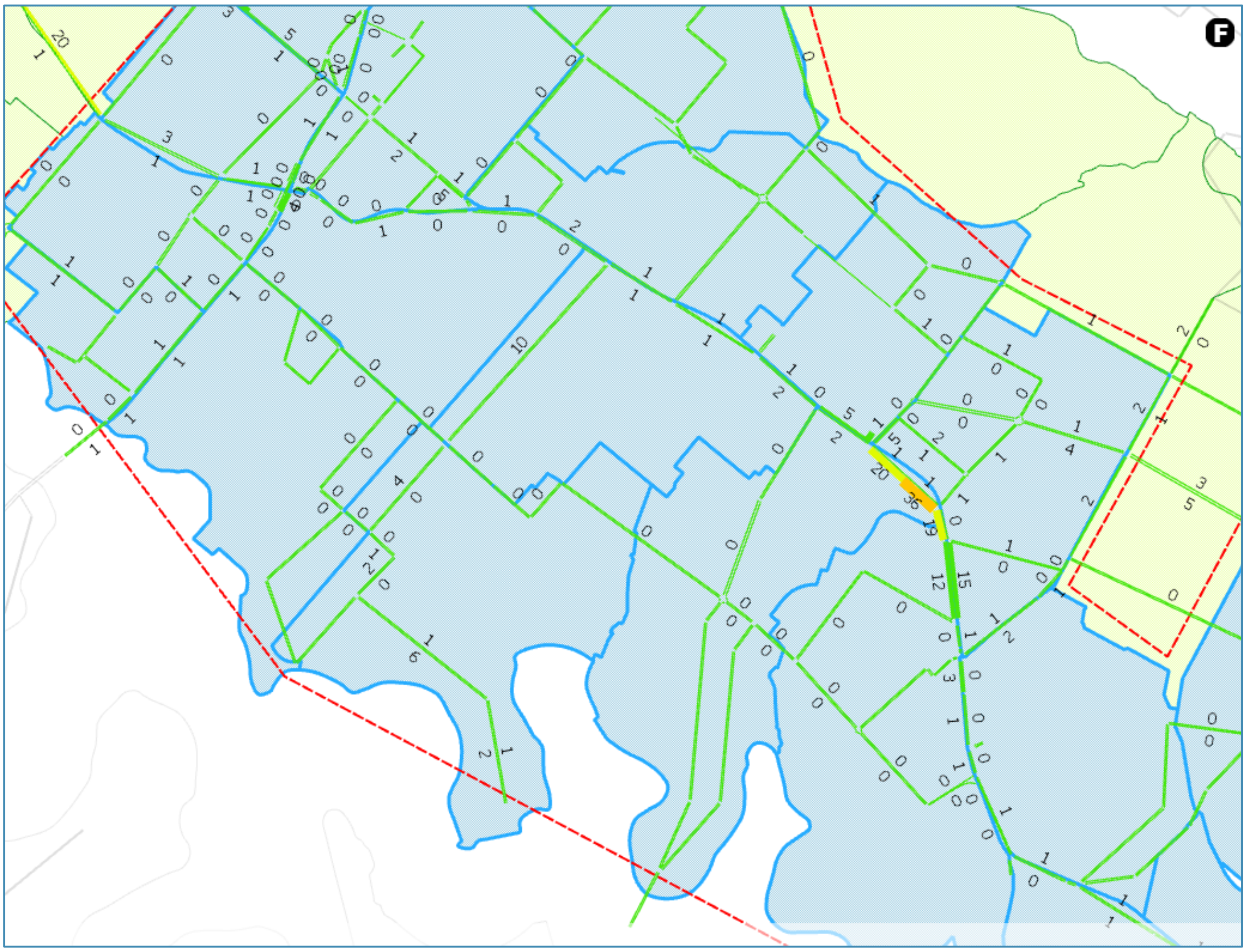
Travel time – all vehicles

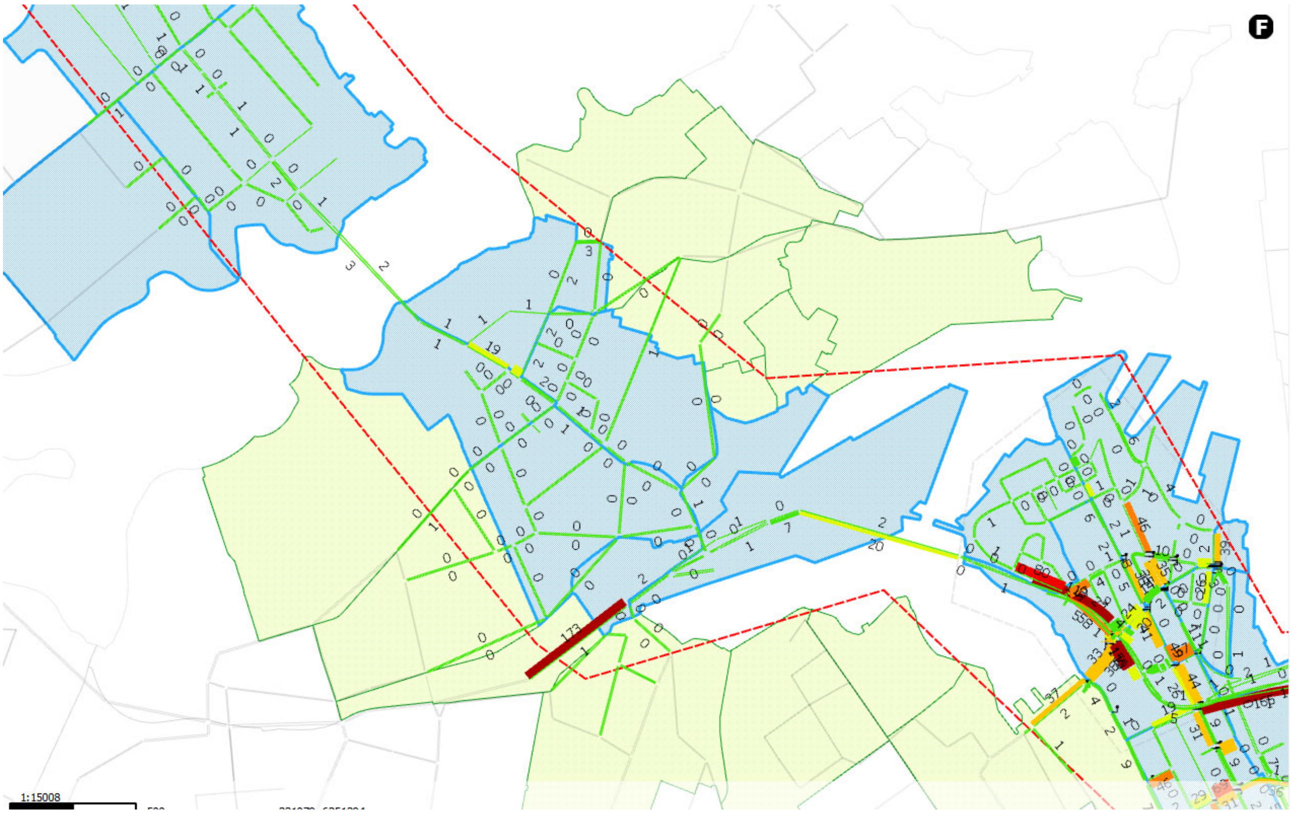


Delay time


Result: 1478960, Name: Incremental Result 1478960
Main
Outputs to Generate
Outputs Summary
Validation
Time Series
Attributes
Path Assignment
DUE Summary
Variables
Action
Delay Time (sec/km)
40
39
38
37
36
35
34
33
7:00 AM
7:15 AM
7:30 AM
7:45 AM
8:00 AM
8:15 AM
8:30 AM
8:45 AM
9:00 AM
Delay Time - All - Incremental Result 1478960 (sec/km)
Help
OK
Cancel


1:30000
1 km
328425, 6253678



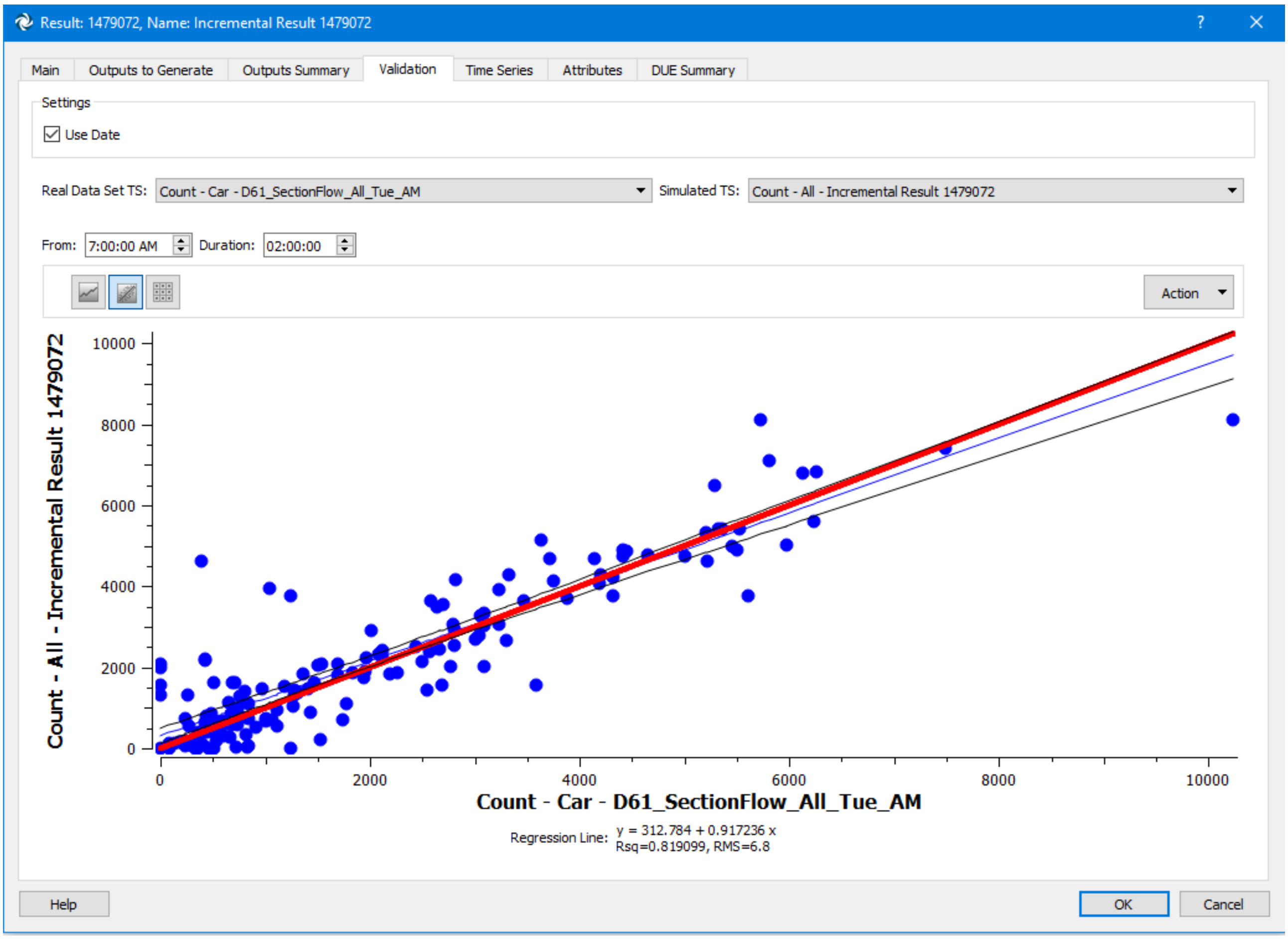

1:15008

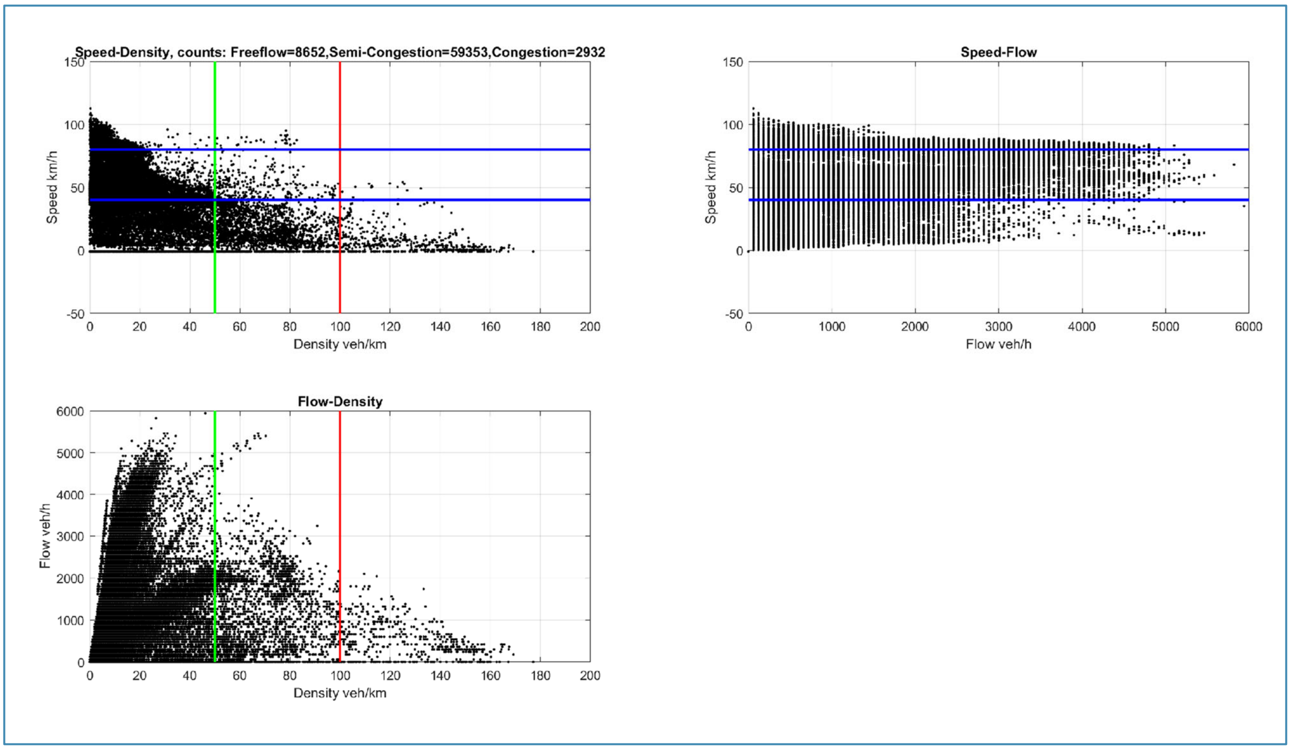
Fundamental Diagram (Stat recorded at 1-min interval)

The Stat was recorded with 1min interval, while other settings remain the same.

Incremental Result 1479072

Validation (for this 1-min interval simulation):

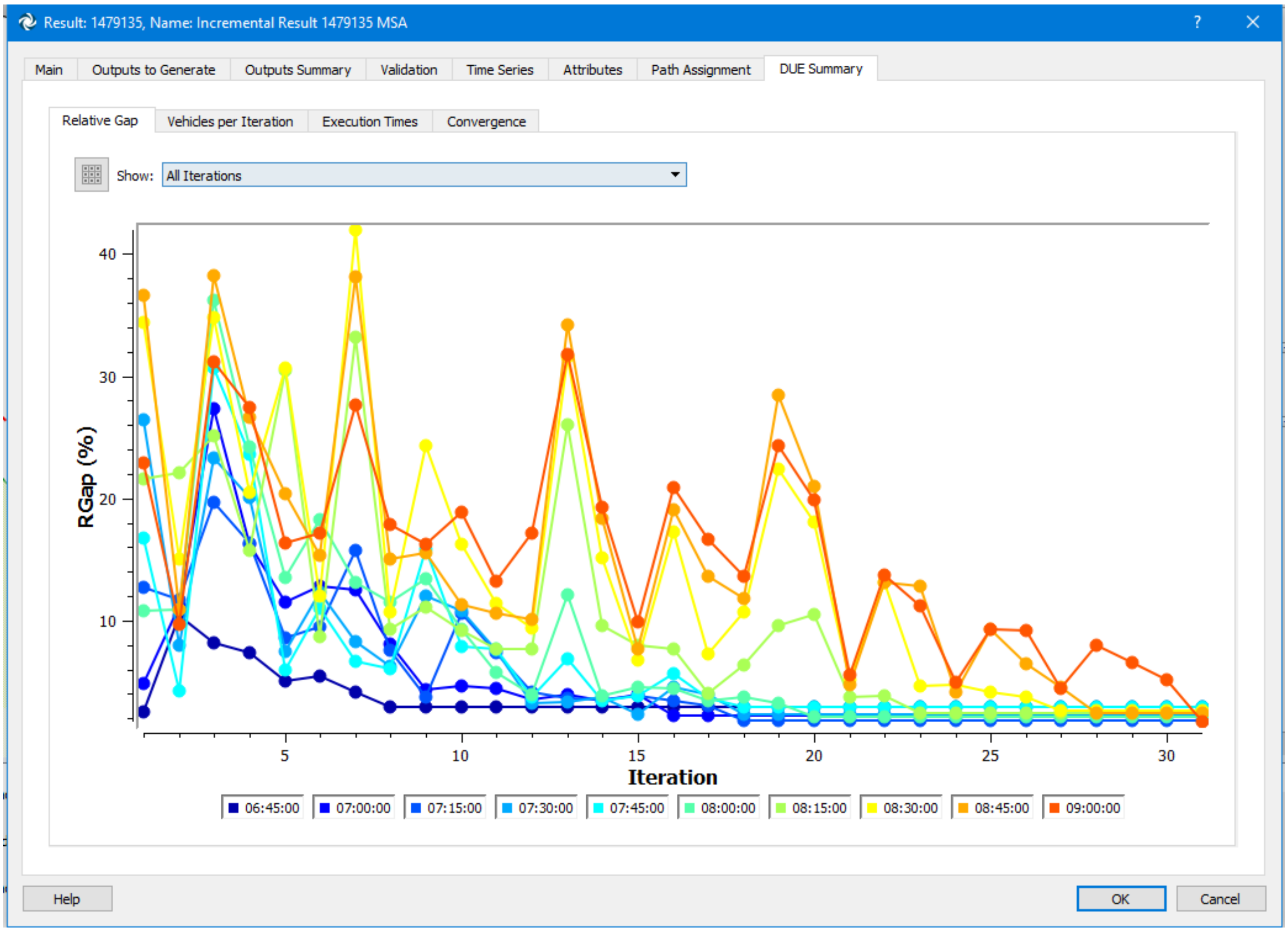




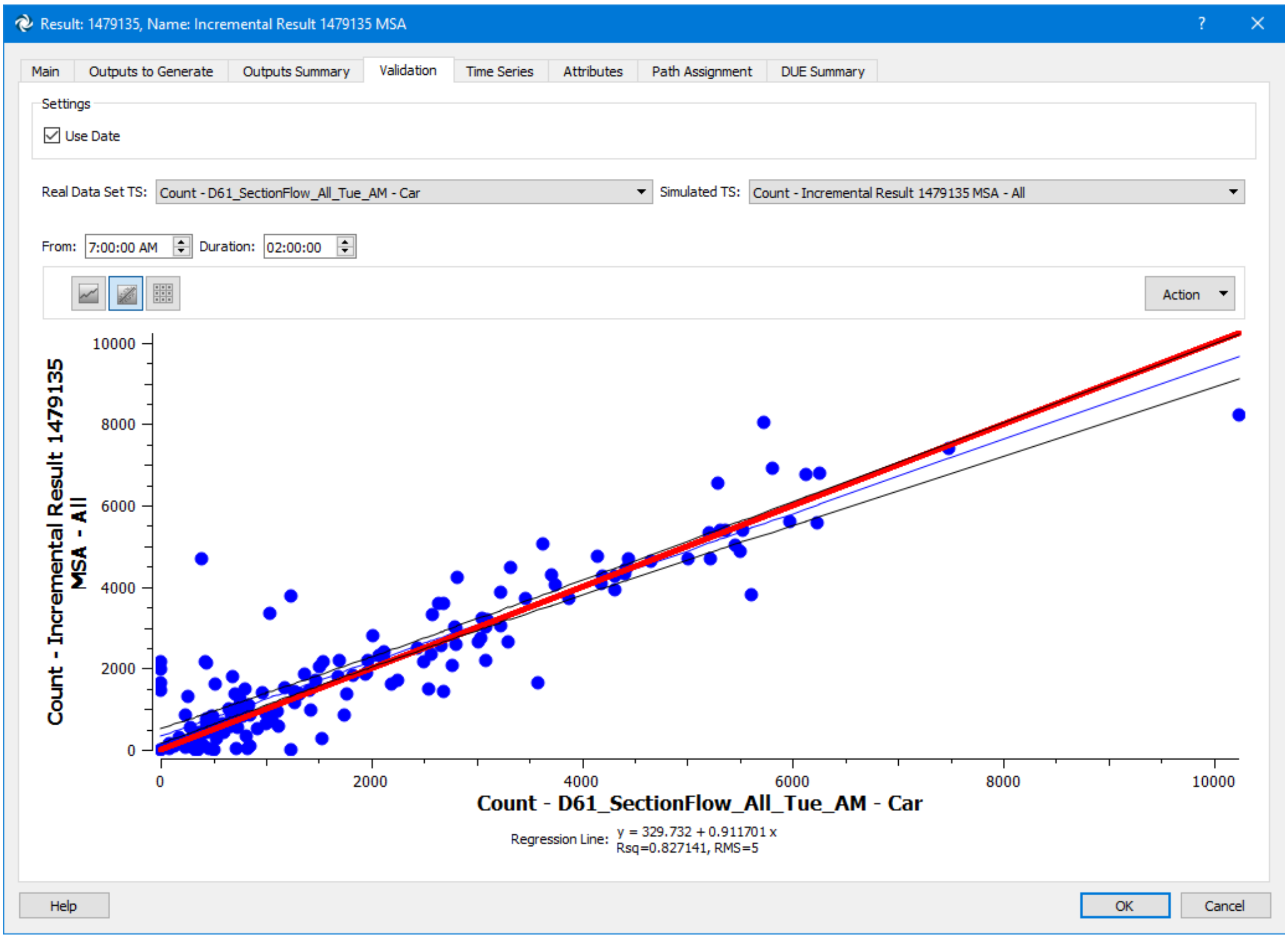
The Fundamental Diagram of the Meso DUE simulation shows a reasonable shape, indicating that the simulation has an overall "correct" result. However, there exist clusters of points where Density equals around 20. Also, fewer points are located at the "congestion area" where Density > 100 and Speed < 40. The reason for such outcome is thought to be caused by the huge number of sections (1067 out of 1994 within the subnetwork) with assigned volumes of zero. Therefore, the author proposed that further examination of the original traffic demand and OD matrix is necessary.

DTA Setting as MSA

Convergence and Validation

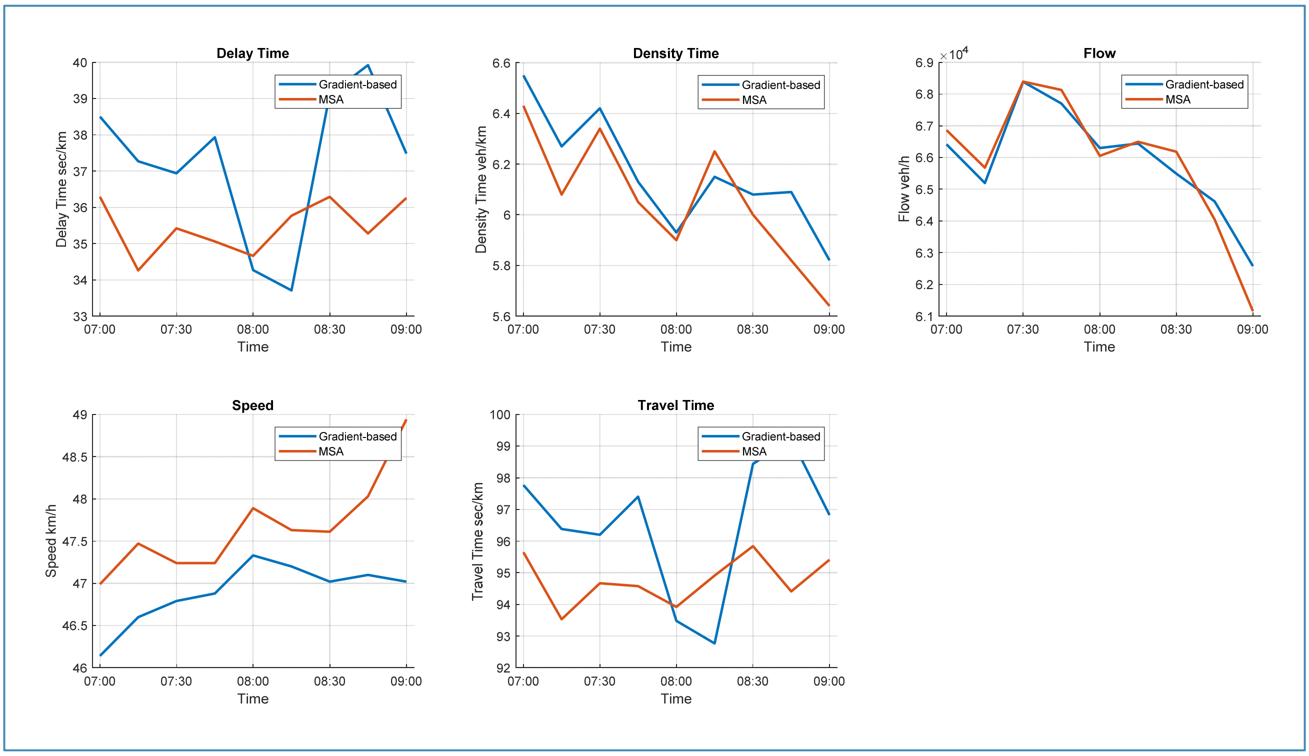


Takes longer to converge (The Gradient case used 20 iterations).

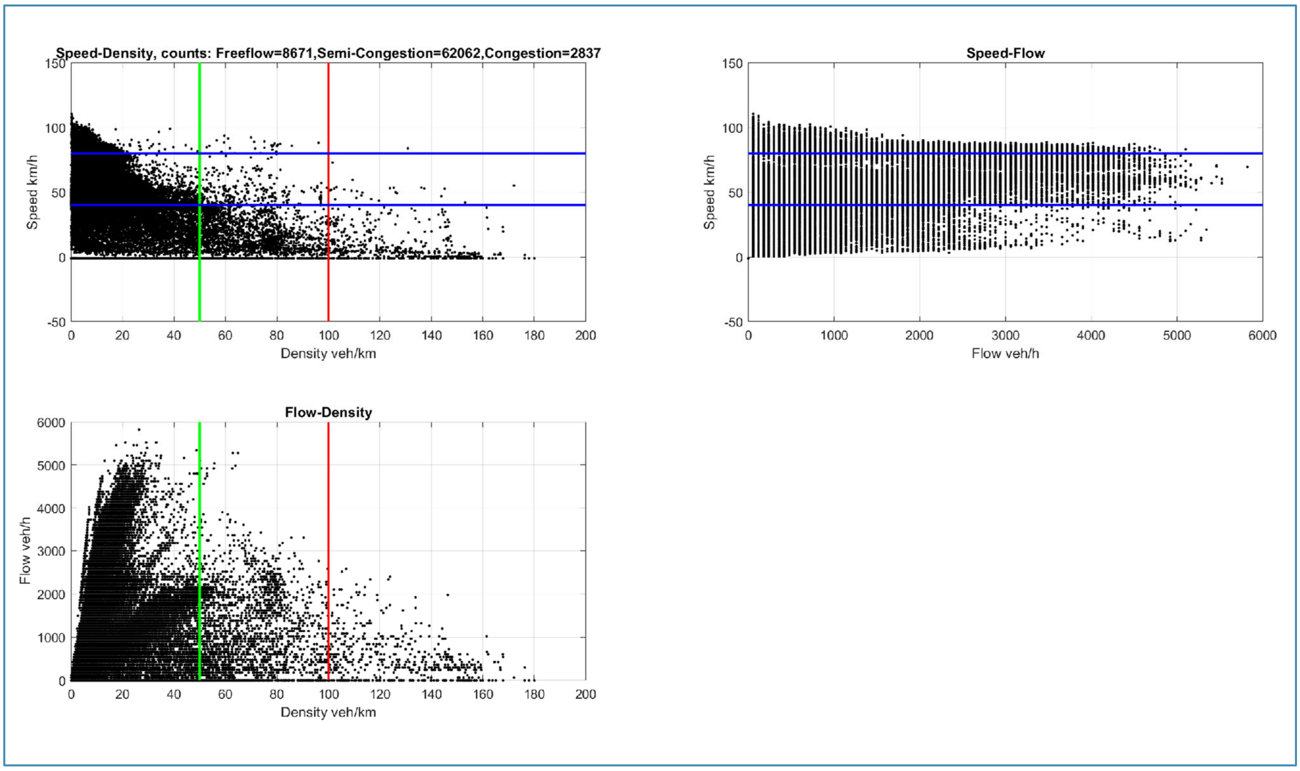


The R2 is slightly better than the previous case (with Gradient).

Time Series comparison with Gradient-based simulation



FD with 1-min interval (MSA)



To summarise, the MSA Dynamic Traffic Assignment has a slightly better result compared to what of the Gradient-boosted method. However, the difference in the R^2 is not significant. Hence, the author chose to continue using the Gradient-boosted method for the sake of consistency.

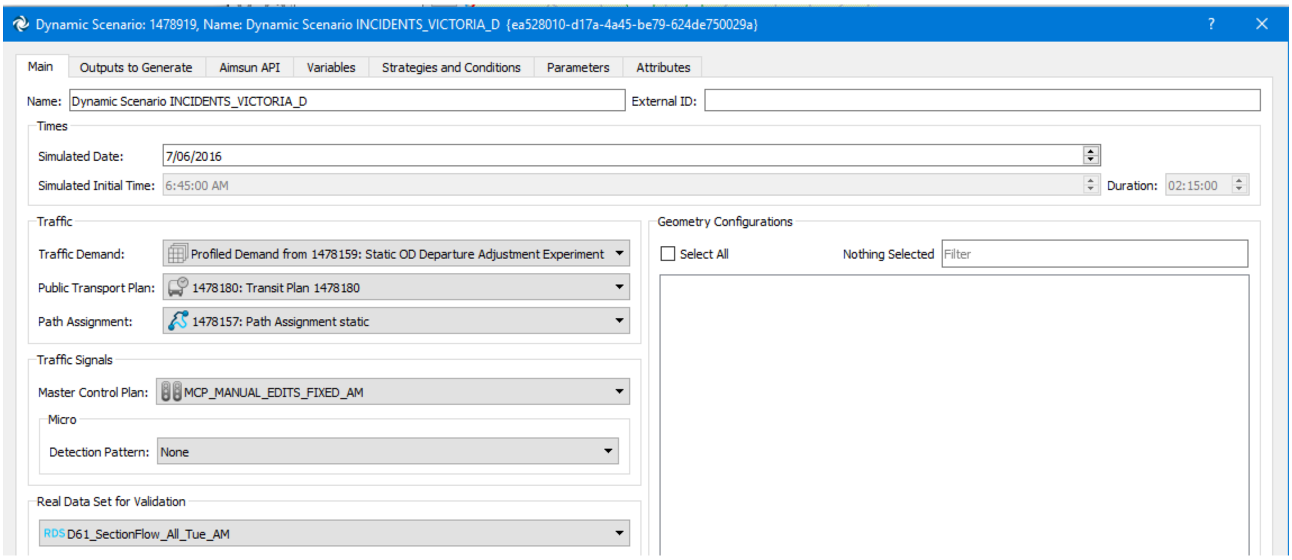
Note that the author chose to discard the Dynamic OD Adjustment as it is extremely time consuming and Aimsun exhibited instability and crashing during the process.

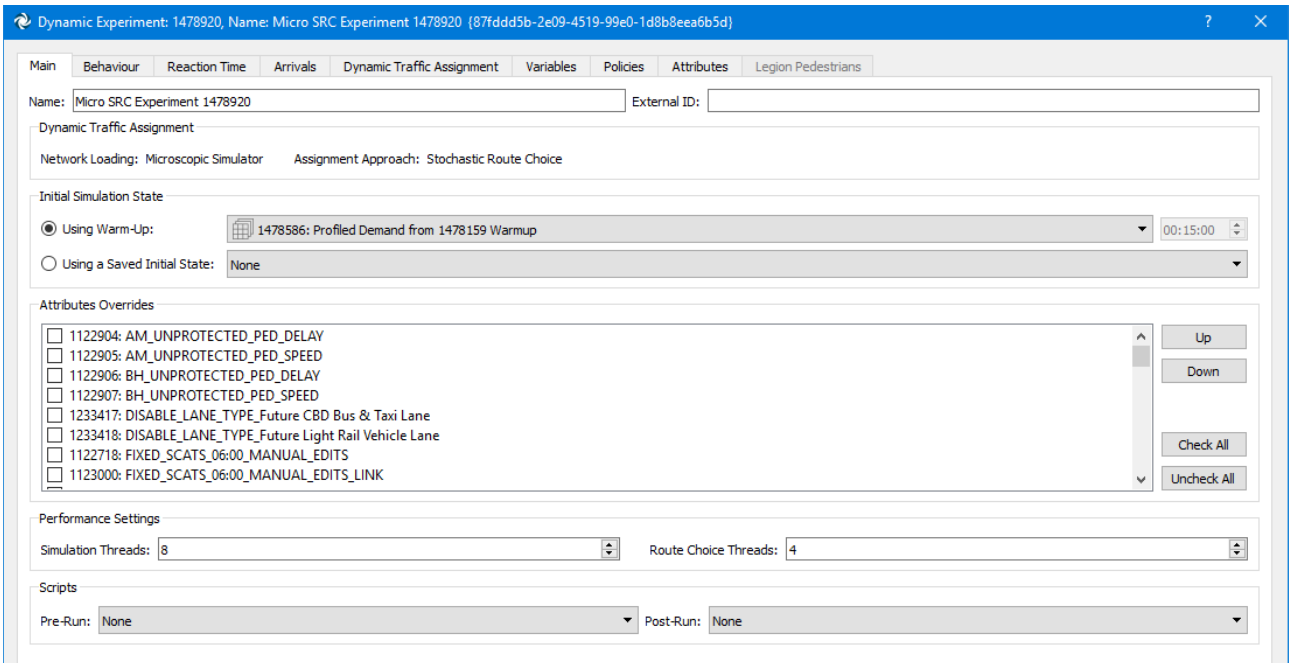
5.2. Microscopic Simulation (compared with Test incident id = 9999)

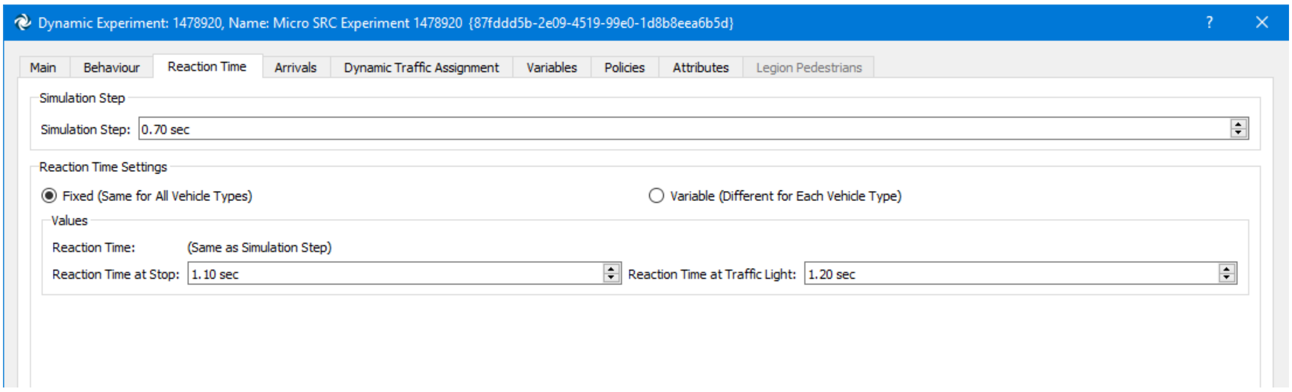
The default setting for the microscopic scenario is: Average of 10 Replication, stat recorded with an interval of 15 minutes.

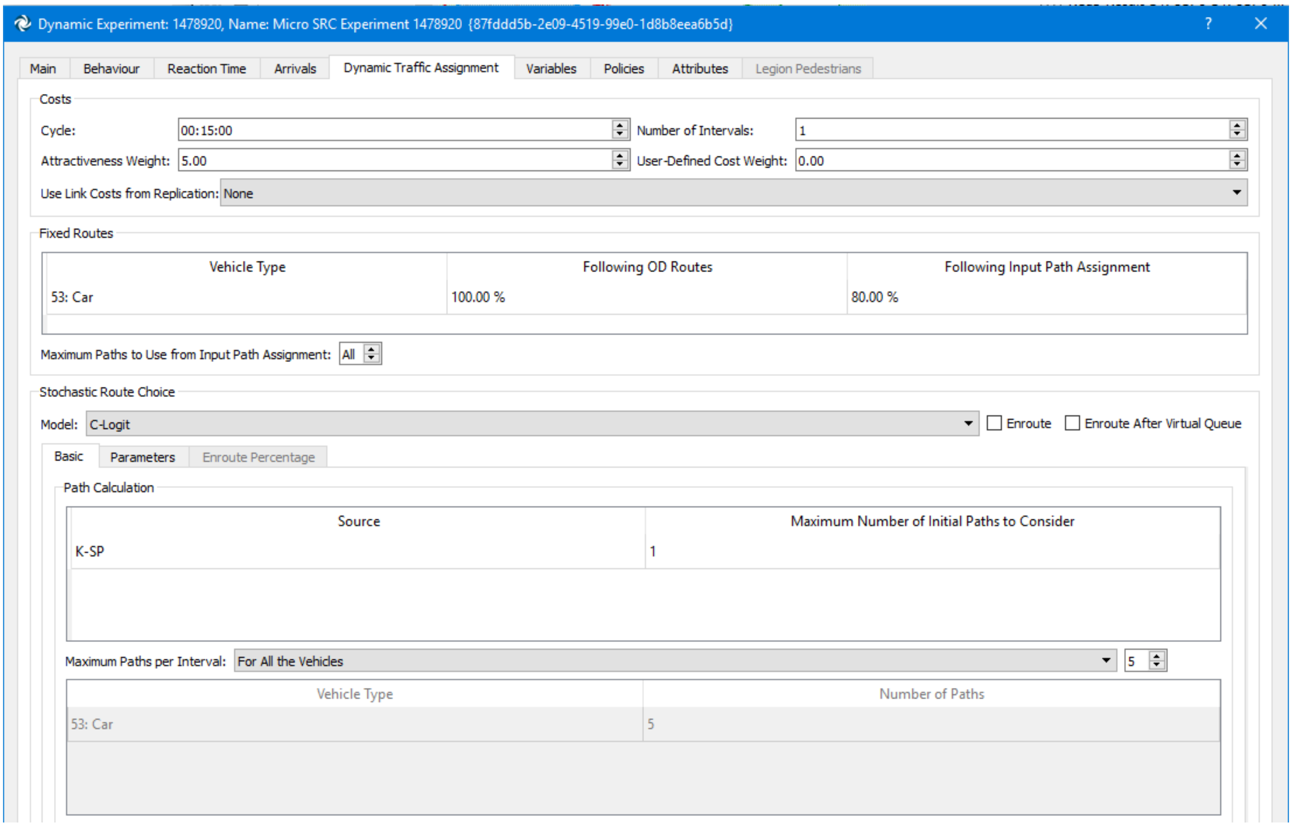
Create Scenarios with and without incident

Create Scenario without incidents (manual)

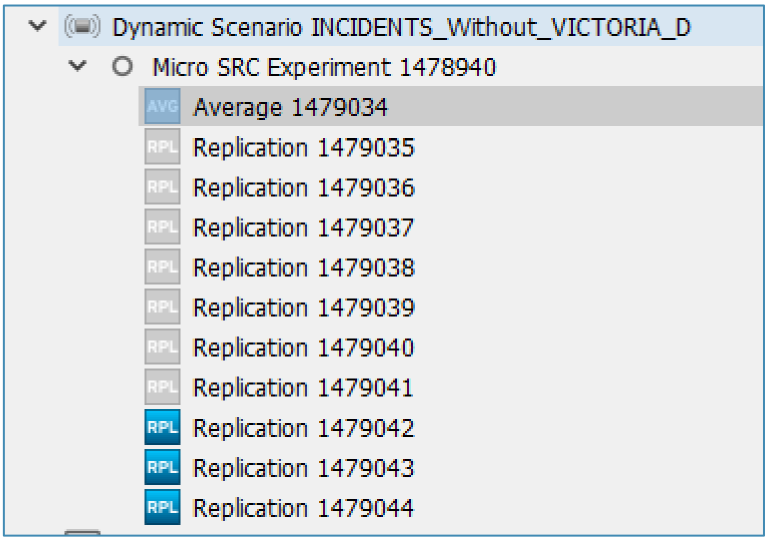



Dynamic Experiment: 1478920, Name: Micro SRC Experiment 1478920 {87fddd5b-2e09-4519-99e0-1d8b8eea6b5d}
Main
Behaviour
Reaction Time
Arrivals
Dynamic Traffic Assignment
Variables
Policies
Attributes
Legion Pedestrians
Name: Micro SRC Experiment 1478920
External ID:
Dynamic Traffic Assignment
Network Loading: Microscopic Simulator
Assignment Approach: Stochastic Route Choice
Initial Simulation State
Using Warm-Up: 1478586: Profiled Demand from 1478159 Warmup
00:15:00
Using a Saved Initial State: None
Attributes Overrides
1122904: AM_UNPROTECTED_PED_DELAY
1122905: AM_UNPROTECTED_PED_SPEED
1122906: BH_UNPROTECTED_PED_DELAY
1122907: BH_UNPROTECTED_PED_SPEED
1233417: DISABLE_LANE_TYPE_Future CBD Bus & Taxi Lane
1233418: DISABLE_LANE_TYPE_Future Light Rail Vehicle Lane
1122718: FIXED_SCATS_06:00_MANUAL_EDITS
1123000: FIXED_SCATS_06:00_MANUAL_EDITS_LINK
Up
Down
Check All
Uncheck All
Performance Settings
Simulation Threads: 8
Route Choice Threads: 4
Scripts
Pre-Run: None
Post-Run: None

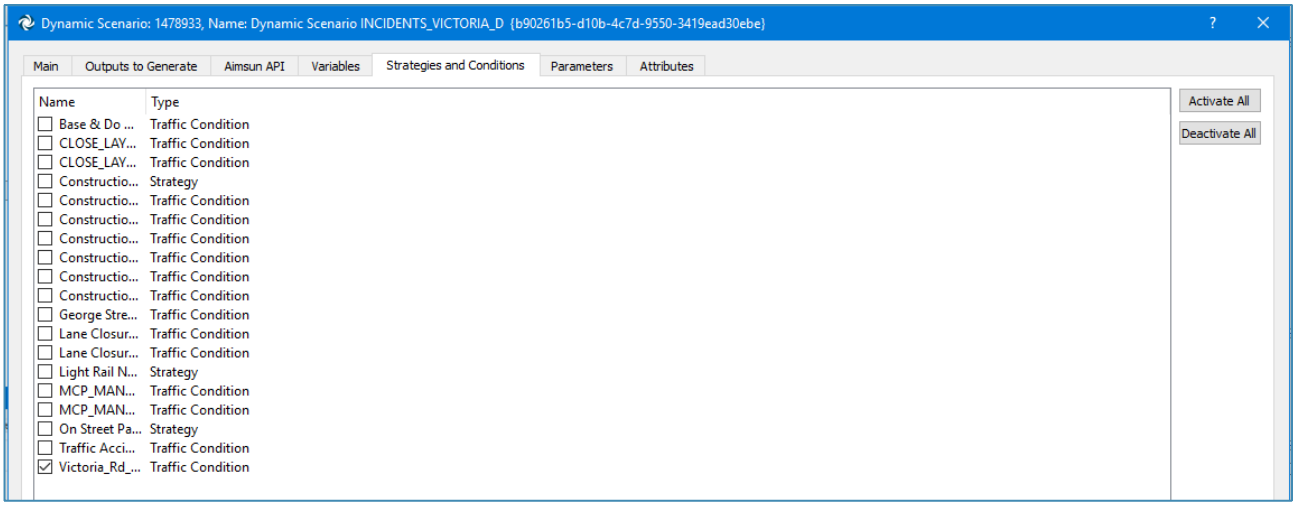

Dynamic Experiment: 1478920, Name: Micro SRC Experiment 1478920 {87fddd5b-2e09-4519-99e0-1d8b8eea6b5d}
Main
Behaviour
Reaction Time
Arrivals
Dynamic Traffic Assignment
Variables
Policies
Attributes
Legion Pedestrians
Simulation Step
Simulation Step: 0.70 sec
Reaction Time Settings
Fixed (Same for All Vehicle Types)
Variable (Different for Each Vehicle Type)
Values
Reaction Time: (Same as Simulation Step)
Reaction Time at Stop: 1.10 sec
Reaction Time at Traffic Light: 1.20 sec

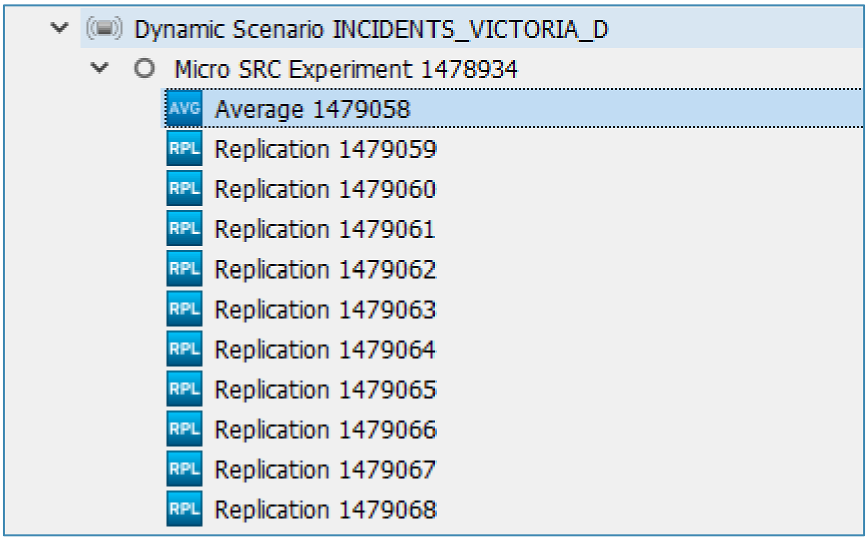

Dynamic Experiment: 1478920, Name: Micro SRC Experiment 1478920 {87fddd5b-2e09-4519-99e0-1d8b8eea6b5d}
Main
Behaviour
Reaction Time
Arrivals
Dynamic Traffic Assignment
Variables
Policies
Attributes
Legion Pedestrians
Costs
Cycle: 00:15:00
Number of Intervals: 1
Attractiveness Weight: 5.00
User-Defined Cost Weight: 0.00
Use Link Costs from Replication: None
Fixed Routes
Vehicle Type
Following OD Routes
Following Input Path Assignment
53: Car
100.00 %
80.00 %
Maximum Paths to Use from Input Path Assignment: All
Stochastic Route Choice
Model: C-Logit
Enroute
Enroute After Virtual Queue
Basic
Parameters
Enroute Percentage
Path Calculation
Source
Maximum Number of Initial Paths to Consider
K-SP
1
Maximum Paths per Interval: For All the Vehicles
5
Vehicle Type
Number of Paths
53: Car
5

Create an Average result for 10 Replications:

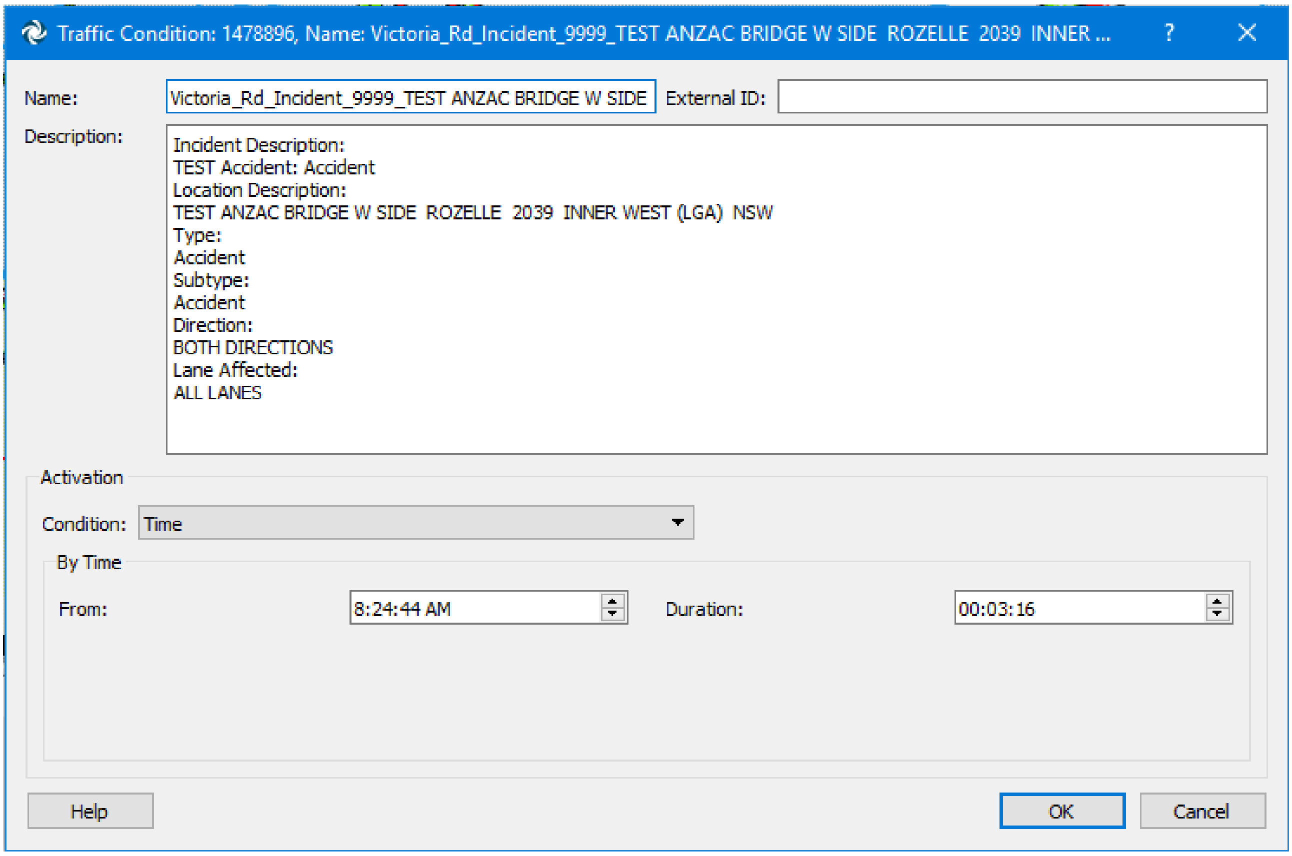


Create Scenario with incidents (script)



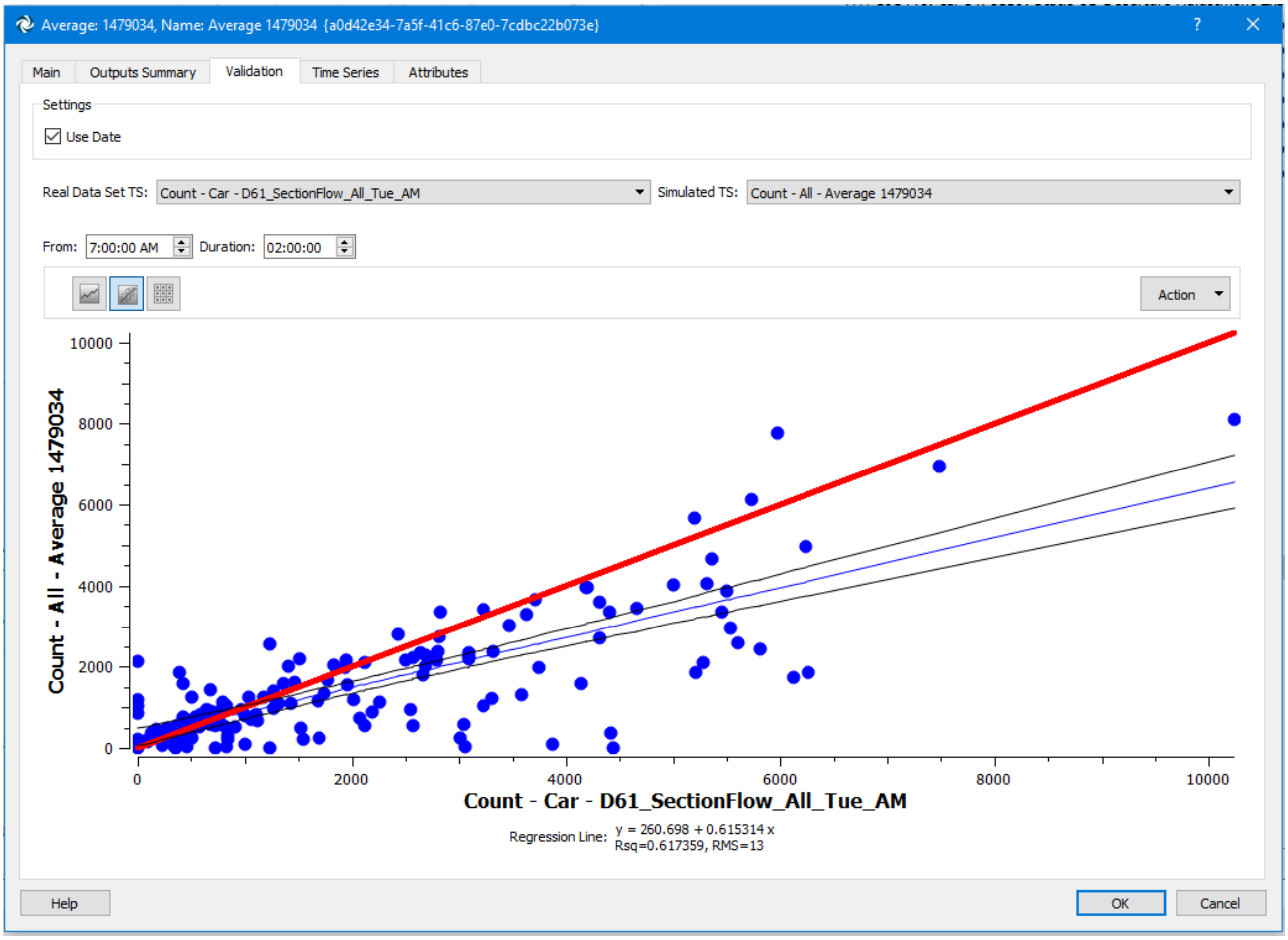


The incident (unique_id: 9999) was manually created by changing the date column; originally it was an incident took place in 2017 on the bridge to the west of Pyrmont.



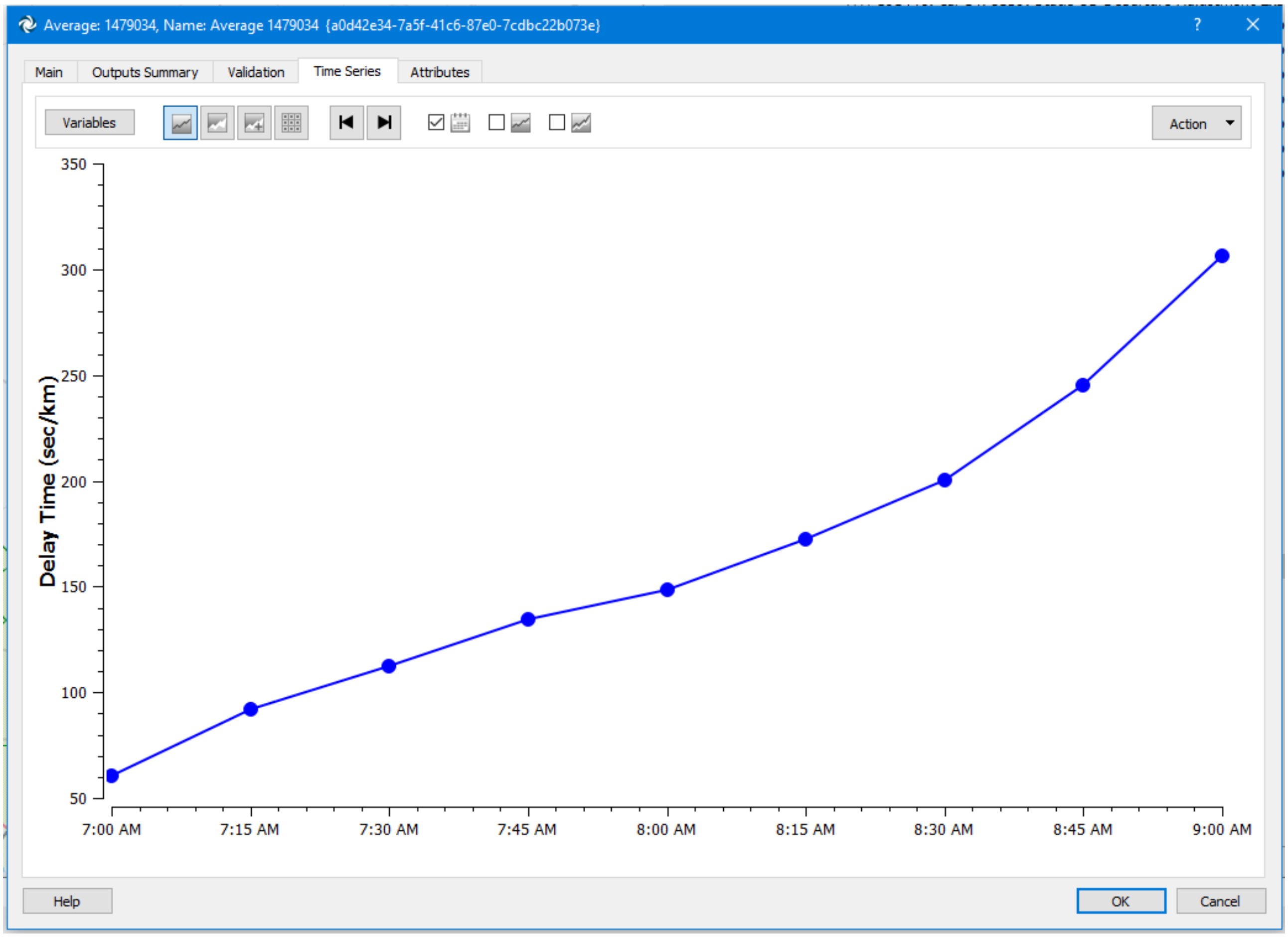
5.2.1 Default case Without incident

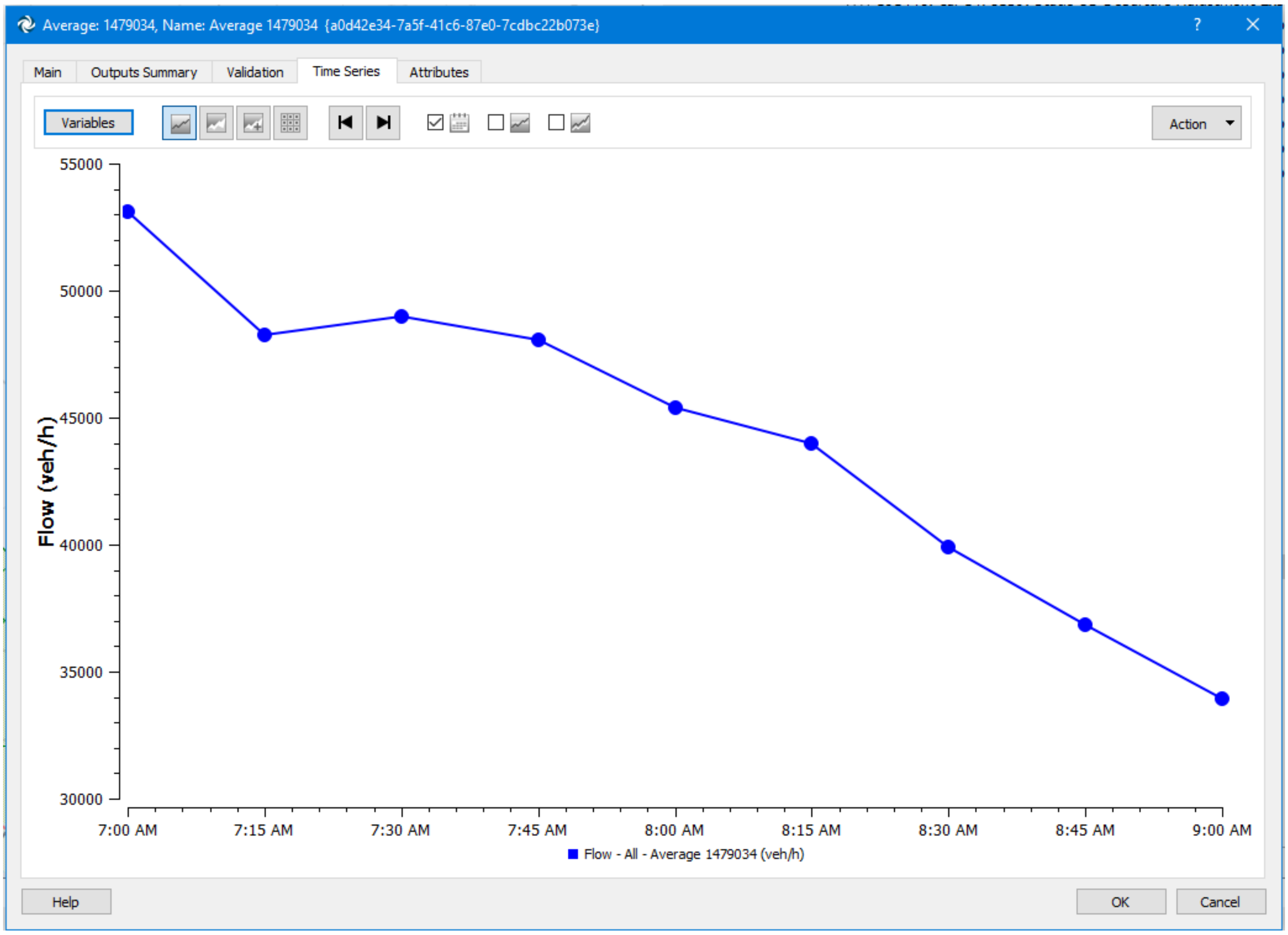
Validation



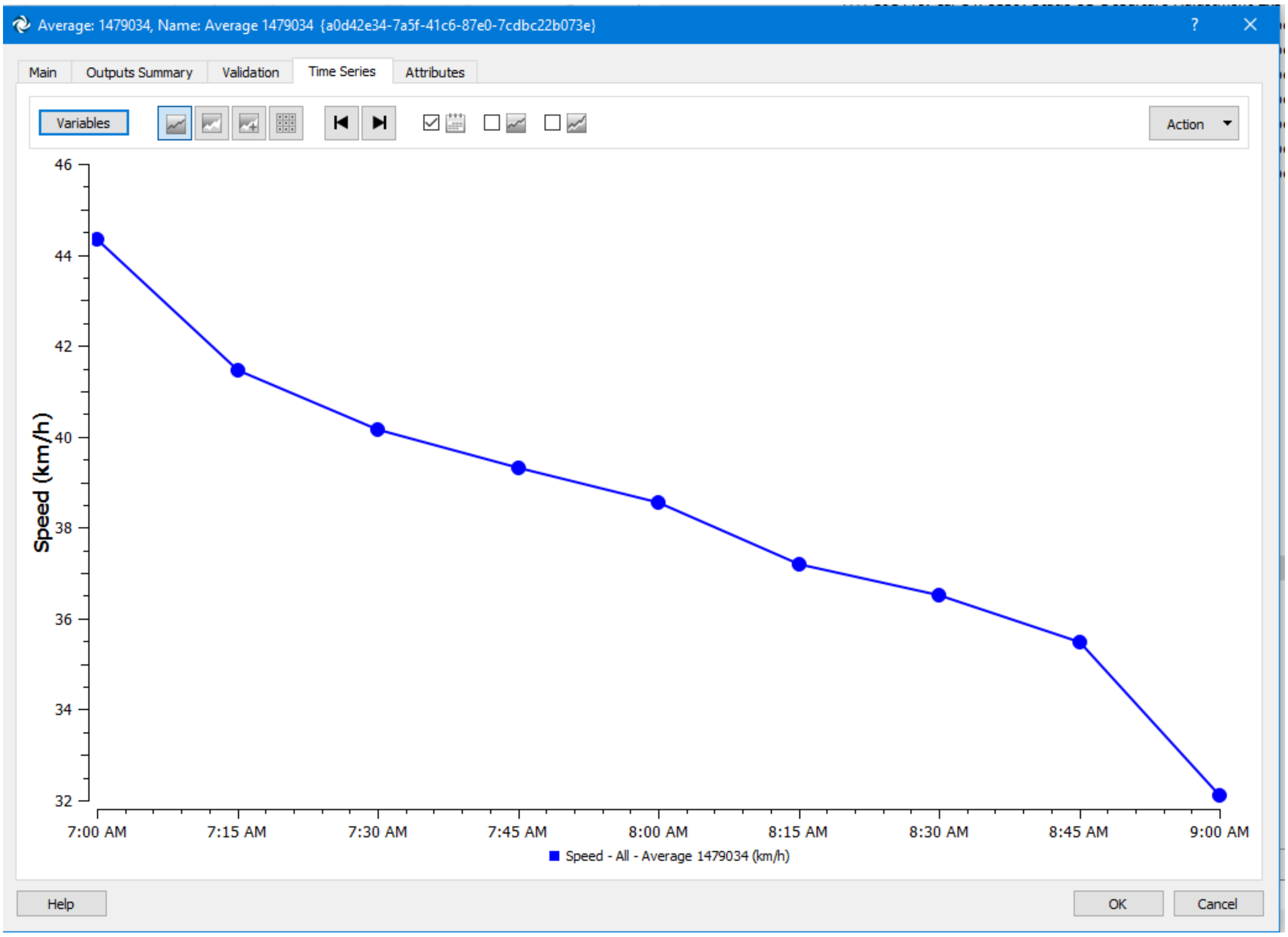
The average result has an R2 of 0.62, which is better than the individual replication recorded in the previous document.

Time-based parameters

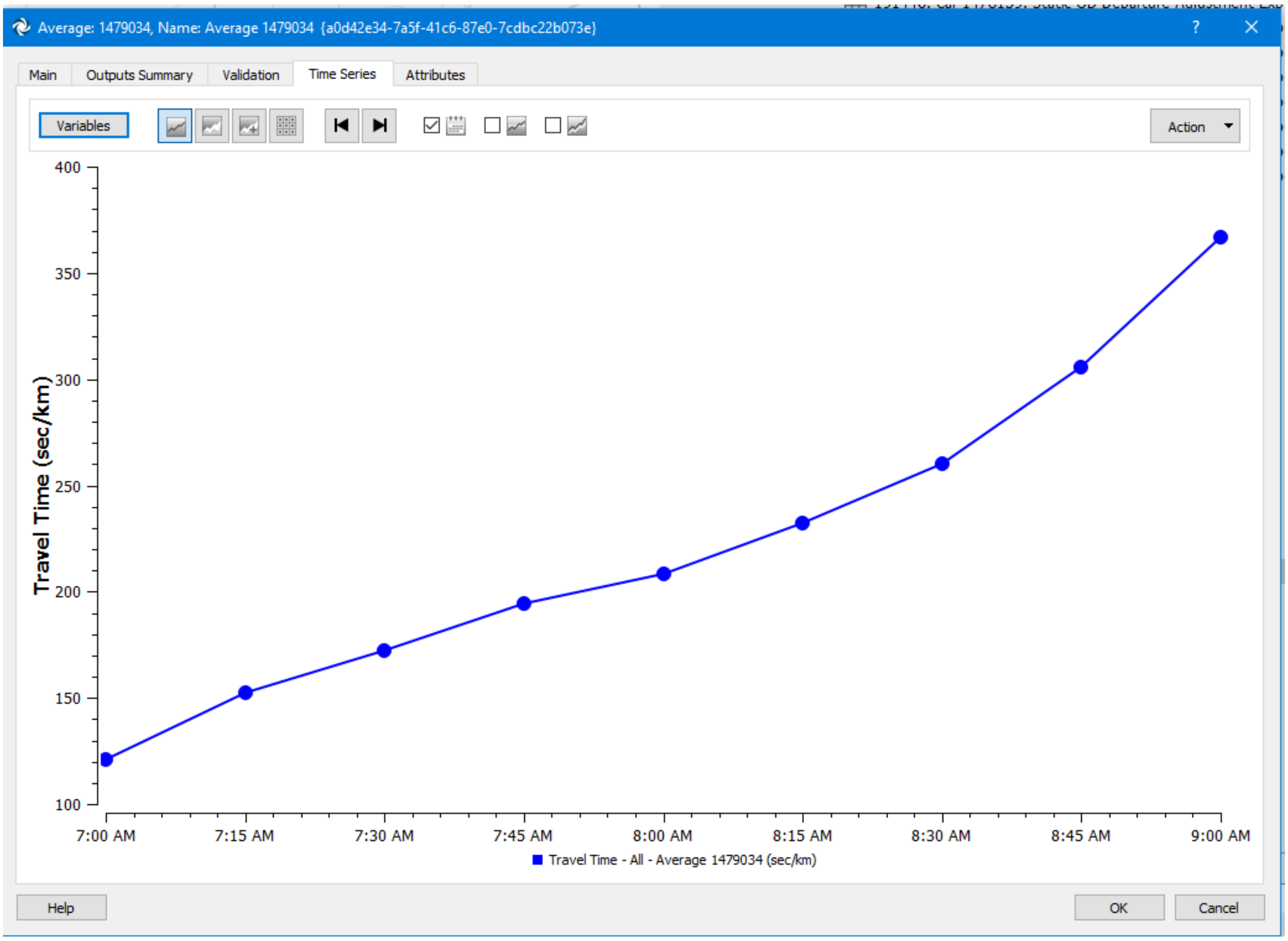




Average: 1479034, Name: Average 1479034 {a0d42e34-7a5f-41c6-87e0-7cdbc22b073e}
Main
Outputs Summary
Validation
Time Series
Attributes
Variables
Action
Flow (veh/h)
55000
50000
45000
40000
35000
30000
7:00 AM
7:15 AM
7:30 AM
7:45 AM
8:00 AM
8:15 AM
8:30 AM
8:45 AM
9:00 AM
Flow - All - Average 1479034 (veh/h)
Help
OK
Cancel

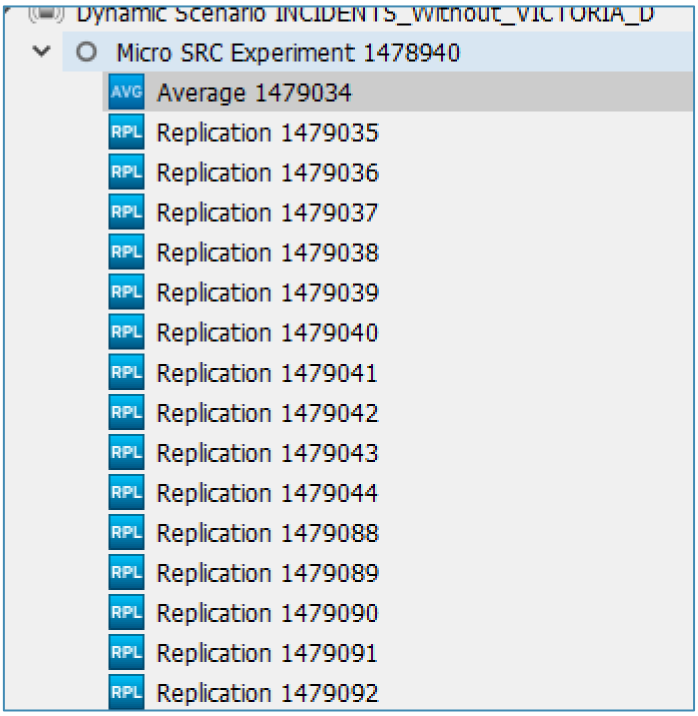


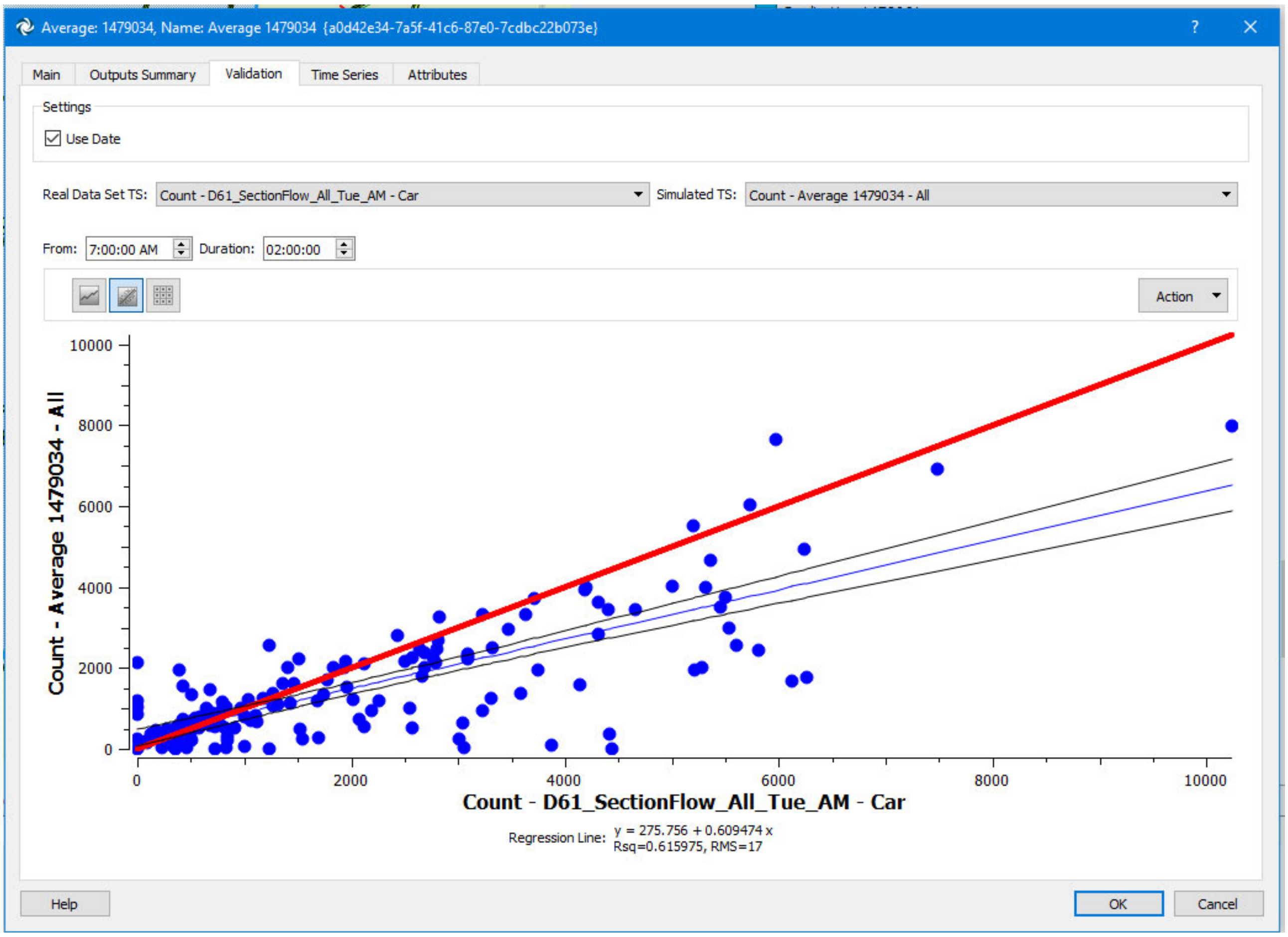
Average: 1479034, Name: Average 1479034 {a0d42e34-7a5f-41c6-87e0-7cdbc22b073e}
Main
Outputs Summary
Validation
Time Series
Attributes
Variables
Action
Speed (km/h)
46
44
42
40
38
36
34
32
7:00 AM
7:15 AM
7:30 AM
7:45 AM
8:00 AM
8:15 AM
8:30 AM
8:45 AM
9:00 AM
Speed - All - Average 1479034 (km/h)
Help
OK
Cancel



Default setting with 15 Replications - Average 1479034

Below is the Validation of the average of 15 Replications.

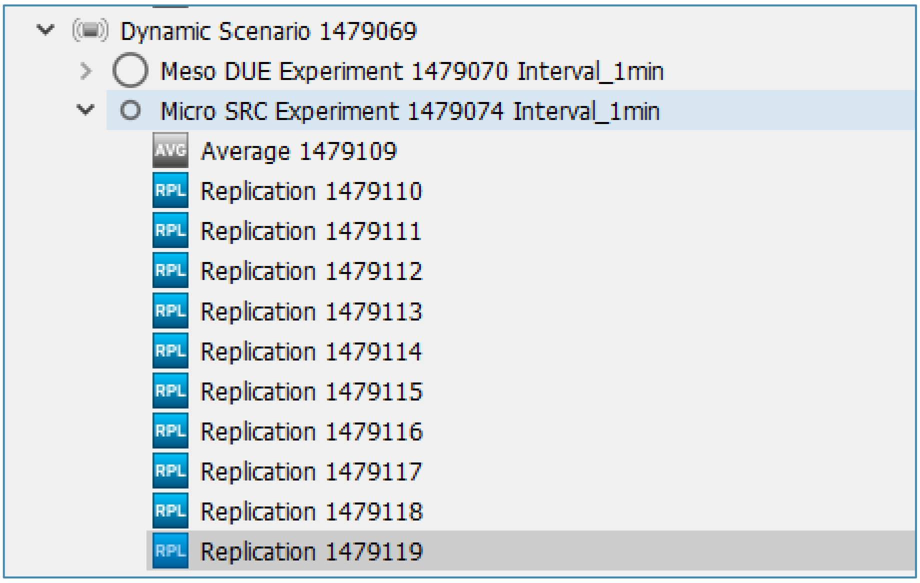




Compared to the Average of 10 Replications, the change in R2 is minimal. Therefore, the author will keep the default 10-Replication setting.

1-min Interval Fundamental Diagram

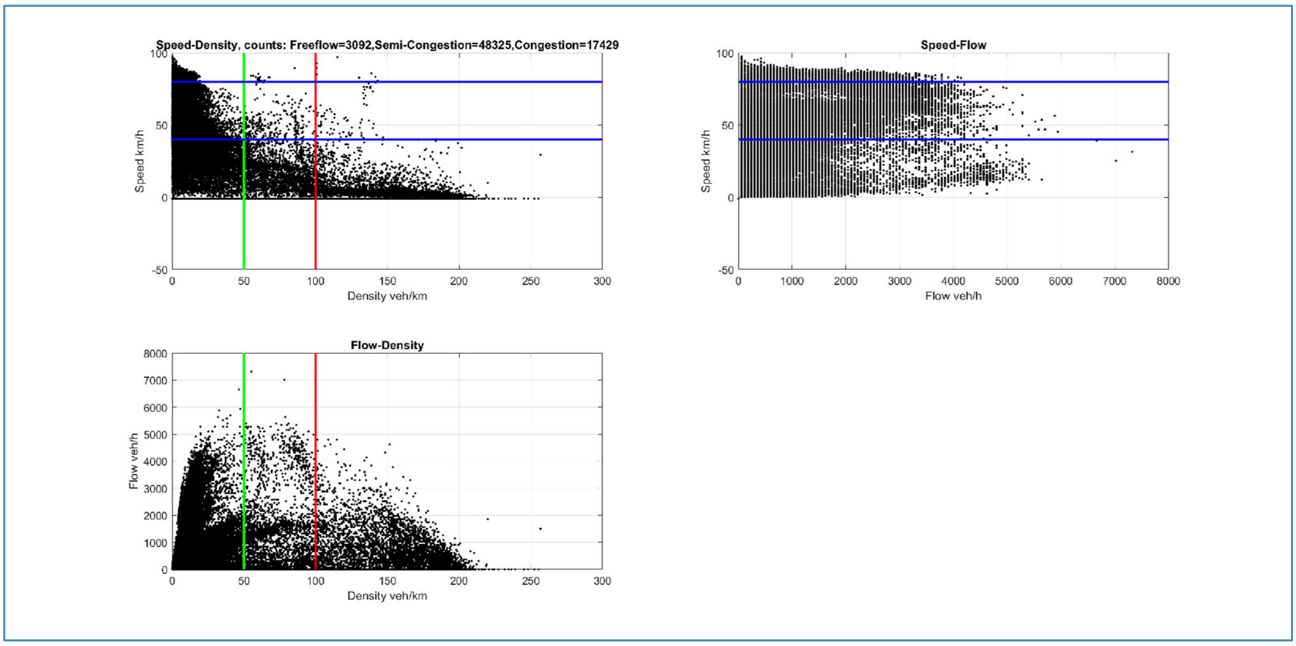
Stat recorded with 1min interval (note: the simulation took hours).



Initially, the author was hoping to obtain an Average of 10 Replications (same as above). However, the calculation could not be performed without Aimsun exhibiting instability and crashing; the reason is thought to be insufficient computer memory for processing the 1-min interval dataset. Therefore, the author was only able to obtain the data for the ten individual Replications.

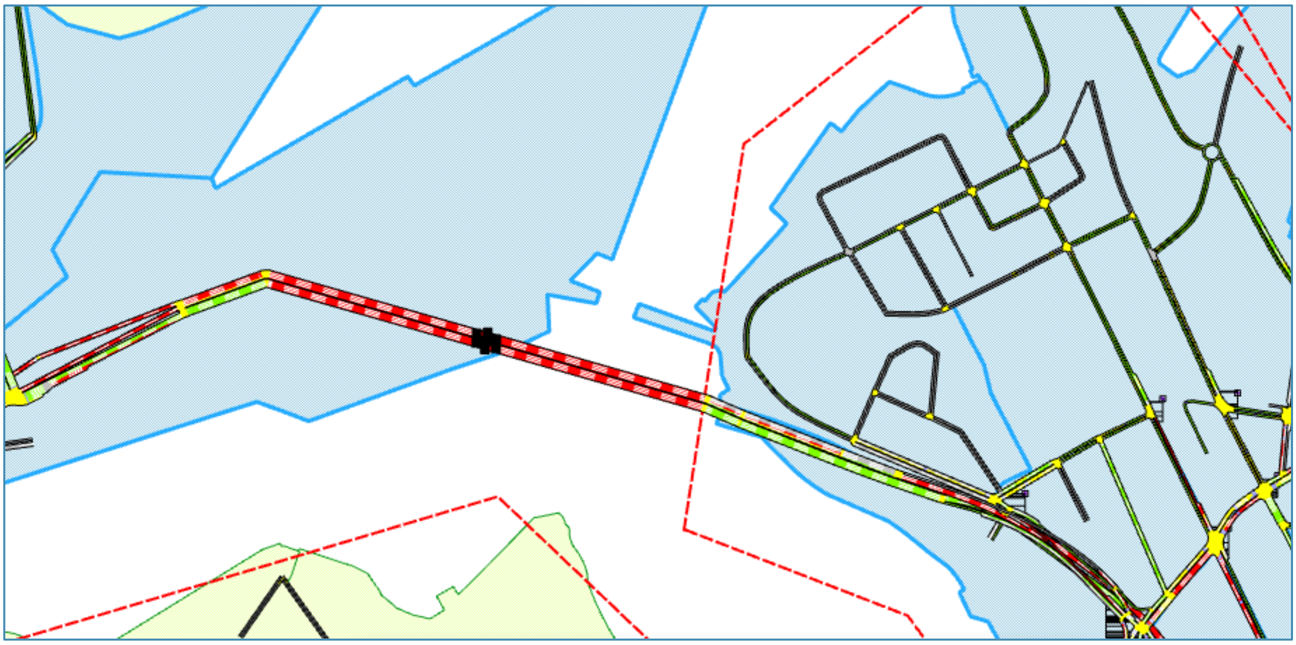
Below is the FD for one of the Replications.

Replication 1479110

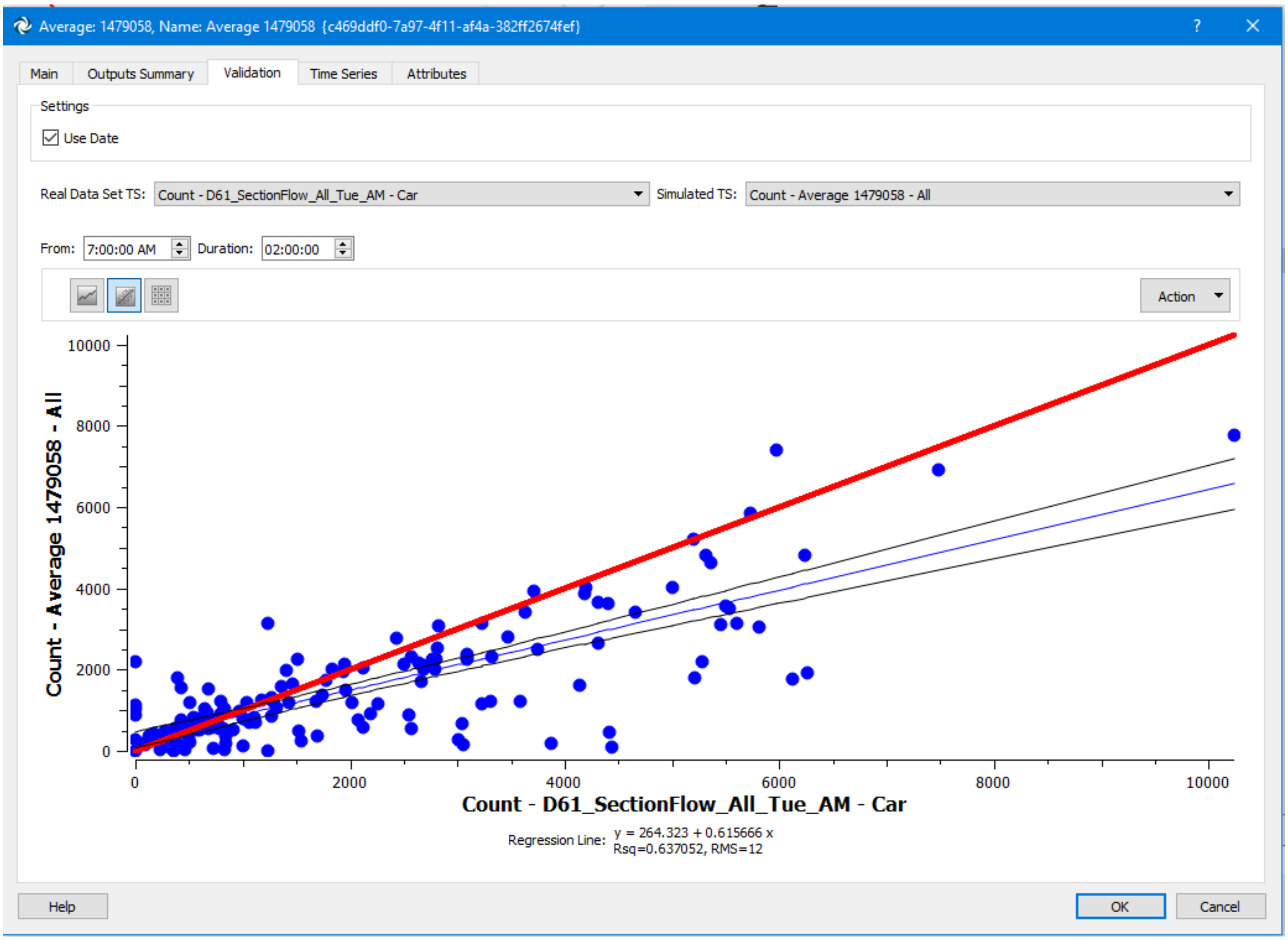


Compared to the Fundamental Diagram of the Meso DUE simulation (section 5.1), the diagram shown above indicates severer congestion (more points with Density>100 and Speed<50), which is more reasonable, given the fact that the simulation is on a weekday from 7 to 9 AM.

Default case with incident (TEST ID 9999)

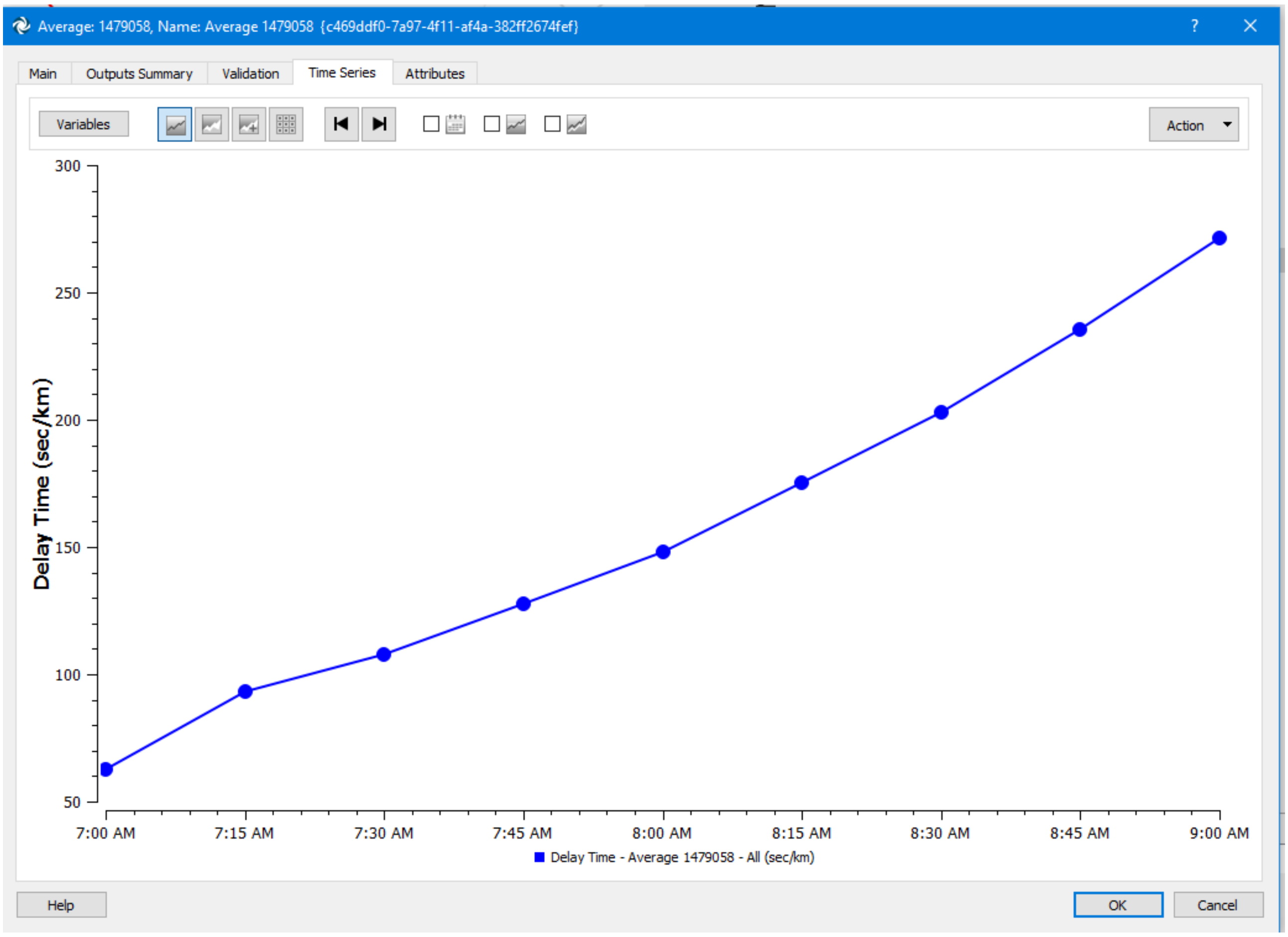


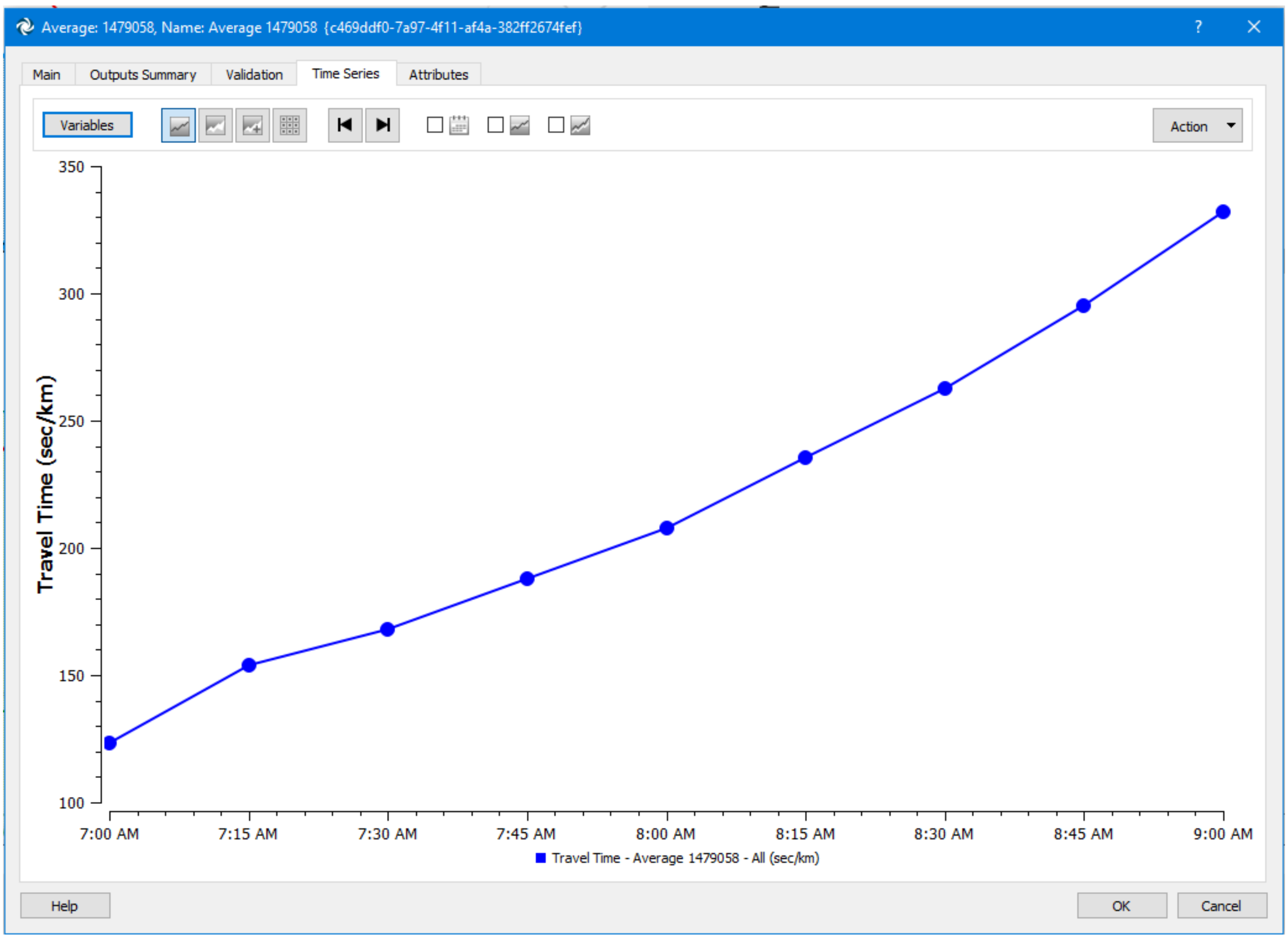
Validation

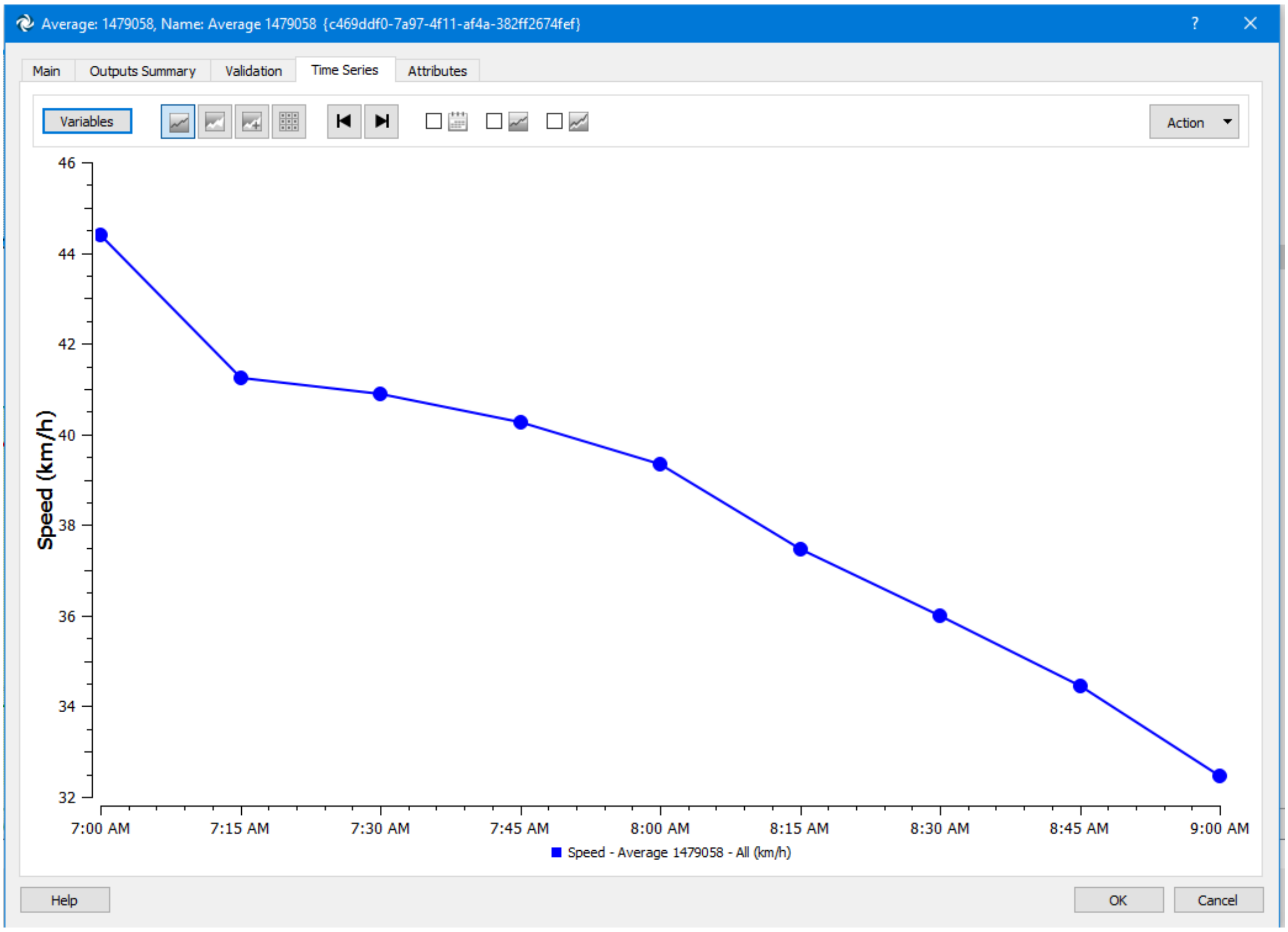


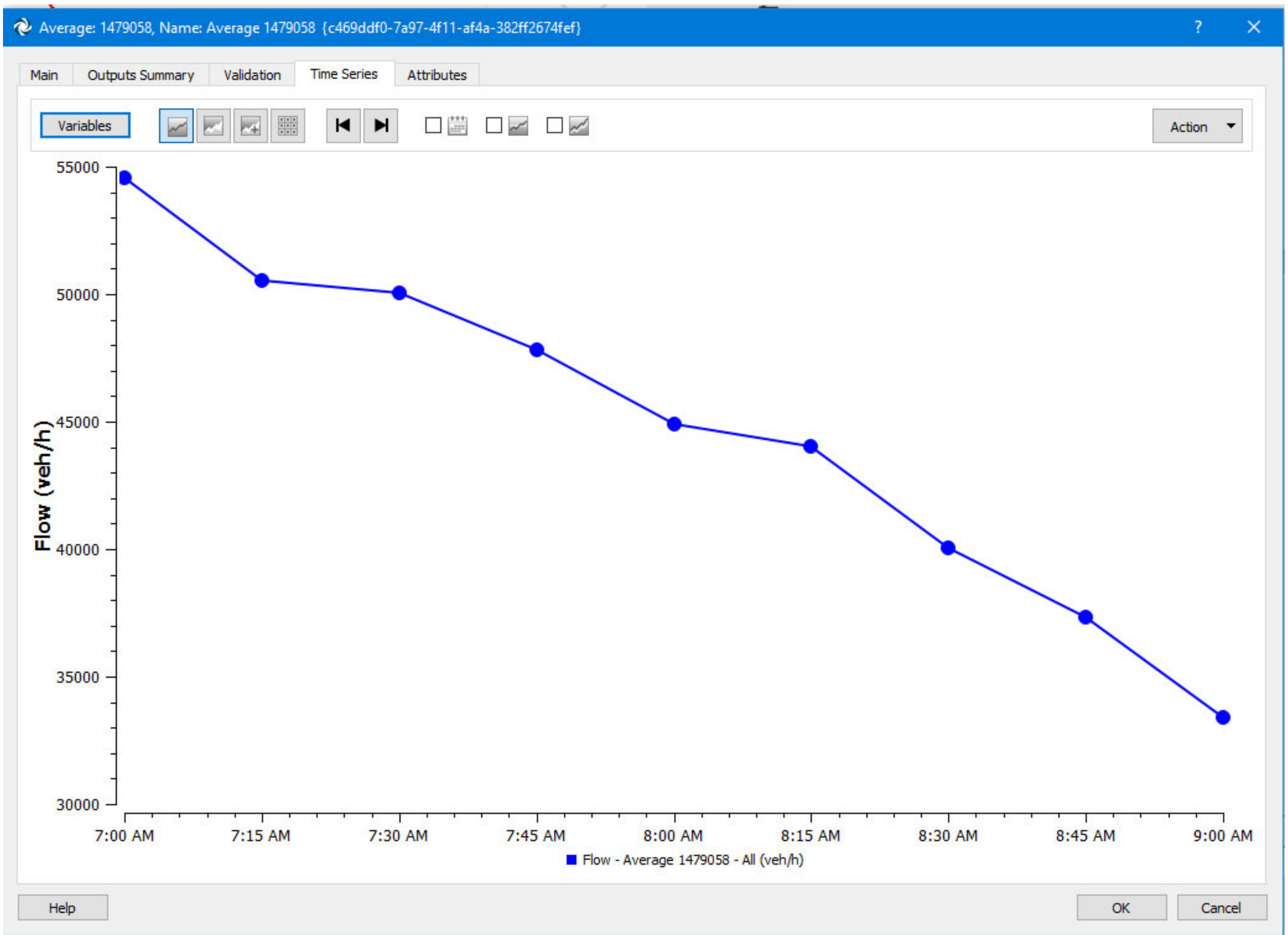
Compared to the Result without Incident, the R2 is slightly larger. This is thought to be caused by randomness and the fact that the original simulation does not fully reflect the actual traffic condition (which ought to have a higher congestion).

Time Series parameters



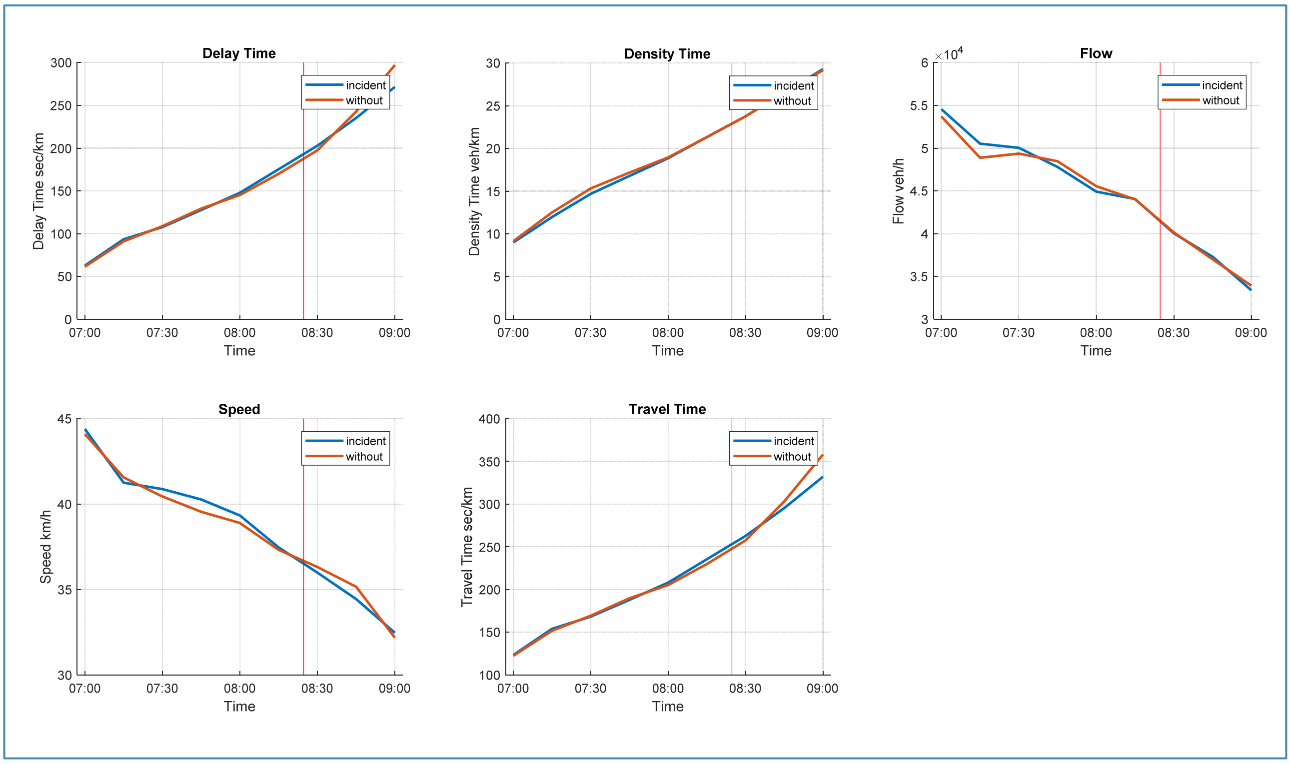



Average: 1479058, Name: Average 1479058 {c469ddf0-7a97-4f11-af4a-382ff2674fef}
Main
Outputs Summary
Validation
Time Series
Attributes
Variables
Action
Speed (km/h)
46
44
42
40
38
36
34
32
7:00 AM
7:15 AM
7:30 AM
7:45 AM
8:00 AM
8:15 AM
8:30 AM
8:45 AM
9:00 AM
Speed - Average 1479058 - All (km/h)
Help
OK
Cancel


Average: 1479058, Name: Average 1479058 {c469ddf0-7a97-4f11-af4a-382ff2674fef}
Main
Outputs Summary
Validation
Time Series
Attributes
Variables
Action
Flow (veh/h)
55000
50000
45000
40000
35000
30000
7:00 AM
7:15 AM
7:30 AM
7:45 AM
8:00 AM
8:15 AM
8:30 AM
8:45 AM
9:00 AM
Flow - Average 1479058 - All (veh/h)
Help
OK
Cancel

Comparison between with/without incident (duration 3 min)

The time-series comparison was plotted by MATLAB.

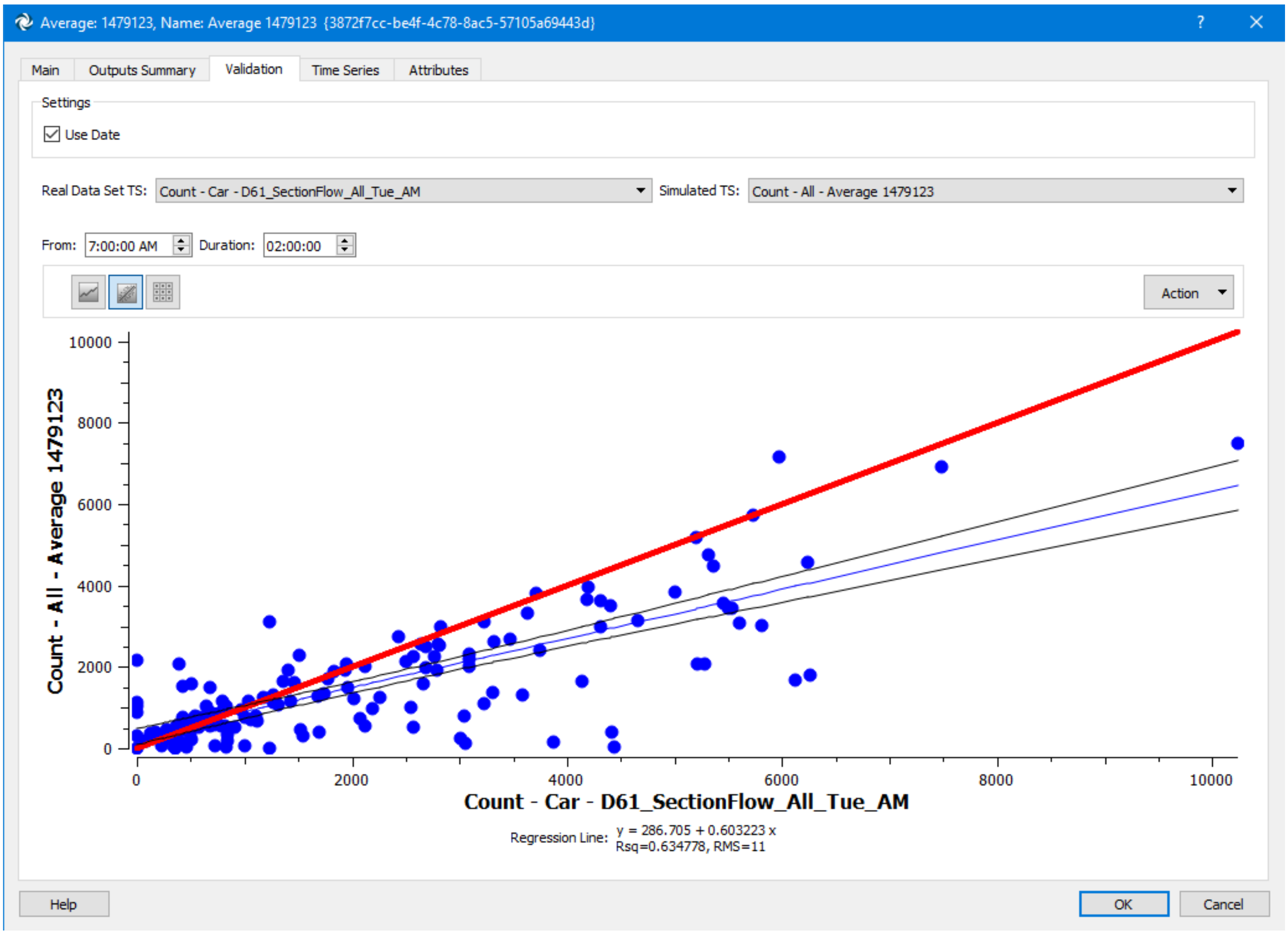


The difference in the time-series parameters is minimal. Several reasons may account for the outcome. First, the randomness of the microscopic simulation may affect the traffic behaviour, as a proof, the trajectories of the plot are different even from the start of the simulation. Also, the impact of a single incident on the entire subnetwork can be small, plus the duration of this incident is only 3 minutes.

To magnify the impact of the incident, the author manually extended the duration for another 10 minutes and performed the following simulation.

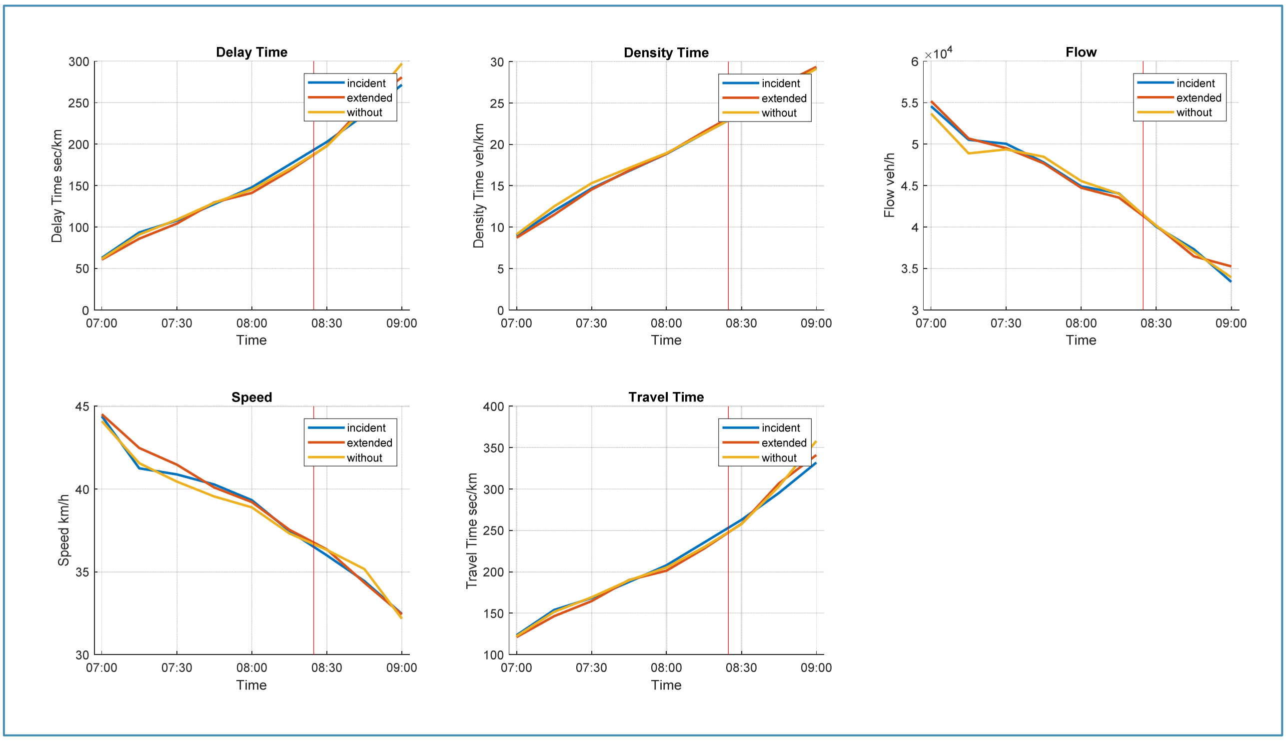
Default setting, with incident - duration manually extended for 10 min

Validation



The Validation plot is similar to what shown beforehand.

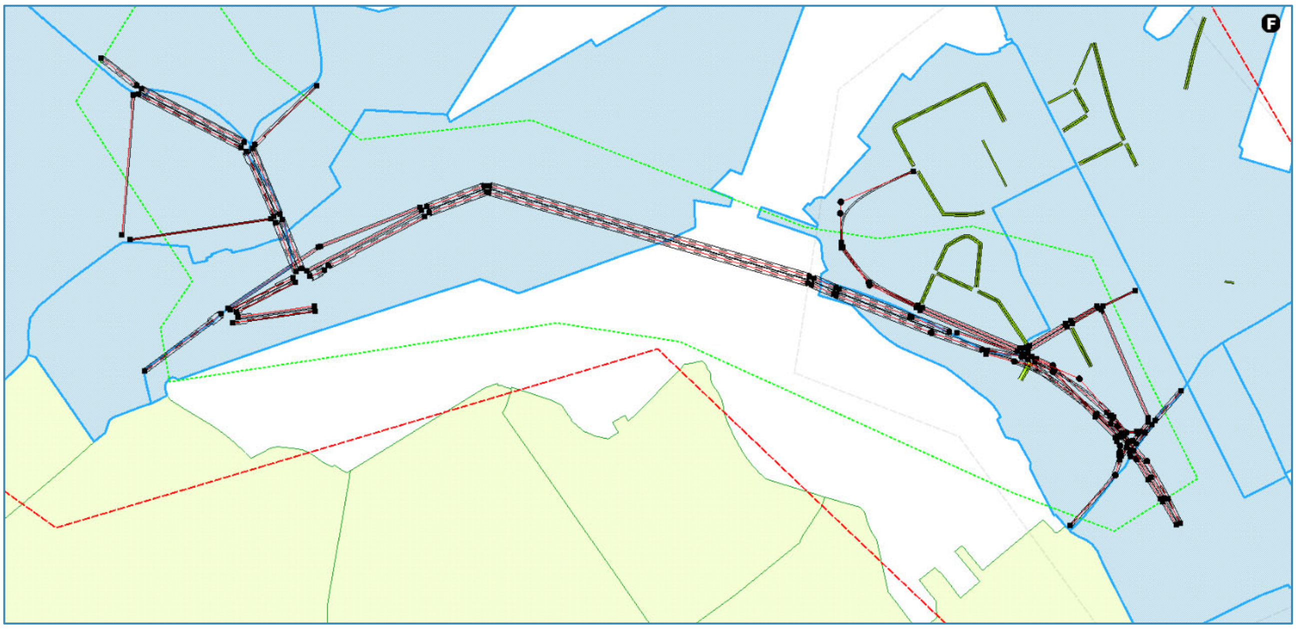
Comparison among the three cases



Overall, there is little difference. In the Speed and Density plots, the results are almost identical. The Delay Time and Travel Time plots show that extending the duration of the incident causes further congestion, which is as expected. However, in the Flow diagram, the extended duration has a surprisingly positive effect on the traffic condition, the reason is thought to be caused by the randomness in the stochastic process (even though these are the Average results of 10 Replications, the simulation outcome is still subjected to some level of randomness).

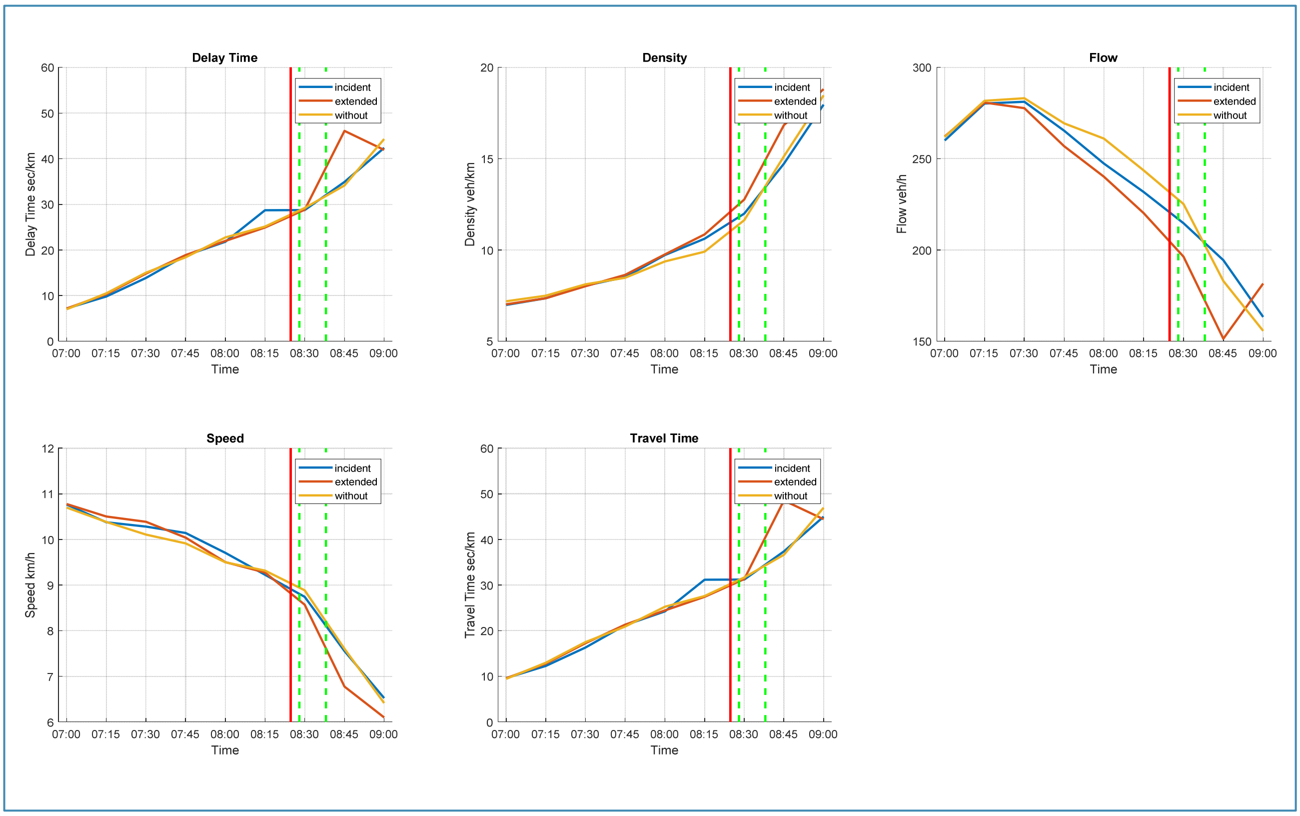
Comparison within the selected Impact Area

The main reason for such small difference in the Time Series plots is thought to be that an individual incident has limited influence on the entire network. Hence, an extra comparison was performed for sections that are close to the incident location, as shown below.



Note that the incident happened on the bridge, and the Time Series data of the selected sections are collected to produce the average value.

The resultant plots are as follows.



As expected, the difference in the time-series parameters is more significant if we only concern the area close to the incident. Specifically, the simulation result of the extended incident shows a reasonable increase in the level of congestion. This can be observed from the trajectories of all the five parameters. However, there is an unexpected recovery in the density, flow and the travel time at the end of the simulation (after the incident has been resolved).

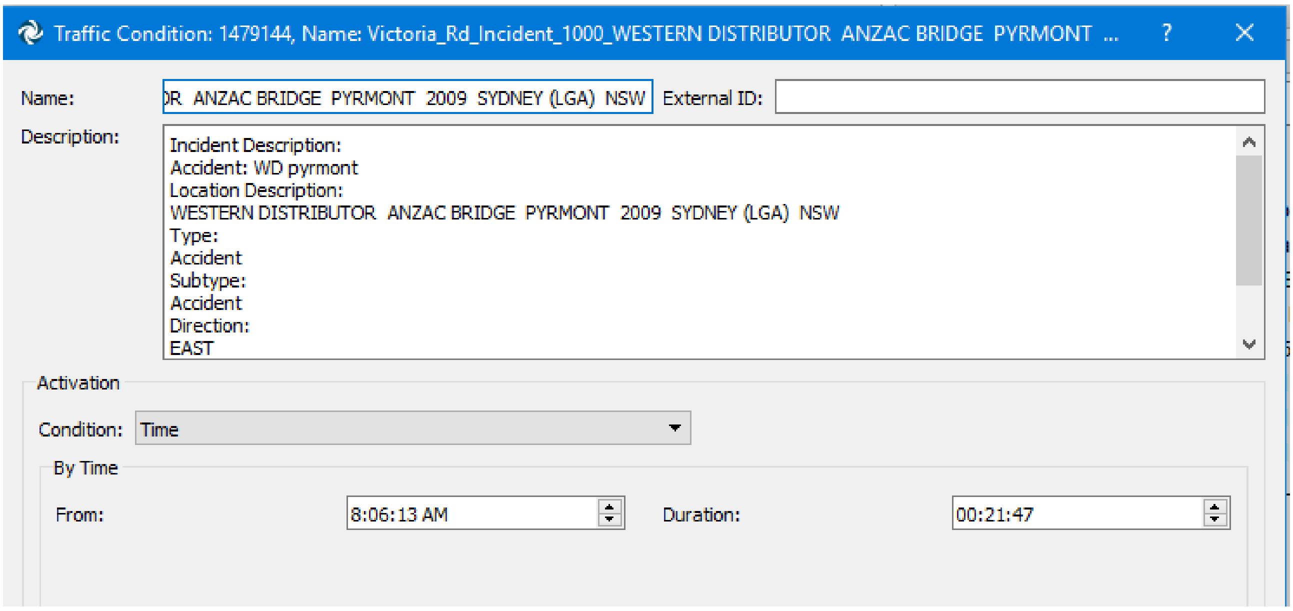
The result of the simulation with the incident of the original 3-minute duration is overall similar to that of the simulation without the incident, except that the flow and the density trajectories have minor differences. By comparing the limited impact caused by the original incident with the *clearly* severer impact of the extended incident, the author concluded that the duration of the incident is indeed a critical factor to be considered.

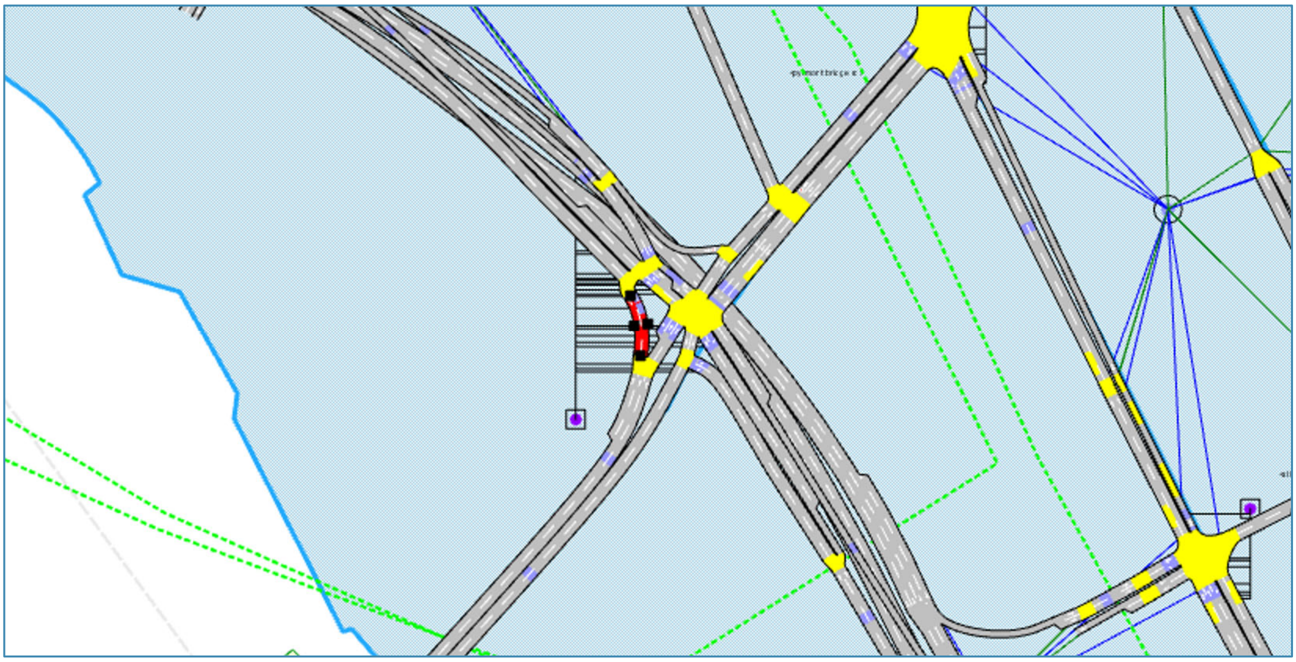
5.3. Microscopic Simulation with 08/03/2017 Incidents (ID = 997, 1000, 1004)

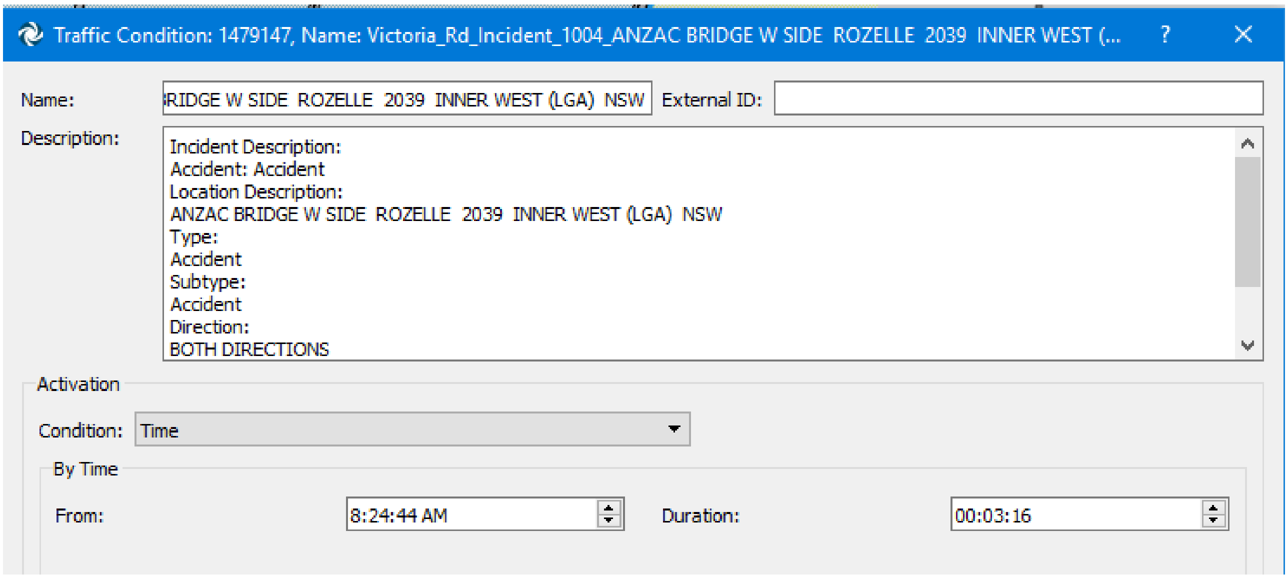
Aside from the TEST fake incident, the author also used three real incidents (dated 08/03/2017) to investigate the traffic behaviour under multiple impacts. Details are as follows.

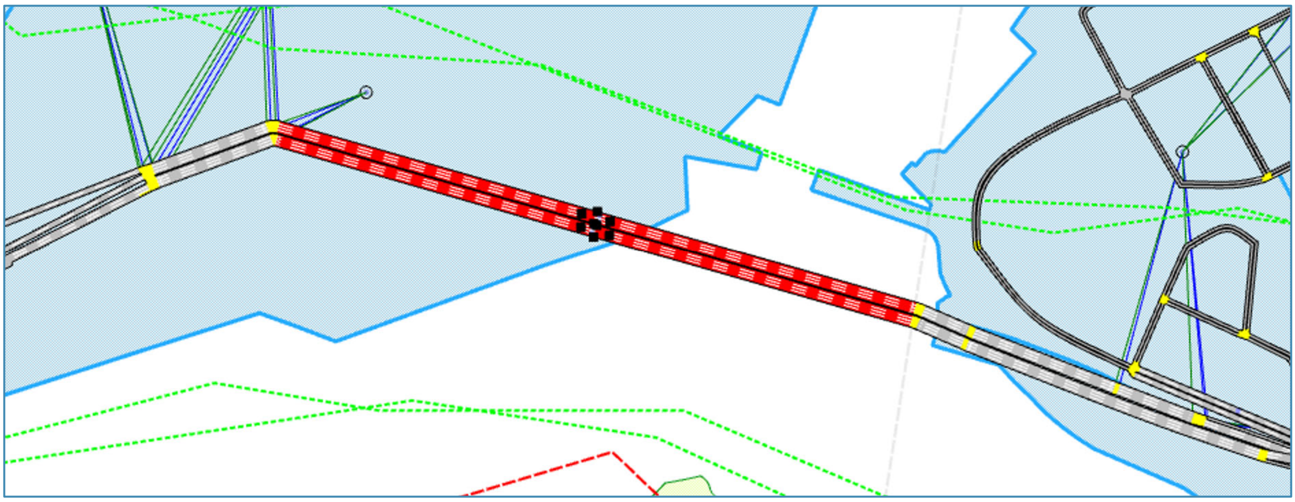
In the following simulations, the author used the default settings mentioned hereinbefore, for the validation results under the default settings, please see section 5.2. The Stats are recorded at 5-min interval.

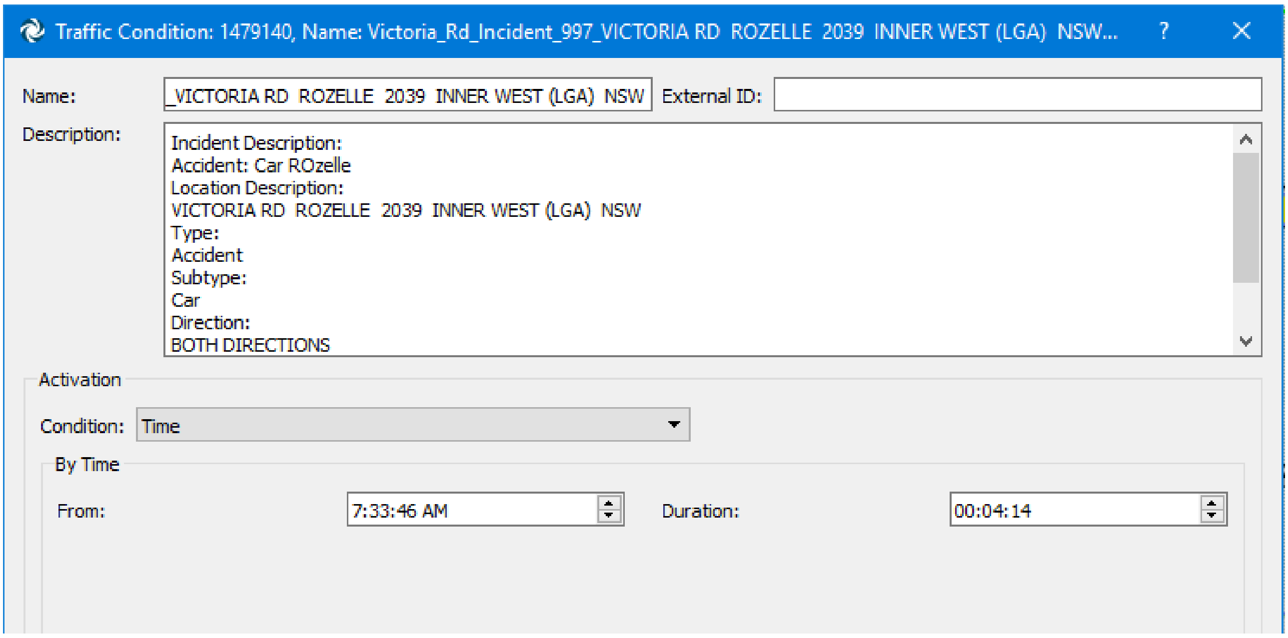
To better illustrate the difference in the simulations, the author chose to simulate the Replications using the same Random Seed. In this way, the behaviour of the simulation will be identical under the same settings. Thus only the incident will induce changes in the time-series trajectories.

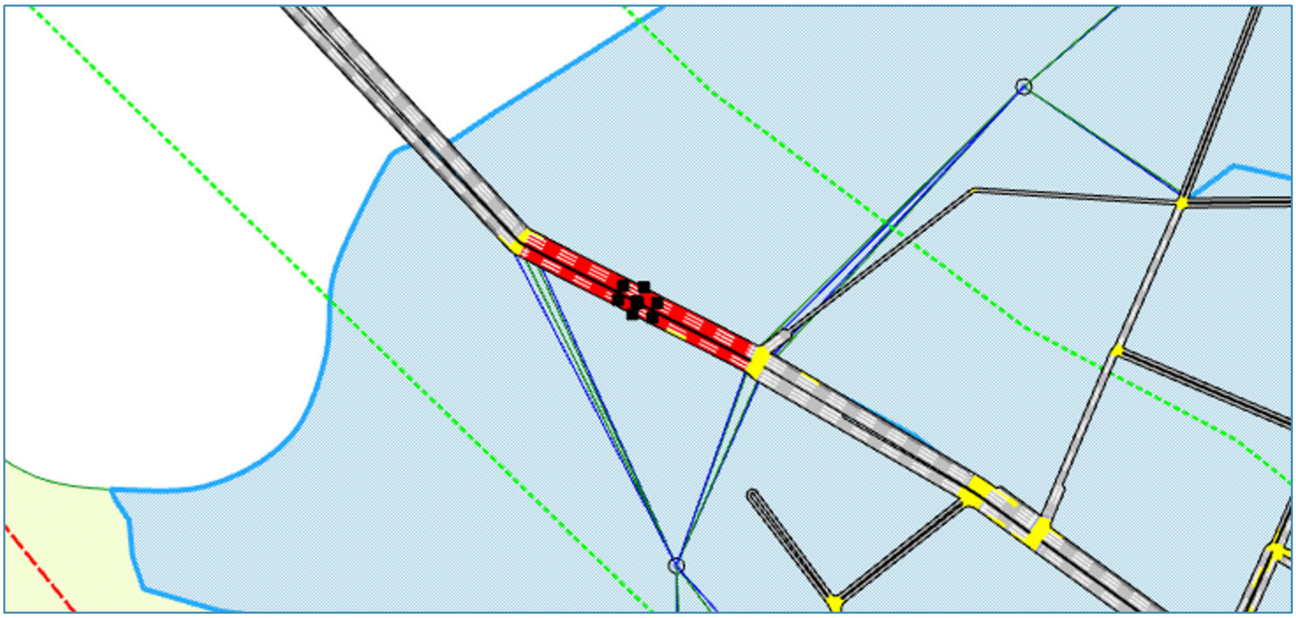

Traffic Condition: 1479144, Name: Victoria_Rd_Incident_1000_WESTERN DISTRIBUTOR ANZAC BRIDGE PYRMONT ...
Name:
)R ANZAC BRIDGE PYRMONT 2009 SYDNEY (LGA) NSW
External ID:
Description:
Incident Description:
Accident: WD pyrmont
Location Description:
WESTERN DISTRIBUTOR ANZAC BRIDGE PYRMONT 2009 SYDNEY (LGA) NSW
Type:
Accident
Subtype:
Accident
Direction:
EAST
Activation
Condition:
Time
By Time
From:
8:06:13 AM
Duration:
00:21:47




Traffic Condition: 1479147, Name: Victoria_Rd_Incident_1004_ANZAC BRIDGE W SIDE ROZELLE 2039 INNER WEST (...
Name:
RIDGE W SIDE ROZELLE 2039 INNER WEST (LGA) NSW
External ID:
Description:
Incident Description:
Accident: Accident
Location Description:
ANZAC BRIDGE W SIDE ROZELLE 2039 INNER WEST (LGA) NSW
Type:
Accident
Subtype:
Accident
Direction:
BOTH DIRECTIONS
Activation
Condition:
Time
By Time
From:
8:24:44 AM
Duration:
00:03:16





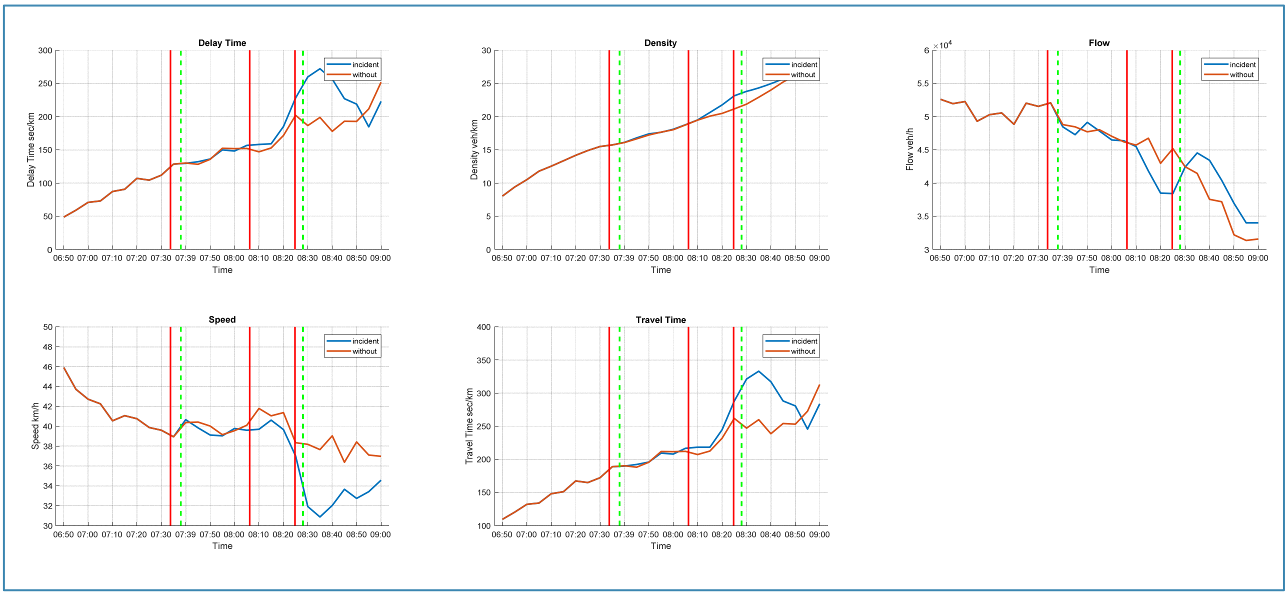


Comparison between with/without incidents

The trajectories were plotted using MATLAB.

Comparison with the entire subnetwork

The author firstly compared the time-series results for the entire subnetwork (similar to what was described in section *5.2 0 Comparison among the three cases*).



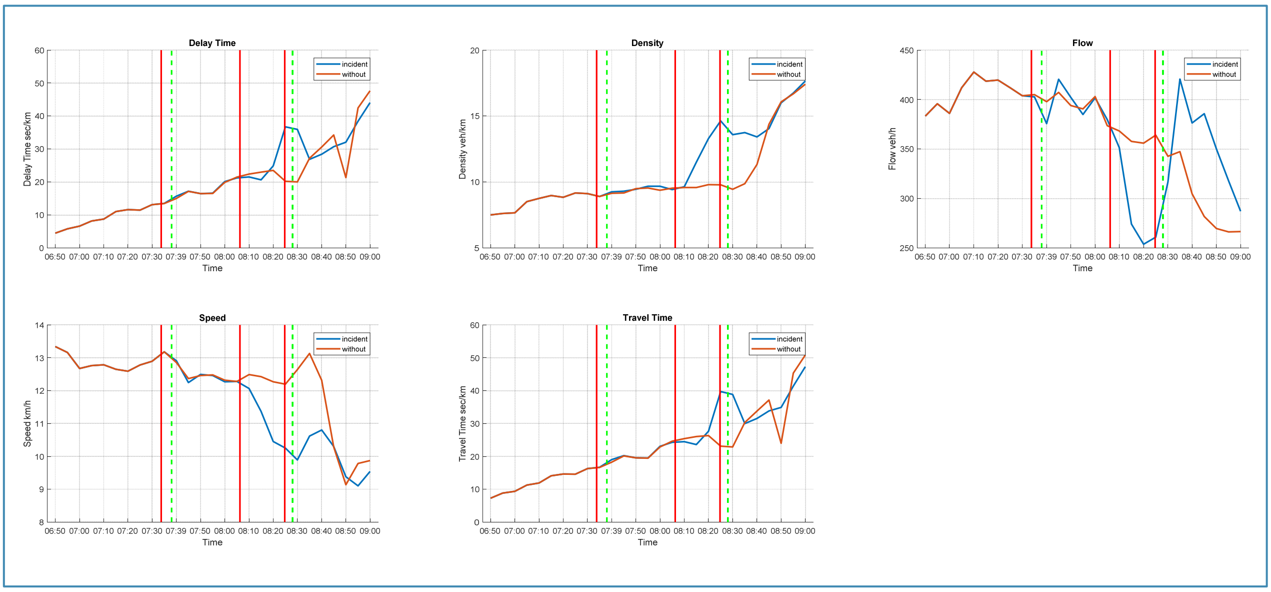
From the plots, one can observe that the first incident (starting from 7:33, lasting for 4 minutes) has a small impact on the entire network. This can be seen from the Delay Time and the Density. Although the Speed, Travel Time and Flow are affected by the event, the trajectories were able to recover within half an hour (before the start of the next incident). This is thought to be that the traffic demand at 7:30 is not high enough to be heavily affected by a single incident.

The subsequent two incidents, however, has a serious impact on the time-series parameters for the subnetwork. Specifically, the Travel Time, Delay Time, Flow and the Speed were all heavily under the influence. Two main reasons can account for this outcome. First, the two incidents occurred between 8:00 and 8:30, at which the traffic demand is already high and congestions have already occurred. Second, the incidents took place at Anzac bridge to the west of Pyrmont, which is a critical pathway for vehicles entering the city. Overall, the simulation result is within expectation. At around 8:40, the time-series parameters show signs of recovery, indicating that the traffic started recovering.

It should be noted that at the end of the simulation, one can observe an unexpected recovery in the Delay Time, Flow and Travel Time, which yield better results compared to the normal simulation without incidents. The reason is unclear, however the author believes that the situation might be caused by the fact that the blocked traffic flow caused by the incident can have a positive effect on reducing existing congestions around the section.

Comparison within the selected Impact Area

Similar to what was described in section *5.2 0 Comparison within the selected Impact Area,* the author also conducted a comparison for a selected area that is near the incidents (sections that are supposed to be influenced heavily).



Compared with the previous trajectories for the entire subnetwork, the trajectories in the plot for Impacted Area have identical behaviours. The only difference is that the trends become more significant, which is as expected.

6. Part IV – Automatic incident simulation

One of the main objectives of this project is to investigate the impact of incidents. To increase the efficiency in the simulation process, the author composed a series of Aimsun Python scripts that could perform automatic incident data importing, simulation and results exporting. Details of the scripts are elaborated on as follows.

6.1. Importing, creating and simulating traffic incidents

The script aims to automatically generate the corresponding incident in the Aimsun model based on CMCS Incident Dataset. After the creation, the script will create and simulate the Microscopic SRC Experiment and illustrate the effect of the incident.

The key aspect is to create the appropriate Aimsun Object and assign the correct attribute values to represent the incident.

For the structure of the program, please refer to Figure 7.

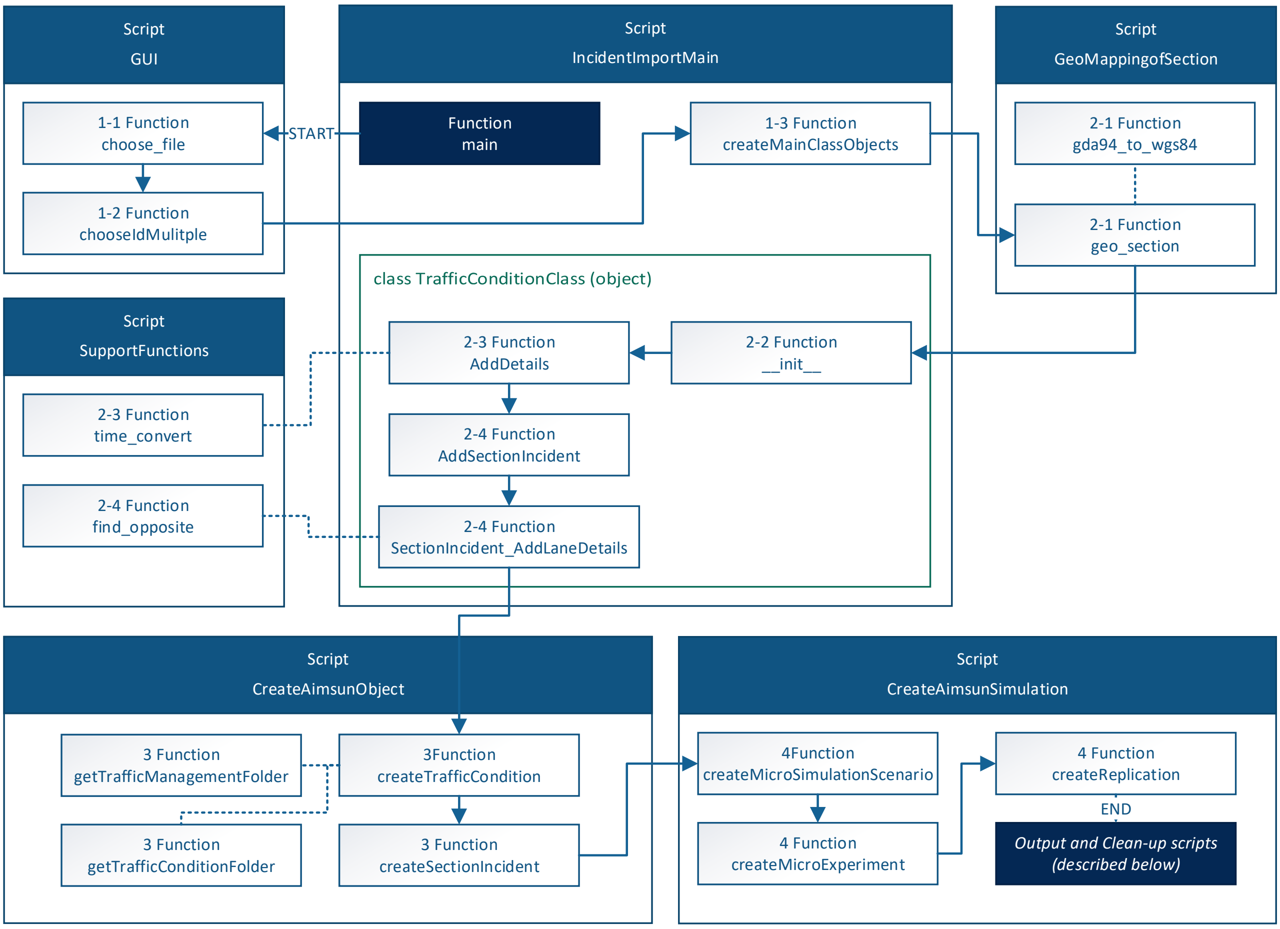


Figure 7 IncidentImportMain - Script and function diagram

Note:

- The main script *IncidentImportMain* imports and calls all the other scripts and functions.
- The Numbering indicates the corresponding Step(s) elaborated on as follows in which the function was called.

Once executed, the script will call the *main* function in the *IncidentImportMain* script. The detailed steps are as follows.

Step 1 Identify the dataset and obtain the incident IDs

Step 1-1 Choose dataset

Call the *choose_file* function and require the user to choose a dataset file, which can be either a csv file or a Microsoft Excel spreadsheet.

- Output: the file directory string, e.g. 'C:\\test.xlsx'

Step 1-2 Choose CMCS incidents based on ID

Call the *chooseIdMultiple* function, which asks the user to input (multiple) incident IDs that are used for the simulation.

- Output: a list of incident IDs in strings, e.g. ['669', '1000', '1004']
- The incident IDs are supposed to be determined from the dataset. Normally, it should be a column of *unique* numbers distinguishing each incident- for the current application; the column is named "*id_unique*", which was created by the author as there were no unique IDs in the original CMCS database.

Step 1-3 Prepare for creating instances in the custom class for all the selected *incidents*

- Call the *CreateMainClassObjects* function
- Input: the dataset file directory, the list of incident IDs in strings from *chooseMultiple* function
- Output: a list of custom class instances created for the incidents
- The function first reads in the dataset, then loops over the input incident ID list and extract the corresponding rows in the dataset. Then, for each incident event, it will call the corresponding functions within the custom class to execute the entire procedure described in Step 2.

Step 2 – Create a list of instances in the custom class to hold the necessary information

For each CMCS incident in the incident list, perform the following operations

Step 2-1 Geo mapping of reported CMCS incidents to the simulation network

- Main function to call: *geo_section*
- Input: CMCS incident data (latitude, longitude), spatial file (*.shp) generated from the Aimsun network
- Output: SCATS section name associated with the incident (the closest one), the name of the three closest sections in a list, and the closest node to the incident and all sections passing through the node (in case the incident affects the intersection)
- Method: using the Python module *shapely*, find the nearest section (a geometric line in the spatial file) to the location of the incident (the longitude/latitude point)
- Limitations:
  - The reported location is not always accurate, which can result in misidentifying in sections
  - There is no detailed data such as affected lane(s) or exact length and position of the incident.
- Perspective:
  - Improve the accuracy of the incident mapping by using Natural Language Processing (NLP) to read the incident location description
    Challenge for NLP:
    - The section names in the simulation model are mostly SCATS names, which is different to the official road names.
      Might be able to solve it by associating the sections with an additional layer of OSM shape file.
    - Also, CMCS operators might report the location descriptions in different styles, example: PYRMONT BRIDGE RD E BND VIC OF PYRMONT ST 137 PYRMONT 2009 SYDNEY (LGA) N.

- ❖ Create GUI for CMCS operators to report incidents
  Unify the format for reporting incidents, so that NLP can identify the location correctly. Example: require the operator to express an intersection incident as "[Street 1], [Street 2]" to indicate that the incident affects all the street connected to the node

Step 2-2 create an instance of the custom Python class

- Call the *__init__* function within the custom class
- Create and initialise the attributes within the instance; the attributes are set to *None* (placeholder)

Step 2-3 Initialise an instance of the custom Python class

- Call the *AddDetails* function within the custom class
- Input: the instance itself that was created in Step 2-2, the dataset row
- Output: void
- Class attributes include: name, LocationDescription, X, Y, StartTime, EndTime, IncidentDescription, Type, SubType, Direction, Lane, id, IncidentComponent
- Most attributes can be directly obtained from the dataset
- The IncidentComponent is a list holding all the necessary information for creating Aimsun traffic incident components such as *Section Incidents*. Since one traffic condition can have multiple incident components (e.g. an incident can result in three section incidents), they are grouped as a list.

Step 2-4 Add Traffic Incident component information to self.IncidentComponent in the instance

- Currently, only one type of incident is supported by the script, which is "*Accident*". The traffic incident component associated with accidents is *Section Incident*.
- First check if the instance Type is *Accident*, if it is:
- Call the *AddSectionIncident* function within the custom class
- Input: the instance itself
- Output: self.IncidentComponent
- The following attributes are required to create a *Section Incident* object in Aimsun: *SectionName*, *FromLane*, *ToLane*, *Position*, *Length*. Note that currently, we are not applying any control actions to the incident, hence ignoring *SpeedReduction* and *VMS* related attributes.
- The sectionName is obtained from Step 2-1. However, additional sections might be added depends on the *Direction* data associated with the incident. Specifically, if *BOTH DIRECTION* or *EAST AND WEST* is recorded, then the opposite direction section should also be affected.
  - ❖ To find the opposite section name, call the *find_opposite* function
  - ❖ Input: the current direction SCATS section name, e.g. "N123_N456"
  - ❖ Output: the opposite direction SCATS section name, e.g. "N456_N123"
  - ❖ Ideally, the opposite direction section should exist in the model. In the case where the script cannot find the opposite section object in Aimsun, it will output a NONE (which is later discarded).
- Meanwhile, if the *Direction* is recorded as *ANY DIRECTION*, it indicates that the section is supposed to occur at a node, and all sections connected to that node should be affected.
  - ❖ Query the model to find the node object using the node name returned from *geo_section*
  - ❖ Find all the sections connected to the node (ExitSections and EntranceSections)
  - ❖ Return a list of section names
  - ❖ Challenge:
    - o The Aimsun model can have multiple sections with the same *Name*, usually because they are of the same street but were cut into several segments, for example: the N10601_N12028 section.
    - o The Aimsun model's *Catalog().findByName()* method cannot find all the objects with the same name, instead it will only return the first object it finds with that name

- Perspective:
  - It has been pointed out that even if Aimsun select the wrong section segment, the incident can still be simulated- as long as the Section Incident can effectively block the traffic.
  - It may be better to create Turn Closures to represent an intersection incident, which is also easier for coding. However, the microscopic behaviours of section incidents and turn closures remain to be investigated. Specifically, the ability of the vehicles to perceive the changes in the traffic might be different.

- (UNSOLVED) If the Direction is recorded as *EAST, WEST, NORTH* or *SOUTH*, we need to find correct direction based on the geometric arrangement of the roads.
  - Potentially, we could identify the angle of the road segment and determine its direction, thus identify the correct section based on the description.
  - Challenge:
    - A road can have turnings and curves in the middle (e.g. the Victoria Road). Therefore, we cannot simply identify the direction of the entire section. However, in the spatial (*.shp) file, roads are represented as continuous LineString objects which contain no segment information, and can only be treated as a whole. Hence, it is impossible to guarantee the correctness of the script for identifying the road direction.
  - Perspective:
    - (same as above) If a GUI can be created for the CMCS operators to allow them to report the incidents directly into our dataset, the issues of identifying sections, directions and lanes will no longer exist.
- The *FromLane* and *ToLane* attributes are obtained based on the *Affected Lane* data in the dataset (which has been recorded as self.Lane while initialising the instance). Since the dataset only records "the number of lanes affected", it is impossible to deduce which lane is under the influence. Hence, the following assumptions are adopted.
  - Call the *SectionIncident_AddLaneDetails* function within the custom class
  - Input: the instance itself
  - Output: *FromLane* and *ToLane* attributes
  - If the *Affected Lane* is recorded as *ONE LANE* or *TWO LANES* or *THREE LANES* or *FOUR LANES*, the script will include the corresponding lanes starting from the leftmost one;
  - If the total number of lanes of the Aimsun section object is smaller than what recorded in the *Affected Lane*, a warning will be prompted, and then the script will block all the sections. This could happen as the Aimsun model might have minor differences compared to real-life situation, and it is impossible to manually check every section and road regularly;
  - If the *Affected Lane* is recorded as *ALL LANES*, then all the lanes are affected;
  - For all the other conditions, such as *NULL* or *NO LANE*, the script will treat them as *ONE LANE.*
- The *Position* and *Length* of the section incident cannot be directly deduced from the dataset. Therefore assumptions are adopted.
  - If the section length is over 30 metres, the *Position* will be set as the middle of the section, and the *Length* will be 29 metres;
  - If the section is shorter than 30 metres, the section incident will affect the entire section;
  - The length of the incident was estimated using common sense.

Step 3 – Create Traffic Conditions in Aimsun for each incident

Use the information held in the instance of the custom class, call the *createTrafficCondition* function and then the *createSectionIncident* function to create the corresponding Aimsun objects in the model

Step 4 – Create Microscopic Simulations

After the incident has been created in the Aimsun model, call the *createMicroSimulationScenario* function, the *createMicroExperiment* and the *createReplication* function to create the Microscopic SRC Experiment.

Step 5 – Simulate

Simulate the experiment.

6.2. Cleaning up of the intermediate objects

The clean-up script aims to delete the intermediate elements created during the simulation process, including the Traffic Condition Objects and all the Incident Components (Section Incident etc.). It will also attempt to deactivate all the traffic conditions that were activated for simulating the incident in the Microscopic Scenario.

The reasons for performing this action are as follows:

- The intermediate objects are no longer useful after we obtain and export the result of the simulation, and to reuse the Microscopic Scenario, we should restore its settings
- Creating and accumulating a large number of objects might slow down the software, or even damage the Aimsun model
- Even if we want to redo the simulation, we can simply recreate the corresponding traffic conditions at a minimum cost

Per the Aimsun scripting document, the execution of the simulation(s) must be the *last call* within the Python script. This is because the Python script will not wait for the completion of the simulation to continue executing the lines of code below. In essence, if the simulation command is *not* the last call, every action that is supposed to take place after finishing the simulation will mostly like be performed right after the start of the simulator, hence disrupt the software. Therefore, the *Cleaning-up*, as well as the Output script, must be written into separate functions that are manually executed after the user obtain the simulation result.

The structure of the program is illustrated in Figure 8.

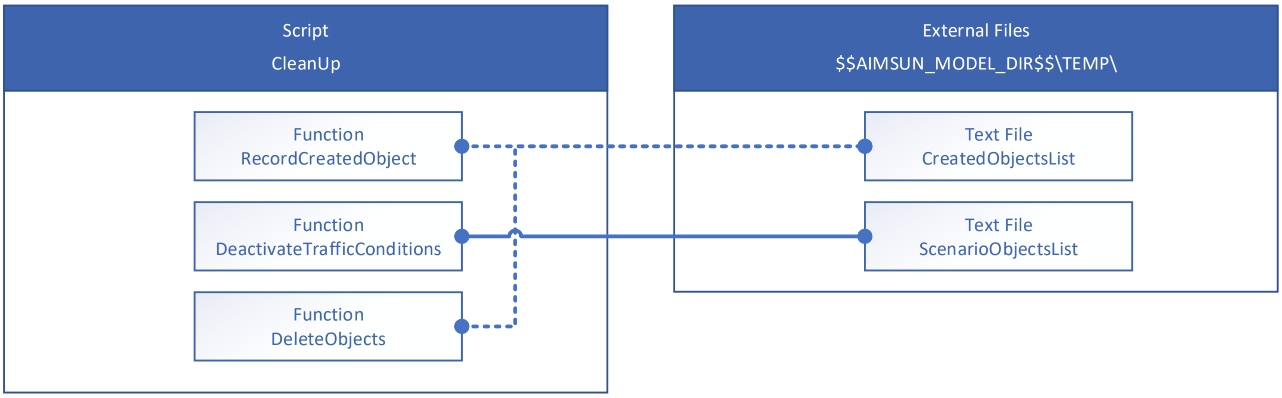


Figure 8 CleanUp Script - Modules and Functions Diagram

The detailed steps are as follows.

Step 1 – Preparation

While executing the scripts for creating and simulating Aimsun incidents, the function *RecordCreatedObject* will be called whenever an intermediate Aimsun Object is created- this includes all the Traffic Conditions and all the Incident Components (Section Incidents etc.).

- Function input: the ID (integer) of the created object;
- The function will append the ID of the newly created object into the local text file *CreatedObjectsList.txt*; this file will then be used when the user wants to delete these objects afterwards.

Step 2 – Delete Objects

After the simulation, if the user would like to clean up the intermediate objects, he/she should manually execute the *Cleanup* script, which will call the *DeleteObjects* function.

- Function input & output: None;
- The function will read in the text file created in Step 0 above as a list of object IDs;
- It will then loop through the list to attempt to identify the corresponding Aimsun Objects and delete them;
- The user can check the information of the deleted objects in the Aimsun Log window.

Note:

Currently, the *DeactivateTrafficConditions* function is not used. This is because once the corresponding Traffic Condition objects are deleted, Aimsun will automatically remove the entry from the *Strategies and Traffic Conditions* Tab in the Simulation Scenario. Therefore, there is no point in manually deactivating them.

6.3. Exporting time-series results

The Output scripts are responsible for exporting the results of the simulations, specifically the five *TimeSeries* data that are crucial for data analysis, including Delay Time, Density, Flow, Speed and Travel Time.

For the similar reason mentioned in the Cleaning-up script introduction in section 6.2, this script must also be separately executed after the Aimsun simulations.

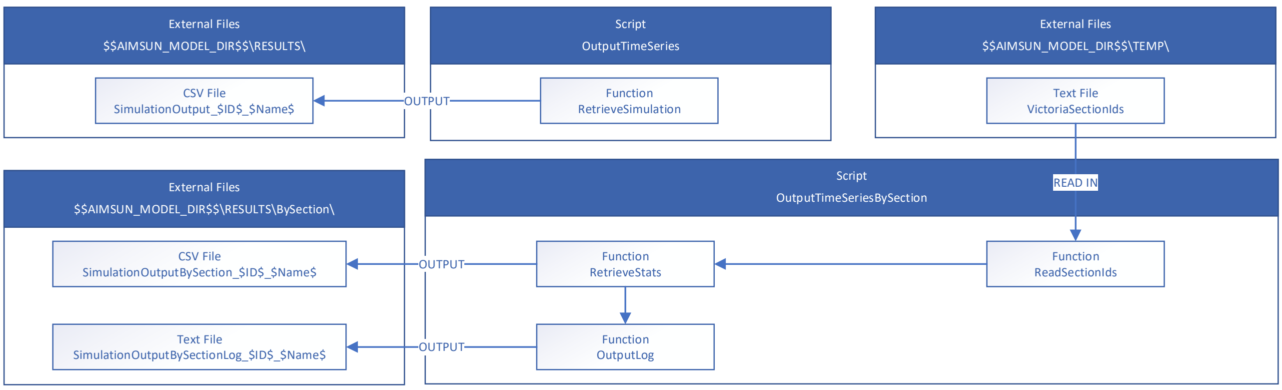


Figure 9 OutputTimeSeries scripts and functions diagram

Output Time-series aggregated results

This function outputs only the TimeSeries results stored in the Replication/Average/Incremental Result object. It does not contain detailed data for each section/segment.

When executing the *OutputTimeSeries* script from a drop-down menu of an Average/Replication/Incremental Result object, the function *RetrieveSimulation* will be called.

- Function Input: The target object (as the script is executed from an Aimsun object);
- The function will extract the corresponding columns within Aimsun and generate a CSV file in the directory *$$AIMSUN_MODEL_DIR$$\\RESULTS* containing the TimeSeries data. An example of such file is as follows;

	A	B	C	D	E	F	G	H
1		SectionId	Time	Delay Time (sec/km)	Density (veh/km)	Flow (veh/h)	Speed (km/h)	Travel Time (sec/km)
2								
3	0	1479034	70000	61.56	9.13	53687.2	44.1	122.27
4	1	1479034	71500	90.98	12.51	48872.8	41.56	151.69
5	2	1479034	73000	108.8	15.31	49355.47	40.45	169.3
6	3	1479034	74500	129.38	17.16	48464.53	39.55	189.67
7	4	1479034	80000	145.15	18.94	45544	38.89	205.19
8	5	1479034	81500	169.81	21.32	44010.13	37.32	229.75
9	6	1479034	83000	197.41	23.76	40150.93	36.32	257.5
10	7	1479034	84500	242.65	26.57	37003.2	35.16	303.17
11	8	1479034	90000	297.17	29.14	33920.8	32.17	358.02

Figure 10 An example TimeSeries output

Note:

- The first column is auto-generated by the Python module - Pandas DataFrame index;
- The *Time* column was recorded without any symbol (e.g. "7:15:00 as 71500"), so that it is more convenient to be processed in MATLAB.

Output Time-series results by section

This function outputs the detailed TimeSeries results for a Replication/Average/Incremental Result object. The exported data contains time-sliced data (time slice depends on the Interval setting) for each section that is within the selected Subnetwork (in this case, the Victoria Corridor).

Step 1 – Read in Section IDs

When executing the *OutputTimeSeriesBySection* script from a drop-down menu of an Average/Replication/Incremental Result object, the function *ReadSectionIds* will be invoked.

- This function will read in the list of all the section IDs within the Victoria Corridor Subnetwork stored in the file $$AIMSUN_MODEL_DIR$$\\TEMP\\VictoriaSectionIds.txt;
- The section list file is manually generated from the Aimsun by using the Table View. It is not part of the automatic process because such one-time procedure is extremely easy to perform in Aimsun.

Step 2 – Export Timeseries data for each section

After the function *ReadSectionIds* have identified all the sections involved in the subnetwork, it will loop over the section list and call the function *RetrieveStats* for each section.

- Function input: the Replication ID and the Section ID;
- Function output: the CSV file containing TimeSeries data stored in the directory: $$AIMSUN_MODEL_DIR$$\\RESULTS\\BySection\\
- This function will append the TimeSeries data of the selected section to the CSV file. The output is similar to that illustrated in Figure 10.

Step 3 – Output LogFile

In addition to the main CSV file, the script will also generate a text file containing some critical information of the data. The log file can be used to check the validity of the simulation results.

After the generation of the CSV file, the function *OutputLog* will be invoked.

- Function input: The Number of sections processed, the number of sections with invalid or empty (0 or -1) TimeSeries values, a list of the questionable sections (their ID)
- Function output: a text file containing the above information stored in the directory $$AIMSUN_MODEL_DIR$$\\RESULTS\\BySection\\
- An example output:

```
Number of Sections recorded:1994
Number of Sections with empty or invalid data:1102

Sections involved:
4328
4359
4662
4695
4744
4965
5152
```

Figure 11 An example output of the Log file

6.4. Additional information on Aimsun scrips

External Packages

The scripts utilise the following external Python packages that are not included in the default Python installation.

PyQt5*, pandas, geopandas, shapely, xlrd*****

In addition, please note that Aimsun (version 8.2.1 R49393 x64) requires the use of Python 2.7 64-bit.

---

* Although it is the recommended choice for Python 2.x, PyQt4 is not compatible with the current version of Aimsun.

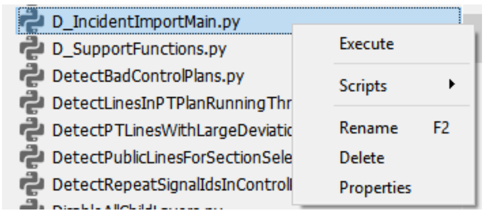
** The dependencies of geopandas include: numpy, fiona, six, pyproj, matplotlib, descartes and pysal.

*** xldr is optional, it is used only if the dataset is saved in *.xlsx spreadsheets.

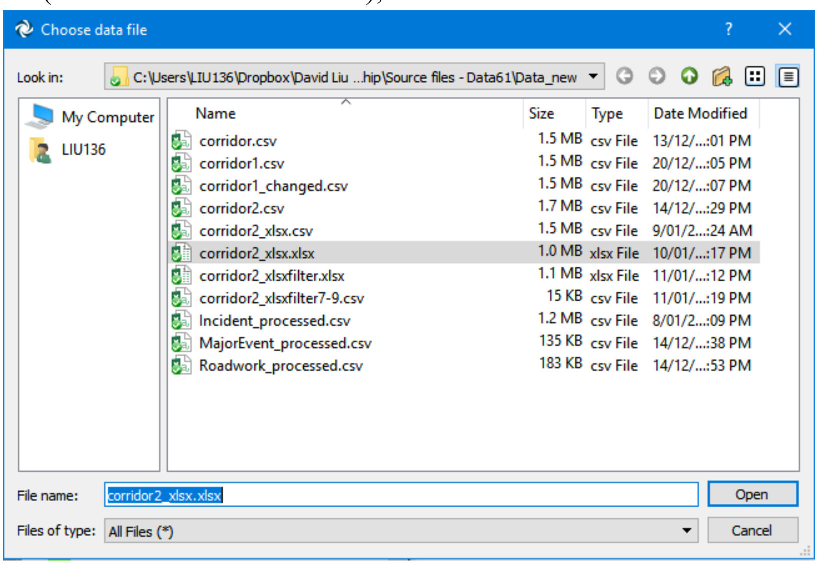
Execution Guide

Create Incident objects and Simulate

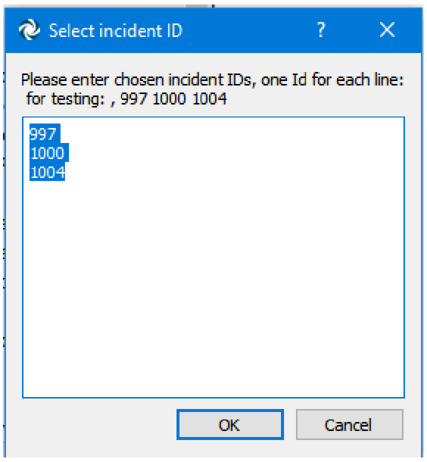
- Right-click on the main script to execute;



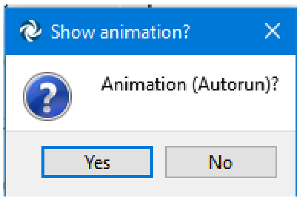
- Choose the dataset (the default is auto-selected);



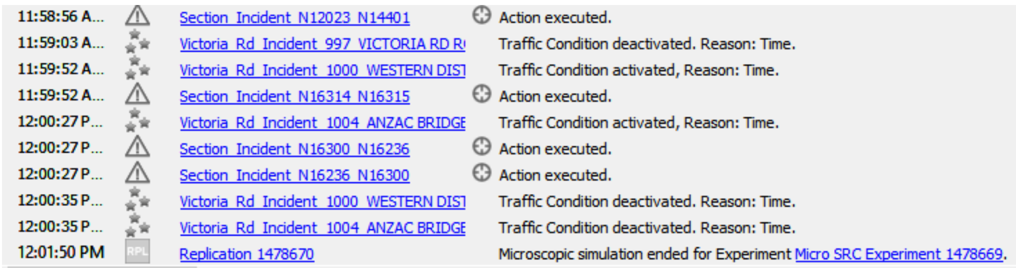
- Enter the CMCS Incident IDs to be created, right now only *Accident* is supported;



- Wait for Aimsun to generate the corresponding Traffic Condition and Section Incidents; after that, Aimsun will automatically create a Microscopic Experiment with the corresponding Traffic Conditions activated.
- Before beginning the simulation, choose whether to show Microscopic Animation;



- Wait for the simulation to end.
- Note that the selected Traffic Condition was automatically activated and deactivated during the simulation.



Output Timeseries results

- After the Simulation is complete, right click on the Aimsun object (can be either an Average/Replication/Incremental Result), select Scripts and then the corresponding Output script (either *OutputTimeSeries* for only the aggregated result, or *OutputTimeSeriesBySection* for every section);

7. Part V – Correlation analysis

As the last part of the project, the author attempted to investigate the correlation between the characteristics of the incident and potential contributory factors using data-driven approaches.

The problem that the author chose is about the duration of the incident. Specifically, the author would like to examine if there is any correlation between potential influential factors (e.g. weather, public holiday, the severity of the incident) and the duration of the incident.

7.1. Data Preparation

The aggregated SCATS flow + other factors

As mentioned hereinbefore in section 3.2, the available dataset at hand includes weather, public holiday, school holiday and events and LiveTraffic data (incident, major events and roadwork data). In addition, the author also has the SCATS flow data obtained beforehand for the calibration of the Aimsun model. However, since different sets of data do not cover the same period, only part of them can be used.

Since the CMCS incident dataset adopted for the simulation is for the year 2017, the author could only include other data for the same year. Specifically, this includes weather information, SCATS flow and public holiday dates. Also, the author was able to extract some characteristics regarding the incident events from the CMCS dataset, including the starting date, starting time, type and severity of the incident. As shown in Figure 12 and Figure 13 below.

	B	C	D	E	F	G	H	I	J	K
1	SubTypeNumbered	TotalFlowDuringIncident	temp_avg	rainfall	wind_9am	cloud_9am	hum_9am	isPublicHoliday	SEVERITY	DayOfWeek
2	4	3014656.255	22.55	2.6	19	7	66	0	4	2
3	4	4293033.557	23	4.4	2	3	61	0	8	6
4	4	6544661.347	25.6	0	9	2	57	0	8	1
5	1	3032912.278	26.95	0.4	9	7	73	0	8	2
6	4	2246891.246	30.85	0	9	4	53	0	1	3
7	4	2344685.973	25.35	0	7	7	68	0	1	4

Figure 12 Partial snapshot of the prepared dataset in the form of CSV file

	1 SubType	2 FlowCount	3 Ave_temp	4 Rainfall	5 wind9am	6 cloud9am	7 hum9am	8 isHoliday	9 DayofWeek	10 Severity
1	4	3.0147e+06	22.5500	2.6000	19	7	66	0	4	2
2	4	4.2930e+06	23	4.4000	2	3	61	0	8	6
3	4	6.5447e+06	25.6000	0	9	2	57	0	8	1
4	1	3.0329e+06	26.9500	0.4000	9	7	73	0	8	2
5	4	2.2469e+06	30.8500	0	9	4	53	0	1	3
6	4	2.3447e+06	25.3500	0	7	7	68	0	1	4
7	4	3.3059e+06	25.3500	0	7	7	68	0	8	4

Figure 13 Partial snapshot of the MATLAB array imported from the CSV dataset

Note that the author pre-processed the dataset to reflect the characteristics of the incident better. Details are listed as follows.

- The *SubType* (e.g. Car, Truck) are replaced by numbers for the same reason, as illustrated as follows;

```
if subtype == 'Accident':
    number_subtype = 1
elif subtype == 'Bicycle':
    number_subtype = 2
elif subtype == 'Bus':
    number_subtype = 3
elif subtype == 'Car':
    number_subtype = 4
```

```
elif subtype == 'Closure':
    number_subtype = 5
elif subtype == 'Motorcycle':
    number_subtype = 6
elif subtype == 'Multi-veh':
    number_subtype = 7
elif subtype == 'Pedestrian':
    number_subtype = 8
elif subtype == 'Truck':
    number_subtype = 9
```

- The *TotalFlowDuringAccident* represents the aggregated SCATS flow of the entire Victoria Corridor subnetwork during the period of the incident, which is generated by a Python script written by the author;
- The *isPublicHoliday* column contains Boolean values indicating whether the date of the incident is a public holiday or not;
- The *Severity* column is calculated based on the *Affected Direction* and the *Affected Lanes* data from the CMCS incident dataset, as illustrated as follows;

```
if affected_lanes == 'ONE LANE':
severity = 1
elif affected_lanes == 'TWO LANES':
severity = 2
elif affected_lanes == 'THREE LANES':
severity = 3
else:
severity = 4

if direction in ['BOTH DIRECTIONS', 'EAST AND WEST']:
severity *= 2
elif direction in ['EAST', 'WEST', 'NORTH', 'SOUTH']:
severity *= 1
else: # ALL DIRECTIONS, NODE INCIDENT
severity *= 4
```

Finally, the duration of the incident (for training) is recorded in the unit of minutes.

	A
1	ed duration
2	13
3	24
4	49
5	11
6	0
7	4
8	76
9	30
10	49
11	89

Figure 14 Partial snapshot of the duration of the incident in the dataset

The real-time SCATS flow + other factors

In addition to examining the aggregated SCATS flow value, the author also attempted to investigate the effect of the *real-time* SCATS flow data. To achieve that, the author created a copy of the above dataset, and replaced the aggregated SCATS flow column with the *real-time* SCATS flow data when the incidents occurred. Note that the SCATS data used in this case is still the summed value for the entire subnetwork.

Models used

For the investigation of the correlation between parameters, the author tried two different approaches in MATLAB- the decision tree (also called CART) and the Artificial Neural Network (ANN), details are described as follows.

7.2. The result of CART (decision tree)

By using the CART, the author hoped to find out the parameters that have the most significant influence on the duration of the incident.

The result of a 15-node result is shown in Figure 15 below. It is clear that the aggregated SCATS flow count is the parameter most related to the incident duration, which is as expected as a longer duration will induce a higher summed flow count. Aside from the SCATS data, other factors that have a noticeable influence on the incident duration include (ordered by significance) rainfall of the day, the average temperature, the severity and the subtype of the incident. However, due to the high number of node in the decision tree, the relationship between the factors and the incident duration is too complicated to summarise.

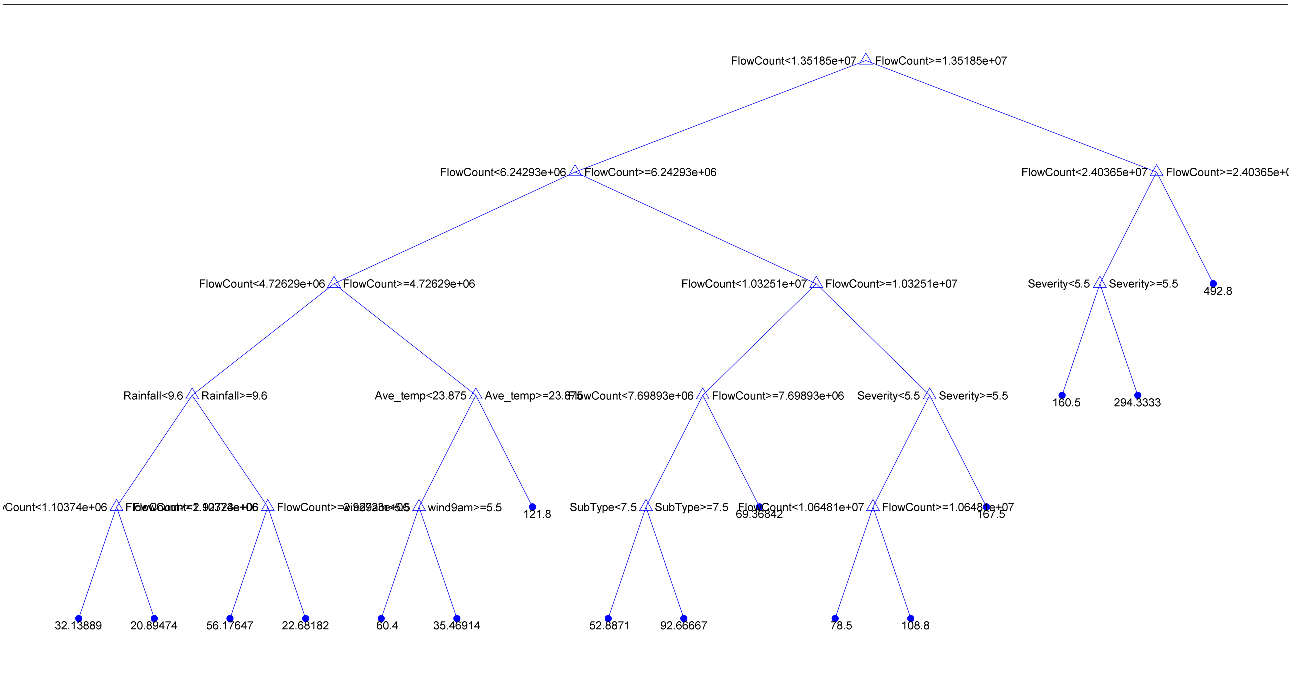


Figure 15 CART result (15 nodes) of the dataset

A simplified CART with 7 nodes was also generated, as shown in Figure 16. The decision tree reveals that the top three variables affecting the incident duration is the aggregated SCATS flow, severity and the average temperature. By examining the branches, the author could also conclude that the average temperature and the severity level are only influential when the aggregated SCATS flow is in the middle of the range (between 4.7×10^6 and 2.4×10^7 veh/h). In essence, when the sum of the SCATS flow is lower or higher than the bound, other parameters does not have significant effect on the incident duration.

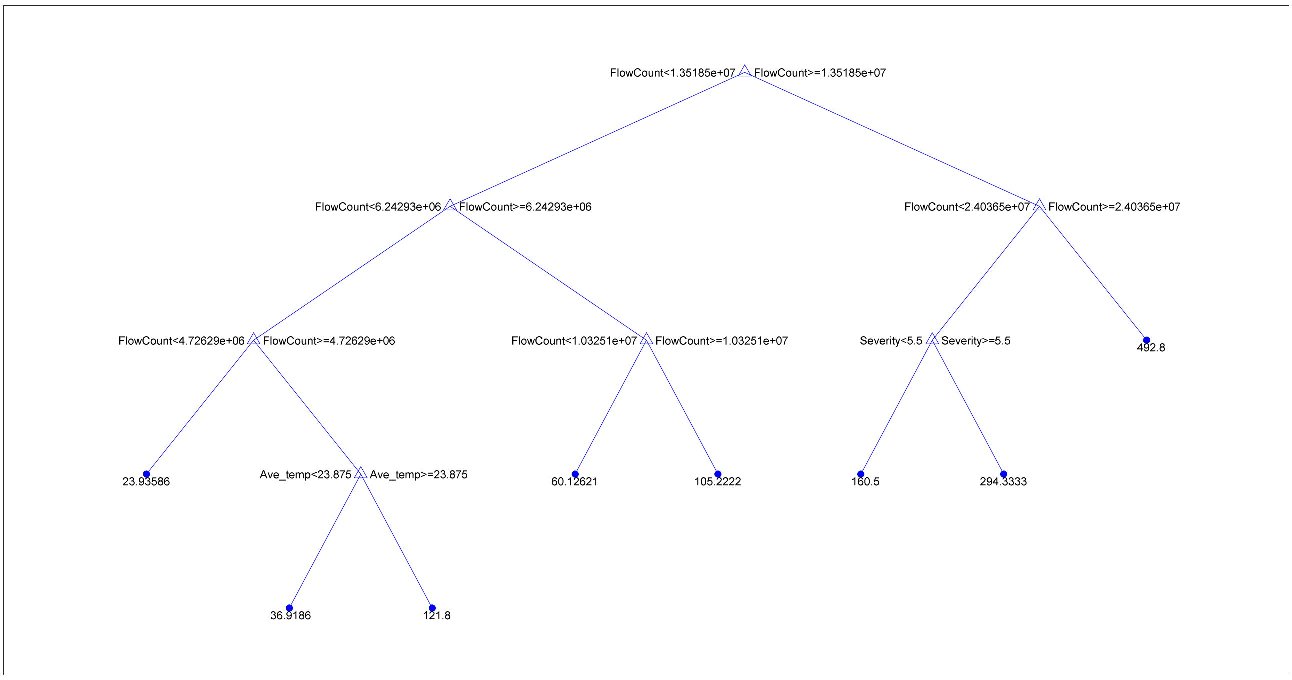
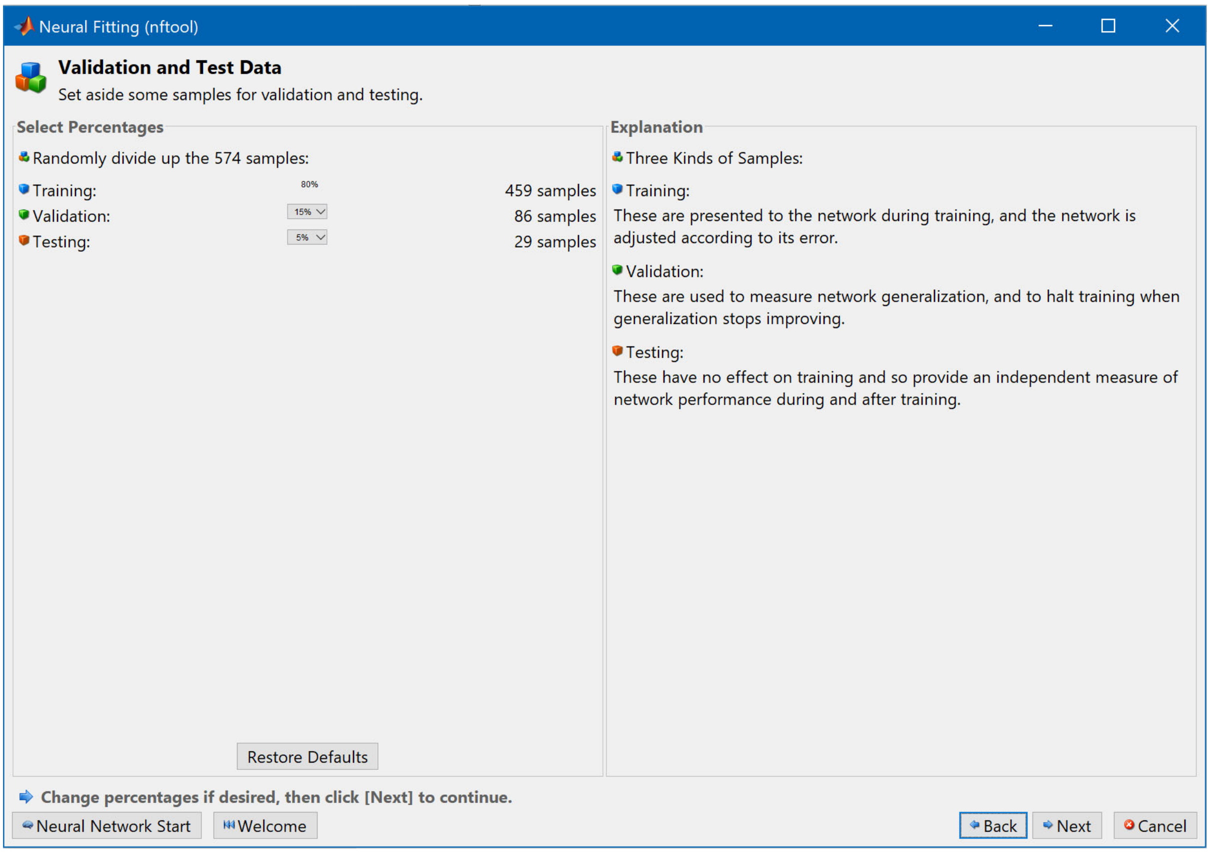


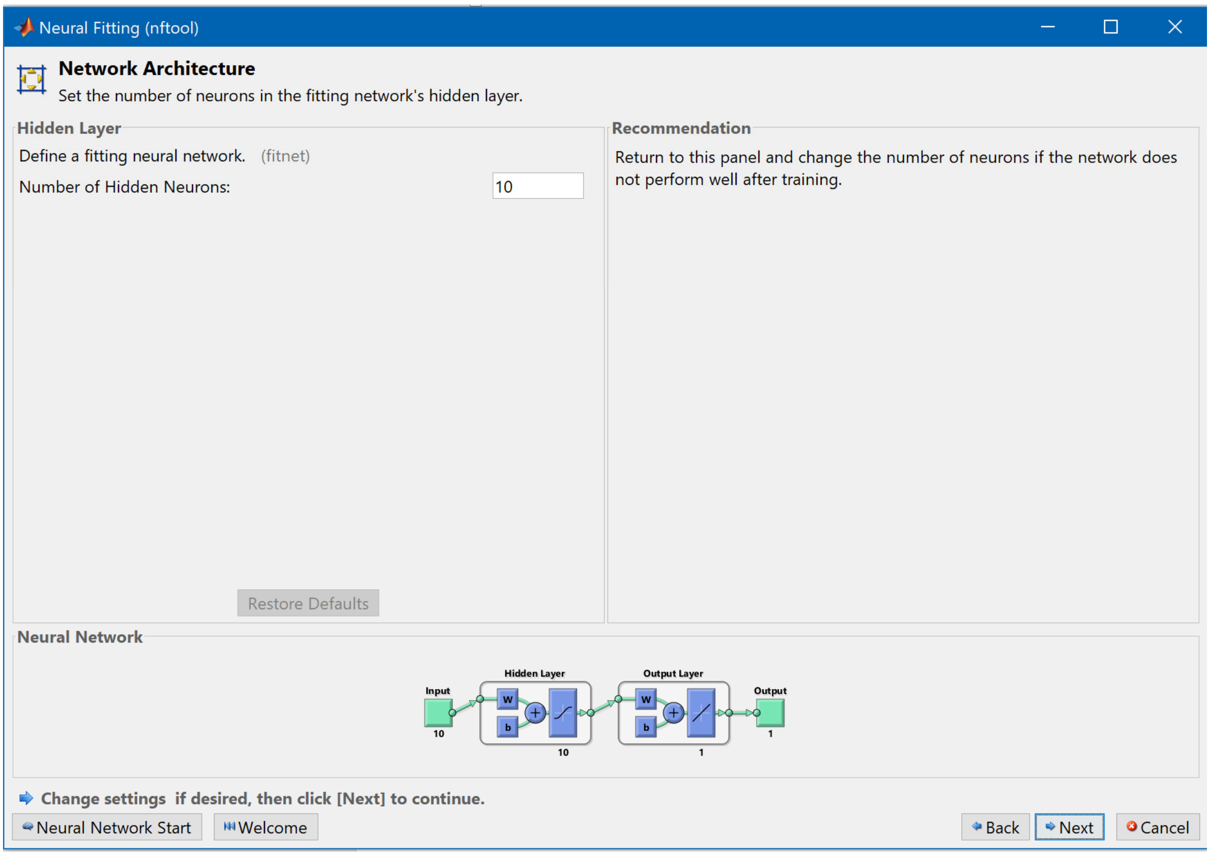
Figure 16 CART result (7 nodes) of the dataset

7.3. The performance of Artificial Neural Network with data of aggregated SCATS flow

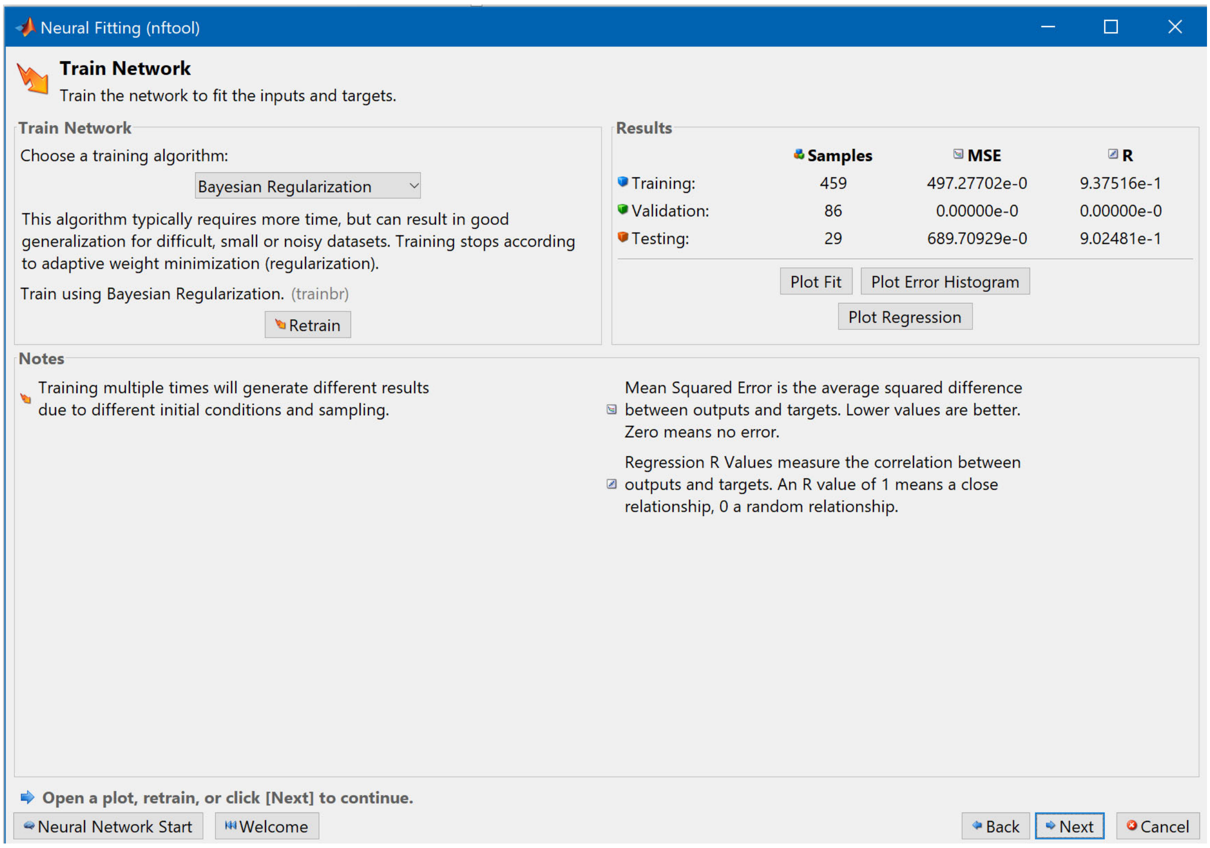
In addition to the CART, the author also tried to use the Artificial Neural Network (ANN) to use the dataset for the prediction of the incident duration. The detailed settings used for this project is attached as follows.



Note: The author modified the default percentage for training, validation and testing for a better regression. However, the smaller number of the testing sample might induce randomness of the performance.



Note: the author tried to increase the number of the hidden neurons from 10 to 15. However, the performance dropped after the modification. The reason is thought to be caused by the limited number of training data and overfitting. Thus, the author restored the default setting of 10 hidden neurons for the training of the network.



Note: according to the MATLAB document, Bayesian Regularization is ideal for the small dataset.

Part V – Correlation analysis

The performance of the ANN for the dataset with the aggregated SCATS flow is as follows.

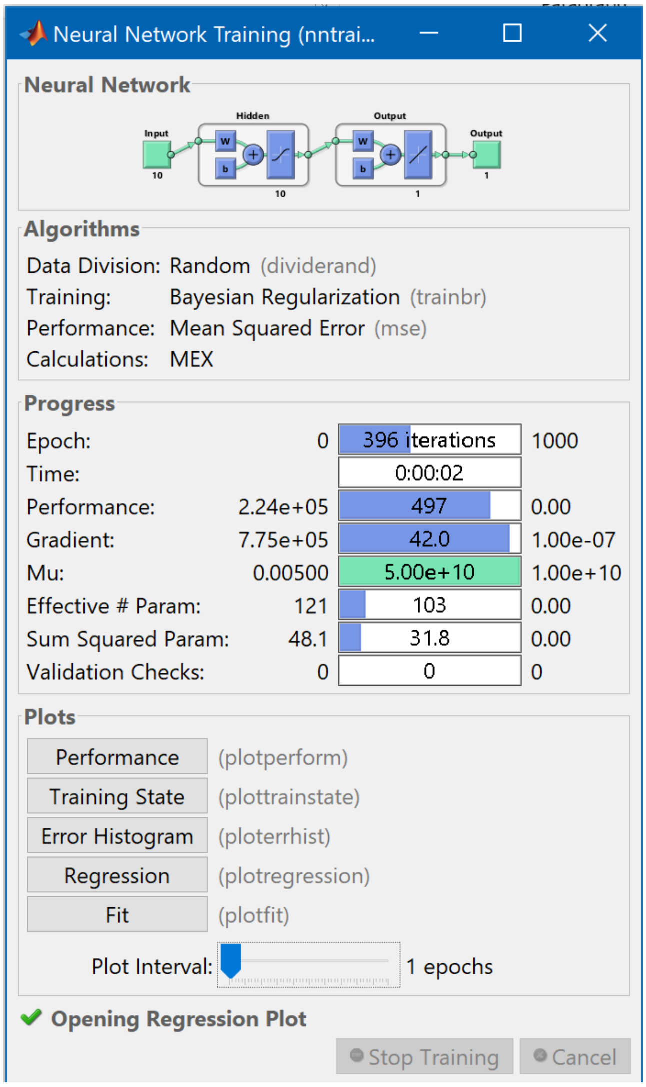


Figure 17 Summary of the performance – ANN, dataset with aggregated flow count

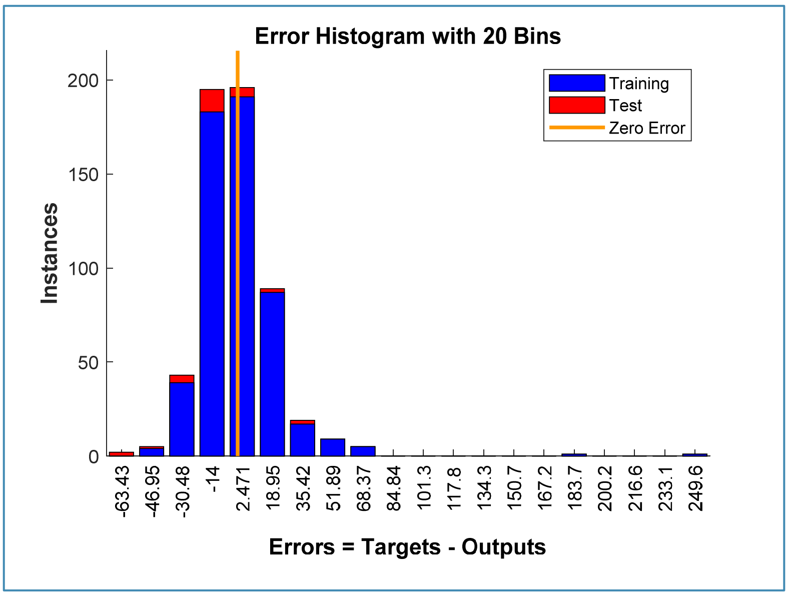


Figure 18 ANN Error histogram, dataset with aggregated flow count

For the error histogram in Figure 18, although most of the errors are within ± 15 minutes, a small number of serious outliers exist in the prediction of the incident duration, with an extreme case of 250 minutes error (≈4 hours). This indicates that the network cannot process certain "rare cases" properly, which suggests that the training sample size is not comprehensive enough.

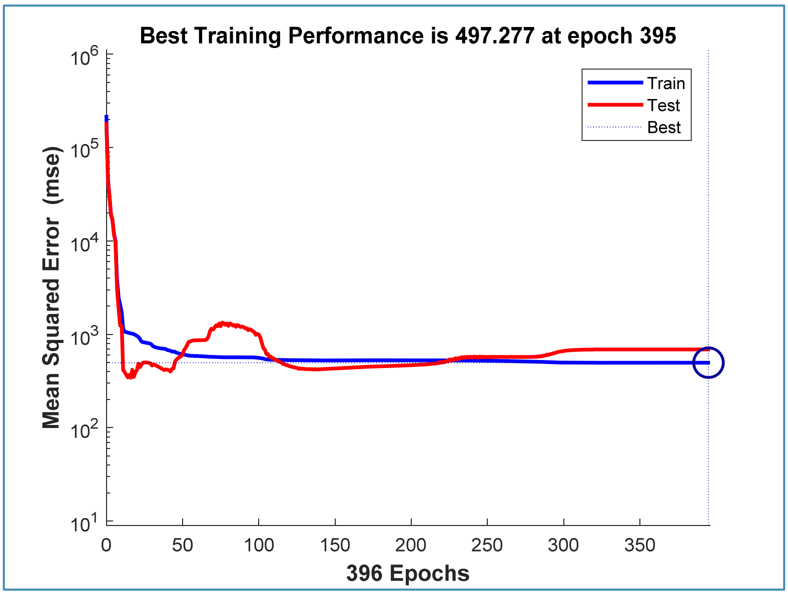


Figure 19 ANN Performance, dataset with aggregated flow count

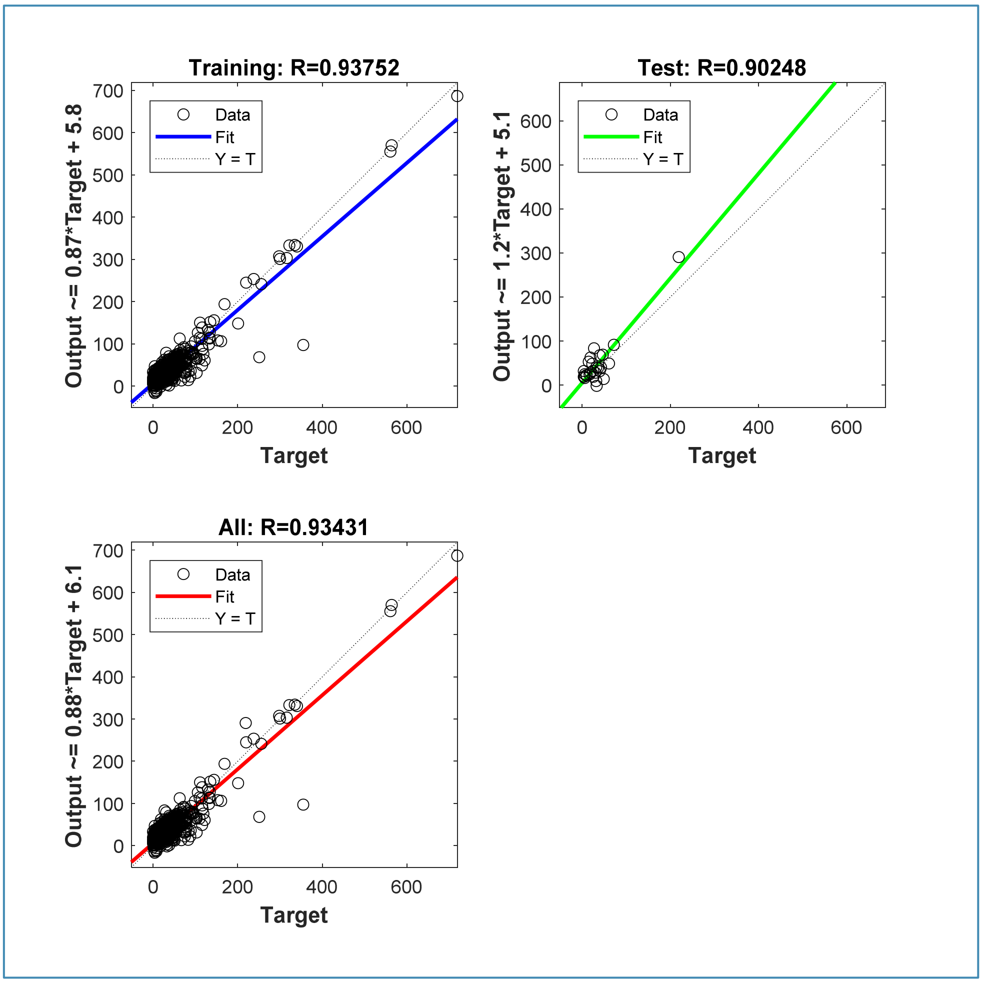


Figure 20 ANN Regression, dataset with aggregated flow count

The testing regression has an R^2 of 90%, which is better than expected given the small dataset. And if combining the training and testing samples, the overall R^2 is around 93%. However, as shown in the testing diagram in Figure 20, the testing samples are heavily concentrated within a certain area (incidents with duration under 100 minutes) and only one sample with long duration (around 200 minutes) are tested. Thus, the author concluded that the regression result might be subjected to randomness, and larger sample size is needed for future testing.

7.4. The performance of Artificial Neural Network with data of real-time SCATS flow

To investigate the correlation between the real-time SCATS flow data with the incident duration, the author used the dataset created in section *7.1 The real-time SCATS flow + other factors* to train another ANN with the same settings. The performance is detailed below in Figure 21, Figure 22 and Figure 23.

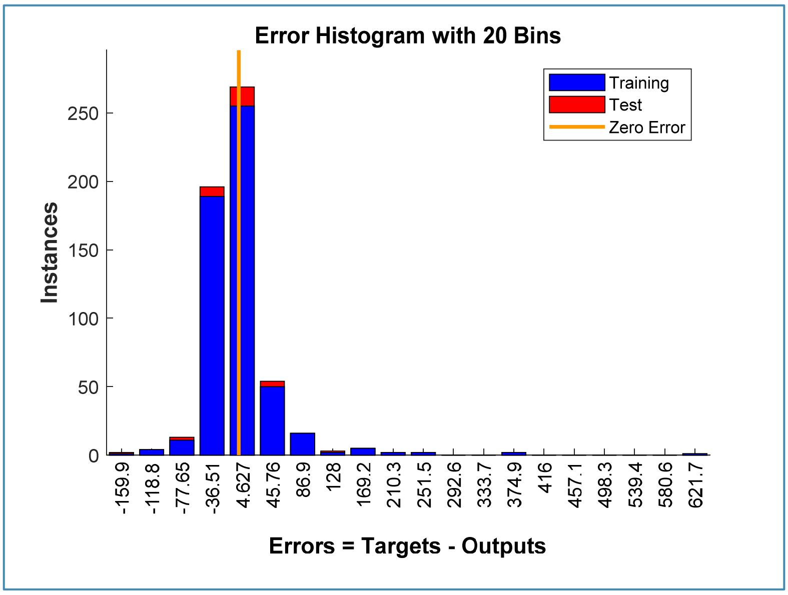


Figure 21 ANN Error histogram, dataset with real-time flow count

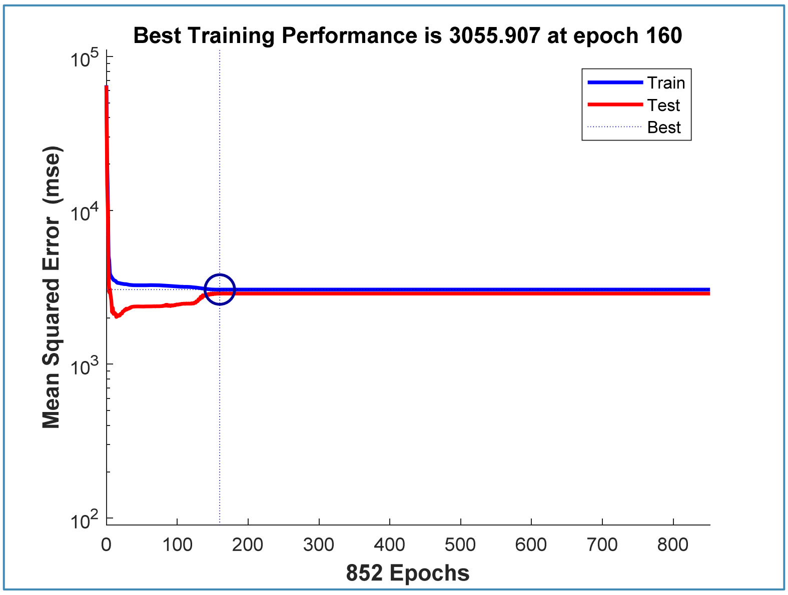


Figure 22 ANN Performance, dataset with real-time flow count

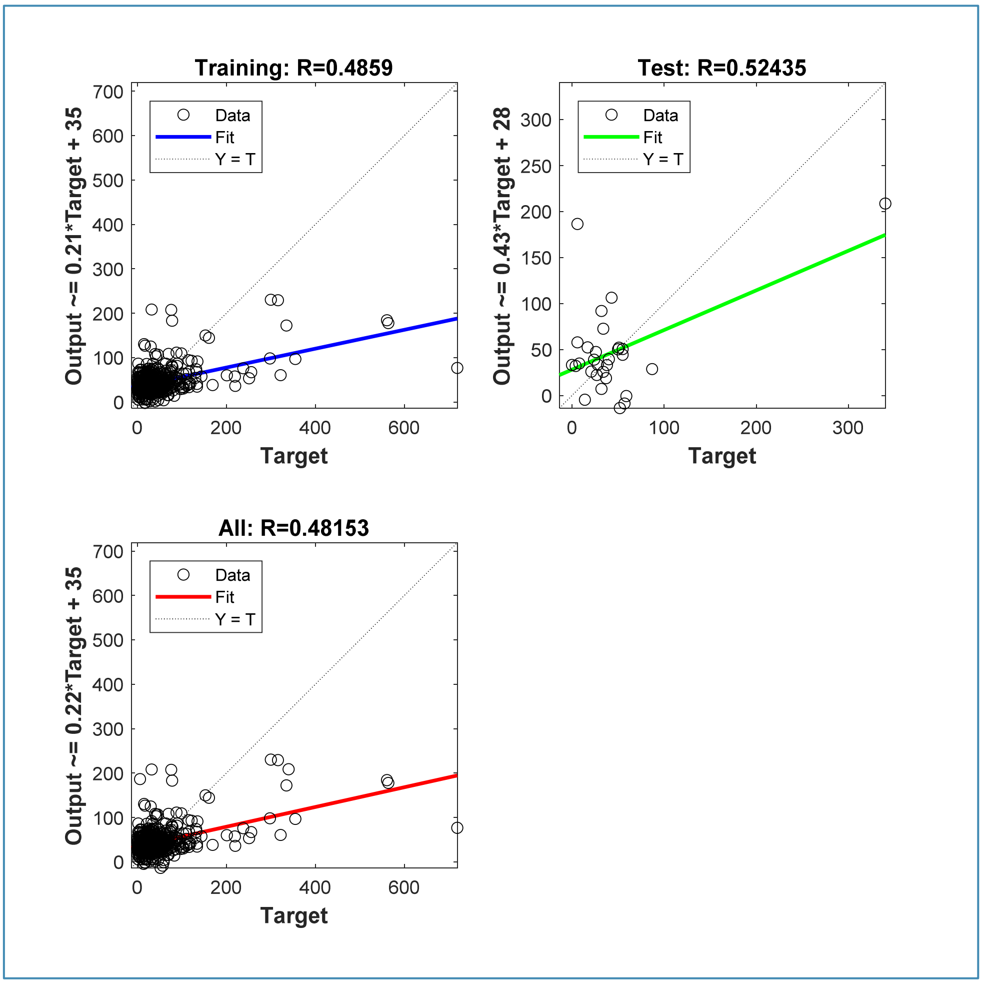


Figure 23 ANN Regression, dataset with real-time flow count

Overall, the performance of the ANN has an R of only around 50%, which is as anticipated as the real-time SCATS flow data has a limited indication on the incident duration. The author anticipated that if the network is trained with a significantly larger number of samples, the performance could be better. However, the resources at hand are limited.

8. Future work

By reflecting on the project outcomes and its flaws, the author proposed the following future work:

- Find alternative sources for incident data, collect additional incident data to expand the sample size

As mentioned hereinbefore, the author failed in obtaining historical incident dataset. As a result, the author could only use scripts to record the real-time data stream from the LiveTraffic data source. However, this has proven to be inefficient, unreliable and costly; as the Python script needs to run 24-hours nonstop, and unexpected error might occur anytime. Therefore, we must find alternative sources for collecting the incident data, preferably from the official channels.

The lack in incident data also affected the data-driven modelling in predicting incident duration. At the moment, the author could only utilise 574 incident events for the training and testing of the Artificial Neural Network, which is no doubt insufficient. It is hopeful that by collecting more incident records, the training sample size can be expanded and a better result can be obtained.

- Refine dataset on contributory factors

Similar to what was described above, the dataset on contributory factors- including weather, public holiday and school events- can be further refined. The main flaws of the current dataset have been discussed in section

3.2. To summarise, school-related data is difficult to organise as various types of school can have different calendars. Moreover, to better utilise school-related data, the exact location of the schools must be obtained. Currently, the author has identified reliable sources for the address of the schools but failed to find an efficient means to extract the information online.

In addition, historical weather data is incomplete at the moment. This is because the online weather database only displays the complete weather records for the past 14 months. However, it is possible to obtain historical data by manually filing requests (charges apply).

- Further calibrate the traffic demand for Aimsun model

The calibration of the Aimsun model can be improved in several ways.

First, the author believes that the original OD Matrix should be double-checked, as it is directly linked to the abnormal performance of the calibration process (low R^2 for Micro SRC) and the irregular behaviour of the traffic model (over 60% of the sections have zero Delay time because of zero assigned volume). In addition, the Profiled Demands generated by the Static OD Departure Adjustment is not entirely reasonable either, with traffic demands decreasing from 7 to 9 AM.

Also, the SCATS flow data can be further refined (if possible). Despite the fact that the amount of SCATS sections cannot be increased in the short term due to physical constraints, the author is still hopeful that by refining the SCATS flow data, the validation of the Meso and Micro scenarios can be improved.

- Investigate DTA settings

The author also noticed that the settings in the Dynamic Traffic Assignment (DTA) could influence the result of the simulation. As an example, the author tested the Meso DUE model settings (Gradient-boosted and MSA) and found that MSA yields a slightly better R^2. Therefore, if given a chance, the author would like to fully investigate the DTA settings and drive an optimal set of settings for the simulations.

- Refine Aimsun incident simulation script

As described hereinbefore, the Aimsun automatic incident simulation script is still incomplete at the moment in the sense that both its functionality and accuracy need to be improved. Details are as follows.

First, the accuracy of *geomapping* must be increased. As documented in section 6.1, the author could only use the geographical relationships to deduce the sections where the incident occurred, which has proven to be inaccurate. Potentially, this issue can be solved by Natural Language Processing, but such means still faces challenges. After discussing with the author's supervisor and Dr Tao Wen, the author concluded that the best approach to remedy the issue is to create standards and GUI for CMCS operators to report incidents in a more coherent style.

Also, at the moment the Aimsun script only supports incidents of type *Accident*, which is incomplete. Therefore, it is important to develop the script further and add more functionalities, so that it can process other types of incident. During this process, it is necessary to investigate the behaviour of different types of incidents so that one could identify the appropriate Aimsun Traffic Condition(s) that could replicate the incident during a simulation.

Finally, since the automatic script created by the author is supposed to contribute to the final objective of fully-automatic model calibration and incident simulation process, the author anticipates that at some point, the script should be integrated with code produced by the ADAIT team to realise the goal.

- Further investigate on data-driven modelling of predicting incident durations

In this project, the author managed to utilise the resources at hand to identify the contributory factors that have the most influence on incident duration. However, due to the limitation on time and dataset (as described beforehand), the outcome is not ideal. Hence, in the future, given more time and more comprehensive dataset, the author expects that one could further investigate the relationships between the major factors and the incident duration. Also, if the sample size is large enough, on might be able to train a data-driven model for predicting incident duration successfully.

9. Conclusion

To conclude, in this project, the author first collected data on incidents and potentially influential factors of congestion. Then, the author performed Aimsun model calibration and scenario testing to evaluate the traffic congestions under the influence of incidents. In addition, the author also attempted to identify the correlation between congestion behaviour and potential contributing factors.

The outcome of the model calibration is overall as anticipated, with an R^2 of around 82% in the mesoscopic level. However, the regression in the microscopic level is unsatisfactory with an R^2 of only 60%. The fundamental diagram of the mesoscopic simulation also shows a lower-than-usual congestion rate in sections. The reason is thought to be caused by the original OD Matrix. Also, the SCATS flow data could be further refined for better validation.

The comparison between simulations with and without incident(s) shows that the incident(s) can have a serious impact on certain sections, but its influence on the entire subnetwork can be limited. To better visualise the incident impact, the author intentionally selected and compared time-series data for sections that are close to the incident location. The comparison shows more significant differences between the two cases.

The data-driven modelling reveals a strong correlation between the aggregated SCATS flow data and the incident duration, which is as expected. It also shows that the average temperature, rainfall, severity and the subtype of the incident have noticeable influences on the duration of the incident. On the other hand, cloud, humidity and the day of week when the incident occurred has no strong effects on the duration. The trained Artificial Neural Network for predicting the incident duration has an R of 90%, which is better than expected.

10. References

1. Mihaita A. S., Benavides, M., Camargo, C. Cai, Predicting air quality by integrating a mesoscopic traffic simulation model and air pollutant estimation models, International Journal of Intelligent Transportation System Research (IJITSR), DOI: 10.1007/s13177-018-0160-z, Online ISSN1868-8659, Online scheduled for 15 May 2019, Volume 17, Issue 2, pp 125–141.
2. Mihaita A. S., Dupont L., Camargo M., Multi-objective traffic signal optimization using 3D mesoscopic simulation and evolutionary algorithms, Simulation Modelling Practice and Theory (SIMPAT), https://doi.org/10.1016/j.simpat.2018.05.005, Volume 86, August 2018, Pages 120-138, (IF = 2.063, H5=49).
3. Wen T, Mihăiţă A-S, Nguyen H, Cai C, Chen F. Integrated Incident Decision-Support using Traffic Simulation and Data-Driven Models. Transportation Research Record. 2018;2672(42):247-256. doi:10.1177/0361198118782270, (IF = 0.695, H5 = 48)
4. Mihaita A.S., Mocanu S., Lhoste, P., "Probabilistic analysis of a class of continuous-time stochastic switching systems with event-driven control", European Journal of Automation (JESA), **July** 2016.
5. Monticolo, D., Mihaita, A.S., Darwich, H., Hilaire, V., "An Agent Based System to build project memories during engineering projects", Knowledge Based Systems Journal (KBS), January 2014
6. Monticolo, D. Mihaita A.S. "A multi Agent System to Manage Ideas during Collaborative Creativity Workshops", International Journal of Future Computer and Communication (IJFCC), vol 3., nr 1, February 2014, P66-71, (extended version of the paper presented in ICFCC 2013).
7. Mihaita A.S., Mocanu S., "Simulation en temps continu pour la commande orientée événements des systèmes stochastiques à commutation", European Journal of Automation (JESA), 45 1-3 (157-172), Oct 2011.
8. Mihaita A. S., Dupont L., Cherry O., Camargo M., Cai C., Air quality monitoring using stationary versus mobile sensing units: a case study from Lorraine, France, 25th ITS World Congress (ITSWC 2018), Copenhagen, Denmark, 17-21st of September 2018.
9. Wen Tao, Mihaita A.S., Nguyen Hoang, Cai Chen, Integrated Incident decision support using traffic simulation and data-driven models. Transportation Research Board 97th Annual Meeting (TRB 2018), Washington D.C., January 7-11, 2018.
10. Mihaita A. S., Tyler Paul, Wall John, Vecovsky Vanessa, Cai Chen, Positioning and collision alert investigation for DSRC-equipped light vehicles through a case study in CITI, 24th World Congress on Intelligent Transportation Systems (ITSWC 2017), Montreal, Canada, 29 October - 2 November 2017.

11. Mihaita A. S, Cai Chen, Chen Fang, Event-triggered control for improving the positioning accuracy of connected vehicles equipped with DSRC, International Federation of Automatic Control World Congress (IFAC WC 2017**)**, 9-14 July 2017, Toulouse, France.
12. Mihaita A. S, Tyler Paul, Menon Aditya, Wen Tao, Ou Yuming, Cai Chen, Chen Fang, "An investigation of positioning accuracy transmitted by connected heavy vehicles using DSRC", Transportation Research Board 96th Annual Meeting **(TRB 2017)**, Washington D.C., January 8-12, Paper number 17-03863, 2017, https://pubsindex.trb.org/view/2017/C/1438533.
13. Mihaita A. S., Benavides, M., Camargo, M., "Integrating a mesoscopic traffic simulation model and a simplified NO2 estimation model", 23rd World Congress on Intelligent Transportation Systems (ITSWC 2016), Melbourne, Australia, 10-14 October 2016.
14. Mihaita A.S., Camargo, M., Lhoste, P. , " Evaluating the impact of the traffic reconfiguration of a complex urban intersection ", 10th International Conference on Modelling, Optimization and Simulation (MOSIM 2014), Nancy, France, 5-7 November 2014 (accepted on 18th of July 2014).
15. Mihaita A.S., Camargo, M., Lhoste, P. "Optimization of a complex urban intersection using discrete-event simulation and evolutionary algorithms", International Federation of Automatic Control World Congress (IFAC WC 2014), Cape Town, Africa, 24-29 August 2014.
16. Issa, F., Monticolo, D., Gabriel, A. , Mihaita, A.S., "An Intelligent System based on Natural Language Processing to support the brain purge in the creativity process", IAENG International Conference on Artificial Intelligence and Applications (ICAIA 2014), Hong Kong, 12-14 March, 2014.
17. Monticolo, D., Mihaita A.S., "A Multi Agent System to manage ideas during Collaborative Creativity Workshops", 5th International Conference on Future Computer and Communication (ICFCC 2013), Phuket, Thailand, 26 May 2013.
18. Mihaita A. S., Mocanu S., "Un nouveau modéle de l'énergie de commande des systèmes stochastiques à commutation", Septième Conférence Internationale Francophone d'Automatique (CIFA 2012) Grenoble, France, 4-7th of July, 2012.
19. Mihaita A. S., Mocanu S., "An Energy Model for the Event-Based Control of a Switched Integrator", International Federation of Automatic Control World Congress (IFAC WC 2011), Milano, September 2011.
20. Rizoiu, M. A., & Velcin, J. (2011). Topic extraction for ontology learning. In W. Wong, W. Liu, & M. Bennamoun (Eds.), Ontology Learning and Knowledge Discovery Using the Web: Challenges and Recent Advances (pp. 38–60). IGI Global. https://doi.org/10.4018/978-1-60960-625-1.ch003
21. Rizoiu, M. A., Xie, L., Caetano, T., & Cebrian, M. (2016). Evolution of privacy loss in wikipedia. In WSDM 2016 - Proceedings of the 9th ACM International Conference on Web Search and Data Mining (pp. 215–224). New York, New York, USA: ACM Press. https://doi.org/10.1145/2835776.2835798
22. Rizoiu, M.-A., & Xie, L. (2017). Online Popularity under Promotion: Viral Potential, Forecasting, and the Economics of Time. In International AAAI Conference on Web and Social Media (ICWSM '17) (pp. 182–191). Montréal, Québec, Canada. Retrieved from https://aaai.org/ocs/index.php/ICWSM/ICWSM17/paper/view/15553
23. Mishra, S., Rizoiu, M.-A., & Xie, L. (2018). Modeling Popularity in Asynchronous Social Media Streams with Recurrent Neural Networks. In International AAAI Conference on Web and Social Media (ICWSM '18) (pp. 1–10). Stanford, CA, USA. Retrieved from https://arxiv.org/pdf/1804.02101.pdf
24. Kong, Q., Rizoiu, M.-A., Wu, S., & Xie, L. (2018). Will This Video Go Viral: Explaining and Predicting the Popularity of Youtube Videos. In The Web Conference 2018 - Companion of the World Wide Web Conference, WWW 2018 (pp. 175–178). Lyon, France: ACM Press. https://doi.org/10.1145/3184558.3186972