



Planning and Transport Research Centre (PATREC)

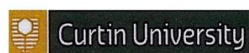
## Planning intermodal and general logistics infrastructure for the future needs of Perth:

### Telemetry Systems for Tracking Road Freight Activity

Prepared by	Tim Hoffman, Daniel Cowen, Rachel Cardell-Oliver & Sharon Biermann
Date	December 2019
Project details	iMOVE Project 2.001, incorporating Reports no. 6.2 & 7.1 (appended)
Version	Final



THE UNIVERSITY OF  
**WESTERN  
AUSTRALIA**



---

## EXECUTIVE SUMMARY

The larger project of which this report forms a part, comprises a suite of related research streams to support the state of Western Australia (WA) and the Westport Taskforce in particular, in the planning for landside logistics infrastructure and services, including a possible new container berth in Kwinana, to support container trade growth. This major long-term infrastructure planning process provides an opportunity for the development of logistics infrastructure including road and rail corridors and terminal facilities to ensure the most cost-efficient and environmentally sympathetic supply chains for businesses depending on imports and exports. Further, it provides the opportunity to consider the impacts of unfolding global trends in logistics and supply chains on the development of port areas and industrial zones throughout Perth. The project also explores the potential to use freight transporters' operational data already being generated via in-cab GPS monitoring systems for public policy purposes, particularly in short-haul urban environments.

The wider project: Planning intermodal and general logistics infrastructure for the future needs of Perth, comprises three components:

- Intermodal Systems for Perth
- Global Supply Chain Trends and Local Perspectives
- **Telemetry Systems for Tracking Road Freight Activity**

This report concerns the third component: Telemetry Systems for Tracking Road Freight Activity, involving the study of road freight movement activity, using GPS fleet management systems, aggregated for mapping and analysis via a software tool developed by the Bureau of Infrastructure, Transport and Regional Economics (BITRE). The aim was to pilot, adapt and test BITRE's methodology for short-haul urban environments to support BITRE's continued national roll-out and to ensure that local WA needs are met in the longer term. Success would enable cost-effective annual or ongoing monitoring of truck movement activity between the major surveys for longitudinal review and modelling

### **Research question**

*Can BITRE's YULO software be adapted to monitor trucking activity in short-haul urban environments to supplement major commercial vehicle surveys and inform local and state transport planning and policy?*

### **Key objectives**

- To pilot BITRE's YULO software in Perth and adapt the software for local context and specific purposes (Appendix A)
- To test the adapted YULO software using local data (Appendix B)

### **Key findings**

- BITRE's YULO framework is viable for application in analysing short-haul, urban freight activity (Appendix A)
  - YULO is able to succinctly analyse and visualise aggregated and single trip information for use in policy and planning.
  - As part of the adoption of the framework from regional activity analysis to urban / short-haul freight analysis in Perth, the following adaptation were implemented and shared with BITRE:
    - Re-implementation of most of the existing front-end into Python to facilitate faster prototyping

- Creation of a directed road network 'mini-segments' shapefile for better visualisation of urban routes
  - Implementation of directionality of travel on road segments in the Neo4j back-end
  - The ability to save interactive maps as stand-alone files for sharing with stakeholders
  - Development of a weighted routing algorithm
  - Initiated shapefile merge, periodicity mining and top-k paths of length-p algorithms
- The adapted YULO software was tested using GPS data for Oversize-Overmass (OSOM) escort vehicles made available by Main Roads WA, Heavy Vehicle Services (Appendix B).
  - Demonstrated the power of GPS telemetry data-bases for future governmental use in the management of the road network, vehicle registration system and administration of the freight transport industry
  - Specifically for Main Roads WA, this analysis provides pointers as to how the OSOM escort fleet management data routinely produced and stored, might be useful for other purposes, both within the narrow management interest, and into the realm of highway planning.
  - Escort vehicle data is significant, as it provides proxy information on routes taken by a subset of heavy freight vehicles in Perth and WA, as well as details on the performance of vehicles under escort, and the average speeds achievable by freight vehicles on all sections of the highway network.

### **Key Challenges**

The primary difficulty was to source useful data from transport firms, despite their willingness in principle to assist the research. Some of the reasons for this are:

- many firms have no reason to keep trip data for very long
- many firms do not utilise or know how to utilise the full capacity of their fleet management systems
- some systems do not present data in compatible format
- there is little incentive for firms to assist in research in this direction

### **Acknowledgements**

*The YULO software framework and methodological information provided to PATREC by BITRE has been critical to the project and the research team is grateful for this assistance from BITRE.*

*The research team also wishes to thank the Western Roads Federation for their assistance in promoting the research to members and encouraging them to make their data available.*

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

## TABLE OF CONTENTS

<b>EXECUTIVE SUMMARY .....</b>	<b>2</b>
<b>APPENDIX A.....</b>	<b>5</b>
<b>1 INTRODUCTION .....</b>	<b>6</b>
<b>2 BITRE YULO SYSTEM.....</b>	<b>7</b>
2.1 System Overview.....	7
2.2 Performance .....	8
2.3 Yulo vs Commercial Telematics .....	10
2.4 Further Developments of Yulo.....	10
<b>3 GPS TRACE INPUTS.....</b>	<b>10</b>
3.1 Project Data Sets .....	10
3.2 GPS Input Requirements .....	11
3.3 Data Security and Privacy.....	12
3.4 Sourcing Data: Lessons Learned.....	12
<b>4 RESULTS.....</b>	<b>13</b>
4.1 Aggregated Trips and Stops .....	13
4.2 Single Trips.....	17
4.3 Map Smoothing .....	17
4.4 Shortest Path Routes .....	19
4.5 Future Work: Freight Fleet Flows .....	20
<b>5 PROJECT OUTPUTS.....</b>	<b>22</b>
<b>6 STUDENT PROJECTS.....</b>	<b>24</b>
<b>7 CONCLUSION AND FUTURE WORK .....</b>	<b>25</b>
<b>8 REFERENCES.....</b>	<b>26</b>
<b>APPENDIX B.....</b>	<b>28</b>
<b>1 INTRODUCTION .....</b>	<b>29</b>
1.1 Project aims and background .....	29
1.2 The SWAN YULO system.....	29
1.3 MRWA escort vehicle Navman data set.....	30
<b>2 ANALYSIS .....</b>	<b>31</b>
<b>3 SAMPLE RESULTS.....</b>	<b>31</b>
3.1 Aggregated trip counts .....	31
3.2 Individual trip description.....	33
3.3 Aggregated speed calculation .....	34
3.4 Other analytical samples.....	35
<b>4 FUTURE DIRECTION .....</b>	<b>38</b>
<b>5 ACKNOWLEDGEMENTS .....</b>	<b>39</b>

## APPENDIX A



Planning and Transport Research Centre (PATREC)

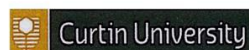
### Planning intermodal and general logistics infrastructure for the future needs of Perth:

#### Telemetry for Tracking Road Freight Activity - Pilot to Monitor Short-Haul Urban Trucking Activity Using GPS-Based Truck Management Systems

Prepared by	Tim Hoffman, Daniel Cowen, Rachel Cardell-Oliver & Sharon Biermann
Date	July 2019
Project details	iMOVE Project 2.001, Report no. 6.2
Version	Final



THE UNIVERSITY OF  
**WESTERN  
AUSTRALIA**



---

# 1 INTRODUCTION

The Yulo Framework has been in development by the Bureau of Infrastructure, Transport and Regional Economics (BITRE) from 2017. It was developed as a pilot project to “use vehicle telematics data to provide information about the Australian road freight network and to assist decision making by firms, government and the public” [18]. BITRE has expressed interest in collaborating with researchers to extend and adapt its pilot methodology to a specific short-haul urban environment. Separately, the Western Roads Federation (WRF) has advised that they are interested in a study to use similar data to monitor service-related truck movement in urban environments.

This report describes the results of the iMove sub-project: Monitoring Short-Haul Urban Trucking Activity Using GPS-Based Truck Management Systems - Pilot. The project ran from January to May 2019. The project has three main goals [8]:

- G1: To pilot BITRE’s methodology by adapting the method for short-haul urban environments to support BITRE’s continued national roll-out and to ensure that local WA needs are met in the longer term. Success would enable cost-effective annual or ongoing monitoring of truck movement activity between the major surveys for longitudinal review and modelling. The pilot would focus one or more particular service industries eg. courier services, waste removal and/or retail deliveries.
- G2: Serve a short-term goal to enable daily truck monitoring as input to local area transport planning especially truck monitoring in CBDs and other high intensity urban environments, in order to plan and advocate for challenges relating to increasing land use intensity and more on-line deliveries, including issues such as the provision of parking spaces etc.
- G3: Enable data driven analysis of truck movements to discover hub and trip characteristics of movement (as has recently been done using TransPerth SmartRider data for hubs, journeys and passengers).

This report presents the main results of this project. Section 2 describes the functionality of BITRE’s Yulo system and its components. Performance measurements for the framework are presented and we conclude with a discussion of limitations of Yulo for the urban datasets we tested. Section 3 describes the GPS trace data sets that were sourced for this project, the required format for raw GPS pings used by Yulo and the data security and privacy principles we applied. The section concludes with some lessons learned about sourcing GPS data. Section 4 is the main part of the report, detailing our results in five project areas: aggregated trips, single trips, map smoothing, shortest path routing, and fleet routing. Section 5 is a dictionary of the software tools developed in this project and Section 6 outlines three related Masters student research projects that are continuing in 2019. Section 7 summarises our findings and outlines future work.

---

## 2 BITRE YULO SYSTEM

This section describes the functionality of BITRE's Yulo system and its components. Performance measurements for the framework are presented and we conclude with a discussion of limitations of Yulo for the urban datasets we tested.

### 2.1 System Overview

The Yulo Framework consists of a series of computer scripts, programs and documentation. It was originally written by the Bureau of Infrastructure, Transport and Regional Economics (BITRE) as part of a "joint pilot study with the Australian Bureau of Statistics to assess suitability of freight vehicle telematics data for providing statistical information about road freight vehicle movements and network usage in Australia" [19]. Figure 1 shows examples of the input data for Yulo and output results. Yulo takes as input raw, unprocessed, GPS pings with vehicle IDs. Output comprises summary statistics on the number and speed of vehicles on road segments.

Figure 2 gives a high-level view of the Yulo tool chain. Yulo uses a Linux toolchain and is written mostly in Python, R and Java. The first processing step takes input sequences of GPS pings and aggregates them into groups of pings representing discrete trip segments, stops and residuals. Residuals are possibly trips, but they require more information to make a final determination.

GPS pings are mapped to road segments by a modified version of an open-source application called Barefoot [4]. Barefoot was developed by BMW and was designed chiefly to correct displayed locations of vehicles on their in-vehicle displays. It does this by receiving data from a central Barefoot server or by conducting in-vehicle processing of batches of recorded data. Barefoot performs the essential 'matching' of, potentially inaccurate, GPS pings to road segments (provided by OpenStreetMap [17]). Barefoot uses a Hidden Markov Model to map pings to road segments, ensuring that pings are matched to the most likely segment based on previous pings, segments matched, and distance from the ping is to its closest segment.

The next step re-joins fields to the dataset that were not used by Barefoot and the results uploaded into a central database, Neo4j where they can be filtered and summarised using database queries.

Yulo has a visual front end that can present individual and aggregated data on an interactive map in a similar fashion to Google or Bing Maps. This presentation layer is built using the R tool, ShinyLeaflet. Examples of analyses that can be performed include average speed of all vehicles on road network, average speed or time per trip, average speed of road segment, number of unique trips or vehicles on any part of the road network, or average stop time. The presentation layers are written entirely with open-source tools and so they can easily be updated or developed further depending on analysis requirements. For example, in this project we have built visualisation tools using the Python Folium library.

We have successfully installed and tested the Yulo Framework using a large dataset sourced from a transport company in Perth. Section 4 presents the results including screen shots from Yulo's presentation layer.

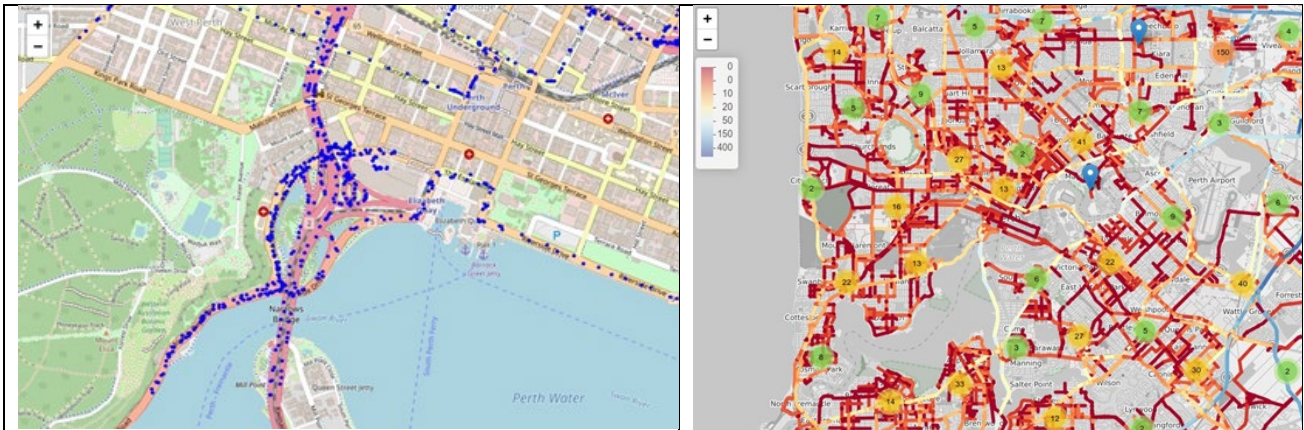


Figure 1: Examples of GPS input data (left) and aggregated outputs from Yulo (right)

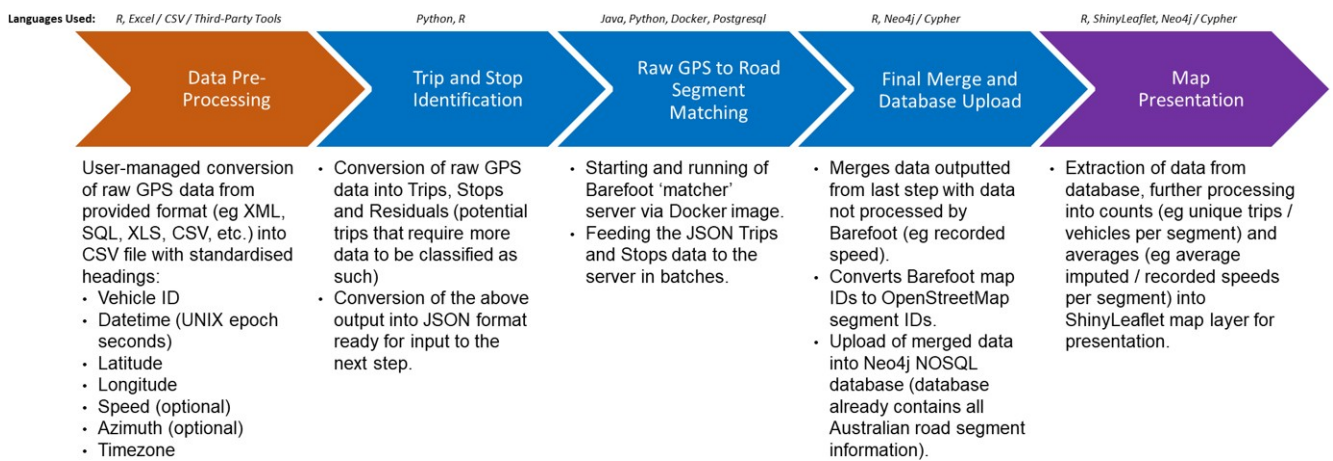


Figure 2: High-level overview of Yulo processing steps

## 2.2 Performance

This section evaluates the performance of Yulo: how long it takes to analyse input data and how different properties of the datasets affects performance. In order to do this, we partitioned a large GPS data set from a transport company into different sized test data sets.

Data set 10K 10,000 GPS pings, approx. one per 4 seconds, from 1 vehicles over 1 months covering all of the greater Perth area (53 x 40 kms). Input file size approx. 727KB.

Data set 100K 100,000 GPS pings, approx. one per 4 seconds, from 2 vehicles over 1 months covering all of the greater Perth area (53 x 40 kms). Input file size approx. 7.3MB.

Data set 1M 1 million GPS pings, approx. one per 4 seconds, from 5 vehicles over 3 months covering all of the greater Perth area (53 x 40 kms). Input file size approx. 72.8MB.

Data set 10M 10 million GPS pings, approx. one per 4 seconds, from 12 vehicles over 18 months covering all of the greater Perth area (53 x 40 kms). Input file size approx 728MB.

Step	10 K points	100 K points	1 M points	10 M points
<b>Trip and Stop Identification</b>	16.8s	84.5s	605.2s	6,533.3s
<b>Road and Segment Matching</b>	63.3s	71.2s	354.2s	3,011.1s
<b>Final Merge and Database Upload</b>	11.5s	38.8s	319.9s	700.4s
<b>Map Presentation</b>	65.4s	134.6s	239.2s	Program crash
<b>Total</b>	<b>157.0s</b>	<b>329.1s</b>	<b>1,538.5s</b>	<b>N/A</b>

Table 1: Execution times for Yulo processing steps for different sizes of input data

**Performance Testing Machine Specification:** All measurements reported in this section were performed using a desktop computer running an Intel i7-7700 CPU at 3.6GHz, 24GB DDR4 RAM, 256GB SSD, clean install of Linux Mint v19.1 (Ubuntu Bionic) and Chromium Browser (for the presentation front-end).

**Real vs User vs System Time:** The times listed above were calculated using Real time. Whilst a sum of System and User time is often the preferred method of calculating execution times, it has not been used in this case for the following reasons:

- The framework employs heavy use of parallel processing across the 4 CPU cores (with 2 threads per core) in the framework test machine. This parallel processing results in some programs completing in a lower Real time than User time (as User time is additive across the execution time in each core / thread).
- Some programs in the framework will issue commands to a Docker image or the Neo4j database and then block until these commands have been executed. In this case the System and User times returned for these processes do not include the execution time for these commands (as the processing of these commands occur on a separate process), and therefore the Real time measurement is the only accurate indicator of the execution time of this section of the framework.

Note that, in order to keep the Real time measurements as indicative of the framework's execution time as possible only the 'base' operating system and X-Window processes were run on the test machine whilst the tests were performed. The machine was 'left alone' during testing so that the Real execution times were not affected by user multitasking eg web browsing or development of other areas of the framework.

**Ping Frequency:** Yulo has been tested for GPS ping frequencies as small as one second and as large as one minute. On both of these edge cases Yulo was able to produce accurate trip and stop mappings onto its presentation layer.

**Input File Sizes:** Yulo has been successfully tested with up to 70MB (approx. 1 million GPS pings) of raw data (ie CSV) input file sizes. Above this number of pings, Yulo's presentation layer becomes unstable and inconsistent as to whether it crashes or completes successfully. It has also been observed that when very large file sizes (with over 10 - 20m GPS pings) are input to Yulo it will lead to memory requirements above the installed RAM amount during its pre-presentation back-end processing, resulting in use of swap memory and page thrashing. It has been found, however, that large input files can be split into smaller components. This allows back-end processing of very large datasets to succeed. However, the instability of the presentation layer continues to occur when the total amount of pings fed into Yulo exceeds 1 million.

**Storage Requirement for Processed Data:** It has been observed that database storage requirements appear to be approximately 3-10x the raw GPS input file. For example, if a 100MB of raw GPS pings are inputted to the framework, 300MB - 1GB of database storage will be required.

**New Instance Setup Times:** Yulo requires an approximate additional 21 minutes (1260 seconds) when setting

up a new / clean instance. This does not include additional time required to download and extract shapefiles of approx 400MB. This time includes compilation of the Barefoot docker image with Australian roads and extraction of the relevant roads into segments and subsequent upload of these segments into the Neo4j database.

## 2.3 Yulo vs Commercial Telematics

Green and Mitchell compare the functionality offered by Yulo with that of commercial telematics providers:

Major proprietary telematics providers, including Google and HERE, already develop this sort of information based on the telematics data of their customers, but this data is not open and is expensive to access, meaning industry, government and community cannot make full use of this information. It is also based on a sample that is dominated by passenger vehicles. Freight vehicles, however, have different network usage patterns, behaviour, needs and impacts that require a dedicated source of data. [18]

## 2.4 Further Developments of Yulo

In testing Yulo on urban freight data we identified several areas where further development would be beneficial.

L1: Performance bottlenecks: Occur in the road and segment matching and map presentation stages. These bottlenecks can be a problem when dealing with very large datasets. However, it is understood that the BITRE Yulo team is working to increase the scalability of Yulo's back-end.

L2 Presentation layer instabilities for large datasets: Above 70 to 100MB ( that is 1 to 1.5m GPS pings) the Yulo presentation layer exhibits instability including slow user interface and crashing whilst running the Global.R pre-processing script. However, as with the above, it is understood that the BITRE Yulo team is also working to increase the scalability of Yulo's front-end.

L3 Presentation layer granularity limitations: The current presentation layer shows mostly aggregate data (speeds, total number of unique vehicles and unique trips per road segment). However, the data is much finer- grained and more detailed in the database back-end. This data could be mined for new insights. Student projects in Section 6 are investigating some extensions.

L4 Complex tool chain: The existing system requires several different programming languages and database tools to be installed. Setting up the full system for data analysis is a non-trivial task. During this project significant parts of the toolchain were re-written from R to Python in order to simplify the toolchain. The project team notes that there still exists scope for further simplification.

---

# 3 GPS TRACE INPUTS

## 3.1 Project Data Sets

Goal G1 is to pilot BITRE's methodology using urban data sourced from Western Australian firms. The following data sets were sourced for this project. As a condition of using this data we agreed that reference to the datasets would be only by the industry segment of the supplying firm. For this reason, we do not use any company names in this report.

**Transport:** 36 months of GPS pings from approximately 100 vehicles. Pings were at an interval of 2-3 seconds. The data was assessed as appropriate for Yulo analysis, and has been processed in Yulo. Results from this data set are shown in Section 4.

**Building Supplies:** 4 months of GPS pings from 12 vehicles. Pings were at an interval of 60 seconds. The data was assessed as appropriate for Yulo analysis, and has been processed in Yulo. Results from this data set are shown in Section 4.

**Rubbish Collection (sample data):** One day of GPS pings from 3 vehicles. Pings were at an interval of 60 seconds. The data has been identified as appropriate for Yulo analysis, but a larger sample is required for useful analysis in Yulo.

**Removalists (sample data):** One day of GPS pings from 1 vehicle. Pings were at an interval of 60 seconds. The data has been identified as appropriate for Yulo analysis, but a larger sample is required for useful analysis in Yulo.

**Removals (sample data):** One day of GPS pings from 1 vehicle. Pings were at an interval of 60 seconds. The data has been identified as appropriate for Yulo analysis, but a larger sample is required for useful analysis in Yulo.

### 3.2 GPS Input Requirements

Yulo inputs sequences of raw GPS observations, GPS pings, from one or more files. An input file typically has the following fields [18]:

- A unique vehicle identifier (this allows Yulo to tie the pings together into trajectories) (required)
- The time and date of the ping (required)
- The latitude and longitude of the ping (required)
- The recorded speed at the time of the ping (optional)
- The bearing, heading or direction of the ping (otherwise known as Azimuth, also optional)

The file format for input to Yulo is a CSV file containing the following fields (datatypes in brackets). Figure 3 shows example inputs.

- Vehicle ID (string with no spaces) (required)
- Datetime in UNIX epoch seconds (integer) (required)
- Latitude (floating point number) (required)
- Longitude (floating point number) (required)
- Recorded speed in km/h (floating point number) or “N/A” (string) if not available
- Azimuth in degrees (floating point number) or “N/A” (string) if not available
- Timezone (string) eg “Australia/Perth” or “Australia/Melbourne” (required)

Note that whilst the framework only accepts CSV files in the above format, raw GPS data can be, and often is, stored in a variety of formats. These include XML, JSON, XLS and SQL. The project team notes that it is often fairly trivial to convert the data in such formats into the required CSV format.

Yulo’s back-end processing engine has been successfully tested with input file sizes up to 1GB. Above this size and Yulo chiefly becomes limited by the amount of RAM available on the computer. Larger data sets have successfully been processed in batches. Yulo’s presentation layer does start to exhibit unstable behaviour when displaying datasets larger than 70 to 100MB.

	A	B	C	D	E	F	G	H
1	Vehicle	<u>datetime</u>	<u>lat</u>	<u>lon</u>	Speed	Azimuth	<u>TZ</u>	
2	AAA1	1375291841	-30.506483	115.062841	NA	NA	Australia/Perth	
3	AAA1	1375292294	-30.506534	115.06268	NA	NA	Australia/Perth	
4	AAA1	1375292378	-30.506677	115.06261	NA	NA	Australia/Perth	
5	AAA1	1375292469	-30.506456	115.062986	NA	NA	Australia/Perth	
6	AAA1	1375292592	-30.506622	115.062659	NA	NA	Australia/Perth	

Figure 3: Raw CSV pings for Yulo input

### 3.3 Data Security and Privacy

Firms sharing their GPS fleet data are, rightly, concerned about the security and privacy of that data. This research funded by iMove and managed by the UWA Planning and Transport Research Centre (PATREC). Both organisations are fully compliant with all Australian privacy legislation including the Privacy Act 1988.

Additionally, raw GPS datasets sourced for this project are managed according to the following principles for secure storage of the data and to preserve the anonymity of companies and vehicles.

**Secure storage:** All data will be stored on the University of Western Australia’s secure Institutional Research Data Store (IRDS). Password protected access to the data will only be made for project researchers. For more information about this facility see <http://www.it.uwa.edu.au/service-catalogue/data-storage/irds>.

**Companies:** Company names will be removed at source and not be used in any presentations or publications of the data. Data will only be identified by its business sector eg shipping, towing, grocery delivery, etc. using a sector description agreed by the data owner.

**Vehicles:** Prior to input into the program, company-specific vehicle identifiers will be changed to a standardised vehicle identifier used by the program.

**Locations:** Trip and stop records that shows routes / stops unique to a specific company or vehicle will be removed before presentation. This is to remove any company-specific indicators such as depots and unique routes between them.

The following additional data privacy options could also be implemented if required:

**Deletion:** All project GPS data will be deleted within one month of the project completion.

**Vehicle Obfuscation:** In addition to anonymised vehicle identifiers, vehicle identifiers can be changed for every day and / or week of data to avoid tracing of particular vehicles. This type of data mixing is designed to prevent tracing data back to the source by using network identifiers.

**Other:** Any other reasonable requests that will enhance data privacy and security.

### 3.4 Sourcing Data: Lessons Learned

Obtaining suitable GPS data for analysis from iMove partners was one challenging aspect of this project. Here we document some of the difficulties so they can be taken account of for future research project planning.

Although many commercial companies operate vehicles with GPS trackers and they are able to access informational reports on this data (usually supplied to them by their GPS tracker provider), they do not always have access to the raw GPS traces that are required for the Yulo analysis.

In some cases, firms were keen to share their data, but their fleet data was managed by a third party

company and the fleet owners did not necessarily have access to the GPS data.

Another problem is that GPS trace data sets are typically very large. Even if firms are able to access this data as it is generated, they often have no infrastructure to then store this data for later extraction and independent processing. Obtaining and sharing their GPS data thus presented technical difficulties for these companies.

Security and privacy of the data is another area of concern. Vehicle tracking data contains sensitive information and so firms can be very protective of their data. Internal resistance to its supply can also occur even if the same firm is outwardly committed to enabling the development of insights from it to strengthen their advocacy activities. Section 3.3 details the policies we adopted to address these concerns.

---

## 4 RESULTS

This section describes the results of applying Yulo visualisation to the WA data sets, and also results using additional functionality that was developed in this iMove project. Ongoing research is also reported.

### 4.1 Aggregated Trips and Stops

Goal: Given a map (a set of road segments) and vehicle movements on those segments (a set of trips), calculate aggregates for the average speed, number of trips and number of unique vehicles for each road segment in the map. Display the results on a map.

This goal is supported by Yulo's existing presentation layer. The main views of this presentation layer are: average speeds, number of trips, number of unique vehicles per road segment. Stops are represented by the number of vehicles and stop times for the map locations where the GPS data indicates that a vehicle has stopped. Figure 4 shows all trips and stops from the transport company data set using 3 years of trips by 100 vehicles. The colours of the roads represent the average speeds of all vehicles from the data set on that road. The coloured, numbered circles represent stopping places for the vehicles. Each circle represents a cluster of stops which, when zoomed in expand as blue markers. The colour legend is non-standard for this type of numeric range data, using red for low speeds and blue for high speeds as on the freeway. One task for future work is to implement this view using a standard heat map scale (Masters project with iMove scholarship, Alex Fletcher).

Figure 5 shows the same data as Figure 4 but zoomed in on a section of the Perth CBD. The key on the left corresponds to the average speed of all vehicles and trips on each street segment. Hovering the mouse pointer over any segment shows a detailed (aggregated) statistical view for this segment. The line graph on the right shows the number of vehicles on this segment by hour of the day or day of the week for the selected segment.

Figure 6 shows two further Yulo views: the number of trips (left) and the number of unique vehicles (right) on any given road segment. The left hand figure is from the GPS dataset of an eastern-suburbs building company. The blue road segments on the right of the left hand figure indicate a large number of trips on these road segments, where as most other road segments are red or yellow for a lower number of trips on non-arterial roads. The number of unique vehicles view (right) is within the Perth CBD for a transport company with approximately 100 unique vehicles in their fleet. Again the main and arterial roads have a high unique vehicle count (dark blue) where as side and capillary roads (red or yellow) are used by fewer vehicles.

A stop within a GPS trace is defined a repeated number of GPS pings within a geo-fenced area (default is 100m) that persist for a minimum time (default if 5 minutes). Stops derived from the raw GPS data are visible

as blue markers on the interactive map when zoomed in, and clustered together into coloured circles when zoomed out. Figure 7 is an example of the stops view, showing a zoom to the same position as the average road segment speed view, but this time focussing on vehicle stop data. The coloured circles show where vehicles are stopping and how many stops. The spiral of blue markers at the show an expanded view of each stop in this section of Roe Street. Clicking on one of these markers shows information including the latitude and longitude of the stop, its start and end times and total duration.

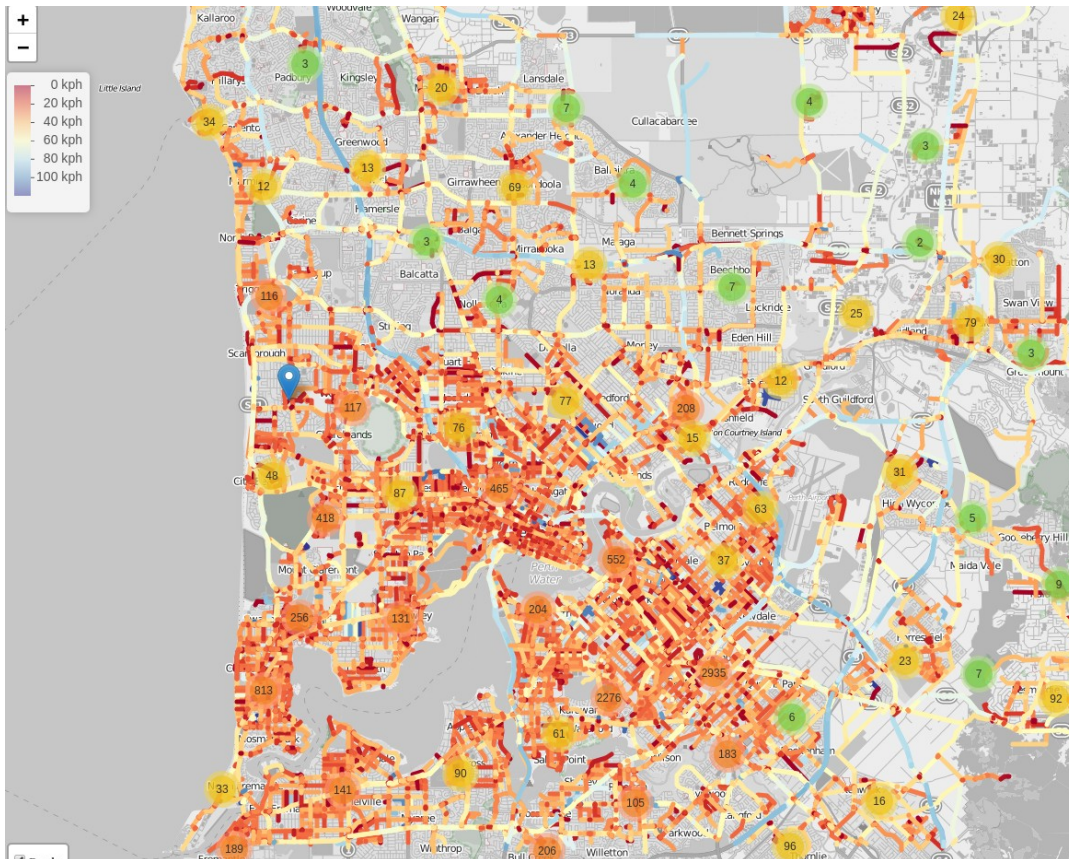


Figure 4: Vehicle movements in greater Perth for a transport company



Figure 5: Average speed per road segment view

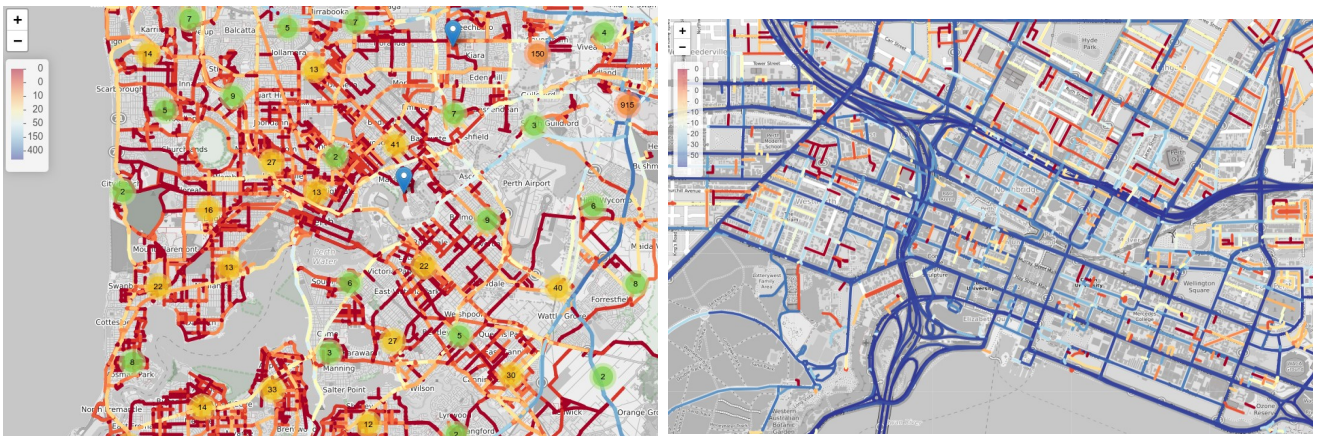


Figure 6: Number of trips (left) and unique vehicles (right)

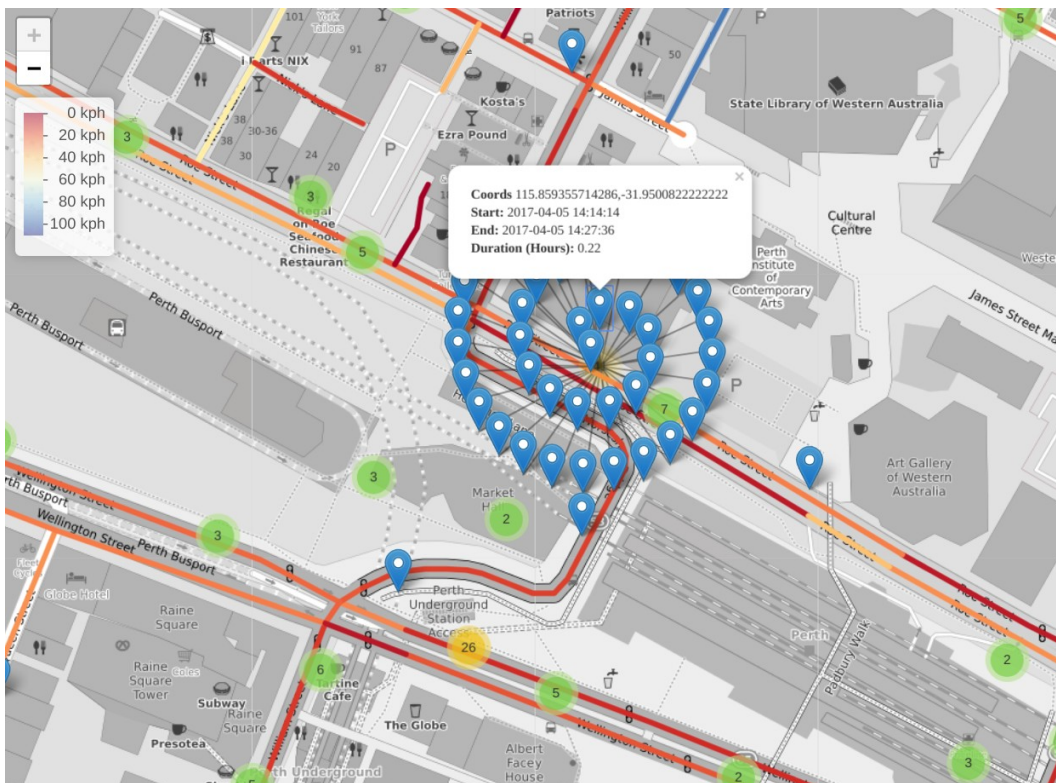


Figure 7: Yulo stops view

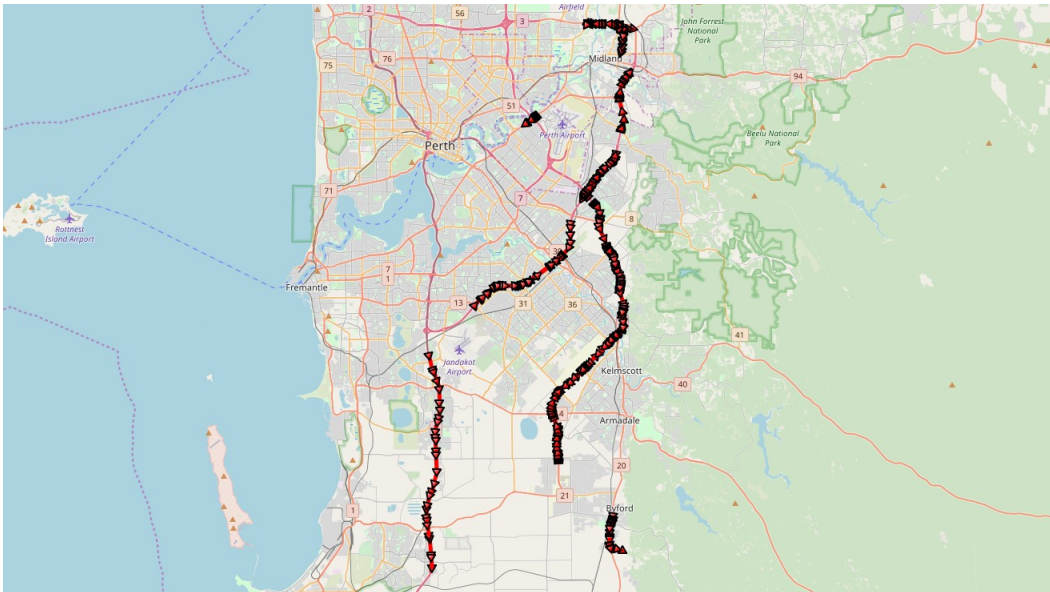


Figure 8: Critical trip components for a building company

**Future Work:** The existing views for aggregate data could be extended in several ways.

By applying a lower bound threshold to filter the Number of Trips view, companies could identify the most significant road segments for their fleet. This information on the 'most used' or 'most critical' road segments for the member's business could be used to inform the member's road development advocacy strategy. Additionally, it would provide evidence on the demand for these segments to support the advocacy.

Critical components are sub-graphs of the road network comprising sequences of connected segments. The sub-graph can be a single path or include branching roads. The idea of a critical component is to provide a greater insight into how the member's vehicles use the road network when it spans many segments. An example of advocacy activities based on critical components is the improvement of traffic flow at certain congested intersections or on/off-ramps. Figure 8 shows a zoomed section of a proof of concept to show the top-20 (directional) paths of length-20. This proof of concept shows the most travelled, and therefore most critical, 20-segment paths that the vehicles in a member's fleet travels on.

Western Australia's Restricted Vehicle Access rules define which road segments may (or may not) be used by different types of freight vehicle. Filtering road segment usage views for a subset of the vehicle fleet would allow monitoring of the level of compliance with road use restrictions.

Sums of all travel times and stop times for a given vehicle across a fixed time period, can be used to report the utilisation of individual vehicles.

The distribution of all stopping durations at each stop can be used to classify the type of stop: short stay delivery, long term parking, or rest stop. The distribution of the times of day of stops can be used as an indicator of the stop utilisation.

## 4.2 Single Trips

Goal: Given a single trip as a sequence of road segments leading from an origin to a destination location, calculate trip metrics to characterise that trip and display the trip on a map. The original Yulo Framework displayed only aggregated trip and stop information. We have written a Jupyter workbook to visualise a single vehicle trip. Figure 9 shows two single trips with a start point (green marker), trip route (blue lines) and end point (red marker). Figure 9 (right) has some additional 'overhangs' or 'dog legs' on some of the trip roads. This is caused by the OpenStreetMap road network map. The map comprises a set of road segments, each of which is a list of GPS points. Some road segments cross over intersections. Yulo Barefoot assigns a vehicle GPS points to road segments, creating a list of segments visited. The visualisation colours the whole map segment, even though a vehicle may only have travelled on part of that segment. These extra sub-segments can be seen as overhangs on some parts of the route. A solution to this problem is described in Section 4.3.

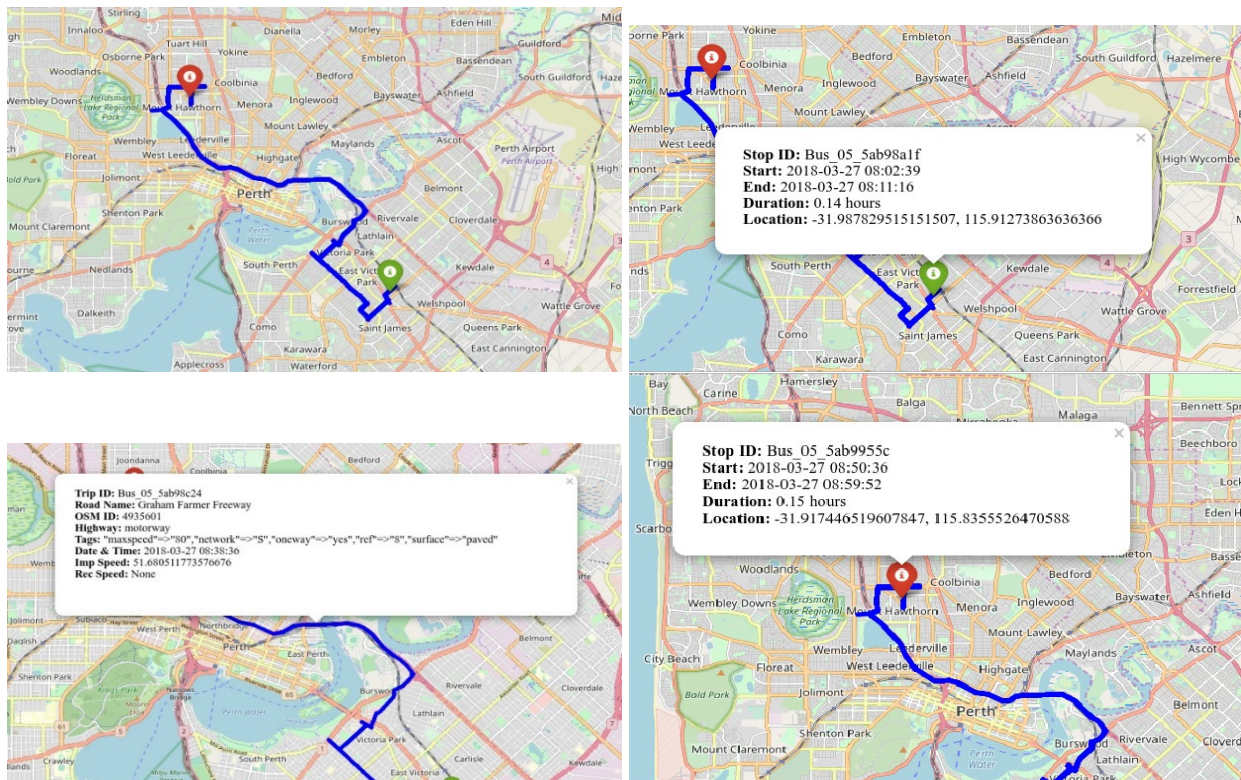


Figure 9: Single trip visualisations

Figure 9 shows different views of a single trip, including the route itself and the trip information that is displayed when the user clicks on parts of the trip. Clickable trip information includes: exactly where trip started and how long the vehicle was stopped there before starting this trip; the exact route taken by the vehicle, including road segments travelled on, the imputed speed for each road segment; where the trip stopped, and for how long until the vehicle began its next trip.

## 4.3 Map Smoothing

Maps are an important part of freight tracking systems. They are used to process the inputs and for displaying outputs. Yulo uses a standard representation of a map as a shape file. However, using existing shape file maps leads to several problems. This section explains some problems that arise with existing maps, and discuss ways to transform those input maps into more usable maps.

Goal 1: Given a map represented as a set of segments, each a sequence of GPS points, generate a new map

as a set of mini-segements so that, no mini-segment crosses over an intersection; and the set of all mini-segments covers all roads in the original map.

Goal 2: Given a map represented as a set of mini-segments, each a sequence of GPS points, generate a (weighted) directed graph representation of the map in which vertices represent intersections and each edge represents a road segment. Optionally, the graph map may include edge weights representing a cost value for travelling on that segment.

**Dog Legs:** A problem we encountered is that of dog legs when a freight trip is displayed on a map. Figure 9 illustrates the problem of these extra segments in a route map. As explained above in Section 4.2 these dog legs are an artifact of the way OpenStreetMaps represents road segments. In this project, we implemented a software module to dissolve a given map (such as OpenStreetMaps) into a tidier map. Each of the OpenStreet map segments (list of GPS points) are recast as 'mini-segments' that span only from intersection to intersection. See Section 5 for more information about how the project team created this map. This mini-segment map provides clean visualisations for short-haul, urban trips.

**Directed Graph Maps:** Applications such as route planning (see Section 4.4) rely on a map represented as a (weighted) directed graph. A mini-segment map can be used for this purpose, where every GPS point is a vertex and each pair of GPS points in a mini-segment sequence forms an edge. However, there are many advantages to using a higher level graph, meaning one with fewer edges and vertices. Most importantly, the space and time requirements of routing algorithms are proportional to the number of edges and vertices in a graph, and so high level maps are much more efficient to process. A high level graph model can be derived from a mini-segment map by setting the vertices of the graph to be the set of all intersections in the original map. A (directed) edge between two vertices is added to the graph whenever there is a (directed) mini-segment between the two intersections.

**Merging maps:** Some applications in freight trip analysis may require more than one map shape file as input. This situation arises, for example, when using the Restricted Access Vehicle maps [RAVrefneeded] to identify vehicle access to certain roads, alongside OpenStreetMaps for other roads. Unfortunately, the shapes used to represent road segments by different map providers are not perfectly aligned. Technically, this problem can be solved using algorithms for graph merging [9, 6]. Graph merging is a complex problem that we plan to address in future work. We have considered two options.

For the first option, each shape file map is converted into a smooth mini-segment map and so to a graph map representation as described above. Then graph merging algorithm is applied to create a single graph from the two. The problem will be a constrained version of the general graph merging algorithm, since representations of intersections (vertices) in the underlying map models, should be very close, and so a standard nearest neighbours clustering algorithm can be used to match vertices in conjunction with some basic metadata matching (eg by matching road name data that exists in each shapefile). Once the vertices are matched, the edges of the super graph are simply the union of all edges in the component graphs.

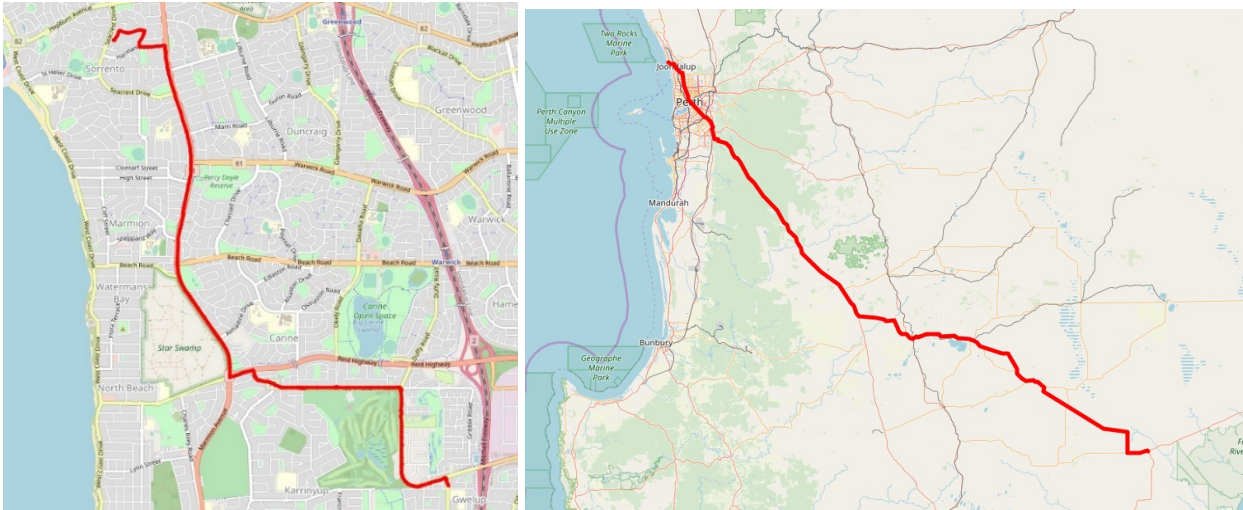


Figure 10: Examples of shortest routes

A second option would be to map the GPS points in one map onto the second map using the existing Yulo Barefoot tool. Road segments are represented as line geometries in each shapefile. Therefore each road segment in the shapefile containing the subset of road segments (to be matched onto the shapefile superset of segments) could be thought of as a short 'trip' of latitude, longitude points. This 'trip' could be run through the Yulo Barefoot tool, with each point being matched to a segment on the superset shapefile. Initial testing with this option has shown some matching errors to the superset shapefile segments (possibly due to the short nature of the subset 'trips') but the project team believes more investigation into this method is required to determine its overall potential.

#### 4.4 Shortest Path Routes

An important task for fleet management is optimising the selection for routes taken by freight vehicles [20, 10, 3, 13, 15, 12]. This problem can be solved when the input map is provided as a graph map as described in the previous section.

Goal: Given origin and destination locations and a map represented as a weighted directed graph, calculate the route of the shortest trip from the origin to destination as a sequence of the road segments to be travelled.

The shortest path problem is to calculate the "best" route from one node of a weighted graph to another. Each road segment is associated with a weight that represents the cost of using that road segment. Common costs include the expected speed on that segment, length of the segment, or other cost indicators such as expected congestion, vehicle restrictions, or real-time local traffic information on congestion, road works or accidents. The shortest path problem is a well known problem in computer science and there are several classic algorithms for solving it [5]. Although the problem is, in general computationally expensive, there are many efficient heuristics for finding the shortest path [7, 14]. We have implemented one of the best known algorithms, Dijkstra's algorithm, and tested it on our data sets.

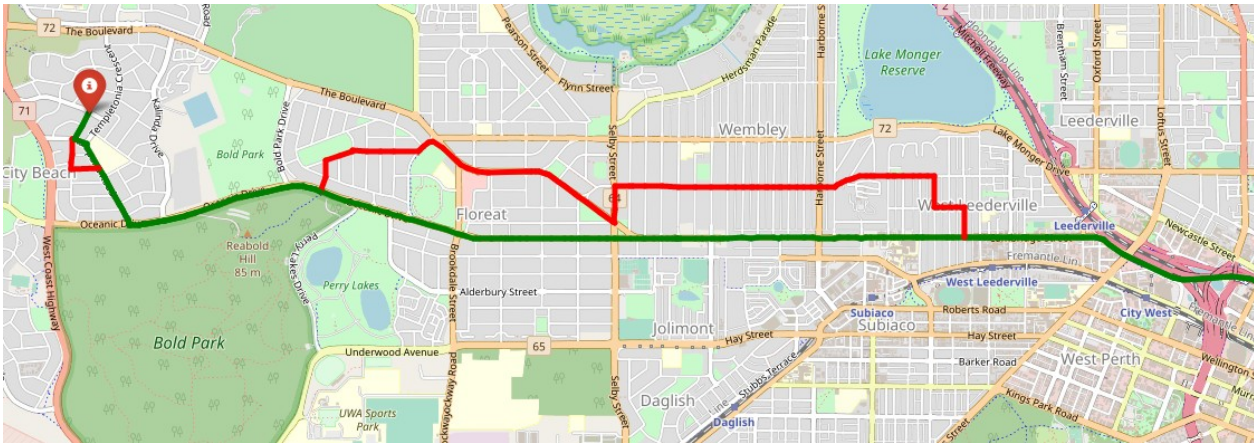


Figure 11: Example of optimal route vs actual route taken

Figure 10 shows the output of our shortest path program. The shortest trip is determined by travel time based on road speed limits and distance. The left hand Figure is the best route between an address in Gwelup and an address in Sorrento. The right hand Figure shows a longer trip calculated from Jerramangup to an address in Burns Beach. Note that optimal routes generated can also be compared against actual routes taken as well. Figure 11 visualises the comparison of an optimal route (in green) vs the actual route (in red when the vehicle deviates from the optimal route) taken.

**Future Work:** A useful extension to the simple shortest path algorithm would be to add more ways of weighting the cost of each road segment [5]. Then best routes could be targeted to local conditions (time of day, day of week), business-specific factors (eg restricted access or height-restricted vehicles). The main task to support this goal is to learn information about the weight of each of the 1000s of road segments in the map. There may be opportunities to discover weights from existing data sets such as Open Street Maps, Google Maps, Bing Maps, or TomTom. Another approach would be to learn weights such as average speeds from a freight dataset using Yulo as described in the previous sections. For road segments that are critical to a business, the effort of learning accurate weights for routing would be repaid by the opportunity to plan more efficient utilisation of the fleet.

Our work considers the problem of routing a single vehicle between a given origin and destination location. The more general problem of planning a loading and travel schedule for a fleet of vehicles in order to minimise the total fleet costs for a known set of deliveries is outside the scope of our work.

#### 4.5 Future Work: Freight Fleet Flows

Goal: Given an origin and destination matrix showing number of trips to be taken between each possible origin and destination pair, calculate the shortest path route for each pair, and so generate the number of vehicles using each road segment. The map is modelled as a weighted directed graph in which vertices represent intersections, each edge represents a road segment, and edge weights represent the cost of travelling on that segment. Additionally, there may be multiple weights and different routing policies may be possible.

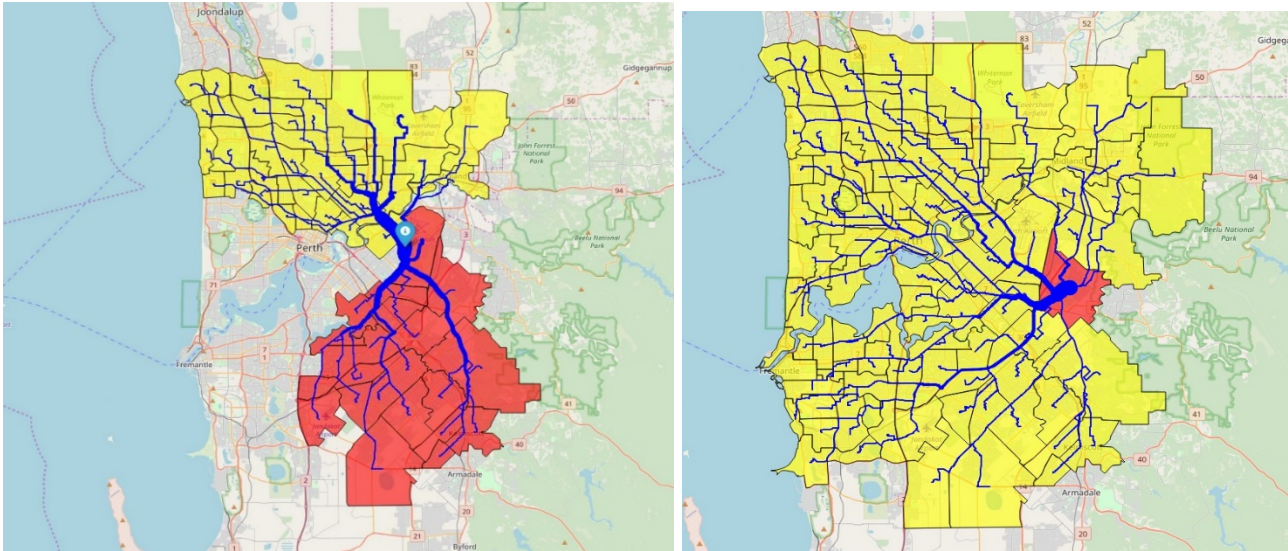


Figure 12: Freight fleet flows

Figure 12 shows two proof of concept maps representing the number of freight movements for a fleet of vehicles on different road segments. For this example, we generated a sample origin-destination matrix with random numbers of trips between each origin-destination pair. For this example the origins and destinations were centroids of SA2 zones. The shortest path algorithm described in the previous section was used to plan routes for each pair, which gives the total number of vehicles using each road segment. In this case, the edge weights used for routing were 'fastest journey' between the centroid (closest road segment to the centre) of one SA2 region to another. Figure 12 (left) selects all freight flows include the northbound carriageway of the Tonkin Highway near the Perth Airport between SA2 regions in the Perth Metro area. Routes that did not use this part of the road network are not shown. In this instance the freight flow volumes between SA2 regions were chosen randomly, but with sufficient real trip data these flows could be extracted from the Yulo back-end database. Alternatively, the flow volumes could be generated from a freight demand model. Figure 12 (right) shows the weighted flows originating from an imagined freight terminal in the Forreestfield SA2.

---

## 5 PROJECT OUTPUTS

This section lists the software modules written for this project.

**Travel Direction on Roads:** Many urban roads in the OpenStreetMap road network database are only represented as one series of points servicing two directions. This is in contrast with many highway, freeway and dual-carriageway roads in this same database that are represented as two series of points, one series for each direction. For short-haul freight analysis activities, the project team fast concluded that Yulo would have to understand and track which direction a vehicle is travelling on for any two-way roads that are only represented as on series of points in this database.

When this issue was raised with BITRE, their Yulo team immediately updated the Barefoot backend to produce a direction that a vehicle was travelling on each OpenStreetMap road segment. With Barefoot now producing this information, the project team updated the Neo4j upload scripts so that this information was saved into the database and made available for further analysis.

The updated upload script will be made available to BITRE for merging into their latest Yulo version.

**Python Folium Maps:** A large part of the functionality in Yulo's aggregation front-end was written in ShinyLeaflet in R. To speed up the prototyping of potential Yulo additional features, certain parts of this front-end were re-written in Python using the Folium module. Advantages of Python and the Folium module include availability of more efficient inbuilt data structures (eg Python dicts), superior error-checking functionality and faster potential prototyping through the use of the Jupyter IDE, which can display generated interactive maps inline immediately instead of saving to html file or running a pseudo-webserver.

This front-end partial re-write is available as a Python script as part of this project.

**Saving Maps:** We implemented a 'download' button for the ShinyLeaflet front-end. This allows interactive maps generated by this front-end to be saved as standalone html files that can then be shared with stakeholders via a simple email attachment. The interactive maps contain most of the functionality as the existing Yulo front-end maps including stop and trip information, but do not contain the interface on the right that allows for changing the road segments view (eg from average speed to number of trips), though this can be mitigated by saving a map of each of these views and sharing them all as an archive, or the graph information per-hour-of-the-day or per-day-of-the-week (as this is pulled from the Neo4j database when requested). The project team found that this allowed stakeholder firms to receive maps and files of their aggregated data that they can browse and interpret on their own time, which in turn led to more excitement around and engagement with the project.

This contribution takes the form of updated 'server.R' and 'ui.R' ShinyLeaflets script that contain this additional functionality. It will be made available to the BITRE Yulo team for merging into their latest version of this script.

**Development of a Mini-Segments OpenStreetMap Road Network to Assist with Short Trip Visualisation:** We found that road segments in the OpenStreetMap database are usually longer than 'smallest-logical-units.' A 'smallest-logical-unit' in a road network sense is a segment that spans from intersection to intersection. Having segments with this property would mean that, when a vehicle travels on this segment, it will travel the entire span of the segment (unless it starts or stops its trip on the segment), and hence its true route is possible to be 'nicely' mapped as a list of these segments. Conversely, having segments without this property (as in OpenStreetMap segments) means that if a trip is mapped as a list of such segments then the visualisation of this trip will contain segment 'dog legs' and overhangs (as well as inaccuracies in ancillary calculations such as distance, average speed, etc.) that decrease the user experience and the quality of

insights that can be gained from the data as shown in Figure 9.

As with the 'Directionality' contribution above, segments spanning several intersections is not as much of a problem for visualisation of mapping regional trips. Such trips often travel very large distances, many multiple times more segments and take comparatively many fewer turns than urban trips. However, for mapping of urban trips, the project quickly determined that the OpenStreetMap road segments database needed to be dissolved into a smallest-logical-unit road segments database. The project hereafter referred to this as a 'mini-segments' database.

To achieve this dissolution from OpenStreetMap segments to mini-segments we wrote a Python script that performs the following:

- Accesses the OpenStreetMap ("OSM") segments database (via OSM's Australian shapefile) and creates a new Python empty dictionary.
- Iterates through each OSM segment. For each OSM segment, the program then iterates through each coordinate tuple that is part of the segment's geometry (note that OSM segments' geometries are always Linestrings).
- For each coordinate tuple the program will store it as a key (if an entry for this tuple does not already exist) in the empty Python dictionary. For the value it stores a 1-item list containing the OSM Segment ID. If the tuple already exists as a key in the dictionary then the OSM segment ID is appended to its value list. (in OSM roads segments that intersect, notwithstanding intersections with restrictive turns, will share a common coordinate from their geometry mappings, where as roads that do not intersect (eg if one road passes over another via bridge) then they will not, so this step is building a dictionary of segment IDs that share a coordinate tuple).
- Once all OSM segments and their coordinate tuples are iterated over, the program then performs this iteration again. However, this time, for each coordinate tuple the program checks how many segment IDs are in its value list (except for the first and last coordinate tuples in the OSM segment). Once the program comes across a coordinate tuple with more than one segment ID in its value list, it means that it has reached an intersection in this segment. The program then stores all coordinate tuples with only 1 segment ID (ie the current segment ID) iterated over before this coordinate tuple into a new dictionary with a unique mini-segment identifier (a variable that starts at 1 and is incremented whenever a new mini-segment is identified) as the key, and the list of coordinates (plus the starting coordinate from the last identified intersection) as its value.
- Once the above iteration is finished the program saves the mini-segment dictionary information into a new 'mini-segments' shapefile. Some additional metadata (eg corresponding OSM ID to the new mini-segments ID) is also saved in this shapefile.

This mini-segments shapefile is likely compatible with Yulo's Barefoot back-end and hereafter vehicle trips could be mapped directly onto these mini-segments rather than the OpenStreetMap segments database. A mapping of mini-segment to the OpenStreetMap segment could be provided on the front-end visualisation to encourage data consistency. We recommend the implementation and testing of this mini-segments shapefile within the Yulo framework as a future work activity.

The above "mini-segments" database does not capture information such as restrictive turns which is needed for routing applications. Incorporation of this metadata into the mini-segment shapefile is recommended for future work.

**Development of a weighted routing algorithm:** To support the testing and analysis performed in section 4.4

of this report the project implemented a routing calculation and visualisation program that can utilise the OpenStreetMap road segments database or the mini-segments database outlined above. This program is written in Python and implements the well-known Dijkstra's algorithm to determine optimal routes. Routes are nominally determined the route with the minimum sum of segment weights. Segment weights used were easy-to-calculate, common existing routing weights, eg distance of segment (which can be determined using the Python library Shapely) or theoretical time of travel (ie segment distance divided by posted speed limit - the latter of which is freely available for most road segments from OpenStreetMap).

Note that this program could easily be extended to derive weights from more complex single-objective or multi- objective inputs. For example, freight-specific segment-time-of-travel weights could be derived by using Yulo aggregated data for average speed of that road segment, and further refined by using only the aggregated data corresponding to the time-of-day and day-of-week that the route is planned for.

Once the route is derived the program will visualise it on a Folium map that is saved to the user's hard drive as a html file.

This program will be made available as part of the project.

**Additional Scripts:** The following program scripts are currently under development.

- Shapefile merging scripts - The project has explored and written a preliminary shapefile polyline matching script in order to match the OpenStreetMap mini-segments shapefile with various Restricted Access Vehicle network shapefiles. This script is written in Python and can produce partially merged, but not yet fully merged, shapefiles.
- Periodicity mining script - The project has written a preliminary script to identify trips that travel on repeated road segments in the last five minutes of a vehicle's trip. This script was used to explore whether an indicator that shows time vehicles waste looking for parking could be produced. The script is written in Python. The script successfully identifies trips with repeated road segments in the last five minutes, however it was determined that a 'looking for parking' activity would likely require more complex analysis than only looking for travel on repeated road segments.
- Top k-paths of length-p script - To support bullet point 2 under the Future Work in section 4.1 the project developed a preliminary script that visualises and returns the top k-paths of length-p of all data in Yulo's back-end. If only one company's data is loaded into Yulo then this script visualises and returns the most critical road paths of length-p to this company's business. This script is written in Python.

---

## 6 STUDENT PROJECTS

This section summarises three related student projects by Masters of Professional Engineering students that are running from February to October 2019. The first two of these projects have been awarded iMove studentships.

**Congestion evolution by Alex Fletcher (Masters of Professional Engineering):** Road congestion in cities leads to longer journey times and increased pollution. Although many systems (such as Google Maps and Tom Tom) can identify and report areas of congestion in real time, there is a lack of understanding about how congestion evolves over time. Better decisions about journey routing (short term) and road planning (long term) could be made with knowledge about the way congestion propagates through the road network, based on measured road speeds that can be obtained from vehicle GPS data. This project evaluates a

congestion tracking algorithm proposed by Anwar [2, 1] using GPS fleet data from Perth-based companies. The congestion tracking algorithm will be implemented around the Yulo system and a visualisation layer will be developed to allow users to see how congestion evolves. This project is supported by an iMove studentship.

***Economic Indicators from GPS Freight Traces by Johnson Lim (Masters of Professional Engineering):*** Freight movements can be used as a proxy for various economic indicators of business activity. But historically freight movement data needed to be gathered from expensive, and rare, surveys. Fleet GPS systems offer the potential to generate up to date indications from real freight movements. This project reviews existing work on economic indicators from freight (such as [11]). It will build on the Yulo system to calculate and report indicators such as vehicle and stop utilisation. This project is supported by an iMove studentship.

***Multi-objective Route Choices for Bike Riders by Daniel Tizoc Gonzales Esquer (Masters of Professional Engineering):*** Active transportation, even in moderate amounts, is thought to improve public health and psychological well-being and reduce obesity rates. Increasing the number of bicycle trips used for leisure and commuting also has the beneficial side-effect of reducing vehicle congestion on the roads. However, studies have identified many barriers that prevent travellers changing to cycling. For example, some of the main barriers are the distance and elevation of a trip, not having a bicycle, the confidence or sense of security, and the need to carry paraphernalia or impedimenta. These impediments can be mitigated by providing better information for travellers about cycling options, so enabling individuals to tailor a trip to their specific needs [16]. To this end, this project investigates a multi-objective route planning service for cyclists. The project will use web sources such as Open Street Maps, Main Roads WA and information from bicycle clubs such as the bike incident web site in order to develop a routing graph map and populate various weights for its edges. Users will be able to state their preferences for a route, and the system will then search for and report the best route to meet those objectives.

---

## 7 CONCLUSION AND FUTURE WORK

This project successfully tested the Yulo framework, written by the Bureau of Infrastructure, Transport and Regional Economics (BITRE), for its potential in analysing short-haul, urban freight activity.

We demonstrated that Yulo is able to succinctly analyse and visualise aggregated and single trip information that could be used in various policy advocacy scenarios. As part of the adoption of the framework from regional activity analysis to urban / short-haul freight analysis the project made the following contributions:

- Re-implementation of most of the existing front-end into Python to facilitate faster prototyping.
- Creation of a directed road network 'mini-segments' shapefile for better visualisation of urban routes.
- Implementation of directionality of travel on road segments in the Neo4j back-end.
- The ability to save interactive maps as stand-alone files for sharing with stakeholders.
- Development of a weighted routing algorithm.
- Ongoing work on shapefile merge, periodicity mining and top-k paths of length-p algorithms.

During the project, meetings were held with many freight companies in Perth. Given their input and our analysis, the following topics are recommended for future work.

- Visualisation and modelling of congestion evolution (see Section 6)
- Economic indicators including vehicle utilisation and stop time analysis (see Section 6)

- Multi-objective optimal route planning (see Section 6)
- Visualisation of freight fleet flows between defined regions
- Identification of most significant road segments for a specific business
- Restricted Access Vehicle network compliance evaluation and route planning
- Stop categorisation
- Freight fleet flows

---

## 8 REFERENCES

- [1] Tarique Anwar et al. "Capturing the Spatiotemporal Evolution in Road Traffic Networks". In: *IEEE Transactions on Knowledge and Data Engineering* 30.8 (2018), pp. 1426–1439. issn: 10414347. doi: 10.1109/TKDE.2018.2795001.
- [2] Tarique Anwar et al. "Temporal Tracking of Congested Partitions in Dynamic Urban Road Networks". In: *Transportation Research Record: Journal of the Transportation Research Board* 2595.1 (2016), pp. 88–97. issn: 0361-1981. doi: 10.3141/2595-10.
- [3] Moshe E. Ben-Akiva et al. "Freight data collection using GPS and web-based surveys: Insights from US truck drivers' survey and perspectives for urban freight". In: *Case Studies on Transport Policy* 4.1 (2016), pp. 38–44. issn: 22136258. doi: 10.1016/j.cstp.2015.11.005.
- [4] BMW. Barefoot Github Repository. url: <https://github.com/bmwcarit/barefoot>. (accessed: 18.03.2019).
- [5] Adi Botea, M Müller, and Jonathan Schaeffer. "Near optimal hierarchical path-finding". In: *Journal of game . . .* (2004), pp. 1–30. url: <http://abotea.rsise.anu.edu.au/data/hpastar.pdf>.
- [6] Horst Bunke. *Recent Developments in Graph Matching*. 2000.
- [7] C Chitra and P Subbaraj. "Multiobjective Optimization Solution for Shortest Path Routing Problem". In: *Signal Processing* 4.1 (2010), pp. 77–85.
- [8] iMove CRC Project Agreement. *Planning intermodal and general logistics infrastructure for the future needs of Perth*. 2018.
- [9] M. Ferrer, E. Valveny, and F. Serratos. "Median graph: A new exact algorithm using a distance based on the maximum common subgraph". In: *Pattern Recognition Letters* 30.5 (Apr. 2009), pp. 579–588. issn: 0167-8655. doi: 10.1016/J.PATREC.2008.12.014. url: <https://www.sciencedirect.com/science/article/abs/pii/S0167865509000026>.
- [10] Kun He et al. "Congestion Avoidance Routing Based on Large-Scale Social Signals". In: *IEEE Transactions on Intelligent Transportation Systems* 17.9 (2016), pp. 2613–2626. issn: 15249050. doi: 10.1109/TITS.2015.2498186.
- [11] Zach Johnson. "Developing Freight Performance Measures Using GPS Truck Data". In: *Trb* (2014).
- [12] Zhongwei Li, Guangtao Xue, and Hongzi Zhu. "Traffic-Known Urban Vehicular Route Prediction Based on Partial Mobility Patterns". In: *2009 15th International Conference on Parallel and Distributed Systems*. IEEE, 2009. doi: 10.1109/ICPADS.2009.129. url: <https://ieeexplore.ieee.org/abstract/document/5395296>.

- [13] Xiaolei Ma et al. "Understanding Freight Trip-Chaining Behavior Using a Spatial Data-Mining Approach with GPS Data". In: *Transportation Research Record: Journal of the Transportation Research Board* 2596 (2016), pp. 44–54. issn: 0361-1981. doi: 10.3141/2596-06. url: <http://trrjournalonline.trb.org/doi/10.3141/2596-06>.
- [14] Jos'e Maria, A. Pangilinan, and Gerrit K. Janssens. "Evolutionary algorithms for the multiobjective shortest path planning problem". In: (2007), pp. 54–59. url: <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=69972D897A117FDB255B001B150DB23A?doi=10.1.1.193.4446%7B%5C%7Drep=rep1%7B%5C%7Dtype=pdf%7B%5C%7D5Cnhttp://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.193.4446>.
- [15] Sabya Mishra. *CFIRE A Guidebook for Freight Transportation Planning using Truck GPS Data*. Tech. rep. 2016. url: [http://www.wistrans.org/cfire/documents/CFIRE%7B%5C\\_%7D0904%7B%5C\\_%7DFinalReport.pdf](http://www.wistrans.org/cfire/documents/CFIRE%7B%5C_%7D0904%7B%5C_%7DFinalReport.pdf).
- [16] Walter Musakwa and Kadibetso M. Selala. "Mapping cycling patterns and trends using Strava Metro data in the city of Johannesburg, South Africa". In: *Data in Brief* 9 (2016), pp. 898–905. issn: 23523409. doi: 10.1016/j.dib.2016.11.002. url: <http://dx.doi.org/10.1016/j.dib.2016.11.002>.
- [17] OpenStreetMap. url: <https://www.openstreetmap.org/>. (accessed: 18.03.2019).
- [18] Green R and Mitchell D. "Adapting truck GPS data for freight metrics". In: *Australasian Transport Research Forum 2018 Proceedings*. 2018. (available from: <http://www.atrf.info>).
- [19] Australian Bureau of Statistics, Bureau of Infrastructure Transport, and Environment. "ABS-BITRE road freight telematics data pilot project - key findings". In: (2016). (available from: [https://bitre.gov.au/data\\_dissemination/priority\\_projects/road\\_freight\\_telematics\\_data\\_collection.aspx](https://bitre.gov.au/data_dissemination/priority_projects/road_freight_telematics_data_collection.aspx)).
- [20] Jing Yuan et al. "T-drive: Enhancing driving directions with taxi drivers' intelligence". In: *IEEE Transactions on Knowledge and Data Engineering* 25.1 (2013), pp. 220–232. issn: 10414347. doi: 10.1109/TKDE.2011.200.

## APPENDIX B



Planning and Transport Research Centre (PATREC)

# Planning intermodal and general logistics infrastructure for the future needs of Perth:

## Telemetry Systems for Tracking Road Freight Activity – Application using Escort Vehicle GPS Data

Prepared by	Tim Hoffman, Daniel Cowen and Rachel Cardell-Oliver
Date	December 2019
Project details	iMOVE Project 2.001, Report no. 7.1
Version	Final



THE UNIVERSITY OF  
WESTERN  
AUSTRALIA



---

# 1 INTRODUCTION

## 1.1 Project aims and background

This project aims to demonstrate the potential application of a customised route data aggregation software to road freight analytics. Road freight activity is notoriously difficult to measure and analyse, due to the absence of any systematic activity reporting or data collection obligations on road transporters or users.

Many road transport firms now routinely use GPS telemetry systems to track the real time location of their vehicles, and to record vehicle and driver performance in some very detailed measures. As yet, however, there is no regulatory requirement on any such data to be kept or submitted for any policy, research or regulatory purposes. Most transport firms only use a small amount of data potentially available from this technology, and have no incentive to make any of it available for any external purpose.

Consequently, it is difficult to obtain data from transport companies for any research purposes. During the course of this research project in 2019, we have developed a robust analytical tool, which is able to aggregate and interpret large volumes of GPS telemetry data, and to generate accurate route maps and analysis of speed and usage density parameters.

The system developed for this project is based on the YULO platform, developed recently by the Bureau of Infrastructure, Transport and Regional Economics (BITRE). This platform has been enhanced with some additional functionality by staff at UWA's Department of Computer Science and Software Engineering, and is provisionally named SWAN YULO.

Despite several promising meetings with freight transport companies, only one was willing and able to provide a suitable dataset to allow testing and demonstration of the SWAN YULO system. Additionally, a comprehensive set of data from a non-freight transporter was also used for the same purpose.

Main Roads WA then offered to make available a set of GPS data from its management system for the operation of Over-size Over-mass (OSOM) escort vehicles. This data is collected for the operation of the entire fleet of 30 escort vehicles, and generally used for internal management purposes (eg driving hours compliance and payroll purposes).

A set of six months data for the entire escort fleet has been made available, and has been treated as a source of secondary data on the use of WA roads by a subset of the freight industry – OSOM permitted vehicles carrying extra-ordinarily large loads.

This data set is large enough to thoroughly test the performance of the software and to establish the potential for its future use in targeted road planning and asset performance measurement. More significantly, in the near to medium term, road user charging initiatives are likely to replace current vehicle registration charging systems, in response to increasing urban road congestion. State government agencies will need to develop programs for the collection and analysis of GPS data from a variety of road users. This research program and the MRWA escort vehicle data might be valuable as a means of furthering research into this emerging policy field.

## 1.2 The SWAN YULO system

The software used in this project combines the YULO route tracking methodology with an open-source mapping program in order to match geo-coded, timed, GPS location records with roads identified in the mapping program. The program uses a set of algorithms to calculate the most likely route taken by a vehicle

from the provided set of GPS pings for each vehicle trip (trips are also identified using the provided GPS location records), which allows maps of a vehicle’s most likely route to be shown.

The SWAN YULO system improves on the mapping presentation, and also provides a facility for counting the number of vehicles and / or average speeds recorded using specific road segments or sections within a database, and demonstrating intensity of use via flow diagrams or colour. Some of this functionality may be available within commercial telemetry program systems such as NAVMAN, but is not generally by transport fleet managers; roads authorities are more likely to be interested in the traffic volumes on a road network. Unfortunately, however, roads authorities do not have access to much telemetry data to track the movement of freight vehicles.

### 1.3 MRWA escort vehicle Navman data set

MRWA Heavy Vehicle Services Division is responsible for the administration of permits for large dimension loads on the WA road network. Certain types of very large loads require official escort along the length of the route via departmental vehicles.

The division has 24 escort vehicles, of which 23 are currently based in Perth, with one in Port Hedland. (The number in Port Hedland will be increased shortly). A GPS-based fleet tracking system (designed by Teletrac Navman) is used to record the locations of all escort vehicles at all times when they are in operation. The system includes a fleet of GPS units permanently attached to all vehicles, which send latitude/longitude ‘pings’ a server located at the Division’s Redcliffe headquarters. Pings are collated into a geographical record of each journey by an escort vehicle.

The main use currently made of this system is to support the administration of the escort function, confirm compliance as per issued permits, and to record driver activity for payroll, staffing conditions and road safety compliance purposes.

The data compiled in the system, and customised for this research, includes the fields shown in this edited sample:

Vehicle	Event Type	Event Date Time	Speed	Odometer	For	Driver	Location	Latitude	Longitude
2001-1QCB538	Logon OK	1/07/2019 6:13:41	16.0 km/h	13756.9 k	0 min	Janice	[HVS], 533 Great Eastern	-31.92459	115.95391
2001-1QCB538	Driving	1/07/2019 6:13:42	20.0 km/h	13756.9 k	0 min	Janice	[HVS], 533 Great Eastern	-31.92455	115.95395
2001-1QCB538	Timed Update	1/07/2019 6:18:42	68.0 km/h	13760.0 k	0 min	Janice	Great Eastern Hwy Bypass,	-31.91575	115.97295
2001-1QCB538	Timed Update	1/07/2019 6:23:42	60.0 km/h	13764.1 k	0 min	Janice	160 Lakes Rd, Hazelmere WA	-31.91482	116.01479
2001-1QCB538	Timed Update	1/07/2019 6:28:42	80.0 km/h	13769.2 k	0 min	Janice	State Route 3, Middle Swan	-31.87193	116.02838
2001-1QCB538	Timed Update	1/07/2019 6:33:42	71.0 km/h	13775.0 k	0 min	Janice	736 Great Northern Hwy,	-31.82887	116.01658
2001-1QCB538	Timed Update	1/07/2019 6:38:42	0.0 km/h	13780.3 k	0 min	Janice	Yagan Bridge, Upper Swan	-31.78095	116.02231
2001-1QCB538	Timed Update	1/07/2019 6:43:42	0.0 km/h	13783.0 k	0 min	Janice	1455 Great Northern Hwy,	-31.76377	116.03094
2001-1QCB538	Timed Update	1/07/2019 6:48:42	0.0 km/h	13783.0 k	0 min	Janice	1455 Great Northern Hwy,	-31.76377	116.03095

For each vehicle the system records timed pings from its GPS unit (once every 5 minutes when the vehicle is in operation) as well as other significant events including ignition start, ignition stop, health check (once every 12 hours that the vehicle is not being used). For each of these pings the system records the date and time of the ping, its latitude and longitude and some additional metadata including speedometer speed, odometer reading and imputed location. Most, but not all, of this information is fed into the YULO framework in order to produce the insights provided in this report.

---

## 2 ANALYSIS

After some processing of the escort vehicle data, the SWAN YULO system provides the capability for many aspects of the trip record to be analysed.

The data set provided by MRWA included all activities of the full fleet of 26 vehicles over a six month period from 1 April to 30 September 2019.

A total of 291,890 pings were recorded, typically at 5 minute intervals. There were 2,316 recorded trips. A trip is defined as a period during which the vehicle was in motion, including any stops (defined as no observable movement outside of a 100 metre radius) of less than one hour duration. (The program allows the temporal and spatial definitions of a stop to be set according to need). There were 2,325 recorded stops (of more than one hour in length). The data set used for this analysis did not include task identification ie the origin and destination of the permitted full journey of the escorted vehicle, but this is assumed to be available (awaiting confirmation).

Since the time and location of each ping can be plotted, it is possible to calculate an imputed average speed for an escort vehicle over the course of a journey. Differences in the imputed speed for a vehicle on rural highway segments can be used to differentiate between trips where a vehicle is engaged in an escort task, versus non-duty trips eg returns to home base.

In the urban setting, the ability to calculate average speeds over a section of highway at different times of the day is potentially useful. (Escorted OSOM vehicles are not permitted to operate within the Perth Metropolitan area during morning and afternoon peak hours.)

---

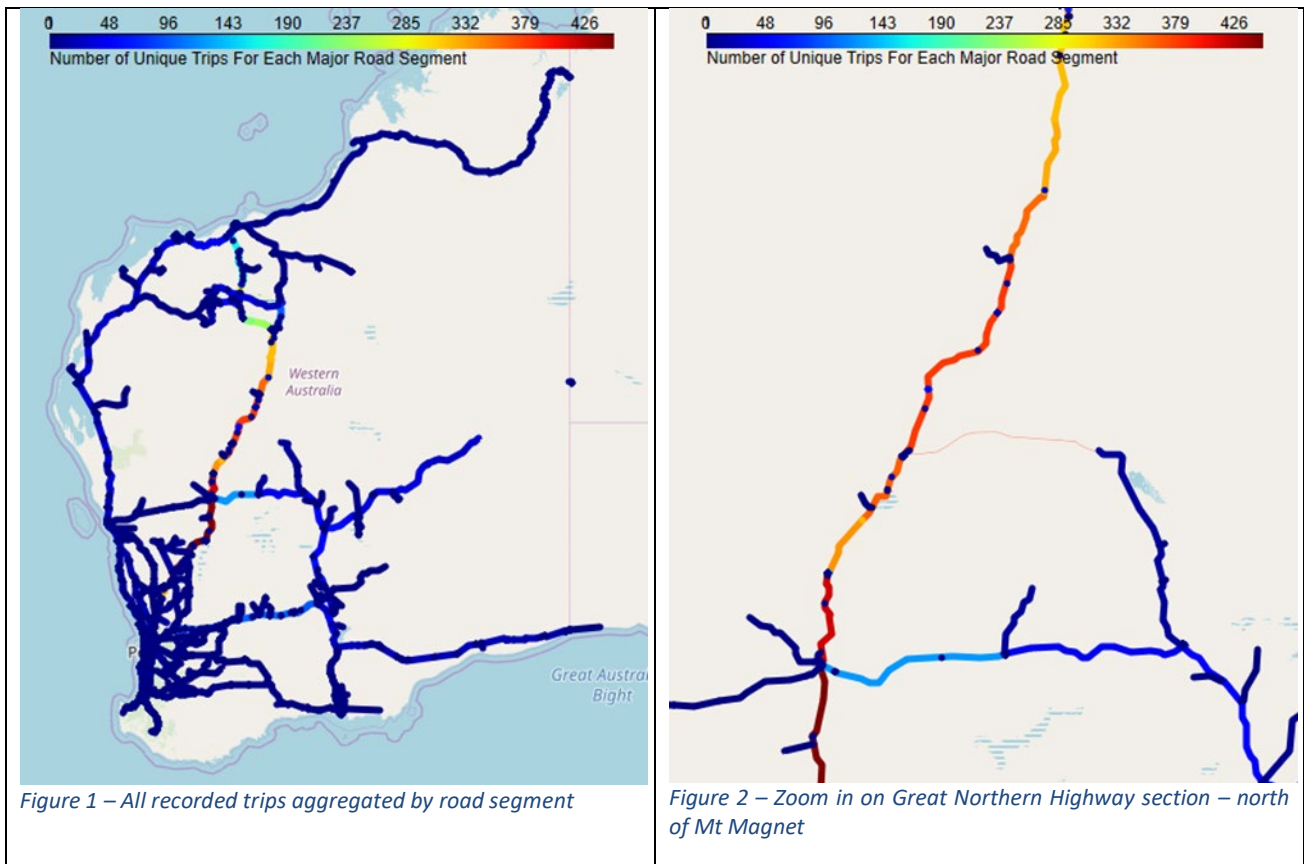
## 3 SAMPLE RESULTS

The data base is quite comprehensive and is conducive to analysis of individual journeys as well as large groupings of journeys. Individual journeys can be examined to determine details about the performance of a particular road section in relation to average truck speed, whereas the larger samples will identify issues such as relative concentration of OSOM permits on all major highway routes.

The best format for viewing route details is in the form of interactive maps, which rely on internet connection to access the underlying geographical context. For this report, indicative snapshots of some of these maps will be shown.

### 3.1 Aggregated trip counts

Examples of maps for aggregated trip data and individual trips follow:



In this data-base, a trip is defined as the passage of a vehicle across a road segment or any part of a road segment. If a long stop takes place during the course of transit across the entire road section, two separate trips will be counted. This accounts for the different colours along various sections of the Great Northern Highway in Figure 2 (above).

The blue dots at various locations along this stretch of highway relate primarily to the use of short road sections joining the highway for small numbers of trips, rather than the highway itself. Some of these records could also be the result of failures by the software to correctly locate the ping with the nearby road/highway.

The number of discrete vehicle one-way trips recorded at various points along the 616km stretch of the Great Northern Highway between Mt Magnet and Newman varies between 328 and 383.

To the east, the number of trips on the Great Eastern and roads in the Goldfields are lower, with around 94-100 discrete one-way trips along the length of the Highway.

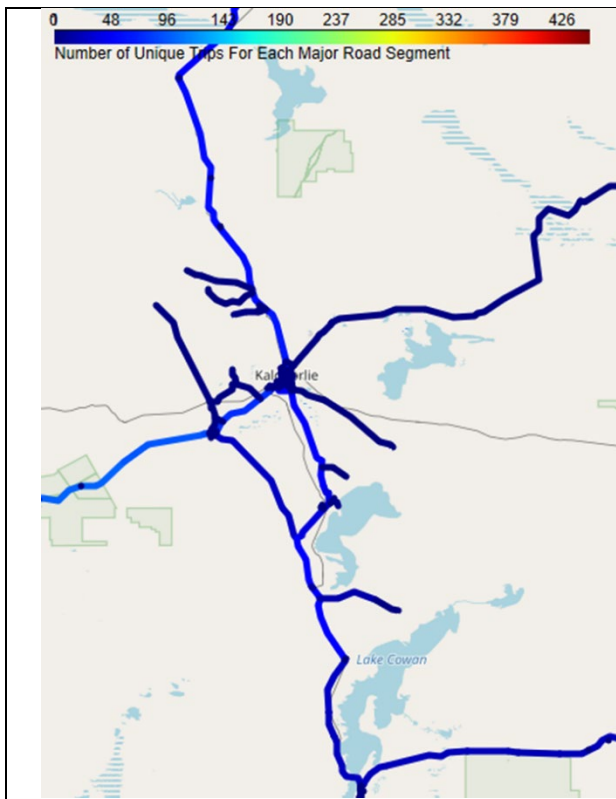


Figure 3 – all recorded trips aggregated by road segment – Goldfields region

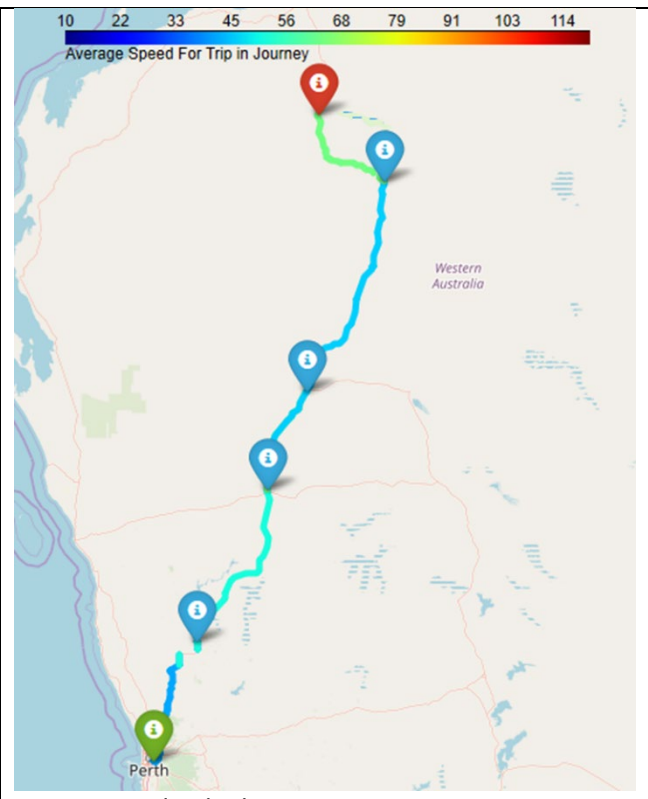


Figure 4 - Individual trip

### 3.2 Individual trip description

This trip took place over a three day period in May 2019, commencing in an industrial Welshpool (Perth) location, and ceasing at a roadhouse on the Great Northern Highway in the Pilbara. The map shows the average speeds attained by the escort vehicle over each section of the journey between major stops. Any stops of less than one hour in duration are not counted as stops, and the duration of any such short stops is not subtracted from the overall time taken, thus affecting the average calculated speed.

The map shows the route taken along the Great Northern Highway, the points at which long stops (more than one hour) were taken (overnight and for lunch break), and average speeds achieved between the stops. The colour bar indicates the average speed for each trip section, and exact records for each road segment can also be accessed if required. Blue markers indicate the location of long stops.

#### Average speeds by section:

Journey section	Average speed (km/h)
Welshpool to Bindi Bindi (near Miling)	42
Bindi Bindi to Mt Magnet	52
Mt Magnet to Cue	67
Cue to Newman	46
Newman to Roadhouse	64

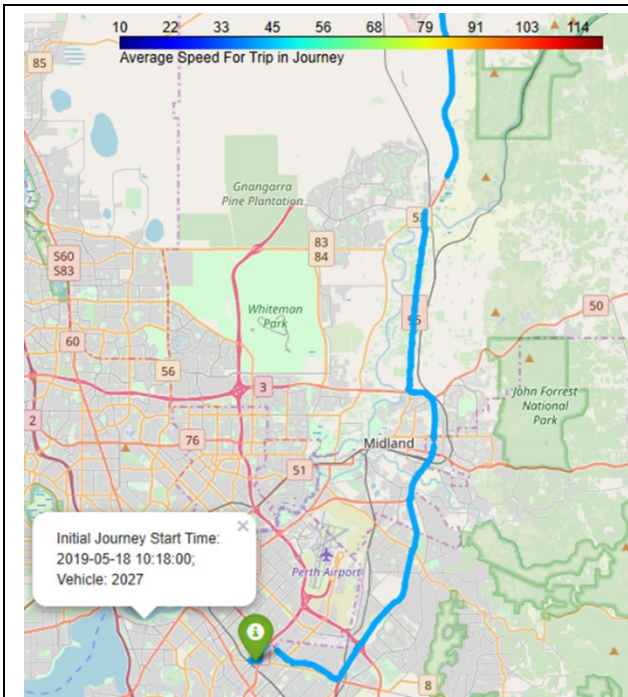


Figure 5 - Trip commencement - Welshpool

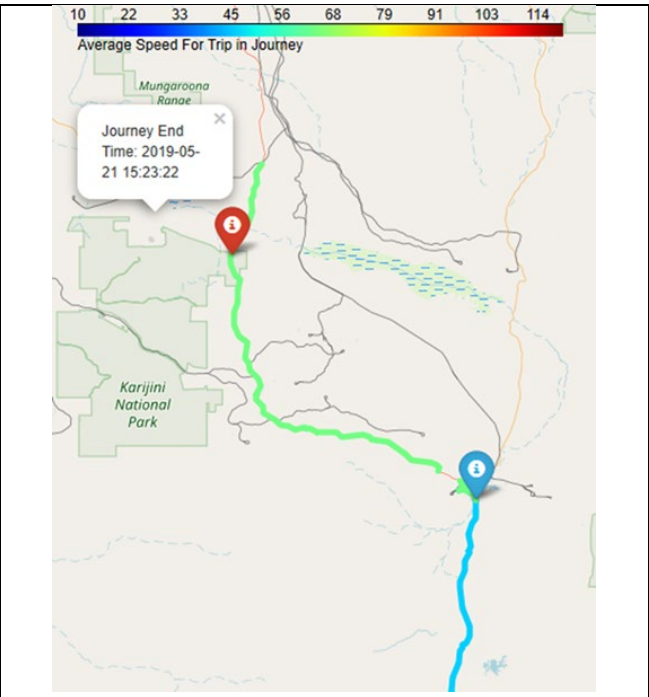


Figure 6 - Trip conclusion - Pilbara

Note: The missing segment in the depiction of the route above has arisen from a minor map database version mismatch between the YULO front-end and SWAN YULO front-end applications. When OpenStreetMap (which both applications use to store and display road segments) changes its road segment identifiers for any reason as part of a periodic update these changes are required to be migrated into both applications. Accordingly this missing segment will have affected the calculation of the average speed figure of the Bindi Bindi to Mt Magnet journey section (albeit the affect is likely to have been only minor in this circumstance).

### 3.3 Aggregated speed calculation

The average speed for any vehicle recorded traversing a road segment is readily calculated and the data aggregated so that the performance of each road segment can be graphed or mapped. The map at Figure 7 highlights the average speeds recorded for escort vehicles using the major metropolitan highway network.

The performance of any of these highways or road sections can be readily analysed in isolation.



Figure 7 – Average imputed speeds for metropolitan highway road segments

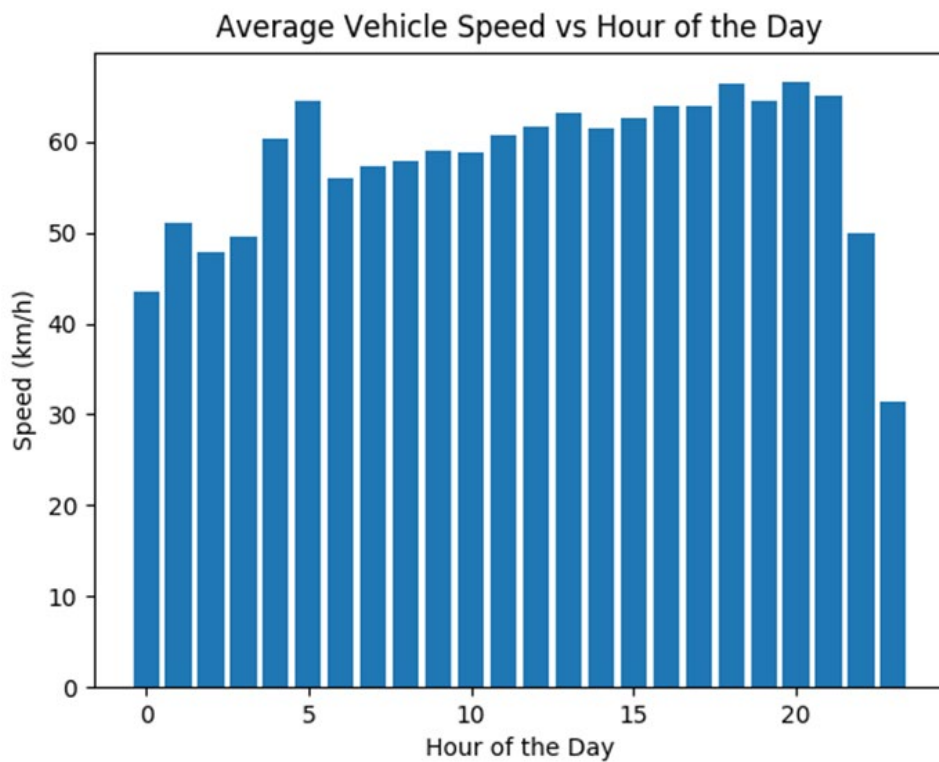
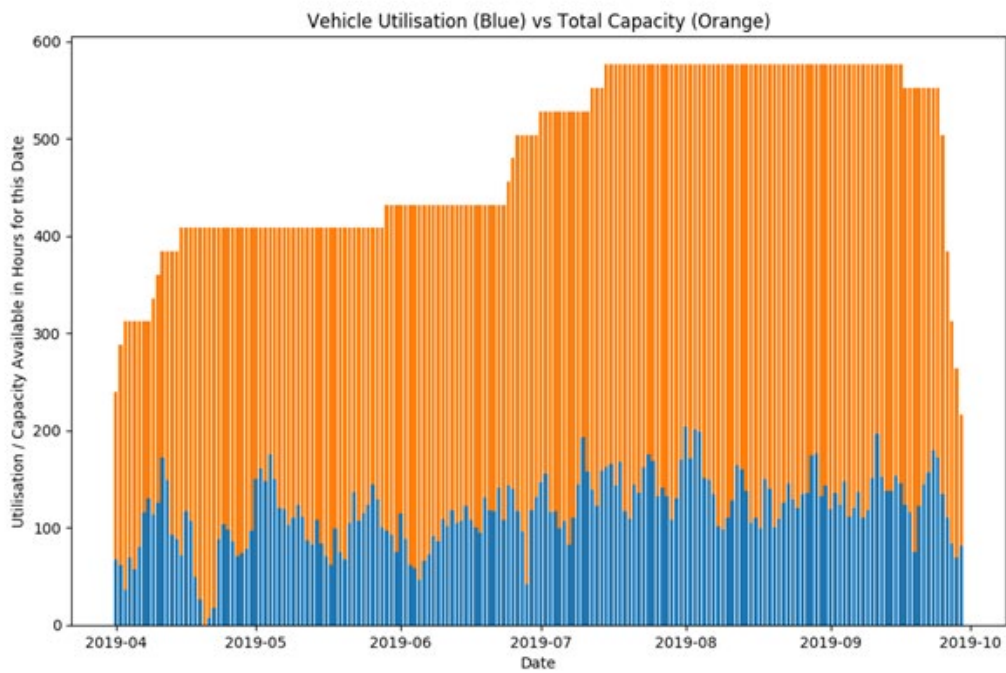
### 3.4 Other analytical samples

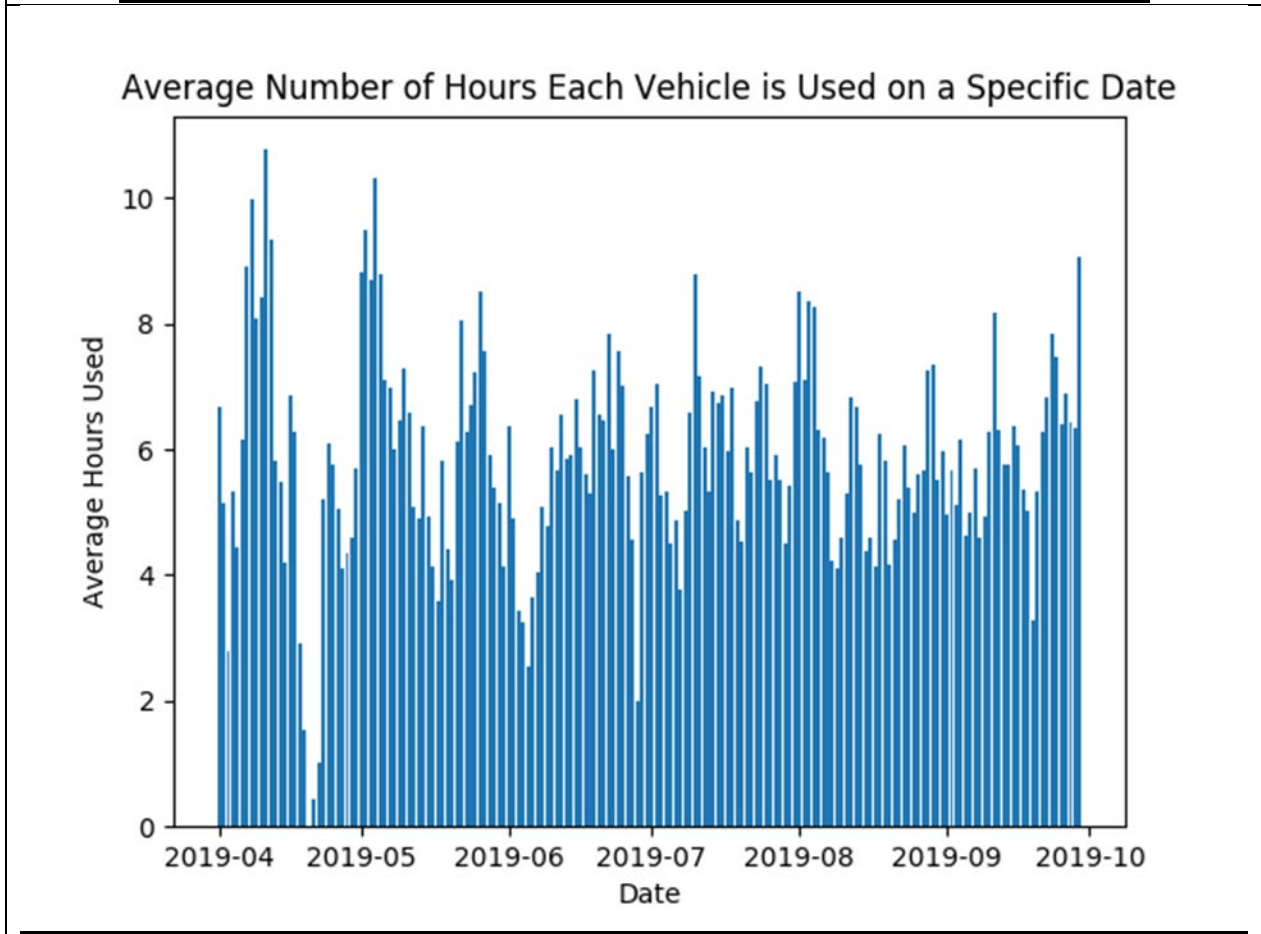
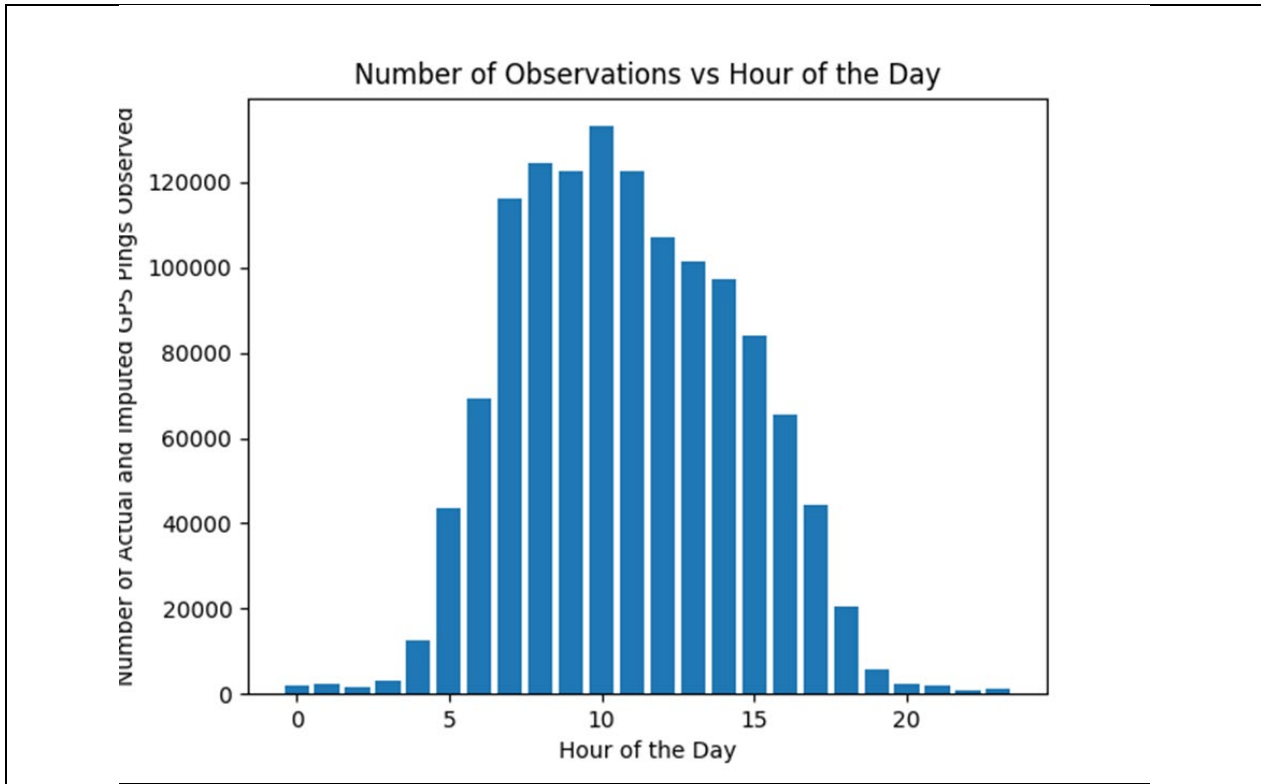
The data is amenable to analysis of various performance parameters as illustrated in the following graphs.

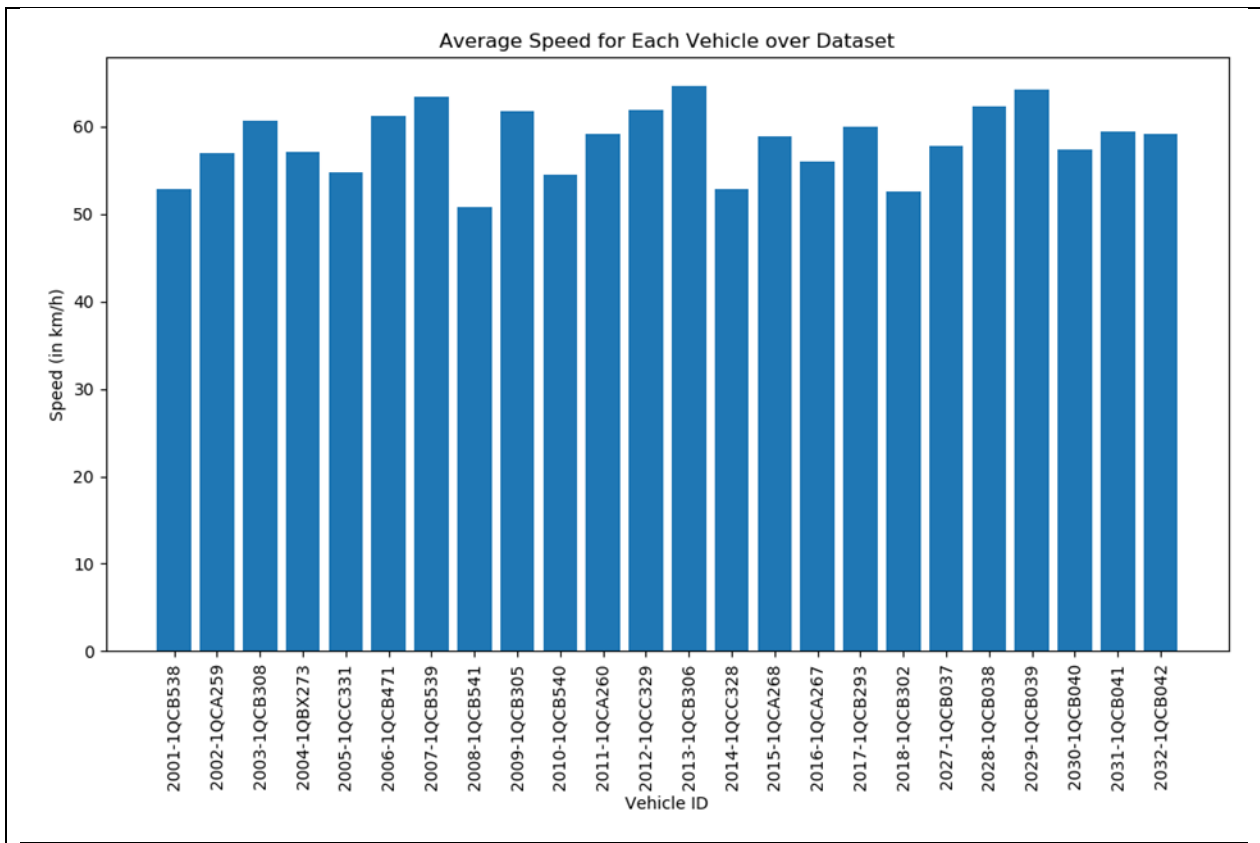
At this stage, the values and findings themselves are not significant, since there are some quirks in the data that need to be identified and managed in order for any measurement outcomes to be valid. Primarily these relate to the definition of trips and the treatment of stops in any analysis.

Inclusion of task IDs (relating vehicle ‘trips’ to full origin-destination escort tasks) in future data-sets would improve the utility of this type of analysis.

Sample graphs:







## 4 FUTURE DIRECTION

The preliminary analysis in this report provides a demonstration of the power of GPS telemetry data-bases for future governmental use in the management of the road network, vehicle registration system and the administration of the freight transport industry.

For MRWA, this preliminary analysis provides some pointers as to how the OSOM escort fleet management data routinely produced and stored, might be useful for other purposes, both within the narrow management interest, and into the realm of highway planning.

In time, it is reasonable to assume that some level of reporting of road usage by the freight industry will become mandatory, filling a large void at present. Road user charging has been on the horizon for some time, since the enabling technology became widely available, but has not been taken up by Australian governments to date. The politics of a shift away from flat registration charges to user charges are intimidating.

However, growing urban congestion issues in Australian cities are likely to lead to traffic monitoring and charging initiatives aimed at incentivising a more even spread of road usage across available daily hours. There will also be pressure to develop the means of funding rural road maintenance costs from dominant users rather than consolidated revenue streams. Fuel excise is becoming a less fruitful source of road maintenance revenue, as fuel usage falls due to improved fuel efficiency and take up of alternative energy sources.

When the time comes for new user charging systems to be trialled, certain types of freight vehicles, already subject to some forms of regulatory control such as OSOM permits, are the most likely guinea pigs. The strengths and weaknesses of various forms of data collection and processing for use in vehicle activity

monitoring will need to be established ahead of any such trials. Development of a system such as the SWAN YULO system would be a useful first step in this process.

---

## **5 ACKNOWLEDGEMENTS**

PATREC would like to thank staff at BITRE for sharing of the YULO software and co-operation and assistance throughout the course of this research.

PATREC would also like to thank MRWA Heavy Vehicle Services staff for the provision of OSOM escort vehicle data to help in the development of the SWAN YULO system.

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