Opturion Dynamic Transport Optimiser
Getting Started August 20, 2025

OPTURION
Intelligent Decision Support

Contents

1 Overview

The Opturion Dynamic Transport Optimiser (DTO) generates solutions to transportation optimisation problems. It takes an input, called a *scenario*, that consists of a set of orders, a set of locations, a set of vehicles (the *fleet*) and a set of business rules governing the assignment of orders to vehicles. The optimiser assigns orders in the scenario to vehicles in the fleet, producing a schedule of pickups and deliveries for each vehicle. These schedules are guaranteed to satisfy the business rules, as well as minimise costs.

This guide contains step-by-step instructions on how to set up and optimise a *pickup-delivery problem* (PDP) using the DTO. It should be read together with the accompanying example Excel workbooks and the "Opturion Dynamic Transport Optimiser – Data Formats" document.

The demonstration version of DTO is *only* available with map data for Australia. It does *not* work with locations outside of Australia.

If you want to trial DTO using locations outside of Australia, then please contact Opturion.

2 Interactive web interface

The DTO interactive web interface is accessible at the following URL:

https://dto-demo.opturion.com

The interactive web interface is compatible with recent versions of most web browsers including Google Chrome, Firefox, Microsoft Edge and Safari. It is *not* compatible with Microsoft Internet Explorer or the legacy version of Microsoft Edge.

After clicking on the above link, you will be prompted to enter your username and password. These are provided by Opturion when you register for DTO. If you are not registered, you can do so by going to the DTO website and clicking on the registration link in the navigation bar.

After your username and password are correctly entered, you will automatically be taken to the Landing Page interface shown in Figure 1.

2.1 Landing Page

Figure 1: Dashboard interface.

Selecting the “New” option will take you to the Dashboard interface shown in Figure 2.

2.2 Dashboard Interface

Figure 3 shows the dashboard control buttons.

The *Choose File* button lets you upload an input file in either the DTO-JSON or DTO-Excel input format. On a successful upload, the map is updated to display the locations in the input. Otherwise, a dialog box giving you the choice to cancel the upload or view the errors will be displayed. If you choose to view the errors, then a new window or tab will be opened in your browser and the errors displayed in it.

The *Run* button starts the optimisation. Once started, it changes into the *Stop* button, allowing you to stop the optimisation and obtain a solution.

If the *Warnings* button appears, then it means that the optimiser has detected potential problems with the scenario. These problems can cause orders to not be assigned or vehicles to not be used. Click the button to see what the potential problems are.

The *Export* button lets you download the solution as an Excel workbook.

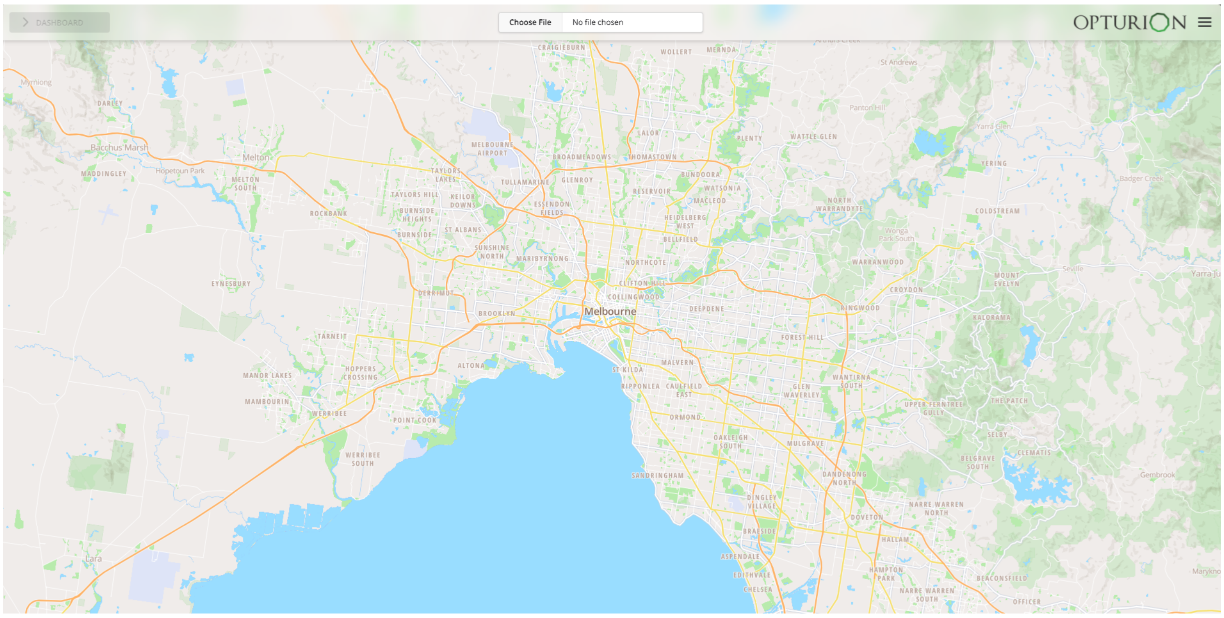


Figure 2: Dashboard interface.

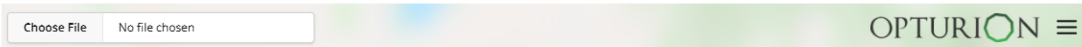


Figure 3: Dashboard control buttons.

Figure 6 shows a solution in the dashboard.

The map option menu (Figure 4) lets you select what is displayed on the map. Selecting the *Locations* option shows each location on the map. Blue markers show locations where all orders are assigned. Green markers show locations where there is at least one unassigned order. The other options in the drop-down let you display the route of each vehicle on the map.

Clicking on a map marker opens an information box containing details of what is scheduled to occur at that location. An example of this can be seen in Figure 5.

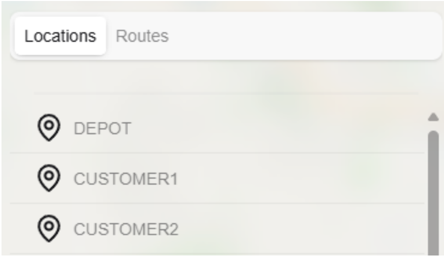


Figure 4: Map option select.

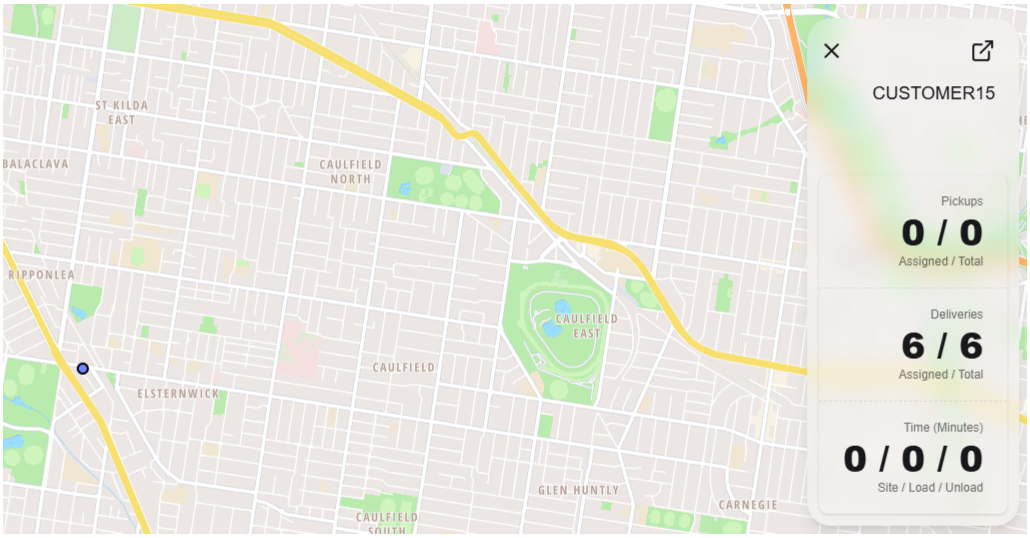


Figure 5: Location select.

Key performance indicators (KPIs), such as total cost and fleet usage, are shown in the top right corner. The graph along the top of the window shows the KPI values as the optimisation progresses.

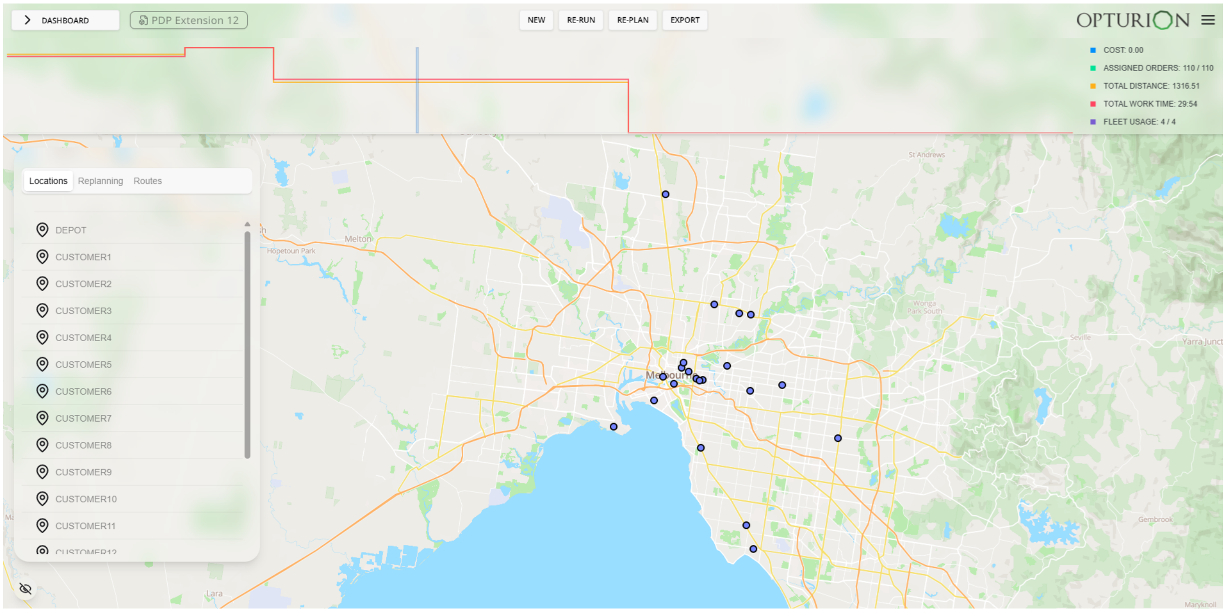


Figure 6: Optimisation solution.

2.3 Job Manager Interface

The *Job Manager* interface lets you run multiple optimisation jobs simultaneously. The progress of the optimisation jobs is updated every ten seconds. Figure 7 shows the control buttons at the top of the job manager interface.

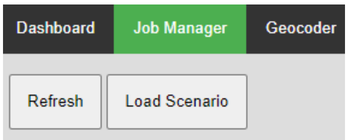


Figure 7: Job Manager control buttons.

The links labelled *Dashboard* and *Geocoder* will take you to the Dashboard and Geocoder interfaces respectively.

The *Refresh* button updates the status of any optimisation jobs. Note that this does not affect the automatic refresh timer, which will continue to update progress every ten seconds.

The *Load Scenario* button lets you submit a new job for optimisation in the DTO-Excel input format. If the scenario contains errors, then optimisation will not commence and instead an *Invalid Instances* table will appear at the bottom of the screen. This lets you open a new window or tab that displays the errors.

If the *Warnings* button appears, then it means that the optimiser has detected potential problems with the scenario. These problems can cause orders to not be assigned or vehicles to not be used. Click the button to see what the potential problems are. Figure 8 shows the interface when both errors and warnings are present.

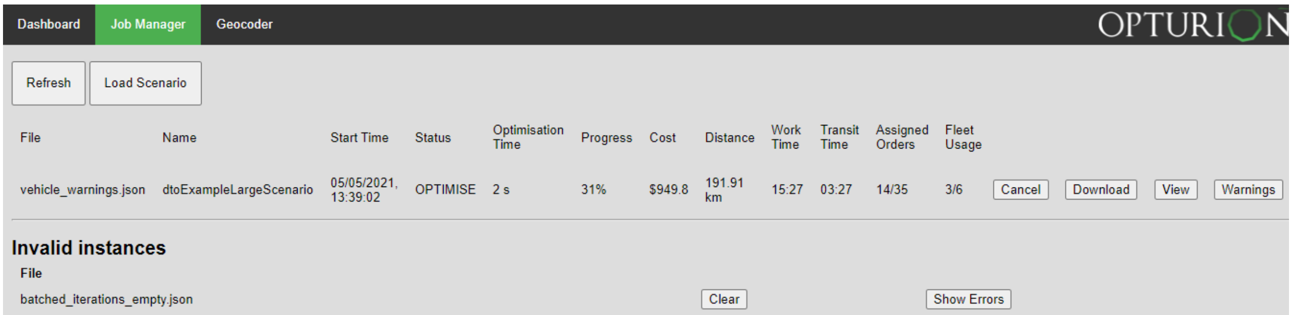
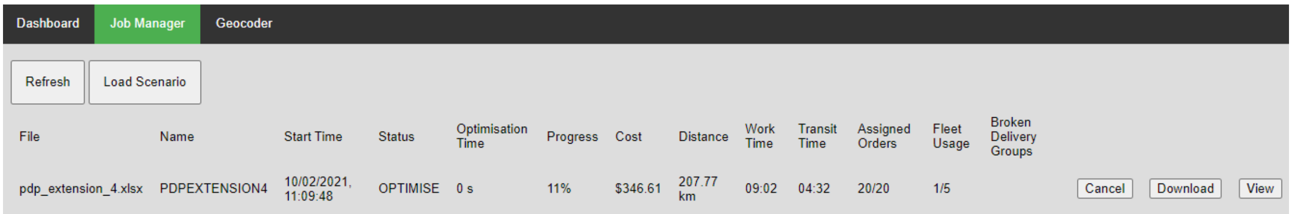


Figure 8: Job Manager errors and warnings.

The job manager displays KPIs for each in-progress optimisation job. This can be seen in Figure 9. The *Download* and *View* buttons only appear for an optimisation job after an initial solution has been generated.



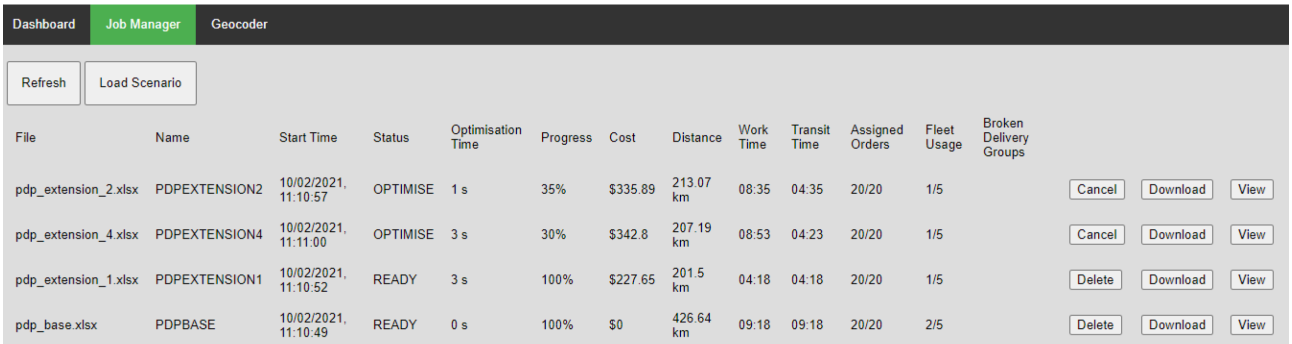
Dashboard | Job Manager | Geocoder

Refresh | Load Scenario

File	Name	Start Time	Status	Optimisation Time	Progress	Cost	Distance	Work Time	Transit Time	Assigned Orders	Fleet Usage	Broken Delivery Groups			
pdp_extension_4.xlsx	PDPEXTENSION4	10/02/2021, 11:09:48	OPTIMISE	0 s	11%	$346.61	207.77 km	09:02	04:32	20/20	1/5		Cancel	Download	View

Figure 9: Job Manager KPIs.

Figure 10 shows the job manager interface when optimising multiple scenarios.



Dashboard | Job Manager | Geocoder

Refresh | Load Scenario

File	Name	Start Time	Status	Optimisation Time	Progress	Cost	Distance	Work Time	Transit Time	Assigned Orders	Fleet Usage	Broken Delivery Groups			
pdp_extension_2.xlsx	PDPEXTENSION2	10/02/2021, 11:10:57	OPTIMISE	1 s	35%	$335.89	213.07 km	08:35	04:35	20/20	1/5		Cancel	Download	View
pdp_extension_4.xlsx	PDPEXTENSION4	10/02/2021, 11:11:00	OPTIMISE	3 s	30%	$342.8	207.19 km	08:53	04:23	20/20	1/5		Cancel	Download	View
pdp_extension_1.xlsx	PDPEXTENSION1	10/02/2021, 11:10:52	READY	3 s	100%	$227.65	201.5 km	04:18	04:18	20/20	1/5		Delete	Download	View
pdp_base.xlsx	PDPBASE	10/02/2021, 11:10:49	READY	0 s	100%	$0	426.64 km	09:18	09:18	20/20	2/5		Delete	Download	View

Figure 10: Optimising multiple scenarios.

If a scenario has no name in its *General* sheet, then the name displayed will be the *job identifier* assigned by the optimiser.

For a description of all the KPIs, see “Opturion Dynamic Transport Optimiser – Data Formats”.

Clicking on the *View* button opens the *Solution Viewer*. This lets you explore the current solution on a map. Select a vehicle from the drop-down menu to plot its route on the map. Clicking on a location marker, represented by small blue squares on the map, will display details of what is scheduled to occur at that location.

Figure 11 shows a vehicle route displayed in the *Solution Viewer*.

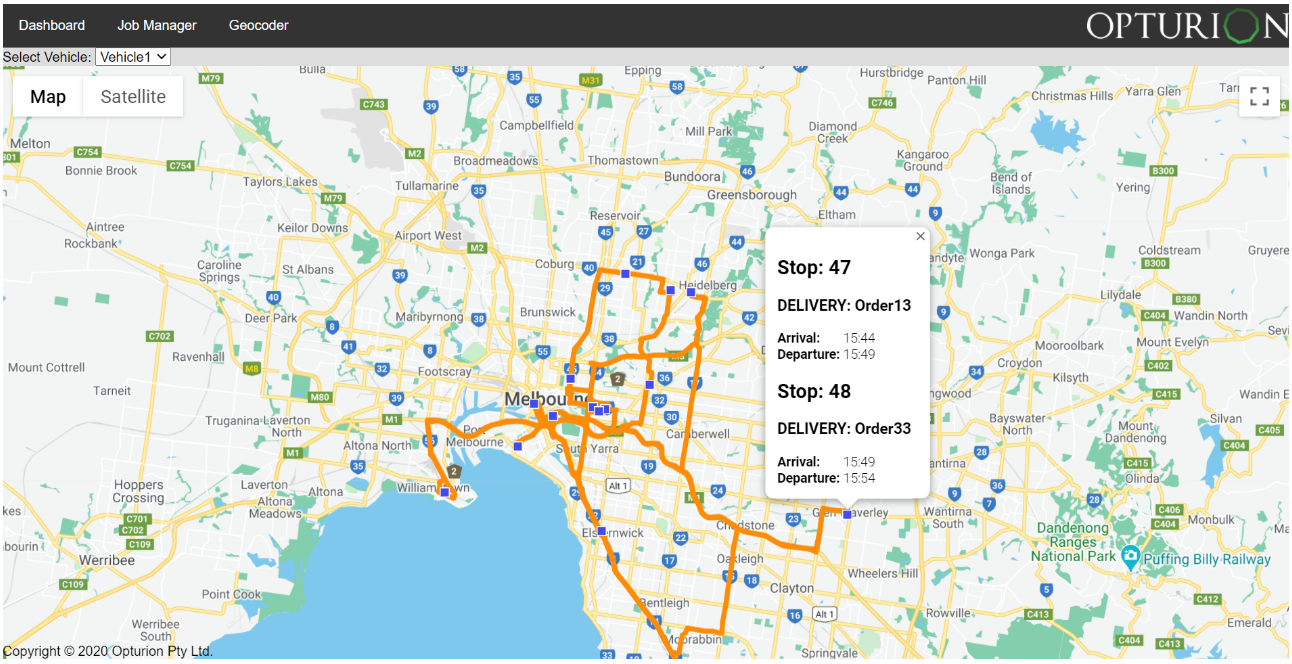


Figure 11: Vehicle route in Solution Viewer.

3 Worked Examples

This section contains a series of worked examples, with the first one introducing a basic *PDP* and subsequent examples increasing the complexity and adding new features.

Each example is intended to be read with its accompanying Excel workbook. These may be downloaded from the DTO website files page.

3.1 Basic Example

Accompanying Workbook: pdp_base.xlsx.

Opturion Freight is a small (and fictitious) transport company operating across greater Melbourne with a fleet of five trucks. Each day, orders must be picked up from their depot and delivered to customers around the city.

The orders have varying weights and volumes. Every assignment of orders to trucks must ensure that the capacities of the trucks are *never* exceeded.

The depot is open from `06:00` to `18:00`. Each vehicle must start and finish its day at the depot. Furthermore, customer locations may have their own operating hours, outside of which Opturion Freight cannot deliver.

Currently, the owner of Opturion Freight plans each day's pickups and deliveries by hand the evening before. This is a time-consuming process and the owner feels that potential efficiencies are being missed.

In these examples, we are going to automate the creation of the daily transport plan for Opturion Freight using DTO. We begin by creating an input workbook for use with DTO.

3.1.1 Excel Data Conventions

Before stepping through the input to the optimiser, we describe how various types of data required by the optimiser are represented in Excel.

- **Identifiers**
  Entities in the optimiser input, such as orders, locations and vehicles, must have an *identifier* (Id) that uniquely identifies them.

  These identifiers are used to refer to those entities in the optimiser output.

  Identifiers are represented using text or number cells in Excel. Their names must satisfy the following rules:

  1. They must contain at least one character.
  2. They must not contain the semicolon ("`;`") character.

  Identifiers are not case sensitive or "underscore" sensitive. Whitespace in the identifier name is *not* significant. For example: "`S1`", "`s1`" and "`S_1`" are all treated as a reference to the *same* entity by the optimiser.

- **Times**
  In Excel, times are *stored* as decimal numbers, where `0` is `00:00` and `1` is `24:00`.

  The integer component of the decimal represents a 24-hour period and is used to represent times greater than `23:59`. When using this representation, it is useful to think of the integer component as the day on which the time occurs, with times less than `24:00` occurring on *day zero*. For example, `1.50` would be `36:00`, or `12:00` on the second day.

  Cells containing times may be formatted and *displayed* in a variety of ways by Excel. For example, `hh:mm`, `d hh:mm` or `h:mm AM/PM` and so forth. Times may be given to the optimiser in any of the Excel time formats.

  Excel automatically converts times entered on the keyboard into decimal format. For example, if `06:00` is entered into a cell then the value of that cell is `0.25` and its format is set to `hh:mm`.

  Note that schedules are generated to the minute by the optimiser. Times and durations below a minute are rounded up.

- **Durations**
  In Excel, durations are like times and are *stored* as decimal numbers.

  Cells containing durations may be formatted and *displayed* in a variety of ways. For example, `[hh]:mm`, `hh:mm` and so forth. Durations may be given to the optimiser in any of the Excel formats.

- **Addresses**
  Addresses must be contained in a single cell. If they are split across multiple columns, then you can use an Excel formula like the following to combine the address information into a single cell.

A	B	C	D
Street Number	**Street**	**Postcode**	**Address**
12	Example St	3000	=A21&" "&B21&" "&C21&", Victoria, Australia"

  Here the "&" character is used to combine the cell values with either spaces or the additional information that the address is in "Victoria, Australia". While the above example only joins three cells to form an address, it can be extended for any number of cells. This formula evaluates to "12 Example St 3000, Victoria, Australia".

  Address information is *only* needed if you do not have latitude and longitude coordinates for locations in your input. In that case, addresses need to be converted into their corresponding latitude and longitude. This is done using a process called *geocoding*.

  Geocoding your input is done as a separate step *before* you run the optimiser. We describe how in the Geocoder section.

  For the worked examples in this guide, the inputs have already been geocoded.

3.1.2 Optimiser Input

For this example, our Excel workbook contains four sheets. They are:

1. **General**
   This sheet lets you control the settings used by the optimisation. It has two columns: the *Parameter* column names an optimiser parameter whose value you want to set, and the *Value* column contains the value for that parameter.

   Here, we set two parameters.

   - **Scenario Name**
     This is an optional parameter that lets you give a name to the scenario. If a name is provided, then it is used in the output.
   - **Iterations**
     This is an optional parameter that lets you set the number of iterations that the optimiser will run for. Using a greater number of iterations *may* yield a better solution, at the cost of a longer runtime. If this parameter is not specified, then the optimiser defaults to using `3000` iterations.

2. **Locations**
   This sheet contains details of the locations used in the scenario. It has the following columns:

   - **Id**
     The unique identifier for the location.
   - **Name**
     The name of the location (e.g. a site or store name). This may be left blank.
   - **Address**
     The street address of the location.
   - **Longitude**
     The longitude, in decimal degrees, of the location.
   - **Latitude**
     The latitude, in decimal degrees, of the location.
   - **Opening Time**
     The opening time of the location. No vehicle can make pickups or deliveries at the location *before* this time.
   - **Closing Time**
     The closing time of the location. No vehicle can make pickups or deliveries at the location *after* this time.

   Opening Time and *Closing Time* must be entered as Excel times.

3. **Orders**
   This sheet contains details of the orders that we want to assign to vehicles in the fleet. It has the following columns:

   - **Id**
     The unique identifier for the order.
   - **Pickup Location**
     The identifier of the location where the order will be picked up.
   - **Delivery Location**
     The identifier of the location where the order is to be delivered.
   - **Weight**
     The weight of the order.
   - **Volume**
     The volume of the order.

   It is not necessary to have both *Weight* and *Volume* columns in the *Orders* sheet, but at least one of them *must* be present.

   The units of measurement used in the *Weight* and *Volume* columns should be the same as those used for the vehicle capacities.

4. **Fleet**
   The sheet contains details of all the vehicles in our fleet. It has the following columns:

   - **Id**
     The unique identifier for the vehicle.
   - **Start Location**
     The identifier of the location where the vehicle begins its route. If blank, then the vehicle begins its route at the first pickup location.
   - **Finish Location**
     The identifier of the location where the vehicle must end its route. If blank, then the vehicle ends its route at the last delivery location.
   - **Maximum Weight**
     The maximum weight of orders allowed on the vehicle at any time.
   - **Maximum Volume**
     The maximum volume of orders allowed on the vehicle at any time.
   - **Earliest Start Time**
     The earliest time that the vehicle can begin work.

The units of measurement used in the *Maximum Weight* and *Maximum Volume* columns should be the same as those for the order *Weight* and *Volume* columns.

Earliest Start Time is entered as an Excel time.

For further details, such as additional sheets or columns not described here, refer to “Opturion Dynamic Transport Optimiser – Data Formats”.

3.1.3 Running The Optimiser

In this guide, we will run the optimiser using the dashboard interface. Instructions for accessing the dashboard interface can be found here. After logging in, you will see the screen in Figure 12.

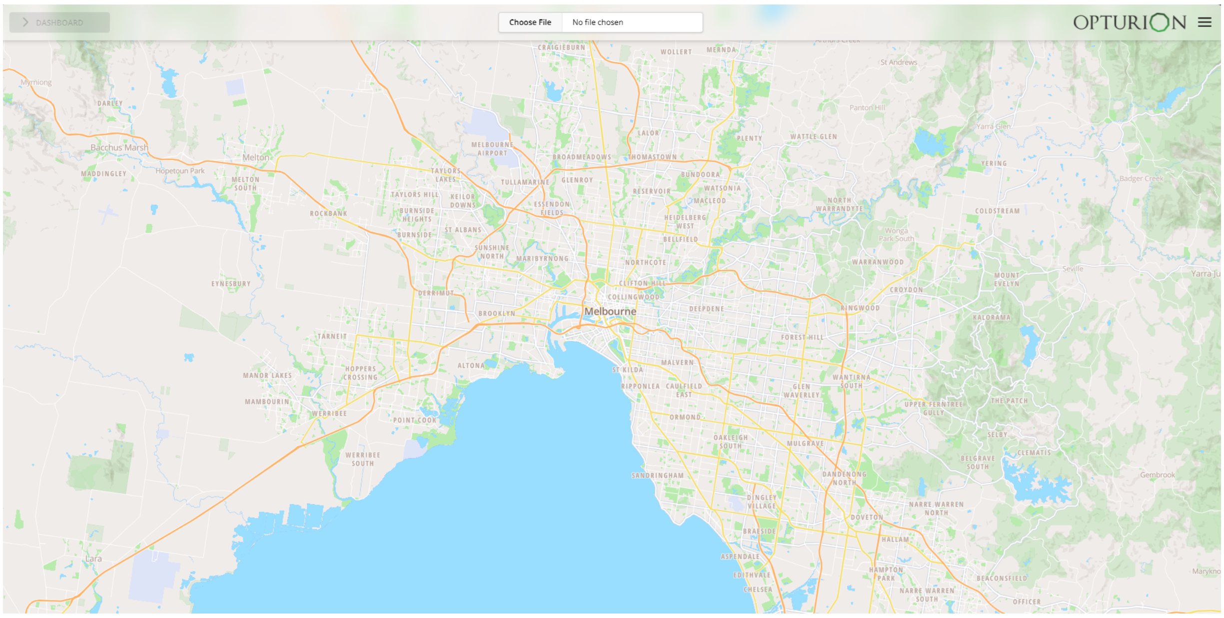


Figure 12: Dashboard interface.

To optimise a scenario, it must first be uploaded to the DTO by clicking the *Load Scenario* button and selecting the workbook file. Once the scenario has loaded, its locations are plotted on the map. In Figure 12, the red squares on the map show the locations. The scenario is now ready for optimisation. The optimisation can be started by clicking the *Solve* button in the top left corner of the screen. The optimiser will run until it has completed all the iterations set in the General sheet of the workbook.

Once the optimiser has started, the dashboard interface displays solution KPIs and routes for the vehicles. The KPI graph is plotted against the number of iterations. It lets you see how the solution improves as the optimisation proceeds. Plots for individual KPIs can be turned on or off using the check boxes in the top right corner of the screen. This can be seen in Figure 13.

When a solution has been found by the optimiser, you can view a vehicle's route by selecting it from the drop-down menu. Clicking on a location, indicated by a square on the map, will open an information box containing details about what is scheduled to occur at that location. Selecting the *Orders* option from the drop-down menu will display all locations on the map.

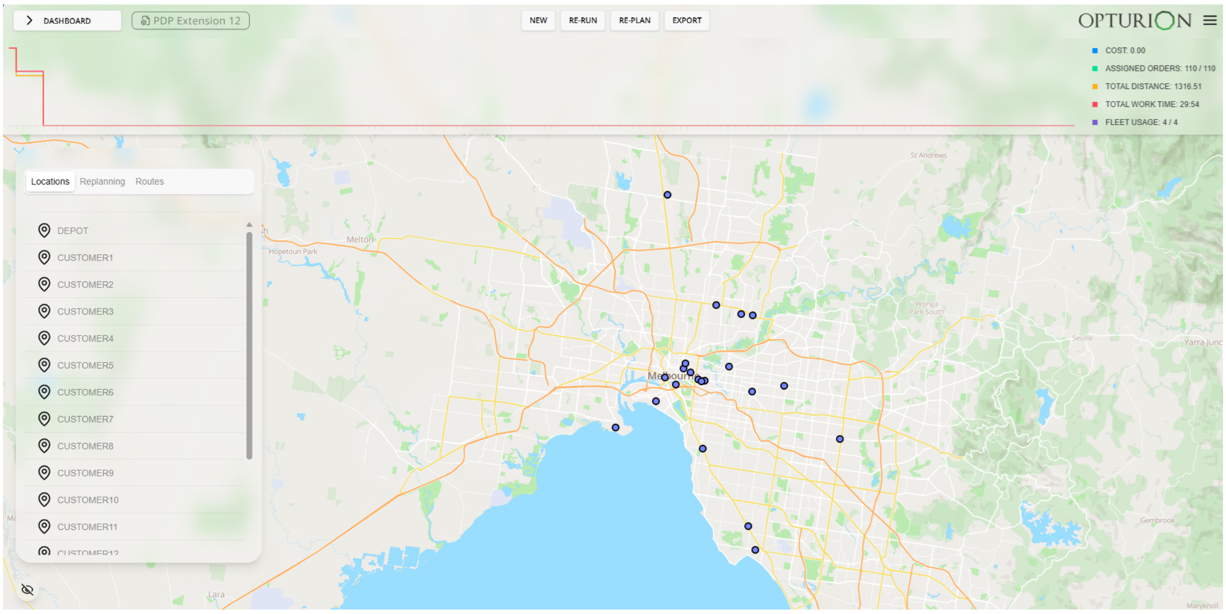


Figure 13: Dashboard interface: solving.

When the optimiser has finished, the *Export Scenario* button will become active. Clicking this button to download an Excel workbook containing the solution found by the optimiser.

You can halt the optimisation at any stage by clicking the *Stop* button. This will also cause the *Export Scenario* button to become active and you can download the best solution the optimiser found before it was halted.

3.1.4 Solution Excel

After you have downloaded the solution, you will have an Excel workbook containing four sheets. These are:

1. **Solution Summary**
   This sheet contains summary KPIs for the entire solution. Table 2 shows a subset of this.

2. **Runsheet**
   This sheet contains a detailed breakdown of each vehicle's schedule. Each row represents a stop and contains the following:

   - **Vehicle Id**
     The identifier of the vehicle.
   - **Load**
     The load number. For this scenario, this is not enabled. It will be relevant for Extension 7: Batched Loads.
   - **Stop Number**
     The stop number.
   - **ETA**
     The estimated time of arrival at the stop.
   - **ETD**
     The estimated time of departure from the stop.
   - **Stop**
     The type of the stop: `START`, `FINISH`, `PICKUP`, `DELIVERY` or `BREAK`.
   - **Order**
     If the stop type is `PICKUP` or `DELIVERY`, then this is the identifier of the order that is being picked up or delivered.
   - **Address**
     The identifier of the location where the stop occurs.
   - **Street Address**
     The address of the location where the stop occurs.
   - **Volume**
     The volume on the vehicle at the departure from the stop.
   - **Delta Volume**
     For `PICKUP` or `DELIVERY` stops, the change in volume at the stop.
   - **Weight**
     The weight on the vehicle at the departure from the stop.
   - **Delta Weight**
     For `PICKUP` or `DELIVERY` stops, the change in weight at the stop.
   - **Transit Time**
     The travel time between the previous and current stop. If the transit time is zero, then this cell is blank.
   - **Transit Distance**
     The travel distance between the previous and current stop. If the transit time is zero, then this cell is blank.

3. **Vehicle Summary**
   This sheet contains KPIs for each vehicle.

4. **Load Summary**
   This sheet contains KPIs for each load.

Notice that in this solution we are underutilising the capacity of the vehicles. Also, each vehicle is only scheduled to work for half a day. This is because we have not assigned running costs to the vehicles. In this case, the

Vehicle Id	Load	Stop Number	ETA	ETD	Stop	Order	Address	Street Address	Volume	Delta Volume	Weight	Delta Weight	Transit Time	Transit Distance
Vehicle1		0		6:56	START		DEPOT	18 Kavanagh St Southbank						
Vehicle1	1	1	6:56	6:56	PICKUP	Order8	DEPOT	18 Kavanagh St Southbank	4	4	320	320		
Vehicle1	1	2	7:00	7:00	DELIVERY	Order8	CUSTOMER8	201 SPENCER ST	0	-4	0	-320	0:04	2.11
Vehicle1	2	3	7:05	7:05	PICKUP	Order17	DEPOT	18 Kavanagh St Southbank	3	3	300	300	0:05	2.27
Vehicle1	2	4	7:05	7:05	PICKUP	Order11	DEPOT	18 Kavanagh St Southbank	5	2	420	120		
Vehicle1	2	5	7:05	7:05	PICKUP	Order9	DEPOT	18 Kavanagh St Southbank	6	1	490	70		
Vehicle1	2	6	7:05	7:05	PICKUP	Order16	DEPOT	18 Kavanagh St Southbank	12	6	690	200		
Vehicle1	2	7	7:05	7:05	PICKUP	Order14	DEPOT	18 Kavanagh St Southbank	13	1	702	12		
Vehicle1	2	8	7:12	7:12	DELIVERY	Order11	CUSTOMER11	Station St, Sandringham	11	-2	582	-120	0:07	4.09
Vehicle1	2	9	7:18	7:18	DELIVERY	Order16	CUSTOMER16	17 Walpole St, Kew VIC 3101	5	-6	382	-200	0:06	3.96
Vehicle1	2	10	7:28	7:28	DELIVERY	Order17	CUSTOMER17	126 Union Rd, Surrey Hills	2	-3	82	-300	0:10	7.39
Vehicle1	2	11	7:35	7:35	DELIVERY	Order14	CUSTOMER14	5/57 Station St, Camberwell	1	-1	70	-12	0:07	4.46
Vehicle1	2	12	7:43	7:43	DELIVERY	Order9	CUSTOMER9	300 BRIDGE RD	0	-1	0	-70	0:08	5.44

Table 1: Runsheet example.

KPI	**Value**
Total Cost	0
Total Distance	426.64
Work Time	09:18
Transit Time	09:18
Assigned Orders	20 / 20
Fleet Usage	2 / 5

Table 2: Solution KPIs.

optimiser cannot distinguish between solutions with less working time, fleet usage etc., because those solutions do not improve the cost. We will fix this in the next example by adding running costs to the fleet.

3.2 Extension 1: Vehicle Costs

Accompanying Workbook: pdp_extension_1.xlsx.

In our first example, there were no costs associated with the vehicles. Because of this, when the optimiser was searching for better solutions, it did not account for the number of vehicles used, distance driven or time taken.

We will now add three kinds of cost, shown in Table 3, to each vehicle.

Cost Per Use	Cost Per Km	Cost Per Hour
$100	$0.10	$25

Table 3: Vehicle costs.

These costs have been added to the *Fleet* sheet of the accompanying workbook. Having a *cost per use* will encourage the optimiser to use as few vehicles as possible. Having a *cost per km* will encourage the optimiser to minimise travel distance. Having a *cost per hour* will encourage the optimiser to minimise work time.

KPI	Value
Total Cost	228
Total Distance	201.50
Work Time	04:18
Transit Time	04:18
Assigned Orders	20 / 20
Fleet Usage	1 / 5

Table 4: Solution KPIs.

Table 4 shows the solution KPIs that result after we add the above vehicle costs. We have used one less vehicle, driven 225.14 km less and worked for five fewer hours.

Here, the choice of vehicle is arbitrary because all of them have identical costs, and begin and finish their routes at the depot. In a situation where the vehicles have different costs, the optimiser will assign orders to vehicles in such a way as to minimise the total cost.

3.3 Extension 2: Service and Site Times

Accompanying Workbook: pdp_extension_2.xlsx.

In the solution of the previous example, you can see that the vehicles are loading and unloading instantaneously. In reality, it takes approximately thirty minutes to load a truck at the depot. Each order also requires a check after it has been loaded. This check takes one minute per order. Every delivery takes five minutes to complete.

DTO supports several different timings that can be used to model loading and unloading times. In this example, we introduce *service time* and *site time*.

- **Service time** is the amount of time needed to *service* an order at a pickup or delivery stop. Orders can have separate pickup and delivery service times. Different orders can have different service times.
- **Site time** is the amount of time needed when a vehicle visits a location. It is independent of the number of consecutive pickups and deliveries occurring at the location. It is also independent of any service time for the pickups and deliveries. Site time has many uses: it can be used to model loading times at depots, docking and engagement procedures, vehicle inspections etc. Note that if a vehicle leaves a location and later returns to that same location to do further pickups or deliveries then it will incur the site time again.

Returning to Opturion Freight, to model the thirty-minute loading time at the depot, we add the *Site Time* column to the *Locations* sheet. We set a value of `00:30` against the depot in that column and `00:00` against the other locations. (We could alternatively leave the cells for locations with no site time blank.)

To model the one minute check of each order, we use a pickup service time. We add a *Pickup Service Time* column to the *Orders* sheet and set a value of *00:01* for every order.

Finally, to model the five minute delivery time for each order, we use a delivery service time. We add a *Delivery Service Time* column to the *Orders* sheet and set a value of *00:05* for every order.

The accompanying workbook contains these modifications.

KPI	Value
Total Cost	330
Total Distance	202.89
Work Time	08:24
Transit Time	04:24
Assigned Orders	20 / 20
Fleet Usage	1 / 5

Table 5: Solution KPIs.

After optimising the modified scenario, we can see the additional time taken for loading and unloading in the *Work Time* KPI shown in Table 5. It has increased from that of the previous examples. In the *Runsheet*, you can see that the site and service times are now included in the differences between the stop arrival time (*ETA*) and stop departure time (*ETD*) columns.

3.4 Extension 3: Earliest and Latest Times

Accompanying Workbook: pdp_extension_3.xlsx.

Some of Opturion Freight's customers say that they do not want deliveries to occur too early in the morning, or too late in the afternoon.

Simple time constraints like these can be set up by adding *Earliest Delivery Time* and *Latest Delivery Time* columns to the *Orders* sheet. (Constraining pickup times in a similar way is also supported, but we will restrict ourselves to deliveries in this example.)

Cells in the *Earliest Delivery Time* or *Latest Delivery Time* columns may be left blank if there is no earliest or latest delivery time respectively for an order.

Table 6 shows an excerpt from the accompanying workbook. In it, we impose the following delivery time constraints:

- *Order 1* cannot be delivered after `12:00`.
- *Order 6* cannot be delivered before `14:00`.
- *Order 10* can only be delivered between `10:00` and `15:00`.

Id	Earliest Delivery Time	Latest Delivery Time
Order1		12:00
Order6	14:00	
Order10	10:00	15:00

Table 6: Earliest and latest delivery times.

Examining the solution *Runsheet*, we can see that the optimiser has scheduled the delivery for *Order 1* at `9:59`, the delivery for *Order 6* at `14:00` and the delivery for *Order 10* at `12:19`.

Satisfying these additional timing constraints has affected the solution KPIs, as seen in Table 7.

KPI	Value
Total Cost	344.50
Total Distance	207.84
Work Time	08:57
Transit Time	04:27
Assigned Orders	20 / 20
Fleet Usage	1 / 5

Table 7: Solution KPIs.

3.5 Extension 4: Time Windows

Accompanying Workbook: pdp_extension_4.xlsx.

In the last section we introduced simple earliest and latest delivery time constraints. However, for some of Opturion Freight's customers the situation is more complex. They open early, close for lunch and then re-open after lunch.

Describing these more complex delivery time constraints requires the use of *time windows*.

A time window is a span of time that has a fixed start time and a fixed end time. Order pickups and deliveries may have multiple time windows associated with them. The optimiser will schedule the pickup or delivery so that it occurs inside at least one of the time windows. (It may be scheduled inside multiple time windows if they overlap.)

Time windows can also be associated with *locations*. If a location has time windows, then any pickup or delivery at that location *must* occur inside one of those time windows.

Time windows are defined using the *Time Windows* sheet. Table 8 shows an example of this sheet.

Id	Start	End
Morning	7:00	12:00
Afternoon	12:00	16:00
Not Lunch	7:00	12:00
Not Lunch	13:00	16:00

Table 8: Time window definitions.

Each *group* of time windows is defined by the identifier in the *Id* column. Rows sharing the same identifier belong to the same time window group. Start and end times for the individual windows in each group are given by the *Start* and *End* columns.

The example in Table 8 defines three groups of time windows: *Morning*, *Afternoon* and *Not Lunch*. *Morning* consists of a single span of time, from `7:00` to `12:00`. Likewise, *Afternoon* consists of a single span of time, from `12:00` to `16:00`. *Not Lunch* consists of two spans of time, `7:00` to `12:00` and `13:00` to `16:00`. Both these latter windows are part of the same time window group as they share the identifier, *Not Lunch*.

To use these time window groups to constrain delivery times, we add a *Delivery Time Windows* column to the *Orders* sheet. If a cell in this column contains the identifier of a time window group, then those time windows will apply to the delivery of the order in that row. (Pickup time windows are similarly handled using the *Pickup Time Windows* column of the *Orders* sheet.)

Time windows defined using the *Time Windows* sheet combine with other times, such as earliest and latest delivery times or location time windows. For example, if an order has a delivery time window *and* an earliest delivery time, then it must be scheduled at a time which is both inside the delivery time window and after the earliest delivery time. If this is impossible, for example because the earliest delivery time is after any of the delivery time windows, then the order cannot be assigned by the optimiser. Such orders are reported in the *Unassigned Orders* sheet of the solution workbook.

The accompanying workbook defines and uses delivery time windows.

3.6 Extension 5: Order-Vehicle Compatibility

Accompanying Workbook: pdp_extension_5.xlsx.

Opturion Freight has a new requirement: heavy orders must only be loaded on a truck that has a tailgate loader. This requirement is an *order-vehicle compatibility* rule. It restricts what assignments of orders to vehicles are allowed.

You can describe the properties of vehicles and orders using labels called *attributes*. Attribute names, like identifiers, must not contain semicolons and are not case or *white space* sensitive. A vehicle or order may be labelled with any number of attributes. A vehicle or order identifier is automatically treated as one of its attributes.

Vehicle and order attributes are set in the *Attributes* column of the *Fleet* and *Orders* sheets respectively. Multiple attributes must be separated using semicolons. Order-vehicle compatibility rules are set up in the *Order-Vehicle Compatibility* sheet.

Note: the *Order-Vehicle Compatibility* sheet only allows you define *positive* relationships between order and vehicle attributes. For further details, refer to "Opturion Dynamic Transport Optimiser – Data Formats".

In the accompanying workbook, we have added attributes and set up an order-vehicle compatibility rule that meets Opturion Freight's new requirement. Heavy orders are labelled with the attribute *HEAVY* and vehicles that have at tailgate loader have been labelled with the attribute *TAILGATE*. Table 9 shows the *Order-Vehicle Compatibility* sheet from the accompanying workbook. The order attributes are listed in the first column, and the vehicle attributes are listed in the first row. The presence of `TRUE` in a cell corresponding to the *HEAVY* and *TAILGATE* attributes means the optimiser is only allowed to assign an order labelled with the *HEAVY* attribute to a vehicle labelled with the *TAILGATE* attribute.

We can also use order-vehicle compatibility rules to restrict the locations the vehicles are allowed to make pickups or deliveries at. While we cannot label locations directly with attributes, we can label the orders being picked up from, or delivered to, those locations with location specific properties.

For example, consider the situation where a location lacks a loading dock and vehicles must park on the street. Longer vehicles are not allowed to make pickups or deliveries at this location due to space restrictions. We have included this example in the accompanying workbook: orders that are picked up from, or delivered to, locations without a loading dock are labelled with the attribute *NO DOCK*. Vehicles that may be parked on the street are labelled with the attribute *SHORT*. Finally, in the *Order-Vehicle Compatibility* sheet we have set up a compatibility rule that requires orders with the *NO DOCK* attribute to only be assigned to vehicles with the *SHORT* attribute. This effectively imposes the restriction that longer vehicles cannot visit the location without a loading dock.

		Vehicle Attributes	
		TAILGATE	SHORT
Order Attributes	HEAVY	TRUE	
	NO DOCK		TRUE

Table 9: Order-vehicle compatibility.

3.7 Extension 6: Driver Breaks

Accompanying Workbook: pdp_extension_6.xlsx.

Managing driver breaks are an important part of the delivery schedule, and when Opturion Freight decides to implement a fatigue management strategy, they realise they can use DTO to choose the optimal times to allocate these breaks.

DTO lets you define *break schemes* and assign them to vehicles. Each vehicle can have at most one break scheme but a single break scheme can be assigned to multiple vehicles.

A commonly used break scheme is the *Basic Fatigue Management* hours as seen in Table 10.

Time	**Work**	**Rest**
In any period of...	*A driver must not work for more than a maximum of...*	*And must have the rest of that period off work with at least a minimum rest break of...*
6.25 hours	6 hours work time	15 continuous minutes rest time
9 hours	8.5 hours work time	30 minutes rest time in blocks of 15 continuous minutes
12 hours	11 hours work time	60 minutes rest time in blocks of 15 continuous minutes

Table 10: Basic Fatigue Management.

Id	**Number of Breaks**	**Duration**	**Interval**
BFM	1	00:15	06:15
BFM	2	00:15	09:00
BFM	4	00:15	12:00

Table 11: Example break scheme.

Table 11 shows how these break rules are specified.

- 1 x 15 minute break within 6.25 hours of starting the shift.
- 2 x 15 minute breaks within 9 hours of starting the shift.
- 4 x 15 minute breaks within 12 hours of starting the shift.

The above break scheme is defined in the *Fatigue Management* sheet of the accompanying workbook.

Other aspects of breaks apply to how, and where, they can occur. While a break scheme can be shared between vehicles, these other aspects are controlled on a per-vehicle basis. Each vehicle has the following break related options:

- **Paid Breaks**: The cost per hour is considered when allocating breaks.
- **Forbid Loaded Breaks:** Breaks cannot occur while orders are loaded on the vehicle.
- **Forbid Breaks During Transit:** Breaks cannot occur between locations.

These can be seen in the *Fleet* sheet of the accompanying workbook.

Enabling fatigue management increases the time needed to solve a scenario.

For this example, we have increased the number of orders from twenty to forty to ensure that there is sufficient work available for driver breaks to be needed.

KPI	Value
Total Cost	587.61
Total Distance	267.81
Work Time	14:26
Transit Time	05:41
Break Time	00:15
Assigned Orders	40 / 40
Fleet Usage	2 / 5

Table 12: Solution KPIs.

Table 12 contains a new KPI that shows the total amount of break time in the schedule.

3.8 Extension 7: Batched Loads

Accompanying Workbooks: pdp_extension_7a.xlsx, pdp_extension_7b.xlsx

So far, Opturion Freight has been making deliveries from their depot to customer locations. Wanting to increase their business, they now start picking up orders from customers and delivering them to the depot for next day delivery. In previous examples, the optimiser was running in *batched loads* mode. This is its default mode.

Batched loads are a scheduling constraint that requires that once a vehicle begins making deliveries, it cannot pick up new orders until it has delivered all its currently loaded orders.

A *load* is defined to be a sequence of `PICKUP` stops followed by a sequence of `DELIVERY` stops such that after the first `DELIVERY` stop occurs, no further `PICKUP` stops may occur until the vehicle is empty.
Example:

Stop Number	Load	Not a Load
1	`PICKUP`	`PICKUP`
2	`PICKUP`	`PICKUP`
3	`PICKUP`	`DELIVERY`
4	`DELIVERY`	`PICKUP`
5	`DELIVERY`	`DELIVERY`
6	`DELIVERY`	`DELIVERY`

Table 13: Batched loads example.

For depot-oriented operations, it makes sense to minimise the number of times a vehicle returns to the depot to re-load. (Remember, each -isit to the depot incurs the thirty-minute site time.) Now that there is the potential to pick up orders from locations where a vehicle is already making a delivery, we will turn off *batched loads*. This is done by setting the *Batched Loads* parameter in the *General* sheet to `False`.

Note: if *batched loads* had been turned off in the previous scenarios, then the optimiser might have still found the same solutions but it would have been slower due to the increase in choices available.

Note 2: it is recommended that you have *batched loads* enabled, unless you need to allow pickups between deliveries.

KPI	Non Batched Loads Value	Batched Loads Value
Total Cost	500.43102	530.4748
Total Distance	216.81	238.08
Work Time	11:09	12:16
Transit Time	04:44	05:21
Break Time	00:15	00:15
Idle Time	00:00	00:00
Assigned Orders	30 / 30	30 / 30
Fleet Usage	2 / 5	2 / 5

Table 14: Solution KPIs.

Table 14 compares solutions for the same PDP with batched loads enabled and disabled. We can see that the non-batched loads solution is cheaper, as it requires less distance and time to service all the orders. For this problem, we do not see a large drop in cost because the orders are large in comparison to the capacity of the vehicles. This limits the opportunity to pick up orders "on the fly".

3.9 Extension 8: Subcontractors

Accompanying Workbook: pdp_extension_8.xlsx.

Let us now consider the situation where Opturion Freight has the option to use subcontractors to make deliveries in addition to also using their existing fleet. Subcontractors are paid a flat rate of $25 per delivery.

We can model this situation by adding the subcontractor vehicles to the *Fleet* sheet and setting all of their costs to zero. We then add a new sheet, *Order Vehicle Costs*, to the workbook. Table 15 shows an excerpt from this new sheet.

Order	Vehicle	Cost
Order1	Sub_1	25
Order2	Sub_1	25
Order3	Sub_1	25
Order4	Sub_1	25

Table 15: Order-vehicle costs.

Each row of this sheet specifies a cost that is incurred if the order given in the *Order* column is assigned to the vehicle in the *Vehicle* column. The cost incurred is given by the *Cost* column.

In the accompanying workbook, we have set up a cost of $25 between every order and each of the two subcontractor vehicles, *Sub_1* and *Sub_2*.

KPI	Value
Total Cost	946.48
Total Distance	457.97
Work Time	29:49
Transit Time	10:14
Break Time	01:00
Idle Time	00:55
Assigned Orders	80 / 80
Fleet Usage	4 / 5

Table 16: Solution KPIs.

In Table 16, we can see that the *Idle Time* solution KPI is now non-zero for the first time. Idle time is time spent waiting for a pickup or delivery. It usually results from waiting for a time window. Previous solutions have avoided idle time. However, in this case, because the subcontractors are not paid an hourly rate, it is cheaper to assign some orders to them even though they will have to wait.

3.10 Extension 9: Vehicle Zoning

Accompanying Workbook: pdp_extension_9.xlsx.

Opturion Freight wants to restrict each vehicle to make deliveries only in its own specific area or areas. This is called *vehicle zoning*. To do this, we label each order with an *attribute* identifying the zone in which its delivery location is located. We then label each vehicle with attributes identifying the zones it is allowed to deliver in. Finally, we set up order-vehicle compatibility rules between the zone attributes on the orders and vehicles. Attributes and order-vehicle compatibility are described in Section 3.6.

Table 17 shows the delivery zones that Opturion Freight wants to use. Listed next to each zone are the vehicles that are allowed to make deliveries in that zone. Here, *Vehicle1* is only allowed to make deliveries in *Zone_1* or *Zone_2*. Deliveries in *Zone_2* may be made by either *Vehicle1* or *Vehicle2*.

Zone	Locations		Allowed Vehicles
Zone_1	CHELTENHAM	MENTONE	Vehicle1
Zone_2	SANDRINGHAM	HAMPTON	Vehicle1, Vehicle2
Zone_3	MOORABIN	HIGHET	Vehicle2
Zone_4	ORMOND	BENTLEIGH	Vehicle2
Zone_5	BALACLAVA	RIPPONLEA	Vehicle3
Zone_6	WINDSOR	PRAHRAN	Vehicle3
Zone_7	SOUTH YARRA	RICHMOND	Vehicle4
Zone_8	HAWTHORN	AUBURN	Vehicle4
Zone_9	IVANHOE	HEIDELBERG	Vehicle5
Zone_10	TOORAK	ARMADALE	Vehicle5

Table 17: Zone locations and vehicles.

In the accompanying workbook, we set up the above zoning by first adding the zone attributes to the *Attributes* column of the *Orders* sheet.

Id	Delivery Location	Attributes
Order1	CHELTENHAM	Zone_1
Order2	MENTONE	Zone_1
Order3	SANDRINGHAM	Zone_2
Order4	HAMPTON	Zone_2
Order5	MOORABIN	Zone_3

Table 18: Order attributes for zones.

Table 18 shows the first five orders in the *Orders* sheet. The attribute identifies what zone an order's delivery location is in.

We then label the vehicles in the *Fleet* sheet with attributes identifying what zones the vehicle is allowed to deliver to.

Id	Attributes
Vehicle1	Zone_1;Zone_2
Vehicle2	Zone_2;Zone_3;Zone_4
Vehicle3	Zone_5;Zone_6
Vehicle4	Zone_7;Zone_8
Vehicle5	Zone_9;Zone_10

Table 19: Vehicle attributes for zones.

Table 19 shows an excerpt from the *Fleet* sheet in the accompanying workbook. The *Attributes* column lists which zones each vehicle can deliver to. Vehicles that can deliver to multiple zones have each zone attribute separated by a semicolon.

Compatibility between the *order* and *vehicle* attributes is defined in the *Order-Vehicle Compatibility* sheet. The *order* attributes are in the left column and the *vehicle* attributes are in the first row.

Note: the *Order-Vehicle Compatibility* sheet only lets you define *positive* relationships between attributes. For further details, refer to "Opturion Dynamic Transport Optimiser – Data Formats".

Table 20 is a summary of the delivery locations visited by each vehicle. Each location visited by a vehicle is in one of the zones the vehicle is allowed to deliver to.

Vehicle	**Delivery Locations**					
Vehicle1	CHELTENHAM	MENTONE				
Vehicle2	BENTLEIGH	HAMPTON	HIGHET	MOORABIN	ORMOND	SANDRINGHAM
Vehicle3	BALACLAVA	PRAHRAN	RIPPONLEA	WINDSOR		
Vehicle4	AUBURN	HAWTHORN	RICHMOND	SOUTH YARRA		
Vehicle5	ARMADALE	HEIDELBERG	IVANHOE	TOORAK		

Table 20: Delivery locations visited.

In this example, we used zoning to restrict where vehicles can deliver. We can apply zoning to pickups in a similar manner.

3.11 Extension 10: Load Zoning

Accompanying Workbook: pdp_extension_10.xlsx.

In the previous example, we used *order* and *vehicle* attributes to define vehicle delivery zones. Sometimes, we do not want to restrict where a vehicle operates but do want to ensure that all the deliveries in a load occur in a given area. This is called *load zoning*.

As with *vehicle zoning*, we label the orders with zone attributes but this time we use the *Order-Order Compatibility* sheet to require that only orders with compatible zones are loaded on the vehicle at the same time.

The *Order-Order Compatibility* sheet must be symmetric. The order in which attributes are listed in the first column *must* match the order in which they are listed in the first row.

Table 21 shows an excerpt from the *Order-Order Compatibility* sheet of the accompanying workbook. A cell containing `TRUE` means that the orders with the attributes in that column and row are compatible with each other. In this example, *Zone_2* is compatible with *Zone_3*. This means that orders labelled with *Zone_2* and *Zone_3* are allowed to be loaded on the vehicle at the same time. Note that *Zone_3* is also compatible with *Zone_4*, but because *Zone_4* is *not* compatible with *Zone_2*, orders from *Zone_2* and *Zone_4* cannot be loaded on the vehicle at the same time.

	Zone_1	**Zone_2**	**Zone_3**	**Zone_4**	**Zone_5**
Zone_1	TRUE	TRUE			
Zone_2	TRUE	TRUE	TRUE		
Zone_3		TRUE	TRUE	TRUE	
Zone_4			TRUE	TRUE	TRUE
Zone_5				TRUE	TRUE

Table 21: Order-order compatibility.

Load	**Order**	**Zone**
1	Order9	Zone_5
	Order12	Zone_6
	Order10	Zone_5
	Order11	Zone_6
2	Order1	Zone_1
	Order4	Zone_2
	Order2	Zone_1
	Order3	Zone_2
3	Order8	Zone_4
	Order6	Zone_3
	Order5	Zone_3
	Order7	Zone_4
4	Order16	Zone_8
	Order15	Zone_8
	Order14	Zone_7
	Order13	Zone_7
5	Order20	Zone_10
	Order18	Zone_9
	Order17	Zone_9
	Order19	Zone_10

Table 22: Load zones visited.

Table 22 is a summary of the orders in each load. You can see that each order is zone compatible with all the other orders in its load.

3.12 Extension 11: Order Precedences

Accompanying Workbook: pdp_extension_11.xlsx.

Opturion Freight has won a new contract that requires orders to be picked up from a factory in Albury via line haul, cross-docked at their depot and then delivered to customers in metropolitan Melbourne. Cross-docked orders take at least one hour to process at the depot.

To handle this, we treat each *stage* of a cross-docked order as a separate order within the optimiser. Here, we will have:

1. Stage 1: Albury → Depot.
2. Stage 2: Depot → Customer.

We need to ensure the following:

1. Stage 1 must be completed before stage 2 begins.
2. There is at least one hour between the delivery of stage 1 to the depot and the pickup of stage 2.
3. Stage 2 can only be assigned to a vehicle if stage 1 has also been assigned to a vehicle.

Figure 14 illustrates the handling of each cross-docked order.

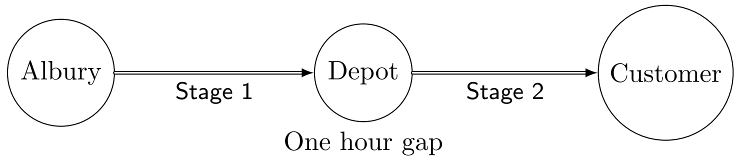


Figure 14: Cross-docking with order precedences.

We use *order precedence* relationships to do this. An order precedence relationship is a pair of orders, together with an optional duration called the *gap*. We call the orders in the pair *before* and *after*. The optimiser schedules the delivery of *before* so that it will be completed prior to the pickup of *after*. The *gap* is the minimum amount of time that must elapse between the completion of the delivery of *before* and the pickup of *after*. The optimiser will only generate solutions where both orders are assigned, or where both orders are not assigned. You cannot have a solution where, for example, *after* is assigned but *before* is not.

In the accompanying workbook, we have added a sheet named *Order Precedences*. This sheet lets you define order precedence relationships. It has three columns: the *Before* and *After* columns contain the identifiers of the orders in the precedence relationship and the *Gap* column specifies the duration of the gap. Here, we set the gap to 1:00 because at least one hour must elapse between the completion of the delivery of *before* and the pickup of *after*.

Table 23 shows the delivery completion times for the "Stage 1" orders, the pickup times for the "Stage 2" orders and the resulting gap. You can see that the pickup of every "Stage 2" order commences at least one hour after the completion of the delivery of the corresponding "Stage 1" order.

Order	**Stage 1 Delivery Completion**	**Stage 2 Pickup**	**Gap**
Order1	09:57	11:27	01:30
Order2	10:02	13:22	03:20
Order3	10:07	13:22	03:15
Order4	10:32	13:22	02:50
Order5	10:17	11:27	01:10
Order6	10:27	11:27	01:00
Order7	10:22	11:27	01:05
Order8	10:12	11:27	01:15
Order9	10:37	13:22	02:45
Order10	10:42	13:22	02:40

Table 23: Solution precedence gaps.

3.13 Extension 12: Replanning

Accompanying Workbook: pdp_extension_12.xlsx.

Opturion Freight are using the DTO to plan the next day's delivery schedule. However, sometimes it is necessary to update or extend a schedule generated by the optimiser. Reasons for this include:

- Orders being added.
- Orders being cancelled.
- Order details (e.g. weights, volumes, time windows etc.) changing.
- Vehicles being added to the fleet.
- Vehicles being removed from the fleet.
- Vehicle or driver availability changing.

When this happens, it is often desirable to maintain as much of the existing schedule as practicable. For example, if loads are already being assembled at the loading docks of a warehouse when changes occur, altering the original schedule too much may unduly affect warehouse operations.

This usage of the optimiser is called *replanning*. When replanning, a scenario is extended with information about the assignments and sequencing from a previous schedule. The optimiser can be configured to generate a new schedule that minimises the differences with the previous schedule while incorporating any changes.

In the accompanying workbook, we have added a previous schedule by copying and pasting the *Runsheet* sheet from an Excel solution workbook. We have also added some new global parameters to the *General* sheet. These parameters control the extent to which changes from the previous schedule can occur. The new parameters are:

1. **Previous-Vehicle Fixed Reward**

   Gives the strength of the preference for orders to remain on their previously assigned vehicles.

2. **Previous-Vehicle Weight Reward**

   Gives the strength of the preference for heavier orders to remain on their previously assigned vehicles.

3. **Previous-Vehicle Volume Reward**

   Gives the strength of the preference for larger orders to remain on the their previously assigned vehicles.

Each of the above is a form of *hidden cost* and their effect is cumulative.

The output workbook for the new schedule will contain an additional sheet named *Replan Changes*. This sheet contains a summary of changes between the previous and new schedules. Table 24 shows the *Replan Changes* sheet for the schedule generated from the accompanying workbook. You can see that five orders previously assigned to "VEHICLE1" are now assigned to different vehicles in the new schedule because"VEHICLE1" is no longer available.

Order	**Weight**	**Volume**	**Previous Vehicle**	**New Vehicle**	**Previous Pickup Time**	**New Pickup Time**	**Previous Delivery Time**	**New Delivery Time**
ORDER16	5	5	VEHICLE1 <UNAVAILABLE>	VEHICLE4	8:29	7:17	8:41	7:29
ORDER94	5	5	VEHICLE1 <UNAVAILABLE>	VEHICLE2	16:49	14:27	17:03	14:41
ORDER96	5	5	VEHICLE1 <UNAVAILABLE>	VEHICLE3	17:16	13:39	17:28	13:51
ORDER98	5	5	VEHICLE1 <UNAVAILABLE>	VEHICLE2	17:39	16:13	17:47	16:21
ORDER12	5	5	VEHICLE1 <UNAVAILABLE>	VEHICLE3	7:45	13:39	8:07	14:15

Table 24: Replanning Summary.

4 What Next?

Congratulations! You have now used the Opturion Dynamic Transport Optimiser to build and optimise a series of PDP scenarios. Hopefully, you have gained an understanding of the DTO and its capabilities. The DTO has applications in many other situations, for example carrier selection, determining fleet composition and modelling situations where there are separate tractors (prime movers) and trailers.

The examples in this guide use only a small subset of the DTO's features. It has many others, like:

- Compatibility relationships between orders; restricting what orders can be loaded on a vehicle at the same time.
- Vehicles with compartments.
- Time-of-day dependent travel times and speed restrictions.
- Load ordering, for example *last in, first out* orderings.
- Loading bays.
- Soft time windows.
- "Shaping" of vehicle routes.
- Multi-day operation.
- A RESTful web API, allowing integration with third-party platforms, such as an ERP system or TMS.

For details on how to include these features in your scenarios, see "Opturion Dynamic Transport Optimiser – Data Formats"

The examples in this guide already contain the latitude and longitude of each location. If you require geocoding then please refer to Appendix A.

If you are interested in learning more, please contact Opturion to discuss the best options for your business.

A Geocoder

Geocoding is the process of converting a street address, or place name, into latitude and longitude coordinates. These coordinates are needed by the optimiser to calculate the travel times and distances between locations.

If your locations are already geocoded, then you can include their latitudes and longitudes in the Locations sheet of your input workbook and skip this appendix. If you are missing coordinates for some, or all, of the locations, then you can use the Opturion Geocoder to obtain them.

The geocoder has a simple web interface. It accepts input workbooks in the DTO-Excel format. At minimum, the input workbook must contain the General and Locations sheets. Additional sheets may be included, but they are ignored by the geocoder.

In the following sections, we describe the geocoder web interface, define the notion of a *geofence* and specify the subset of the DTO-Excel format that must be present for the geocoder to work.

A.1 Geocoder Interface

The *Geocoder* interface lets you submit multiple jobs for geocoding. Their progress is updated every ten seconds. Figure 15 shows the control buttons at the top of the geocoder interface.

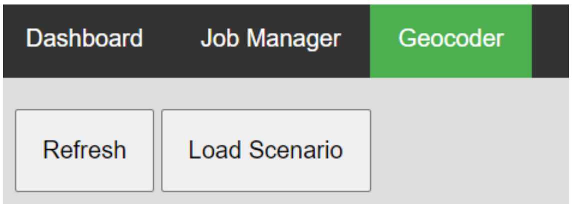


Figure 15: Geocoder interface.

The links labelled *Dashboard* and *Job Manager* will take you to the Dashboard and Job Manager interfaces respectively.

Clicking the *Refresh* button will update the status of in-progress geocoding jobs. Note that this does not affect the automatic timer which will continue to update progress every ten seconds.

The *Load Scenario* button lets you submit new scenarios for geocoding in the DTO-Excel input format.

Figure 16 shows the geocoder interface while a job is in-progress.

The *Download* button appears after a geocoding job is complete. Click it to download the result.

Figure 16: Geocoder progress.

A.2 Geofencing

The optimiser can be restricted to only consider orders and vehicles that are located within a specific geographic area. This area is called a *geofence*. It takes the form of a bounding box that is defined by the latitudes of its southern and northern boundaries, and by the longitudes of its western and eastern boundaries.

The geofence can operate in one of two modes: *hard* or *soft*.

In *hard* mode, the optimiser will report an error for orders that have pickup or delivery locations outside the geofence. It will also report an error for a vehicle that has start or finish locations outside the geofence. In the event of such errors being detected, optimisation will *not* proceed.

In *soft* mode, the optimiser will not assign an order whose pickup or delivery location is outside the geofence. It will not use a vehicle whose start or finish location is outside the geofence. In this mode, optimisation will proceed using only those orders and vehicles that are within the geofence.

The geocoder can be configured with the same geofence as the optimiser. It will report an error for locations outside the geofence.

The geocoder also has an additional geofence that is defined by the map data that the instance of DTO is configured to use. For example, if the DTO instance is set up to use Australian map data then locations that lie outside of Australia are flagged with an error.

A.3 Sheet: General

This sheet is required by the geocoder. It only recognises the following parameters; all others are ignored.

A.3.1 Parameter: Geofence North

Required: No.
Value Type: Number in [-90, 90].
Default: none.
Description: Latitude of the northern boundary for the geofence.

A.3.2 Parameter: Geofence East

Required: No.
Value Type: Number in [-180, 180].
Default: none.
Description: Longitude of the eastern boundary for the geofence.

A.3.3 Parameter: Geofence South

Required: No.
Value Type: Number in [-90, 90].
Default: none.
Description: Latitude of the southern boundary for the geofence.

A.3.4 Parameter: Geofence West

Required: No.
Value Type: Number in [-180, 180].
Default: none.
Description: Longitude of the western boundary for the geofence.

A.3.5 Parameter: Hard Geofence Errors

Required: No.
Value Type: Boolean.
Default: `FALSE`.
Description: If `TRUE`, then any geocoded locations that lie outside the geofence are flagged in the output.

A.4 Sheet: Locations

This sheet is required and must contain at least one location. Only the following columns are recognised by the geocoder; all others are ignored.

A.4.1 Column: Id

Required: Yes.
Type: Text or (integer) number.
Description: The unique identifier for the location.

A.4.2 Column: Address

Required: Yes.
Type: Text.
Description: The full street address of the location.

A.4.3 Column: Longitude

Required: No.
Type: Number.
Description: Longitude of the location, in decimal degrees.

A.4.4 Column: Latitude

Required: No.
Type: Number.
Description: Latitude of the location, in decimal degrees.

A.5 Geocoder Response

The response from the geocoder is a DTO-Excel workbook that contains the contents of the input workbook along with some additional columns added to the Locations sheet. These additional columns are:

A.5.1 Column: Geocoded Address

Always Present: Yes.
Type: Text.
Description: The address as recognised by the geocoding service. Cells in this column will be blank for locations where geocoding errors occurred.

A.5.2 Column: Geocoder Errors

Always Present: No.
Type: Text.
Description: The description of any errors encountered during geocoding.

A.5.3 Column: Longitude

Always Present: Yes.
Type: Number.
Description: Longitude of the location, in decimal degrees. Cells in this column will be blank for locations where geocoding errors occurred.

A.5.4 Column: Latitude

Always Present: Yes.
Type: Number.
Description: Latitude of the location, in decimal degrees. Cells in this column will be blank for locations where geocoding errors occurred.

A.6 Geocoder Example

Accompanying Workbook: geocoder_examples.xlsx.

The accompanying workbook contains a scenario that requires geocoding. It defines a geofence that restricts locations to lie within the state of Victoria, Australia. We have enabled a hard geofence, so an error is reported for any location that lies outside the geofence.

Figure 17 contains the response from the geocoder that results when the accompanying workbook is uploaded. Of particular interest are the three rows that have been highlighted to indicate the presence of a geocoding error.

- **Row 4: "Bad Address"**
  The error in this row means that the geocoder could not recognise the address in the input. The row is highlighted in red to indicate a failure.
- **Row 5: "Sydney, Australia"**
  The error in this row means that although geocoding was successful, the resulting coordinates lie outside of the user-defined geofence. The row is highlighted in amber to indicate a warning.
- **Row 6: "New York"**
  The error in this row indicates that although geocoding was successful, the resulting coordinates lie outside of the geofence defined by the map data with which the DTO instance is configured. The row is highlighted in amber to indicate a warning.

	A	B	C	D	E	F
1	**Id**	**Address**	**Geocoded Address**	**Geocoder Errors**	**Longitude**	**Latitude**
2	DEPOT	18 Kavanagh St Southbank	18 Kavanagh St, Southbank VIC 3006, Australia		144.9660451	-37.8226005
3	CUSTOMER1	145 Studley Rd, Heidelberg VIC 3084	145 Studley Rd, Heidelberg VIC 3084, Australia		145.0592495	-37.7566762
4	CUSTOMER2	Bad Address		Could not geocode address.		
5	CUSTOMER3	Sydney, Australia	Sydney NSW, Australia	Outside User Geofence.		
6	CUSTOMER4	New York	New York, NY, USA	Outside Router Geofence.		
7	CUSTOMER5	3/55 Victoria Parade, Fitzroy VIC 3065	3/55 Victoria Parade, Fitzroy VIC 3065, Australia		144.9756615	-37.8077088
8	CUSTOMER6	33 Ryot St, Warrnambool VIC 3280	33 Ryot St, Warrnambool VIC 3280, Australia		142.4740338	-38.380208
9	CUSTOMER7	101 Drummond St N, Ballarat Central VIC 3350	101 Drummond St N, Lake Wendouree VIC 3350, Australia		143.8464779	-37.5577926
10	CUSTOMER8	100 Barnard Street, North Bendigo VIC 3550	100 Barnard St, Bendigo VIC 3550, Australia		144.2802625	-36.749334
11	CUSTOMER9	77B Railway Cres, Williamstown VIC 3016	Williamstown Hospital, 77B Railway Cres, Williamstown VIC 3016, Australia		144.8926785	-37.8637396
12	CUSTOMER1(	51 FILLO DRIVE, Sommerton, VIC 30062	51 Fillo Dr, Somerton VIC 3062, Australia		144.9562244	-37.6408774

Figure 17: Geocoder response.