

---

# Analytical Business Modelling Assignment 3

---

*Students:*

Louis CARNEC

Vijay KATTA

Adedayo ADELOWOKAN

*Student #:*

15204934

15202724

15204151

April 29, 2017

## 1 Introduction

In the real-world companies are faced with problems regarding the transportation of goods, people or information. A couple of important challenges companies are faced with are finding ways to increase profits, reduce costs and further increase customer satisfaction. One of the many ways companies can save costs is in having an effective supply chain management which involves choosing distribution center locations or improving their delivery system.

The choice of locations can reduce transportation costs, and reduce lead times. By reducing lead times, inventory control becomes easier which in turn increases service level [4]. The factors for choosing a location go beyond lead time and reduction of costs, factors like the surrounding community also have a significant effect. For example, the strength of the infrastructure (road etc.) surrounding the location, reliability, congestion and vulnerability. An area with a strong infrastructure that is further away could be more beneficial to an area that is closer with weak infrastructure. These are the type of factors that could affect the supply chain, in service level, costs etc.

Improving the delivery system would also bring the same benefits as choosing the right location for a depot. In this case, a company would need a Vehicle Routing Problem (VRP) solution. VRP dates to the late 50s, although there has been nearly 60 years of research in this phenomenon large-scale versions of this problem still poses a challenge for the scientific community.

Vehicle Routing Problem tries to find a way to visit several locations given a certain number of vehicles in a cost-effective way. The simplest definition states that every customer is visited by one vehicle, and each vehicle does one trip starting and ending at the depot. The question now is what order should customers be visited while ensuring that the various constraints are considered. This is the problem we will be addressing in this assignment.

## 2 Application: Goods Delivery

The case study problem we have tackled is the delivery of goods throughout Irish cities and towns from a single depot location based in Dublin given a general time window constraint on delivery times. A single vehicle, with a given carrying capacity, must deliver goods to all towns/nodes in the graph and meet each town's demand for goods, that is deliver an amount of goods equal to that city's demand. Given the fact that each vehicle has a given capacity, the transportation vehicle can only take so many goods and must therefore revert back to the depot to pick up more goods to deliver. Further, deliveries must be conducted within a maximum number of working hours and the vehicle must return to the depot. Therefore, each 'trip', if forced by the time constraint on working hours available, is a working day. Here a 'trip' is a succession of edges connecting nodes that leaves and returns to the depot once.

Our problem involves finding the optimal route for the vehicle so that costs are minimised, where costs incurred stem from the the time spent travelling across the route,

the sum of the ‘trips’. This is a Vehicle Routing Problem (VRP), a variant of the Travelling Salesman Problem for which the *order* of nodes in a graph to be visited must be optimised to find the tour with lowest travelled distance.

In his chapter on vehicle routing, Cordeau warns that due to high variability of problems in practice, the objective function and constraints of VRP are highly variable and thus must be tailored to each problem [2].

We initially attempted to solve a slightly different version of the problem solved here and modelled in Section 3; the ‘Capacitated Vehicle Routing Problem’ [7]. In this problem multiple vehicles are used at the same time to deliver goods around the symmetric undirected graph. In our implementation we followed closely the implementation used in section 11.5 of ‘Applications of optimization with Xpress-MP’ [5] for planning flights. The model is similar to the one presented in this report, distance is minimised given that each node/city must only be visited once. To account for the fact that  $n$  trips can be made by  $n$  vehicles, an additional constraint is added to stipulate that the depot must have  $n$  incoming edges. Optimising the model, we were able to create the subtours for the graph to be ‘broken’ into  $n$  subtours. Having Dublin as the depot, the  $n$  closest cities to Dublin were used as outgoing/incoming nodes to be attached to the  $n$  subtours. When it came to making subtour elimination procedure into  $n$  subtours, we were unable to create a function which would allow us to form the subtour containing the depot to then break the smaller subtours.

### 3 Related Work

Combinatorial optimization problems have been studied in huge detail, the Vehicle Routing Problem was initially introduced in 1959 [3]. The needs of the transportation industry were the motivation behind the problem, especially realizing the amount of savings that could be made with small improvements in efficiency. Routing problems always have an abundance of complex constraints and variables, this is evident in the industry and our personal lives. These include traffic conditions, weather in certain situations, road-work etc. There are further constraints that would be specific to the industry which include shift limits for drivers, specified arrival and delivery times. The widely-studied version of VRPs are more simplistic, and utilize less constraints. The simpler versions still pose as a difficult task which shows how difficult it is to solve VRPs.

#### 3.1 Travelling Salesman Problem

The Travelling Salesman Problem (TSP) is one of the oldest known routing problems. It presents a salesman that has a specified number of cities to visit and needs to know the optimal order to visit these cities to minimize the distance needed to travel. TSP is an NP-hard problem [6].

### 3.2 Vehicle Routing Problem

VRP is also an NP-hard problem but it is safe to assume that it is a much harder problem to solve over TSP. VRP can be applied to various fields; logistics, communications, manufacturing, transportation, to list a few. In a VRP there are several vehicles that need to visit many customers. Vehicles start at depot before visiting a section of the customers before returning to the starting depot. Capacitated Vehicle Routing Problem (CVRP) is a classic problem, this involves finding a solution to a transport problem where customers can be reached via multiple identical vehicles with different capacity restrictions. A special case of VRP is Vehicle Routing Problem with Time Windows (VRPTW). VRPTW comes with its own added complexity, each customer has a start and end time which indicates the time the vehicle should be servicing. An example of this application would be in case of a company that delivers heating oil. Although this constraint is hard, the vehicle does not need to arrive between that time, it can arrive before the start time but must remain inactive for a given period. Like TSP, VRP is an NP-hard problem [6] but it is safe to assume that the VRP is a much harder problem to solve than TSP.

## 4 Mathematical Programming Model Formulation

The formulation proposed in this report is an extension of the formulation proposed in the ‘Applications of optimization with Xpress-MP’ book in section 10.4 [5]. The formulation is extended by; adding a general time window which applies to all cities within the graph (as opposed to individualised time windows) and thus a constraint forcing the vehicle to return to the depot within the time window, an additional constraint to specify the number of times the vehicle can return to the depot and a probability parameter simulating the random probability of being stuck in traffic on a given leg of the trip.

### 4.1 Assumptions

Several assumptions are made in modelling this VRP problem.

We assume that the time taken to travel from one city to another is linearly proportional to the distance by air between the two points, as such in the mathematical formulation of the model distance, rather than time is minimised. It is also assumed that no time is lost at each node, that is the time taken for the route is equal to the time taken proportional to the distance. However, a fixed cost is incurred at each stop. This assumption means that for a route of length  $x$ , the time taken is the same whether there were 5 stops or 10 stops along the route. It is assumed that the distance covered over time does not affect cost. Cost is only related to time spent travelling, plus a fixed cost.

## 4.2 Objective Function

The objective function (1) we wish to minimise is the cost of delivering goods throughout the graph.

$$\text{Minimise } Cost = \sum_{i \in cities} \sum_{j \in cities, i \neq j} c_{i,j} p_{i,j} \quad (1)$$

Where,

$$c_{i,j} = (d_{i,j}/kmph_{i,j}) * cph + \bar{c}$$

$c_{i,j}$  is the cost of travelling from node  $i$  to node  $j$ , which is given by the time taken to travel from  $i$  to  $j$  (distance / speed) times the cost per hour  $cph$  for the given leg journey, plus a fixed cost for each stop  $\bar{c}$ .

There are three decision variables in this model,  $p_{i,j}$  which states whether node  $j$  is a successor to node  $i$ , the total quantity at node  $i$  ( $q_i$ ), and the total time spent delivering at node  $i$  ( $t_i$ ).

$$p_{i,j} \in \{0, 1\} \quad , \forall i, j \in cities \quad , \text{where } i \neq j \quad (2)$$

$p_{i,j}$  in (2) is a binary decision variable which states that the vehicle travels from node  $i$  to node  $j$  if  $p_{i,j} = 1$ . The total quantity delivered at  $i$  ensures that the quantity delivered is equal to demand for each node and that each trip in the tour is conducted while respecting the maximal capacity of the vehicle. The total time delivering decision variable is similar to the total quantity delivered variable and ensures that the route respects the fact that there is a general time window to be respected for delivery and that the vehicle must return to the depot within this time window. By solving for these variable, the model should return the shortest route which respect to the constraints presented below.

## 4.3 Constraints

Equations (3) and (4) state that each city (except from the depot, which is always node 1) must be entered and departed exactly once, so that every city is delivered to but that the tour is not unnecessarily extended by visiting nodes which have already been visited. Note that  $i, j \neq 1$  due to the fact that the depot can be entered and departed multiple times. The vehicle should return to the depot once it has delivered all goods within its capacity ( $C$ ) or worked for a given number of hours ( $H$ , see Table 1).

$$\sum_{i \in cities} p_{i,j} = 1 \quad , \forall j \in cities \quad , j \neq 1 \quad (3)$$

$$\sum_{j \in cities} p_{i,j} = 1 \quad , \forall i \in cities \quad , i \neq 1 \quad (4)$$

#### 4.3.1 Increasing Total Quantity

The following constraints are used in finding the optimal tour given a set of constraints while solving for the decision variable total quantity, the total quantity delivered at a given node. The total quantity delivered at a given node ( $q_i$ ) must be less or equal to the maximal capacity ( $C$ ) of the delivering vehicle (7) and less or equal to the demand ( $D_i$ ) / or the amount to be delivered at that node (8). Constraint (7) forces the vehicle to take trips back to the depot if it must deliver a total demand greater than the total capacity of the vehicle.

Since total quantity delivered is a decision variable to be maximised within maximal capacity of the vehicle, constraint (5) ensures that the total quantity delivered if city  $i$  is the first city visited *in the trip* is equal to the demand at  $i$ , otherwise if  $i$  is not the first city delivered to, then the total quantity at  $i$  must be less or equal to the maximal capacity of the vehicle (while is already stated in (7)). That is (5), resets the total quantity variable at each trip to the demand of the first node visited within a new trip. Constraint (6) is a strictly increasing total quantity constraint. It states that if node  $j$  follows node  $i$ , the total quantity delivered at  $j$  must be the total quantity delivered at  $i$  plus the demand at  $j$ . This constraint ensures that at each node successor node the total quantity delivered increases and that each ‘trip’ is optimised.

$$q_i \leq C + (D_i - C) \times p_{1,i} \quad , \forall i \in cities \quad , i \neq 1 \quad (5)$$

$$q_j \geq q_i + D_i - C + C * p_{i,j} + (C - D_j - D_i) * p_{j,i} \quad , \forall i \in cities \quad , i, j \neq 1 \quad , i \neq j \quad (6)$$

$$q_i \leq C \quad , \forall i \quad , i \neq 1 \quad (7)$$

$$q_i \geq D_i \quad , \forall i \quad , i \neq 1 \quad (8)$$

#### 4.3.2 Increasing Total Time Constraint

The concept behind the total time of delivery constraints (9), (10) and (11) is the same as that of the (5) and (6) constraints for total quantity delivered. (9) ensures that the total time delivering at  $i$ , if  $i$  is a successor to the depot (node 1), is equal to the time taken to deliver to that node. Alternatively if  $i$  is not a successor to the depot node, the total time of delivery at  $i$  is less or equal to the hours available for delivering in a day. Constraint (11) is therefore made redundant in this case. Both constraints ensure that the vehicle returns to the depot within the delivering time limit allocated (before the end of the working day). The strictly increasing delivering time constraint (10) ensures that the total time of delivery at  $j$  is equal to the total time of delivery at  $j$ , plus the time taken to travel from  $i$  to  $j$ .

$$T_i \leq H + (t_{1,i} - H) \times p_{1,i} \quad , \forall i \in cities \quad , i \neq 1 \quad (9)$$

$$T_j \geq T_i + t_{i,j} - H + H * p_{i,j} + (H - t_{i,j})p_{i,j} \quad , \forall i \in cities \quad , i, j \neq 1 \quad , i \neq j \quad (10)$$

$$q_i \leq H \quad , \forall i \quad , i \neq 1 \quad (11)$$

#### 4.3.3 Number of times through the depot

Lastly, we present an optional constraint which forces the model to return a route solution with a given number of trips, or returns to the depot.

$$\sum_{i \in cities} p_{i,1} = n \quad (12)$$

Table 1: List of Variables

Decision Variables	Symbol	Description
Precedes	$p_{i,j}$	Binary Variable - 1 if $i$ precedes $j$
Total Quantity	$q_i$	Total Quantity Delivered at $j$
Total Time	$T_i$	Total Time Delivering at $j$
Variables	Symbol	Description
Distance	$d_{i,j}$	Distance from $i$ to $j$
Speed	$kmph_{i,j}$	Average Constant Speed of Vehicle from $i$ to $j$
Cost per hour	$cph$	Cost per hour of delivering
Fixed cost	$\bar{c}$	Fixed Cost per stop
Capacity	$C$	Capacity of vehicle
Demand	$D_i$	Demand / Quantity ordered at city $i$
Hours	$H$	Maximal Delivering Hours
Travel Time	$t_{i,j}$	Travel time from $i$ to $j$
Trips	$n$	Number of Trips

## 5 Mosel Model

The Mosel model used in this report is an application of the mathematical model presented above. It was run using both real and generated data. A probabilistic component was added to the travel between nodes/cities.

The city location and population data used to create the symmetric undirected graph of Irish cities was obtained from Tageo<sup>1</sup>. The data was cleaned in `python` and an

<sup>1</sup><http://www.tageo.com/index-e-ei-cities-IE.htm>

adjacency matrix of the distance ‘by flight’ between cities was calculated using the *vincenty*<sup>2</sup> distance (great-circle distance) from the *geopy* library.

Demand for each city was calculated within *Xpress Mosel* as being proportional to the population of each city with an added random component.

To model the fact that travel time between two cities may be affected by traffic, we added a function which would assign different average speeds to different legs of the route. A node was given an 20% chance of having an average speed of 60km/h and an 80% chance of having a 100km/h average speed.

A printing procedure was created which would output the total distance covered, quantity delivered and number of stop for each ‘trip’.

### **5.1 The model development process; what facilities/ features does the development environment have to aid model development and solution of your problem?**

*Xpress Mosel*, the development environment used in conducting this report, allowed us, the users, to easily check the status of our model which was beneficial in developing and finding a solution to our model.

The info bar, when an error occurs, outputs a range of errors in the code, this allowed us to quickly and easily debug our code. By right clicking on a variable within the text and showing values assigned to the variable, we were able to quickly check that each variable was correctly assigned, this was useful when working with a dummy data while building the model. The ‘Run Bar’ feature, to the right of the screen allowed us to check the solutions being produced when running the model by navigating to the ‘Solutions’ tab. The ‘Stats’ tab enabled us to check the status of the program without having to print it out. Finally the ‘Row View’ feature within the ‘Matrix’ tab allowed to test the effect of different vehicle capacity and time capacity parameters on the slack variables for each constraint.

### **5.2 How easy is it to verify correctness of the model and to separate the problem and its data**

Verifying the correctness of the model was relatively simple. We built the model in *Xpress Mosel* using a dummy dataset containing a 5-by-5 adjacency matrix. Examining the solutions in ‘solution’ tab within *Xpress*, we were able to verify that each of the model’s constraints were being respected and that indeed for the small dummy dataset the result of the minimised objective function was as expected.

Applying the *Mosel* model to our application of choice, after initialising the datafile containing the adjacency matrix for the distances of Irish cities, although the model itself remained the same, some of the model’s parameters had to be altered to fit the new data in order for the model to output a result, otherwise the model would keep running. Constraints (5, 6, 9 and 10), depend on the capacity parameter and on the

---

<sup>2</sup><https://geopy.readthedocs.io/en/1.10.0/>



time available parameter of the vehicle. For some values of these parameters with a given dataset, the model will be infeasible. Additionally, the optional constraint for the number of times the vehicle must pass by the depot must be feasible. For example, if demand at city 1 is 400 units and the capacity of the vehicle is 300 units, the problem will be infeasible and Mosel will not find a solution. The capacity of the vehicle needs to be increased for the model to function. Likewise, if the speed of the vehicle is 20 km/h, the working hours of delivery are of 8 hours, the maximum distance covered in the route is 160 km.

In order to make the program more versatile in implementing with different datasets, we would have like to have created if-loops which would raise an error in case the parameters provided do not allow for a feasible solution to be found. If the program did not stop, this was generally due to a parameter making the problem infeasible. In this case we had to use the ‘Row View’ tab within the environment to inspect each constraint in order to debug the program.

This versatility enables us to solve different problems. For example, we could solve a VRP, for picking up broken Dublin City Bikes from the various stations around Dublin. We would download the locations of the stations, create an adjacency matrix using the same `python` program used in creating the Irish cities adjacency matrix and solve for the quickest route.

### 5.3 How easy is it to either perform sensitivity analysis on the defined problem or to amend/extend the problem?

Amending and/or extending the problem presented here is feasible depending on the changes/extensions the user seeks to make. We have demonstrated that the model can be extended easily, by stipulating some additional constraints such as the number of times the vehicle must return to the depot and respecting time windows are easy. This is possible by altering/adding some constraints.

Further, the model’s parameters can easily be altered; such as the maximal capacity of the vehicle, the number of hours in the working day, the average speed of the vehicle, the probability of the time to go from one node to another increasing or decreasing.

However, for some changes/extensions, we would need a different model. If we wanted to change the model to use a number of  $n$  vehicles to find  $n$  tour for each vehicle so that each city is visited once, we would need to change the subtour elimination constraint completely.

Further, if we were to extend the model to a case study with many more nodes, we would need to adopt a different formulation such as tree search or a heuristic approach [5]. The model we are currently using would work for instances of up to 30 nodes only. Therefore, the Dublin Bikes case study proposed in the previous section would be infeasible as there are 102 stations.

## 5.4 What theoretical principles are demonstrated in your application?

The theoretical principles demonstrated in our application come from *network optimisation* and *stochastic optimization*. The distance between cities, produced in the form of an adjacency matrix, represent a graph of cities where the nodes are cities and edge weights are the distances between them. Using combinatorial optimisation, the optimal set of nodes in a specific order are returned while respecting a set of constraints. At the same time this application is a stochastic optimisation problem as there is the presence of randomness in both the demand and speed variables, that is the result of the optimisation, the total cost of the route, should vary across runs of the model as it is affected by randomness. Using a stochastic optimisation model of this type allows us to more realistically *simulate* the cost of a route. Running the model a number of times and getting a set of statistics which represent the results would enable us, the analysts, in making recommendations on best route to choose to minimise cost.

## 6 Results

We ran the **Xpress Mosel** model to find the optimal vehicle route on datasets of different size - 5, 20 and 30 cities, for two different scenarios -1) With Capacity Constraint and No Time Constraint and 2) With Capacity and Time Constraint. The distance between any two cities remains same in either direction, but the speed of the vehicle can be different and the model takes this into consideration while finding the minimal cost route. First line of the output is the total transportation cost of the goods delivered in all the trips from the warehouse. Second line has five different columns - route of journey (from city -to city), distance between the cities in kilo meters , speed in kilo meters per hour, time in hours and finally quantity of goods delivered. Other statics such as quantity delivered, distance covered and number of stops are for a given a trip from the central warehouse in Dublin. Executing each instance of 5 cities and 20 cities took less than a second and 30 cities took about 5 seconds. It is possible that the run time increases exponentially as the number of cities increases but for all practical purposes of delivering the goods the model works fine.

### 6.1 With Capacity Constraint but No Time Constraint

In the "5 cities" case we have 3 different executions of the model as shown in Figure 1 , Figure 2 and Figure 3. In each scenario total demand, generated randomly based on the population of the cities didn't exceed the maximum vehicle capacity of 1500 units. All the orders can be fulfilled by one trip but it can be seen in Figure 1 that there are three different trips taken from Dublin delivering quantities of 187, 357 and 266 units, covering a total distance of 1296 kilo meters. There are three stops in trip1, five in trip2 and two in the last and final trip. It is obvious that the function is optimising the cost of delivery so the number of trips doesn't matter as loading charges and opening hours are not considered. It is interesting to note that last leg of journey in every trip has a quantity of zero as the model caters for only deliveries from the warehouse to client locations but no

returns. In trip 3, though the distance is same from Dublin to Waterford and Waterford to Dublin, the time taken is 2.22 and 1.32 hours as the speed is different in each direction.

Scenario 2 (Figure 2) differs from scenario 1 (Figure 1) and scenario 3 (Figure 3) in terms of the number of trips taken. There is only one trip costing 640 euros covering a distance of 675 kilo metres, delivering 340 units of goods to 5 different cities. Figure 5, Figure 6 and Figure 7 are the results of applying the model on dataset with 20 cities, each with vehicle capacity constraint of 4000 units. Higher demand of 252 units in Figure 5 can be attributed to the random demand based on the population of Cork city. Output from Run1, Run2 and Run3 of this dataset shows that cost efficient way of fulfilling the demand is by one trip covering all the cities. Examining the reason as to why only one trip is optimal rather than multiple trips to warehouse, it can be concluded that model needs additional constraints such as road types, connectivity on top of existing distance and speed.

Cost optimal vehicle routing path based on 30 cities/towns dataset is shown in Figure 9, Figure 10 and Figure 11. Unlike the previous results using 5 cities and 20 cities, the output generated here in each instance of the run is different for each other. Figure 9 shows that first trip covers 26 cities/towns delivering 818 units covering 1235 kilo meters with two journeys having 60 kilo meter per hour speed and second trip delivers to 5 cities. In reality not all deliveries are of same priority, but we haven't modelled that component in so next location from a given city is based on distance and speed.

## 6.2 With Capacity Constraint and Time Constraint

The only difference in this case is that there is an additional time constraint of 9 hours on top of the capacity constraint of a goods delivery van. It means a vehicle's onward and return journey time from and to the warehouse cannot exceed 9 hours in a single trip. Figures 4, Figures 8 and Figures 12 shows the output of running the model with both constraints enable on datasets with 5, 20 and 30 nodes.

Exceptional cases such as order sizes greater than capacity of a delivery van are not handled but can be added to the model if need.

## 7 Conclusion/Recommendations

### The dependence on Software

*In OR practice and research, software is fundamental. The dependence of OR on software implies that the ways in which software is developed, managed, and distributed can have a significant impact on the field.*

The above statement can be equally applied to Analytical Business Modelling (ABM). The ability to create software to address some of the problems we encountered is very advantageous as trying to solve some of the problems using the simplex tableau would

be much more difficult. Utilizing software allows for the problem types to be more complex, this is evident in ABM as we can utilize different solvers, choosing one that is more suitable in our Mosel model to solve our problems. The inclusion of a debugger in Mosel, allows the developer/model creator to see locations where errors arise [1].

The way the software is managed also could also have an impact as making solutions that are reusable with minimum modification to solve a separate problem. For example, creating procedures that are common to many problem types, that way that one procedure can be used when solve those types of problems. This is important as having to rewrite code will prove to be a waste of time when they can be reused. Mosel also allows for the utilization of libraries which can lead to novel approaches been found out and implemented [1].

How the software is distributed can have an impact on the field, especially if the software if it's one that isn't native all operating systems. Also, the limitation in the available licenses proved to be a limitation, with the student license the max amount of points was 30. This can prove to limitation as there isn't the ability to further test your solution. In other words, we agree with this statement, how software is developed, managed and distributed can have a significant impact on ABM, in some cases positive and others negative.

## Appendix - Results

### Vehicle Routing - 5 Cities

With Capacity Constraint and No Time Constraint

Cost of Goods Delivery (All Trips) : 563.2 Euros

From(City) - To (City)	Distance(KM)	Speed(KMH)	Time(Hours)	Quantity
'Dublin' -> 'Cork'	218	100	2.18	117
'Cork' -> 'Waterford'	100	100	1.00	70
'Waterford' -> 'Dublin'	133	100	1.33	0
Quantity Delivered	: 187			
Distance Covered	: 451			
Number of Stops in Trip	: 3			
---TRIP---				
'Dublin' -> 'Galway'	187	100	1.87	42
'Galway' -> 'Limerick'	73	100	0.73	128
'Limerick' -> 'Cork'	86	100	0.86	117
'Cork' -> 'Waterford'	100	100	1.00	70
'Waterford' -> 'Dublin'	133	100	1.33	0
Quantity Delivered	: 357			
Distance Covered	: 579			
Number of Stops in Trip	: 5			
---TRIP---				
'Dublin' -> 'Waterford'	133	60	2.22	70
'Waterford' -> 'Dublin'	133	100	1.33	0
Quantity Delivered	: 70			
Distance Covered	: 266			
Number of Stops in Trip	: 2			

Figure 1: Goods Delivery Optimal Cost : With Capacity Constraint - Run1

Cost of Goods Delivery (All Trips) : 640 Euros

From(City) - To (City)	Distance(KM)	Speed(KMH)	Time(Hours)	Quantity
'Dublin' -> 'Galway'	187	100	1.87	46
'Galway' -> 'Cork'	158	100	1.58	155
'Cork' -> 'Limerick'	86	100	0.86	85
'Limerick' -> 'Waterford'	111	100	1.11	54
'Waterford' -> 'Dublin'	133	100	1.33	0
Quantity Delivered	: 340			
Distance Covered	: 675			
Number of Stops in Trip	: 5			

Figure 2: Goods Delivery Optimal Cost : With Capacity Constraint - Run2

```

Cost of Goods Delivery (All Trips) : 609.067 Euros
  From(City) - To (City)      |Distance(KM)|Speed(KMH)|Time(Hours)|Quantity
-----|-----|-----|-----|-----
'Dublin'    -> 'Cork'         |      218 |      100 |      2.18 |      190
'Cork'       -> 'Waterford'    |      100 |      100 |      1.00 |       71
'Waterford' -> 'Dublin'       |      133 |      100 |      1.33 |        0
Quantity Delivered      : 261
Distance Covered        : 451
Number of Stops in Trip: 3
---TRIP---
'Dublin'    -> 'Galway'       |      187 |      100 |      1.87 |       40
'Galway'     -> 'Limerick'    |       73 |      100 |      0.73 |       84
'Limerick'   -> 'Cork'        |       86 |       60 |      1.43 |      190
'Cork'       -> 'Waterford'    |      100 |      100 |      1.00 |       71
'Waterford' -> 'Dublin'       |      133 |      100 |      1.33 |        0
Quantity Delivered      : 385
Distance Covered        : 579
Number of Stops in Trip: 5

```

Figure 3: Goods Delivery Optimal Cost : With Capacity Constraint - Run3

### With Capacity Constraint and Time Constraint

Cost of Goods Delivery (All Trips) : 1465.07 Euros

From(City) - To (City)	Distance(KM)	Speed(KMH)	Time(Hours)	Quantity
'Dublin' -> 'Cork'	218	100	2.18	134
'Cork' -> 'Dublin'	218	100	2.18	0
Quantity Delivered	: 134			
Distance Covered	: 436			
Number of Stops in Trip	: 2			
---TRIP---				
'Dublin' -> 'Limerick'	175	100	1.75	106
'Limerick' -> 'Dublin'	175	60	2.92	0
Quantity Delivered	: 106			
Distance Covered	: 350			
Number of Stops in Trip	: 2			
---TRIP---				
'Dublin' -> 'Galway'	187	100	1.87	48
'Galway' -> 'Dublin'	187	100	1.87	0
Quantity Delivered	: 48			
Distance Covered	: 374			
Number of Stops in Trip	: 2			
---TRIP---				
'Dublin' -> 'Waterford'	133	60	2.22	54
'Waterford' -> 'Dublin'	133	100	1.33	0
Quantity Delivered	: 54			
Distance Covered	: 266			
Number of Stops in Trip	: 2			

Figure 4: Goods Delivery Optimal Cost : With Capacity Constraint and Time Constraint

## Vehicle Routing - 20 Cities

With Capacity Constraint and No Time Constraint

Cost of Goods Delivery (All Trips) : 1324.8 Euros

From(City) - To (City)	Distance(KM)	Speed(KMH)	Time(Hours)	Quantity
'Dublin' -> 'Naas'	31	100	0.31	21
'Naas' -> 'Bray'	38	100	0.38	35
'Bray' -> 'Swords'	27	100	0.27	15
'Swords' -> 'Drogheda'	31	100	0.31	38
'Drogheda' -> 'Dundalk'	33	100	0.33	31
'Dundalk' -> 'Navan'	43	100	0.43	23
'Navan' -> 'Athlone'	87	100	0.87	13
'Athlone' -> 'Sligo'	101	100	1.01	18
'Sligo' -> 'Galway'	117	100	1.17	96
'Galway' -> 'Ennis'	46	100	0.46	28
'Ennis' -> 'Limerick'	32	100	0.32	74
'Limerick' -> 'Tralee'	85	100	0.85	18
'Tralee' -> 'Cork'	95	100	0.95	252
'Cork' -> 'Waterford'	100	100	1.00	38
'Waterford' -> 'Clonmel'	41	100	0.41	17
'Clonmel' -> 'Kilkenny'	45	100	0.45	28
'Kilkenny' -> 'Wexford'	63	100	0.63	21
'Wexford' -> 'Carlow'	63	100	0.63	10
'Carlow' -> 'Newbridge'	38	100	0.38	11
'Newbridge' -> 'Dublin'	40	100	0.40	0
Quantity Delivered : 787				
Distance Covered : 1156				
Number of Stops in Trip: 20				

Figure 5: Goods Delivery Optimal Cost : With Capacity Constraint - Run1



Cost of Goods Delivery (All Trips) : 1286.13 Euros

From(City) - To (City)	Distance(KM)	Speed(KMH)	Time(Hours)	Quantity
'Dublin' -> 'Swords'	13	60	0.22	12
'Swords' -> 'Navan'	38	100	0.38	12
'Navan' -> 'Drogheda'	22	100	0.22	42
'Drogheda' -> 'Dundalk'	33	100	0.33	39
'Dundalk' -> 'Sligo'	137	100	1.37	12
'Sligo' -> 'Athlone'	101	100	1.01	12
'Athlone' -> 'Galway'	75	100	0.75	76
'Galway' -> 'Ennis'	46	100	0.46	15
'Ennis' -> 'Tralee'	81	100	0.81	28
'Tralee' -> 'Limerick'	85	100	0.85	119
'Limerick' -> 'Cork'	86	100	0.86	244
'Cork' -> 'Clonmel'	72	100	0.72	23
'Clonmel' -> 'Waterford'	41	100	0.41	35
'Waterford' -> 'Wexford'	45	60	0.75	11
'Wexford' -> 'Kilkenny'	63	100	0.63	13
'Kilkenny' -> 'Carlow'	30	100	0.30	20
'Carlow' -> 'Newbridge'	38	100	0.38	14
'Newbridge' -> 'Naas'	9	100	0.09	14
'Naas' -> 'Bray'	38	100	0.38	35
'Bray' -> 'Dublin'	16	100	0.16	0
Quantity Delivered	: 776			
Distance Covered	: 1069			
Number of Stops in Trip:	20			

Figure 6: Goods Delivery Optimal Cost : With Capacity Constraint - Run2

Cost of Goods Delivery (All Trips) : 1228.8 Euros					
From(City) - To (City)	Distance(KM)	Speed(KMH)	Time(Hours)	Quantity	
'Dublin' -> 'Naas'	31	100	0.31	22	
'Naas' -> 'Newbridge'	9	100	0.09	12	
'Newbridge' -> 'Carlow'	38	100	0.38	25	
'Carlow' -> 'Kilkenny'	30	100	0.30	24	
'Kilkenny' -> 'Wexford'	63	100	0.63	25	
'Wexford' -> 'Waterford'	45	100	0.45	56	
'Waterford' -> 'Clonmel'	41	100	0.41	9	
'Clonmel' -> 'Cork'	72	100	0.72	208	
'Cork' -> 'Tralee'	95	100	0.95	25	
'Tralee' -> 'Limerick'	85	100	0.85	114	
'Limerick' -> 'Ennis'	32	100	0.32	29	
'Ennis' -> 'Galway'	46	100	0.46	67	
'Galway' -> 'Athlone'	75	100	0.75	18	
'Athlone' -> 'Sligo'	101	100	1.01	25	
'Sligo' -> 'Dundalk'	137	100	1.37	26	
'Dundalk' -> 'Drogheda'	33	100	0.33	40	
'Drogheda' -> 'Navan'	22	100	0.22	11	
'Navan' -> 'Swords'	38	100	0.38	30	
'Swords' -> 'Bray'	27	100	0.27	25	
'Bray' -> 'Dublin'	16	100	0.16	0	
Quantity Delivered : 791					
Distance Covered : 1036					
Number of Stops in Trip: 20					

Figure 7: Goods Delivery Optimal Cost : With Capacity Constraint - Run3

## With Capacity Constraint and Time Constraint

Cost of Goods Delivery (All Trips) : 4914.13 Euros

From(City) - To (City)	Distance(KM)	Speed(KMH)	Time(Hours)	Quantity
'Dublin' -> 'Cork'	218	60	3.63	256
'Cork' -> 'Dublin'	218	100	2.18	0
Quantity Delivered	: 256			
Distance Covered	: 436			
Number of Stops in Trip: 2				
---TRIP---				
'Dublin' -> 'Limerick'	175	60	2.92	76
'Limerick' -> 'Dublin'	175	60	2.92	0
Quantity Delivered	: 76			
Distance Covered	: 350			
Number of Stops in Trip: 2				
---TRIP---				
'Dublin' -> 'Galway'	187	100	1.87	52
'Galway' -> 'Dublin'	187	60	3.12	0
Quantity Delivered	: 52			
Distance Covered	: 374			
Number of Stops in Trip: 2				
---TRIP---				
'Dublin' -> 'Waterford'	133	100	1.33	32
'Waterford' -> 'Dublin'	133	100	1.33	0
Quantity Delivered	: 32			
Distance Covered	: 266			
Number of Stops in Trip: 2				
---TRIP---				
'Dublin' -> 'Dundalk'	77	100	0.77	51
'Dundalk' -> 'Dublin'	77	100	0.77	0
Quantity Delivered	: 51			
Distance Covered	: 154			
Number of Stops in Trip: 2				
---TRIP---				
'Dublin' -> 'Bray'	16	100	0.16	38
'Bray' -> 'Dublin'	16	60	0.27	0
Quantity Delivered	: 38			
Distance Covered	: 32			
Number of Stops in Trip: 2				
---TRIP---				
'Dublin' -> 'Drogheda'	44	100	0.44	23
'Drogheda' -> 'Dublin'	44	100	0.44	0
Quantity Delivered	: 23			
Distance Covered	: 88			
Number of Stops in Trip: 2				
---TRIP---				
'Dublin' -> 'Swords'	13	60	0.22	17
'Swords' -> 'Dublin'	13	100	0.13	0
Quantity Delivered	: 17			
Distance Covered	: 26			
Number of Stops in Trip: 2				
---TRIP---				
'Dublin' -> 'Tralee'	261	100	2.61	13
'Tralee' -> 'Dublin'	261	100	2.61	0
Quantity Delivered	: 13			
Distance Covered	: 522			
Number of Stops in Trip: 2				
---TRIP---				
'Dublin' -> 'Kilkenny'	100	60	1.67	21
'Kilkenny' -> 'Dublin'	100	100	1.00	0
Quantity Delivered	: 21			
Distance Covered	: 200			
Number of Stops in Trip: 2				
---TRIP---				
'Dublin' -> 'Ennis'	190	60	3.17	14
'Ennis' -> 'Dublin'	190	100	1.90	0
Quantity Delivered	: 14			
Distance Covered	: 380			
Number of Stops in Trip: 2				
---TRIP---				
'Dublin' -> 'Sligo'	180	100	1.80	22
'Sligo' -> 'Dublin'	180	100	1.80	0
Quantity Delivered	: 22			
Distance Covered	: 360			
Number of Stops in Trip: 2				
----				

Figure 8: Goods Delivery Optimal Cost : With Capacity Constraint and Time Constraint

## Vehicle Routing - 30 Cities

With Capacity Constraint and No Time Constraint

Cost of Goods Delivery (All Trips) : 1690.67 Euros

From(City) - To (City)	Distance(KM)	Speed(KMH)	Time(Hours)	Quantity
Dublin -> Leixlip	16	100	0.16	13
Leixlip -> Celbridge	5	100	0.05	13
Celbridge -> Maynooth	6	100	0.06	10
Maynooth -> Navan	31	100	0.31	23
Navan -> Drogheda	22	100	0.22	17
Drogheda -> Dundalk	33	100	0.33	46
Dundalk -> Letterkenny	133	100	1.33	8
Letterkenny -> Sligo	87	100	0.87	21
Sligo -> Mullingar	112	100	1.12	14
Mullingar -> Tullamore	29	100	0.29	13
Tullamore -> Athlone	34	100	0.34	24
Athlone -> Galway	75	100	0.75	71
Galway -> Killarney	39	100	0.39	12
Killarney -> Ennis	41	100	0.41	12
Ennis -> Limerick	32	60	0.53	104
Limerick -> Tralee	85	100	0.85	13
Tralee -> Cork	95	100	0.95	235
Cork -> Clonmel	72	100	0.72	18
Clonmel -> Waterford	41	100	0.41	53
Waterford -> Wexford	45	100	0.45	17
Wexford -> Kilkenney	63	100	0.63	29
Kilkenney -> Carlow	30	100	0.30	12
Carlow -> Port laoise	33	100	0.33	15
Port laoise -> Newbridge	36	100	0.36	10
Newbridge -> Naas	9	60	0.15	15
Naas -> Dublin	31	100	0.31	0
Quantity Delivered : 818				
Distance Covered : 1235				
Number of Stops in Trip: 26				
---TRIP---				
Dublin -> Greystones	24	100	0.24	7
Greystones -> Bray	8	100	0.08	35
Bray -> Malahide	26	100	0.26	12
Malahide -> Swords	3	60	0.05	33
Swords -> Dublin	13	100	0.13	0
Quantity Delivered : 87				
Distance Covered : 74				
Number of Stops in Trip: 5				

Figure 9: Goods Delivery Optimal Cost : With Capacity Constraint - Run1

Cost of Goods Delivery (All Trips) : 1672.8 Euros

From(City) - To (City)	Distance(KM)	Speed(KMH)	Time(Hours)	Quantity
Dublin -> Malahide	14	100	0.14	8
Malahide -> Swords	3	100	0.03	32
Swords -> Navan	38	100	0.38	16
Navan -> Drogheda	22	100	0.22	41
Drogheda -> Dundalk	33	100	0.33	23
Dundalk -> Letterkenny	133	100	1.33	18
Letterkenny -> Sligo	87	100	0.87	25
Sligo -> Mullingar	112	100	1.12	18
Mullingar -> Tullamore	29	100	0.29	16
Tullamore -> Athlone	34	100	0.34	15
Athlone -> Galway	75	100	0.75	63
Galway -> Killarney	39	100	0.39	9
Killarney -> Ennis	41	100	0.41	23
Ennis -> Tralee	81	100	0.81	23
Tralee -> Limerick	85	100	0.85	76
Limerick -> Cork	86	100	0.86	247
Cork -> Clonmel	72	100	0.72	11
Clonmel -> Waterford	41	100	0.41	43
Waterford -> Wexford	45	100	0.45	16
Wexford -> Kilkenny	63	100	0.63	17
Kilkenny -> Carlow	30	100	0.30	10
Carlow -> Port Laoise	33	100	0.33	5
Port Laoise -> Newbridge	36	100	0.36	21
Newbridge -> Naas	9	100	0.09	25
Naas -> Celbridge	15	60	0.25	23
Celbridge -> Maynooth	6	100	0.06	12
Maynooth -> Leixlip	7	100	0.07	10
Leixlip -> Bray	30	100	0.30	33
Bray -> Greystones	8	100	0.08	11
Greystones -> Dublin	24	100	0.24	0
Quantity Delivered	: 890			
Distance Covered	: 1331			
Number of Stops in Trip:	30			

Figure 10: Goods Delivery Optimal Cost : With Capacity Constraint - Run2

Cost of Goods Delivery (All Trips) : 1645.6 Euros

From(City) - To (City)	Distance(KM)	Speed(KMH)	Time(Hours)	Quantity
Dublin -> Greystones	24	100	0.24	8
Greystones -> Bray	8	100	0.08	40
Bray -> Malahide	26	100	0.26	18
Malahide -> Swords	3	100	0.03	17
Swords -> Drogheda	31	100	0.31	26
Drogheda -> Navan	22	100	0.22	15
Navan -> Dundalk	43	100	0.43	40
Dundalk -> Letterkenny	133	100	1.33	19
Letterkenny -> Sligo	87	100	0.87	25
Sligo -> Galway	117	100	1.17	92
Galway -> Killarney	39	100	0.39	14
Killarney -> Ennis	41	100	0.41	14
Ennis -> Limerick	32	100	0.32	44
Limerick -> Tralee	85	100	0.85	14
Tralee -> Cork	95	100	0.95	154
Cork -> Clonmel	72	100	0.72	13
Clonmel -> Waterford	41	100	0.41	31
Waterford -> Wexford	45	100	0.45	22
Wexford -> Kilkenny	63	100	0.63	18
Kilkenny -> Carlow	30	60	0.50	9
Carlow -> Port Laoise	33	100	0.33	11
Port Laoise -> Tullamore	29	100	0.29	11
Tullamore -> Athlone	34	100	0.34	11
Athlone -> Mullingar	42	100	0.42	20
Mullingar -> Newbridge	53	100	0.53	12
Newbridge -> Naas	9	60	0.15	13
Naas -> Celbridge	15	100	0.15	15
Celbridge -> Maynooth	6	100	0.06	13
Maynooth -> Leixlip	7	100	0.07	18
Leixlip -> Dublin	16	100	0.16	0
Quantity Delivered	: 757			
Distance Covered	: 1281			
Number of Stops in Trip	: 30			

Figure 11: Goods Delivery Optimal Cost : With Capacity Constraint - Run3

### With Capacity Constraint and Time Constraint

Cost of Goods Delivery (All Trips) : 6270.93 Euros

From(City) - To (City)		Distance(KM)	Speed(KMH)	Time(Hours)	Quantity
Dublin	-> Cork	218	100	2.18	104
Cork	-> Dublin	218	100	2.18	0
Quantity Delivered : 104					
Distance Covered : 436					
Number of Stops in Trip: 2					
---TRIP---					
Dublin	-> Limerick	175	100	1.75	113
Limerick	-> Dublin	175	100	1.75	0
Quantity Delivered : 113					
Distance Covered : 350					
Number of Stops in Trip: 2					
---TRIP---					
Dublin	-> Galway	187	100	1.87	59
Galway	-> Dublin	187	100	1.87	0
Quantity Delivered : 59					
Distance Covered : 374					
Number of Stops in Trip: 2					
---TRIP---					
Dublin	-> Waterford	133	100	1.33	30
Waterford	-> Dublin	133	100	1.33	0
Quantity Delivered : 30					
Distance Covered : 266					
Number of Stops in Trip: 2					
---TRIP---					
Dublin	-> Dundalk	77	100	0.77	30
Dundalk	-> Dublin	77	100	0.77	0
Quantity Delivered : 30					
Distance Covered : 154					

Figure 12: Goods Delivery Optimal Cost : With Capacity Constraint and Time Constraint

## Authorship

Adedayo Adelowokan Contribution: Started development on the very first Mosel model before the problem was changed, also offered insights into the new Mosel Model. Sections: 1, 3, 7

Louis Carnec Contribution: Problem research original and final one, Data Generation in python, 50% of Mosel Implementation, sections; 2,4,5

Vijay Katta Contribution: Started research on identifying the problem and changing

Louis Carnec

Vijay Katta

Adedayo Adelowokan

the original one, 50% of Mosel implementation, generating the results, sections; 6



## References

- [1] COLOMBANI, Y., AND HEIPCKE, S. Mosel: An extensible environment for modeling and programming solutions. In *Proceedings of CP-AI-OR* (2002), vol. 2, pp. 277–290.
- [2] CORDEAU, J.-F., LAPORTE, G., SAVELSBERGH, M. W., AND VIGO, D. Vehicle routing. *Handbooks in operations research and management science* 14 (2007), 367–428.
- [3] DANTZIG, G. B., AND RAMSER, J. H. The truck dispatching problem. *Management science* 6, 1 (1959), 80–91.
- [4] GALLMANN, F., AND BELVEDERE, V. Linking service level, inventory management and warehousing practices: A case-based managerial analysis. *Operations management research* 4, 1-2 (2011), 28–38.
- [5] GUÉRET, C., PRINS, C., AND SEVAUX, M. Applications of optimization with xpress-mp. *contract* (1999), 00034.
- [6] LENSTRA, J. K., AND KAN, A. Complexity of vehicle routing and scheduling problems. *Networks* 11, 2 (1981), 221–227.
- [7] TOTH, P., AND VIGO, D. Models, relaxations and exact approaches for the capacitated vehicle routing problem. *Discrete Applied Mathematics* 123, 1 (2002), 487–512.