



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

---



# Real-time Electric Vehicle Routing

System, Control and Mechatronics

Guo Wu   Ruisheng Zhao



# Real-time Electric Vehicle Routing

Guo Wu   Ruisheng Zhao



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

Electrical Engineering  
*System, Control and Mechatronics*  
Electromobility  
CHALMERS UNIVERSITY OF TECHNOLOGY  
Gothenburg, Sweden 2019

Real-time Electric Vehicle Routing  
Guo Wu, Ruisheng Zhao

© Guo Wu, Ruisheng Zhao, 2019.

Supervisor: Rafael Basso, Volvo Group Truck Technology  
Examiner: Balázs Adam Kulcsár, Electrical engineering

Electrical Engineering  
System, Control and Mechatronics  
Electromobility  
Chalmers University of Technology  
SE-412 96 Gothenburg  
Telephone +46 31 772 1000

Cover: Real-time Electric Vehicle Routing.

Typeset in L<sup>A</sup>T<sub>E</sub>X  
Printed by [Volvo Group Truck Technology]  
Gothenburg, Sweden 2019

Real-time Electric Vehicle Routing  
Guo Wu, Ruisheng Zhao  
Electrical Engineering  
Chalmers University of Technology

## Abstract

The growing demands for promoting sustainable fuel solutions in recent years have facilitated the use of electric vehicles (EVs) in the urban area. Traditional vehicle routing technologies for internal combustion vehicles have been very mature today. However, those technologies always focus on calculating the shortest distance or the most time-saving solution, which is not a suitable plan for EV due to its current limitations: limited driving range and long recharge time. Hence some energy-optimal routing methods need to be extended.

In this paper, a real-time electric vehicle routing problem with capacity and battery limitation is introduced, possible recharging motion en-route will also be considered. Various solution models such as tree-searching, integer linear programming and mixed integer linear programming are analyzed. The mixed integer linear programming model is selected for practical application in the end because it can find high quality of solutions within reasonable computation time while facing with a large-scale routing problem.

The practical experiment results are collected by implementing the model on the products distribution tasks within the urban area of Luxembourg City, where the time-dependent traffic congestion exists. The cost graph of the particular urban area in Luxembourg City is estimated by a parameterized energy model, which is provided by [10]. The results indicate that real-time routing solutions based on the dynamic traffic environment can effectively save energy.

Keywords: Real-time, Electric vehicle, Energy estimation, Cost graph, Energy-optimal routing, Mixed integer linear programming, Model predictive control, recharging.



## Acknowledgements

We would like to first express our sincerest gratitude to Rafael Basso, our thesis work supervisor for giving us this precious opportunity to work in Volvo group company and providing indispensable sources to support our work. We would not be able to go this far without his technical advances from beginning to end. We would also like to express our very great appreciation to our examiner Balázs Adam Kulcsár for every time his patient guidance, enthusiastic encouragement and meaningful feedback comments for our thesis work. Special thanks should be given to our friends Min Cheng and Shengwei Deng, we always have lunch together, they fill our working days with interesting moments. Thanks to our colleagues who also work on their thesis in the company, we always get strength and energy from daily greetings and fika chat.

Finally, we wish to thank our parents for their support and encouragement throughout our master study.

Guo Wu, Ruisheng Zhao, Gothenburg, May 2019



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Literature review . . . . .	3
<b>2</b>	<b>Theory</b>	<b>9</b>
2.1	Policy . . . . .	9
2.2	Shrinking horizon window . . . . .	10
2.3	Energy estimation . . . . .	10
2.4	Path Finding . . . . .	11
2.5	Periodic routing . . . . .	12
<b>3</b>	<b>Grid Map</b>	<b>13</b>
3.1	Basic problem description . . . . .	14
3.2	Tree search . . . . .	15
3.3	Linear programming . . . . .	18
3.3.1	Mathematical model - simplified . . . . .	19
3.3.2	Mathematical model - improved . . . . .	21
3.4	Experiments . . . . .	25
<b>4</b>	<b>Real Map</b>	<b>29</b>
4.1	Routing without charging . . . . .	30
4.1.1	Customer graph . . . . .	30
4.1.2	Find the optimal route . . . . .	30
4.1.3	Experiments . . . . .	32
4.2	Routing with charge stations . . . . .	36
4.2.1	Model formulation . . . . .	36
4.2.2	Model adaption for dynamic situation . . . . .	37
4.2.3	Experiments . . . . .	39
<b>5</b>	<b>Conclusion</b>	<b>47</b>
5.1	Work summary . . . . .	47
5.2	Future Works . . . . .	48
	<b>Bibliography</b>	<b>49</b>



# 1

## Introduction

In recent years, the increasing benefits of researching on green energy solutions and the intense demands of reducing the environmental pollution caused by daily road transportation promote the evolution of Electric vehicles (EVs), which makes EVs become an alternative choice for vehicles purchasing. Comparing with the traditional combustion vehicles, EVs have absolute advantages such as cleaner energy resources, quieter behaviors and zero emissions, which make an outstanding contribution to the sustainable development of environment [1]. However, the capacity of EV's battery puts a severe limitation on EVs' drivable range, and a long waiting time is required for fully recharging the battery. These are the most concerning problems of EVs that cannot be avoided by drivers and researchers [2]. In essence, such problems are concluded as indirect energy-saving problems and could be theoretically addressed either by driving smoothly throughout the ways from a subjective motivation perspective, or by optimizing the sustainable distribution plan for vehicles to reduce unnecessary energy waste.

Traditional distribution tasks within workdays always refer to the movement of goods from one location to another, and it is always created and handled as vehicle routing problems (VRPs) by many transportation logistics companies. For the most traditional combustion vehicles, the goal is to search for the minimal transportation cost routing solution regarding to the total traveling time and distance that will visit a number of customers with both the start and the endpoint located at the depot. After decades of innovation and improvement within this field, the existing routing technologies for traditional combustion vehicles have been proven to be advanced and efficient. New requirements for the sustainable development of environment have been proposed, which leads many variations of VRP nowadays to focus on green vehicle routing, such as electric vehicle routing problems (EVRPs), which incorporates the electric energy consumption constraints with traditional routing problems. Therefore, in EVRPs, instead of solving the shortest distance or the least time-consuming routes, the total electric energy cost is a more important benchmark to consider when people are judging the quality of solutions.

Nowadays, most EVRPs are generated from daily transportation that uses electric vehicles to accomplish distribution tasks within urban areas, while the traveling cost would fluctuate due to the complex real-time traffic network conditions like weather, congestion or city topography, which lead our EVs to spend more electric energy than the expectation in many cases. Ignoring such disturbances caused by those uncertain and unforeseen real-time factors when arranging distribution tasks will result in a poor or even dangerous routing solution that put EVs at risk of low battery and have to terminate the current mission somewhere on the roads. If that

emergency happens, the vehicle has to turn to the nearby charging station while such motion is not included in the previous routing plan, which forces additional computation sources to come into service. Terminating the current task temporarily will cause even a more serious trouble to customers because they have to wait for a long time unreasonably. This fact directly points out that solving EVRPs with a static manner is insufficient to guarantee the optimality of the previous routing plan during the whole work day since the input information is evolving all the time in reality. Hundreds or thousands of times updates need to be implemented.

In most real-time traffic applications, the network information is gradually revealed when vehicles are en-route, and vehicles usually can only know the complete input information when the current task has been finished. Consider such real-time property, an intuitive option is to solve the problem with an ongoing manner and continuously update the current solution based on the newly arrived traffic information. This operation requires the routing model to be able to react and process the constraints and conditions quickly.

The thesis work starts by reviewing relevant theory and literature to gain basic ideas of solution methods; after that, the practical implementations are presented step-by-step. Firstly, the feasibility validation of those models is carried out within the fictitious grids world with simple routing problems. Then the feasible model is extended to the real-world application. The introduced electric vehicle routing problem are characterized as: (1) the input data is time-varying, (2) incorporates the weight and battery capacity constraints, and (3) considers the possible charging motion en-route when congestion happens. The goal of thesis work is to develop a model-based routing algorithm to minimize the total energy cost for visiting all customers. A variation of linear programming called mixed integer linear programming is applied to solve the introduced problem dynamically, which further incorporates with the optimization software tool CPLEX to improve the performance. The numerical result proves that our model can update the routing solution continuously with potential energy saving effect and efficiency based on the real-time circumstances. The graphic result part visualizes the solution routes, which provides the foundation for the correctness analysis of the model.

This thesis work is carried out in Electromobility department of Volvo Group Truck Technology Company, guided by our supervisor Rafael Basso and examiner Balazs Adam Kulcsar. The document is organized as the following: The literature review part is presented in the rest of this chapter. Chapter 2 presents the fundamental theories that are used in the proposed models and algorithms. Chapter 3 presents and analyzes the pros and cons of the applying tree-searching method and linear programming methods for VRP and Capacitated vehicle routing problem (CVRP) on grids world based routing problems, followed by experiments with the two methods. Chapter 4 describes the practical application of the dynamic CVRP model as well as considering the charging issue on real map and related experiment results. A short conclusion is given in Chapter 5.

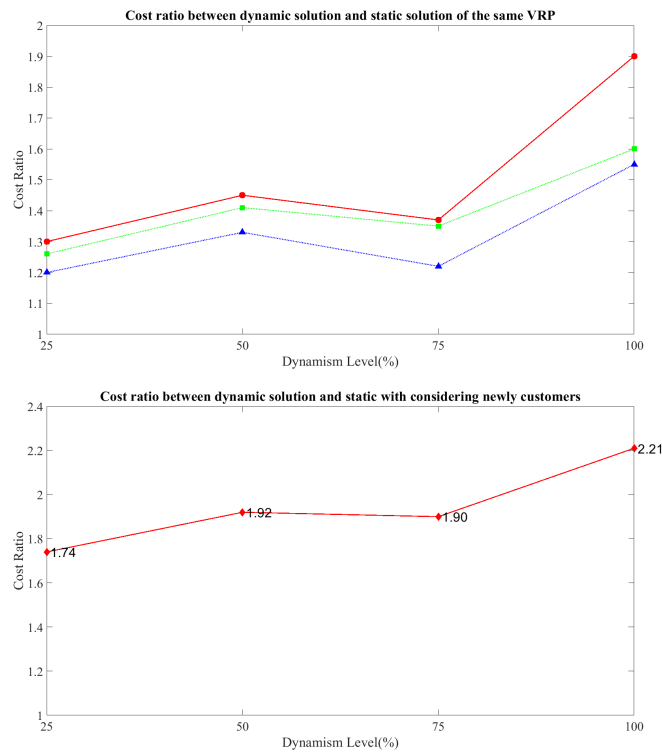
## 1.1 Literature review

The Vehicle Routing problem (VRP) is a large topic composed by a comprehensive optimization and linear programming problem (Mixed Integer linear programming in most cases). In essence, VRP is one representation of logistic management, which decides the control motions on both the internal components operations and the external routes selections of a vehicle [3]. A general form of VRP with single vehicle can simply be seen as a traveling salesman problem (TSP), which can be described as: the dispatchers need to arrange a single vehicle to serve all customers' demands, starting from and ending at the depot [4]. However, although VRP has been widely studied for several decades since 1959, it is still a challenging research topic nowadays if people combine it with complicated real-life applications. Then it requires a more complex and effective model to provide the feasible or optimal traveling routes based on large amounts of information. Since the VRP belongs to the class of NP-hard combinatorial problems, the variation of VRPs that contain more complicated constraints regard to new requirements will be easily turned to an unsolvable problem.

CVRP is one of the most common variation, which imposes a constraint that the total carried cargo weight does not exceed fixed capacity of the vehicle while satisfying all demands of customers. Paolo Toth and Daniele Vigo [5] illustrated the CVRP with asymmetric and symmetric cost matrix conditions, the problem is to find  $K$  routes plan (corresponding to  $K$  vehicles) for pick-up and delivery with minimum cost. In [6] and [7], there are examples of solving CVRP. Another popular variation is electric vehicle routing problem (EVRP) produced by the goal of energy-efficiency [8]. However, the current research area on VRP considers not only the payload limitation of our electric vehicle but also incorporates battery capacity and possible recharging requests into a multi-constrained optimal routes planning problem. Such combination of the CVRP and EVRP was mentioned in [10][11] and [9], where the first paper presents an energy estimation model that can predict the energy consumption between two nodes precisely based on the Newton's laws of motion and topography of the city. The most energy-saving routes is calculated by a '2-stage' algorithm based on those estimated cost of paths. But the proposed algorithm in the papers did not solve the routing problem in a dynamic method.

The study of dynamic vehicle routing problem (DVRP) starts from 1988 when Psaraftis and Harilaos first proposed the definition of DVRP in [12], this topic and its variation has become a popular field to be investigated and developed. Recent articles such as [13] and [14] summarized different types of DVRPs. The former gives a comprehensive review of two types of DVRP: the dynamic and deterministic problem and the dynamic and stochastic problem, and the latter mentioned the DVRPs with stochastic factors. Intuitively, DVRP introduces more complex constraints compared to the conventional VRP since in real life applications, not all information is known in advance, some inputs data are evolving during the execution of routing plan. This fact forces some input parameters in the previous static VRP now become time-dependent [15][16] or stochastic [17], hence the corresponding algorithms for DVRPs are required to make online decisions quite efficiently. The advantages of solving DVRP is that the designed dynamic algorithm has a 'sense'

to the real-time environment, making decisions that depend on the external changes of the traffic conditions, hence such result is more realistic and valuable in most real-world application compared to the static VRP. The numerical comparison of the total traveling cost between static and dynamic algorithm based on the dynamic environment is presented as the figure 1.1. The dynamism level here refers to the amplitude of the fluctuation on the actual travel times of each road, a high level of dynamism means that the travel time of passing through a specific road will change with a large amplitude, and vice versa. The first figure shows the cost ratio trajectories in which the model only considers those known customers, while three different colors indicate three different time intervals between routes adaptation. The red trajectory means that the routes are adapted every hour, the green and blue one mean that the routes adaptations take place every 2 and 3 hours respectively. The second figure shows the cost ratio trajectory in which the model takes into account the newly arrived customers. Both figures show that the periodic routing algorithm performs better on saving cost than only solving the routing problem one time in the beginning.



**Figure 1.1:** The cost ratio between dynamic algorithm solution and static algorithm solution, which is calculated as  $\text{Cost ratio} = (\text{Cost of static route planning}) / (\text{Cost of dynamic route planning})$ . These two figures are reproductions of figure.6 and figure.7 in the paper [18].

Many DVRP examples in the previous literature focus on deciding a routing solution by mainly considering three types of uncertainties: stochastic traveling time, stochastic emerging service requests, as well as stochastic customer's demands, the

routing plan under these uncertain conditions are always decided depending on the history data and the newly collected real-time information while sometimes one can merely know the possibility of the occurrence of the incidents, which are unpredictable in advance. For example, in [19], it presented a real-time model in order to prevent the vehicle from sticking in routes when the urban recurrent or non-recurrent congestion happens. Another variation of DVRP are shown in [20] and [21], the vehicle routing problem with time windows (VRPTW) has been investigated, where the travel time between each two nodes is changing over time. Those papers considered the soft time windows constraints  $[e_i, l_i]$  on arriving customers, that means the dispatched vehicles have to start serving customers within a certain time interval. Missing the deadline will increase the total cost (be punished) significantly. The [3] and [22] described a more intricate VRPTW, in addition to the varying link travel time, new customer requests with time windows will appear continuously during the daytime. In this case, additional calculation is needed to decide which of those new requests should be rejected due to its unreasonable emergence position before inserting them into the executing plan.

Recently, the VRP produced by traditional combustion vehicles have been adapted into electric energy related vehicle routing problems due to the rising of electric commercial vehicles. For example, in [23], it proposed a dynamic programming method to solve single and multiple EVs' routing problem with charging nodes, and in [24], a new type of VRPTW called electric vehicle routing problem with time windows and recharging stations (E-VRPTW) has been discussed. The E-VRPTW incorporates the process of recharging at each station with the charging level depended recharging time, and it also considers the capacity limitation on vehicles as well as the soft time window constraints on customers. More research on this field can be found in [25] and [26]. The former introduced a recharging vehicle routing problem (RVRP) with capacity limitations and time window constraints, and the latter proposed an electric vehicle routing problem with time windows and partial recharging (EVRP-TWPR). To reduce the total recharge cost, the authors considered different recharging technologies (fast recharge and slow recharge) and allowed vehicles to recharge while providing service to customers if there is a charging station at customer node.

However, to the best our knowledge, in all the referred papers that are about solving DVRPs or its variations, the proposed DVRPs were divided into a series of static VRPs with different constraint architectures. Each static VRP was adapted by the external information sample of the current time period (or called time slice in some papers). This fact leads the solution techniques for solving DVRPs to range from the exact methods to heuristics. The exact methods such as Linear programming (LP) and dynamic programming (DP) allow the finding of optimal solutions, but they are often time-consuming because of the dimension curse in solving real-world problems (e.g. the problems with large dimension). Heuristic algorithms seek to produce good-quality solutions within reasonable computation times and the results are always capable for practical application. The trade-off happens when balancing the solution quality and the computation speed [13]. LP technique is capable to solve small or medium-sized logistic problems that involves multiple variables and constraints. It is now becomes possible to apply on more complex routing problems

because the powerful software solvers that are embedded in mathematical platforms such as Matlab [27] and IBM can accelerate the data processing speed [28].

LP formulation and its variations have already commonly been used in many combustion vehicles routing problems based on different demands. In [18], a classic example of using mixed integer linear programming model is developed to solve a dynamic CVRP on multiple vehicles case, where the dynamic routing model adjusts (re-optimizes) the routes plan every 2-3 hours based on the real-time travel time information as well as the newly appeared customer demands. A more complicated mixed integer linear programming problem application could be found in [20], where a dynamic VRPTW with time-dependent travel times in addition to real-time demands (RT-TDVRPTW) was introduced. The difference is that in the latter paper, vehicles are allowed to wait at current customer after the service is completed such that vehicles can serve the next customer within its soft time window constraints, which is a very common scenario in real life. For those readers who want to learn more about using linear programming technique to solve DVRPs or SVRPs can take a look at [29][30]. As for electric vehicles, the linear programming technique is also widely used in EVRPs. For example, in [25] and [26], the charge limitation, electricity consumption are included in constraints of the proposed linear programming model and the recharging waiting cost becomes a part of the objective function.

DP is an iterative technique that can solve routing problems by cutting down the whole traveling plan into a number of "sub-flows". Each sub-flow is aim to solve the lowest cost sub-routes for a sub-optimization problem and hence the origin problem is optimized by combining all those optimal sub-solutions. The typical EVRP for a single electric vehicle with recharging station that solved by dynamic programming was described in [23], but the proposed model only considered the time consuming factor. The model mainly contributed to minimize the value of an iterative equation that consists of the total elapsed time so far as well as the time to go one step further based on the current moment to reach the next node. The algorithm will terminate when the total elapsed time from the depot to the last customer has been calculated. In another paper [19], a real-time routes decision problem is also modeled as a dynamic programming model with look ahead policy. The real-time network scenario is described by state variables of each road link such as the congestion level and the waiting duration that are used to determine the value of the iterative equation at every decision stage (every node). However, the horizon of looking ahead policy in the paper was limited to prevent the exponential growth of the state space.

Metaheuristic techniques such as Ant Colony Optimization (ACO), Genetic algorithm (GA) and Large Neighborhood Searching (LNS) make great contribution to DVRP research. A detailed introduction to some of the metaheuristic methods that used for some combinational optimization problems such as DVRPs is presented in [31].

ACO is a probabilistic technique, nature-inspired metaheuristic, which uses artificial "ants" to simulate the foraging behavior that succeeds in seeking for the optimal path between their colony and source of food. The basic knowledge and simple application of ACO are mentioned in [32]. Recent research shows that ACO is quite an efficient method for solving DVRPs and its variations. In [22], a single colony ACO

algorithm is proposed to solve the DVRPTW with different values of dynamicity (a percentage of  $\%p$  customer demands are known a priori), the cost function is minimized from a multi-objective perspective: the number of vehicles and the total traveled distance. The result shows that the proposed ACO algorithm can obtain a competitive result with small variance when facing a high dynamicity level. In [4] a similar ACO algorithm based model is proposed to tackle the DVRP with stochastic customer demands, but the updating of pheromone is determined by the best performance ants in the last iteration. An enhanced ACO (E-ACO) is introduced in [33], where it combines the traditional ACO algorithm with a crossover operation that is similar in GA, and K-means to separate the known customers into several clusters with the most reasonable total distance. The experimental result in this paper shows that the proposed E-ACO is capable to find high-quality solutions for DVRP.

In [34], GA is mentioned as a nature-inspired metaheuristic algorithm with good efficiency and performance on several applications of DVRPs such as multiple vehicles routing problems with capacity constraints and VRPTW. In [35], the adapted GA algorithm is based on a integer linear programming formulation with the capacity and time window constraints. Additional calculations are generated when applying self-determined strategies on creating initial population (a feasible solution set), crossover process, mutation and individual replacement in order to guarantee the quality of the converged result and accelerate the convergence speed. Similar investigation can be found in [36], where the implemented crossover operator is different from the previous paper, e.g. Partially-Mapped-Crossover (PMX) and Edge Recombination (EDGERC), although the dynamic factor considers in the paper only the newly arrived demands, the average cost from the depicted GA-based model outperforms those of the traditional ACO.

Metaheuristics based on LNS have also shown outstanding result in tackling pickup and delivery transportation service problems. With different heuristic neighborhood searching actions (destroy and re-insert techniques), LNS is likely to find better candidate solutions within a complex neighborhood. A comprehensive introduction about LNS and application examples are presented in [37]. There are similarities between LNS and GA: (1) in order to promise reasonable calculation time and produce superior solutions, both of them need mature rules for creating a high-quality initial population, (2) the crossover process of GA share the same essence with the neighborhood switch of LNS: use heuristic search to find better candidate solutions that satisfy the higher benchmarks. An improved LNS algorithm for real-time vehicle routing problem with hard time windows (DVRPTW) is shown in [21], where the current route plan will be adjusted according to the position and time constraints of the newly arrived demands. Since the traditional LNS algorithm for VRPs and DVRPs is very time consuming when it has a poor initial population, the improved LNS algorithm in this paper mitigates this problem by: (1) improving the quality of the initial solution, and (2) attempting different neighborhood searching strategies in each iteration. In fact, the principle behind (2) is the centerpiece of Adaptive Large Neighborhood Searching (ALNS), which is a popular variation of LNS that allows the algorithm to use multiple neighborhood searching methods in one iteration for various solutions searching. In [38] and [39], the exact examples of

applying ALNS on solving DVRPs are presented.

As for the real-time electric vehicle routing, none of the articles mentioned above covers all aspects of our investigation topic, since the optimization principle of routing does not focus on the shortest-distance or least time-consuming. Instead, the EVRPs are always addressed from the energy cost perspective, developing a solution model which can make routing decisions for electric truck delivery regards to the minimum electric energy cost that based on the network topography as well as the real-time road information simultaneously.

In the next section, the routing problem, as well as the goal of the solution, will be described in detail. In the Methods section, several strategies that are feasible for our case are introduced, which start from realizing our dynamic approaches on rudiment models and verifying their improvement based on a customer graph. The approaches are presented as pseudo code and the results are shown by comparing the routing solution in a grid world. In the Result and Discussion section, the result data is collected to formulate a table for comparison and analysis of the advantages and weaknesses of our designed dynamic model.

# 2

## Theory

In this chapter, the theoretical backgrounds of the thesis work are introduced, including the look ahead policies, the concept of shrinking horizon window, the formula of energy estimation and path structure, which provide technical guidance for implementing the solution methods in the following chapters.

### 2.1 Policy

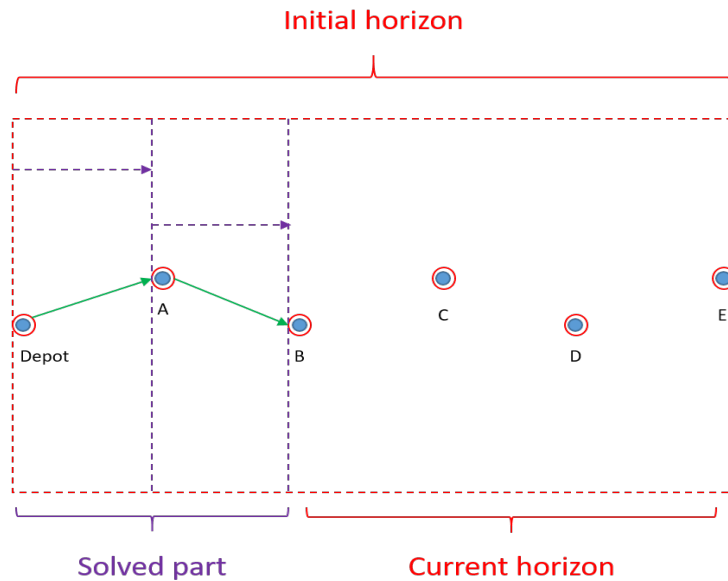
The goal of this project is to design a model that finds the most energy-saving route for an electric delivering truck. The truck departs from the depot, visits all of the customers once and back to the depot. The vast majority of strategies for sequential stochastic optimization problems can be organized with four fundamental classes of policies [40]:

- Myopic cost function approximations: Chose the next action by only considering the current state value. Apparently this policy does not work well in our case since the cost changes and the goal of the thesis is to find the global optimal solution.
- Policy function approximations: Find a well-defined function that matches to the policy function, requiring the structure of the policy to be obvious. This is an analytic function that returns an action given a state but without solving an embedded optimization problem. The idea is well matched to  $(q, Q)$  inventory re-ordering policy or fleet sizing strategy. However, this report does not need to consider any inventory or fleet issue, and it is difficult to find a classic function that can represent the policy.
- Policies based on value function approximations: Here the policy is made after estimating the state values using Bellman's equation. In a word the main part of this idea is estimating the state values (cost), but an energy estimation model has already been provided by [10] that gives the customer graph containing the minimal cost between every two nodes (customers and the depot).
- Look-ahead policies: This one optimizes over the entire future and pick only the next action. A fairly good solution might be found by implementing this policy every time that the vehicle reaches a customer. This is referred to as model predictive control (MPC) in control theory and is the most common approach to tackle dynamic vehicle problem when the customers are all known. Thus in this thesis work, all algorithms within the models will be based on

Look-ahead policy.

## 2.2 Shrinking horizon window

Assuming the number and the position of customers are known beforehand and fixed in the map. The optimization process is to solve a real-time optimal routing problem - a circular trip incorporating all the customers as well as the depot. Comparing with the receding horizon window used in traditional MPC method with fixed window size, the whole time interval is equal to the length of the prediction horizon at the beginning. Therefore, the MPC method applied in our case uses a shrinking horizon window as figure 2.1 shows. Such adaptation is inspired by the fact that the terminal time index is always the last customer. during the periodic optimization process, note that the solved customer pair won't be counted in the optimization process later, so the scale of the problem as well as the prediction horizon are decreased after each time the optimal routes between the current arrived customer and the next customer have been solved. Hence the computational efficiency will be improved gradually.

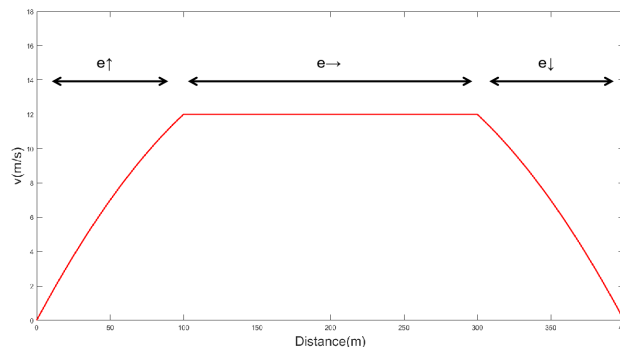


**Figure 2.1:** Shrinking horizon window

## 2.3 Energy estimation

The energy consumption estimation is a vital prerequisite for solving the proposed routing problem. In order to make sure that the model will solve for an energy-optimal routing plan, the exact amount of energy cost of traveling through any road link must be known beforehand. In this paper, the energy estimation work is executed by referring to the energy estimation model in Rafael Basso's paper [10], where for every link, as figure 2.2 shows, the energy consumption for the vehicle to

pass through any specific link can be divided into three parts:  $e^\uparrow$  for acceleration,  $e^\rightarrow$  for moderate driving and  $e^\downarrow$  for final braking.



**Figure 2.2:** The speed trajectory over any links  $(a, b)$  with initial and final speed equal to 0. This figure is a reproduction of Figure 4 in [10].

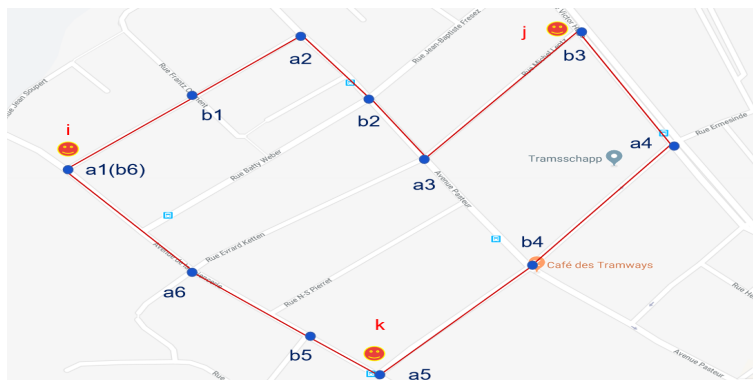
The total energy consumption of a certain link  $(a, b)$  is calculated as:

$$e_{ab} = e^\uparrow + e^\rightarrow + e^\downarrow = \alpha_{ab}m + \beta_{ab}\bar{v}^2 \quad (2.1)$$

$W_{ab}$  is the total weight of the vehicle (including self weight and the payload),  $\bar{v}$  is the average driving velocity on the link  $(a, b)$ . Note that these two parameters are constants while the vehicle is driving on the link  $(a, b)$ ,  $\alpha_{ab}$  is the cost coefficient of mass and  $\beta_{ab}$  is the cost coefficient of velocity. For the detailed derivation of equation (2.1) please refer to [10].

## 2.4 Path Finding

In the real traffic network, a complete path  $(i, j)$  from one node to another is composed by a series of consecutive road links  $(a, b)$  and the whole routing plan is composed by a series of those consecutive paths given a number of nodes, as the figure 2.3 shows.

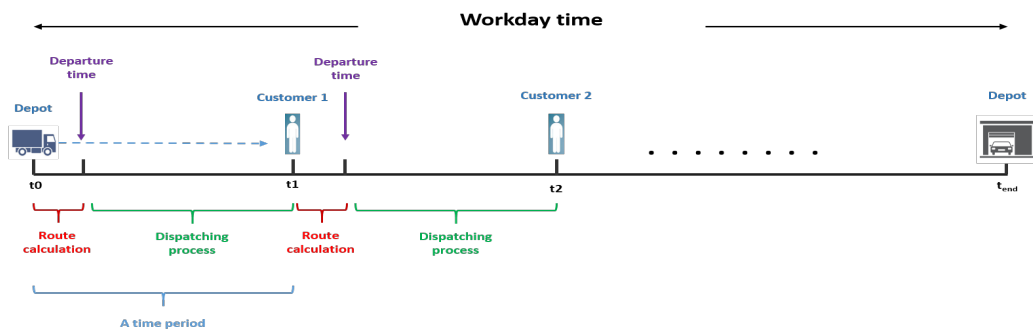


**Figure 2.3:** Paths between customers  $i, j, k$  are composed by road links  $(a_1, b_1)(b_1, a_2)(a_2, b_2)(b_2, a_3) \dots (a_6, b_6)$ . The whole routing plan is composed by  $(i, j)$  and  $(j, k)$ .

In this paper, our ultimate objective is to minimize the total energy consumption during the delivery task. Hence the most energy-saving paths between any two nodes need to be computed at any particular time. Those paths provide the foundation for calculating the optimal visiting order. This assignment is finished by referring to the "Stage One" linear programming model in [10], where it can solve the paths with the energy-optimal combination of links. The output from the "Stage One" model presents the unique optimal path to follow when a specific customer pair is chosen.

## 2.5 Periodic routing

The dynamic routing process in this thesis work essentially means to periodically solve the static routing problem to find a temporary optimal route solution based on the current information. This idea corresponds to the principle of 'dynamic' mentioned in the documents of the literature review section. However, the length of the time interval of each period is not fixed; it depends on the computation efficiency of the model as well as the actual elapsed time during the dispatching process respectively. During each time interval, the new route will be ready before departure from the current customer. The complete process is shown in figure 2.4.



**Figure 2.4:** Periodic routing process.

# 3

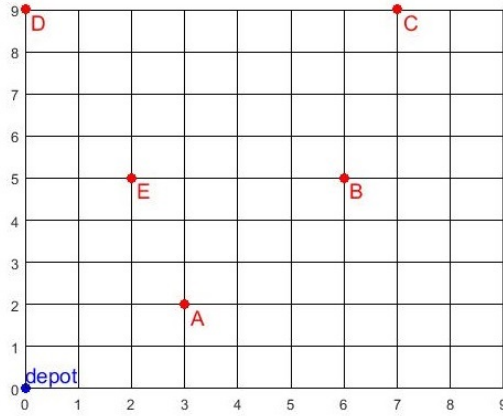
## Grid Map

At the starting point, it is necessary to build and test the model on a simplified case, which is a fast way to see if the model has any contribution or not on DVRP. Grid map has the most compact format as a road network, with only  $X$  and  $Y$  axes, which is a good choice for model design and validation in the initial phase. Grid map can be seen as a simplified version of the real world traffic network, the significance of creating such simple structure map is to initially validate the feasibility of the designed routing models without any external data support. Theoretically, there is no difference in calculating the total cost and the optimal routing plan between the grid map and real map, where the energy cost of driving through the path between customers is given.

This section is divided into three parts, 1: Tree-search model. 2: Linear programming model. 3: Experiments of the linear programming model.

Consider a 10 by 10 grid map, a depot is located at  $(0,0)$ , a certain number of customers will be positioned randomly at the intersections. The task is to have a truck departure from the depot, visit all customers once and then back to the depot. The truck can drive from a node to another node directly and the goal is to find a route that has the entire cost as small as possible. For the purpose of this thesis, the energy cost is random with a normal distribution. The mean is the straight line distance and the standard deviation is proportional to the mean.

Figure 3.1 shows an example of a customer set. The idea is to generate a customer graph (the term was firstly introduced in [10])  $\mathcal{R}$  containing the best path between every two nodes, and only the paths that chose from the customer graph are used to compose the entire route. Since there is no need to follow the grid lines, the best path between node  $A$  and  $B$  is the straight link  $(A, B)$ .



**Figure 3.1:** A  $10 \times 10$  grid map with  $(0, 0)$  the depot, and the customers  $A(3, 2)$ ,  $B(6, 5)$ ,  $C(7, 9)$ ,  $D(0, 9)$  and  $E(2, 5)$ .

Consider the case in Figure 3.1,  $\mathcal{R}^t$  is the customer graph at time  $t$  formed as a two dimensional array with  $N$  rows and  $N$  columns,  $N$  is the number of the nodes. The distance between the  $i_{th}$  node and the  $j_{th}$  node ( $\mathcal{R}_{ij}^t$ ) is placed in the  $i_{th}$  row and the  $j_{th}$  column. Furthermore,  $\mathcal{R}_{ij}^0$  shown in Figure 3.2 is assumed not been affected by the noise (pure distance).

	depot	A	B	C	D	E
depot	0	3.6056	7.8102	11.402	9	5.3852
A	3.6056	0	4.2426	8.0623	7.6158	3.1623
B	7.8102	4.2426	0	4.1231	7.2111	4
C	11.402	8.0623	4.1231	0	7	6.4031
D	9	7.6158	7.2111	7	0	4.4721
E	5.3852	3.1623	4	6.4031	4.4721	0

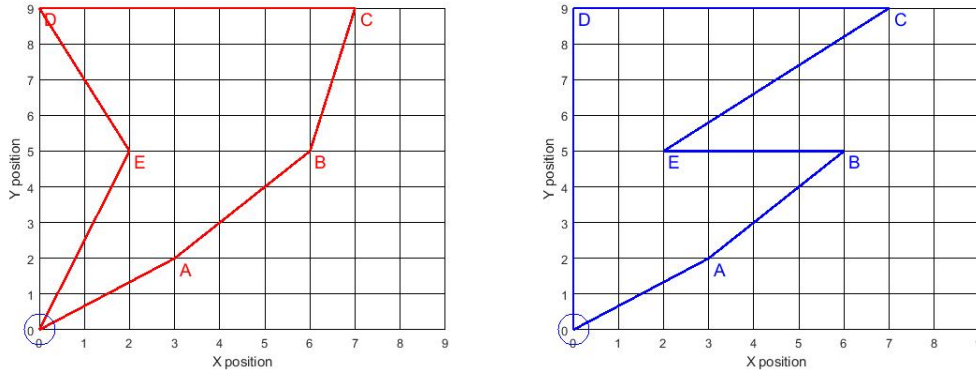
**Figure 3.2:** customer graph  $\mathcal{R}_{ij}^0$ ,  $i, j = 1, 2, 3 \dots N$ .

### 3.1 Basic problem description

Like most single vehicle routing problem in other documents, let  $G = (V, E)$  become the complete directed road graph (all nodes pairs are connected by edges) where  $V = \{0, 1 \dots n\}$  represents the depot and the customer positions respectively. Here, the proposed problem is assumed to only have one depot "0". The coordinates information of these customers is generated randomly inside a  $10 \times 10$  grid world. And  $E = \{(i, j) | i, j \in V, i \neq j\}$  defines the edges between each customer pair.  $V_s$  is a subset of  $E$  that contains the selected edges in the optimal solution. The vehicle will start to distribute goods to customers from the depot, serving every customer exactly once and back to the depot in the end. The basic goal of the proposed real-time electric vehicle routing problem in any particular time period is to find a set of routes to visit those customers with minimum energy cost. Other extra model



### 3. Grid Map



**Figure 3.4:** The graphs demonstrate the two routes in figure 3.3 in the grid map.

Through this strategy, one can find all the feasible solutions that are represented by branches. Next step is to calculate the total cost of each branch. For example the entire cost of the red branch and the blue branch (Figure 3.4) at time  $t = 0$  are:

$$C_{red} = \mathcal{R}_{12}^0 + \mathcal{R}_{23}^0 + \mathcal{R}_{34}^0 + \mathcal{R}_{45}^0 + \mathcal{R}_{56}^0 + \mathcal{R}_{61}^0 = 28.8286$$

$$C_{blue} = \mathcal{R}_{15}^0 + \mathcal{R}_{54}^0 + \mathcal{R}_{46}^0 + \mathcal{R}_{63}^0 + \mathcal{R}_{32}^0 + \mathcal{R}_{21}^0 = 34.2513$$

Here  $C_{red}$  and  $C_{blue}$  represent the entire cost of the red route and the blue route. Since the customer graph is symmetric, the route  $depot - A - B - C - D - E - depot$  has nothing different with  $depot - E - D - C - B - A - depot$ . The red path in this case has the shortest distance in total thus it is the best choice here. The pseudo code is shown as following:

#### Static tree search model

```

% Assume the problem has  $n$  customers and one depot
customer = [c1, c2, ..., cn];
Generate the customer graph  $\mathcal{R} \in \mathbb{R}^{(n+1) \times (n+1)}$ ;
path = [];
The total cost  $C = 0$ ;
optimal_cost =  $+\infty$ ;
optimal_path = [];
Loop 1: for  $i_1$  in customer:
    path = [path;  $i_1$ ];
    customer = customer /  $i_1$ ;
     $C = C + \mathcal{R}_{1i_1}$ ;
    Loop 2: for  $i_2$  in customer:
        path = [path;  $i_2$ ];
        customer = customer /  $i_2$ ;
         $C = C + \mathcal{R}_{i_1 i_2}$ ;
        ...
    Loop n: for  $i_n$  in customer:
        path = [depot; path;  $i_n$ ; depot];
         $C = C + \mathcal{R}_{i_{n-1} i_n} + \mathcal{R}_{i_n 1}$ ;

```

```

        if  $C < optimal\_cost$ :
             $optimal\_cost = C$ ;
             $optimal\_path = path$ ;
        end
        if all situations are considered:
            return  $optimal\_path$ ;
            END the process;
        end
         $customer = [customer; i_n]$ ;
         $path = [path/(depot\&i_n)]$ ;
         $C = C - \mathcal{R}_{i_{n-1}i_n} - \mathcal{R}_{i_n1}$ ;
    end

    ...
     $customer = [customer; i_2]$ ;
     $path = [path/i_2]$ ;
     $C = C - \mathcal{R}_{i_1i_2}$ ;
end

 $customer = [customer; i_1]$ ;
 $path = [path/i_1]$ ;
 $C = C - \mathcal{R}_{1i_1}$ ;
end

```

This model is used to solve static routing problem, when it comes to the stochastic as well as dynamic customer graph, the driving route needs to reoptimize periodically (lookahead policy).

Assume the stochastic  $\mathcal{R}^t$  at time period  $t$  has normal distribution  $\mathcal{N}^t(\mu_t, \sigma_t^2)$ , with the mean value  $\mu_t$  measured from  $\mathcal{R}^{t-1} = \mathcal{N}^{t-1}(\mu_{t-1}, \sigma_{t-1}^2)$ , and the variance proportional to the mean value ( $\sigma_t^2 = \gamma \cdot \mu_t$ ) which means that the change rate will be relatively high/low if the value itself is large/small. It is easy to understand, for example, it takes a long time to drive through a long route and factors like traffic lights, congestion, or car accident could affect the energy consumption in a relatively high extent, vice versa.

Before the departure, find the optimal route by using  $\mathcal{R}^0$ , assume the optimal route at  $t = 0$  is  $depot - A - B - C - D - E - depot$ , pick the first path  $depot - A$  as the first action. Once the truck has reached customer  $A$ , the customer graph  $\mathcal{R}$  changes because of the disturbance and becomes  $\mathcal{R}^1$ . Find the static optimal route for the rest nodes by using  $\mathcal{R}^1$ . Assume the result is  $A - E - C - D - depot$ , pick  $A - E$  as the next action. The process is the same as previous but with time-dependent customer graph. The final cost in total has the expression:

$$Cost = \mathcal{R}_{n_d n_1}^0 + \mathcal{R}_{n_1 n_2}^1 + \mathcal{R}_{n_2 n_3}^2 + \dots + \mathcal{R}_{n_N n_d}^{N+1} \quad (3.1)$$

Where  $\mathcal{R}_{n_i n_j}^t$  is the cost driving from node  $n_i$  to  $n_j$  in time period  $t$ , and  $n_d$  represents the depot. The pseudo code of the model in this case is shown as following:

**Lookahead tree search model**

```

% Assume the problem has  $n$  customers and one depot
customer = [c1, c2, ..., cn];
Generate  $\mathcal{R}^1$ ;
Step 1:
    Find the optimal_path1 by using  $\mathcal{R}^1$ ;
    Take the first customer in optimal_path1 and name it  $i_1$ ;
    Generate  $\mathcal{R}^2$ ;
Step 2:
    Find the optimal_path2(start from  $i_1$  and end up at the depot)
    by using  $\mathcal{R}^2$ ;
    Take the first customer in optimal_path2 and name it  $i_2$ ;
    Generate  $\mathcal{R}^3$ ;
    ...
Step  $n - 1$ :
    Find the optimal_path $n-1$ (start from  $i_{n-2}$  and end up at
    the depot) by using  $\mathcal{R}^{n-1}$ ;
    Take the first customer in optimal_path $n-1$  and name it
     $i_{n-1}$ ;
    Name the unselected customer  $i_n$ ;
    optimal_path = [depot;  $i_1$ ;  $i_2$ ; ...,  $i_{n-1}$ ;  $i_n$ ; depot];
return optimal_path;
END the process

```

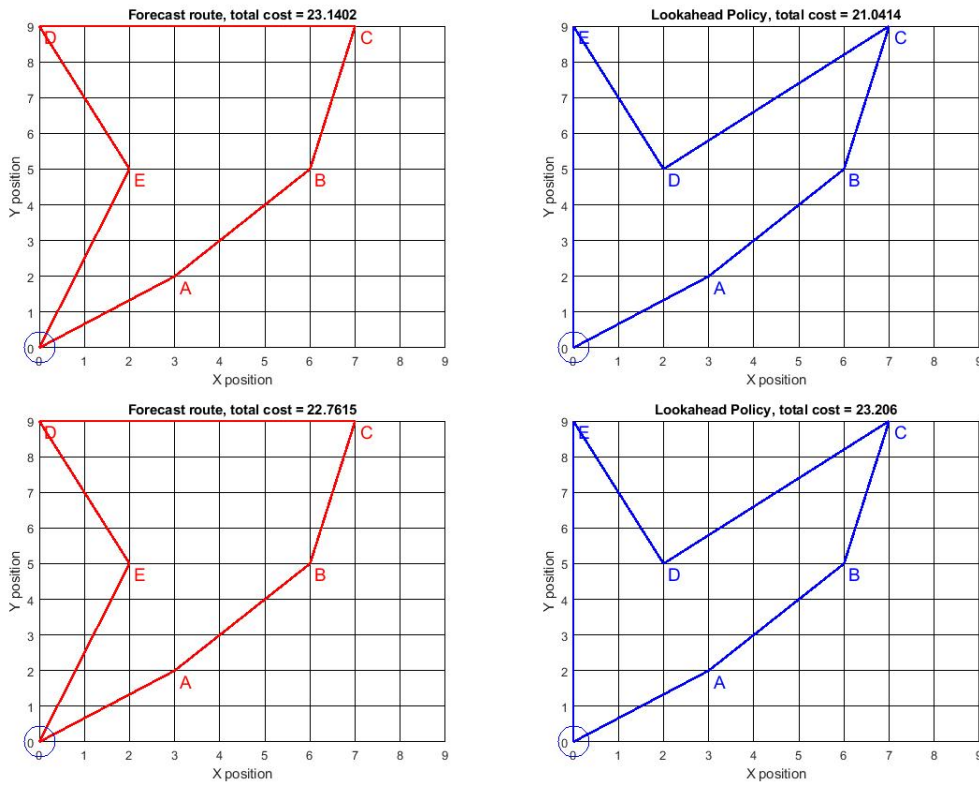
In the process above, the Lookahead tree search model does not return a round route. Instead, it finds the optimal chain route from the current node to the depot incorporating all customers that have not been visited yet.

Consider the case in figure 3.1 again, but this time with noised customer graph. Set the proportion factor  $\gamma = 0.1$  and the customer graph as  $\mathcal{R}^0$ . The solutions are given by simulating two trucks driving parallel in the same map, sharing the same customer graph set but one just follows the route that was provided by the Static model before start (use  $\mathcal{R}^0$  only), while the another keeps updating the route by the Lookahead model. From the results (figure 3.5 upper) the Lookahead model changes the forecasted route and leads to a less cost consumption.

As the number of the customer increases, the feasible solutions increases exponentially and leads to a significant long computation time. The Look-ahead model gives a solution within a second with five customer, however, it takes more than two days to handle ten customers. Finally, tree search model has been abandoned because it can not give an answer in a short time if there are more than six or seven customers, while efficiency is one of the factor to asses a model.

### 3.3 Linear programming

In this section, the traditional VRP and CVRP are solved dynamically based on LP formulation. These two types of routing problem can be seen as the simplified version of EVRP, which does not contain the energy cost related constraints and calculations. The final dynamic routing model for EVs is designed based on LP,



**Figure 3.5:** Solutions by implementing the Static tree search model (red) and the Lookahead tree search model (blue) with different customer graph set (upper and lower).

which is developed step-by-step from the simple initial model for the traditional combustion vehicles.

### 3.3.1 Mathematical model - simplified

The simplified model for solving the regular VRP in the grid world is formulated in this section. Here, since this is the first step of using LP to solve the real-time EVRP, the routing problem only considers a single vehicle and the constraints in other variations of VRP are not included. So the problem can be simply seen as the classic TSP, where the vehicle capacity and electricity associated constraints are not considered at this stage, which means the customer graph is purely calculated based on the Euclidean distance between node pairs. But one can treat it as an 'abstract' energy cost because the way people name it will not cause any effect to the result, only the value of it matters. This idea holds for all the model applications in grid map. The following formulation is the reproduction from [41], which targets to minimize the total cost of visiting  $n$  number of customers. In the report, this

simplified model is named as model 1 (M.1):

$$\min \sum_{i,j \in V} c_{ij} x_{ij} \quad (\text{M.1})$$

$$\text{s.t.} \quad \sum_{j=1}^n x_{ij} = 1, \quad \forall i \in \mathcal{V} \quad (3.2)$$

$$\sum_{i=1}^n x_{ij} = 1, \quad \forall j \in \mathcal{V} \quad (3.3)$$

$$\sum_{i,j \in V} x_{ij} \leq |V_s| - 1, \quad V_s \subset \mathcal{V} \setminus 0, 2 \leq |V_s| \leq V - 1 \quad (3.4)$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in E \quad (3.5)$$

The binary decision variables  $x_{ij}$  to be solved in this formulation specify which edge  $(i, j)$  that will be included in the optimal solution. (3.2) and (3.3) represent the assignment constraints that every known customer has to be served exactly once and the vehicle has to leave from that customer after completing the service. (3.4) is the sub-tour elimination constraints which require the number of selected edges in the optimal solution must be one less than the total number of the reached customer nodes before finally back to the depot. (3.5) is the integral constraints for decision variables.

In Matlab 2017b, such VRP problem can be solved by “intlinprog” function within the embedded optimization toolbox, it’s an integer linear programming solver that used to find the minimum of a given objective function. The solver uses simplex method to solve the problem, where try to find feasible edges combination among vertices within feasible region (a polygon formulated by constraints (3.2) ~ (3.5)), updating the minimal value in an iterative manner. The optimal routing solution of VRP will be find after finite time iterations.

However, (M.1) is difficult to apply on large-scale problems in real world because of the sub-tour constraint (3.4). As the number of customer to be served increases, the number of existing sub-tours will also increase. To the best of our knowledge, the solver does not know which are the sub-tours that must be avoided in the routing result beforehand, hence one has to list all possible sub-tours among the given customers before running the solver. This fact will result in the exponential increase on the number of sub-tour elimination constraints (3.4), which are obtained by listing all possible combinations of decision variables (edges), followed by filtering them with certain judgment conditions to find those combinations with practical meaning (e.g.  $x_{12} + x_{21} \leq 1$  is valuable,  $x_{12} + x_{35} + x_{46} \leq 3$  is not since they are not connected end-to-end). Such operation will severely slow down the speed of routing calculation. For example, to solve the optimal route for 7 customers case once at the beginning, (M.1) takes 1633 seconds to execute our model to generate the sub-tours elimination constraints matrix, and it takes 56 seconds for the solver to solve the optimal routing plan. This fact indicates that such rudiment model is incompetent to handle the practical cases, which requires high demand on quick reaction.

### 3.3.2 Mathematical model - improved

To eliminate sub-tours in solutions and improve the calculation efficiency, redundant operations such as listing all possible sub-tours must be avoided. Thus (3.4) has to be replaced by more feasible constraints. In all real-time case, the payload limitation must also be considered to make sure that an arranged routing plan for the dispatching tasks is realistic and meaningful. Therefore, as the second step of solving the real-time EVRP, additional vehicle capacity constraints in CVRP are introduced into the linear programming formulation in this stage. The vector of decision variables now consists of the binary decision variables  $x_{ij}$  for the optimal edge selections and the a new series of variables  $\omega_{ij}$ . The electric truck is assumed to be fully loaded before departing from the depot, and the weight of the payload is the sum of customer demands. The following mixed integer linear programming (MILP) formulation is the reproduction from [11], which is used to solve the CVRP in its case. In this report, the improved model is called model 2 (M.2).

$$\min \sum_{i,j \in V} c_{ij} x_{ij} \quad (\text{M.2})$$

$$\text{s.t.} \quad \sum_{j=1}^n x_{ij} = 1, \quad \forall i \in \mathcal{V} \quad (3.6)$$

$$\sum_{j=1}^n x_{ji} = 1, \quad \forall i \in \mathcal{V} \quad (3.7)$$

$$\sum_{j=1}^n \omega_{ij} - \sum_{j=1}^n \omega_{ji} = q_i, \quad \forall i \in \mathcal{V} \setminus O \quad (3.8)$$

$$q_j x_{ij} \leq \omega_{ij} \leq (Q - q_i) x_{ij}, \quad \forall (i, j) \in E \quad (3.9)$$

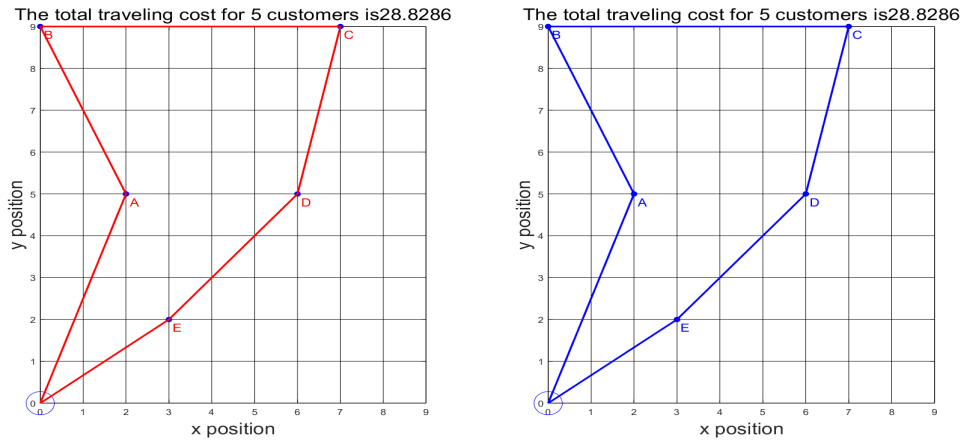
$$x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in E \quad (3.10)$$

$$\omega_{ij} \geq 0, \quad \forall (i, j) \in E \quad (3.11)$$

In this formulation,  $q_i$  represents the cargo weight demand of customer  $i$  and the depot does not have cargo weight demand.  $\omega_{ij}$  indicates the total remaining payload weight left on the vehicle when it is driving on the path between the customer  $i$  to the customer  $j$ . Similarly, (3.6) and (3.7) guarantee that every customer will be visited once and every customer has an incoming link and a corresponding outgoing link. (3.8) is the most ingenious constraints for eliminating sub-tours; it forces the weight of payload to decrease continuously. Noticing that constraint (3.8) is not applied on the edges that start from the depot, it means the weight variables that start from the depot  $\omega_{Oj}$  ( $j \in \mathcal{V} \setminus O$ ) are known, whose value is equal to the total cargo weight at the beginning. Therefore, such operation enables our solver to solve  $\omega_{ij}$  for other edges during the optimal routing process. On the contrary, if any sub-tour exists during the calculation process, then constraint (3.8) has no solution for those  $\omega_{ij}$  within the group of equations of the generated sub-tours. Comparing with the constraint (3.4) in (M.1), (3.8) does not need to spend plenty of time in finding and filtering the possible sub-tour combinations, instead, only the descending relationship constraints of payload between two nodes need to be declared in advance. Both the scale and the complexity of this part of the constraint matrix are simplified. Hence it greatly accelerates the computation process. (3.9)

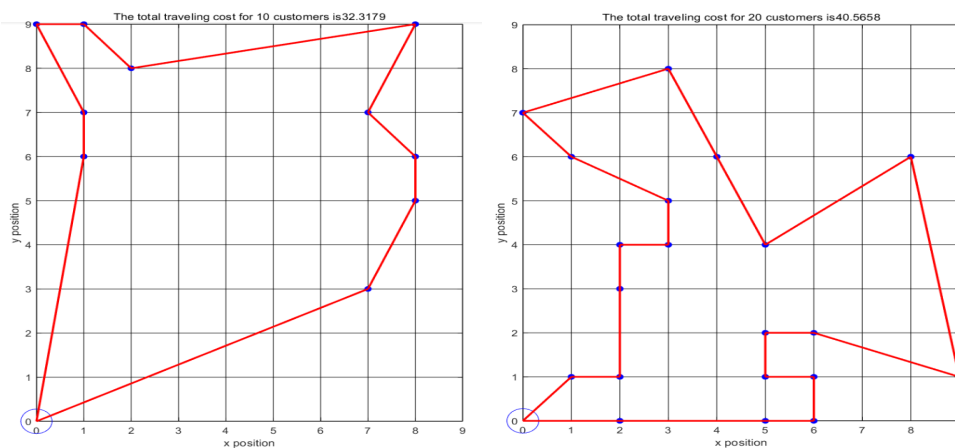
### 3. Grid Map

puts constraints on the maximum payload that carried by vehicle and force  $w_{ij}$  to be 0 if the edge  $x_{ij}$  is not part of the optimal routing solution. (3.10) and (3.11) specify the feasible value range for  $x_{ij}$  and  $\omega_{ij}$  respectively. In figure 3.6, the same test case is applied here to validate the consistency of the optimal solution solved by M.2 formulation above with respect to the designed tree-searching model in the previous section.



**Figure 3.6:** The optimal routing solution solved by M.2 model (left) and the optimal routing solution solved by Treesearching model (right).

Obviously, two optimization methods share the same global optimal result, which includes both the driving routes and the total cost. This fact illustrates that the M.2 formulation is definitely a better choice than M.1 to solve the proposed EVRP later. Besides, without searching for all possible branches as the method did in the tree-searching model, M.2 formulation will save a lot of time when facing an optimization problem with a larger size in reality.



**Figure 3.7:** The result routing solutions of 10 customers (left) and 20 customers (right)

To further increase the calculation speed of the model, the powerful optimization software tool CPLEX is integrated with the previous mixed integer programming

model. To initially test the routing efficiency of the proposed combinatorial model, two customer test sets with larger scales are created, where the number of customers is equal to 10 and 20 respectively. The final routing graph is shown in figure 3.7. The running time for 10 random customers is 0.82s and for 20 random customers is 2.79s. These two results show great computation speed of the model under static conditions. Hence, a significant improvement is presented comparing with the running time of the first M.1 model.

To solve the simplified EVRP in a dynamic manner by using the proposed M.2 model, the whole routing process needs to be decomposed into a series of EVRPs with changeable constraints and edge cost coefficients according to the circumstance information during the current time interval. This transformation operation is the key of converting an EVRP into a dynamic electric vehicle routing problem (D-EVRP). By applying this principle on our case, a new static and deterministic M.2 needs to be solved at each time before the vehicle sets off for the next customer. The new EVRP that generated at the next time index takes both the unvisited customers as well as the latest customer graph  $\mathcal{R}$  into account. The new constraints of this EVRP are adapted accordingly since the number of remaining cargo on the truck has been reduced after serving the last customer. For those customers who have already been visited, the value of their corresponding decision variables  $x_{ij}$  will be fixed permanently. Hence these variables become known to the solver and the vehicle has to go through the corresponding paths in the next time calculation. The solver will solve the optimal paths for the rest of trips based on the paths that the vehicle has already passed through.

Whenever M.2 is executed, it gives a prediction routing result which starts from the current node and end at the depot. The final complete optimal routing solution of a D-EVRP is constructed by selecting only the first step of the visiting order solutions from those EVRPs, which means only the path that reaches the next customer in each static solution is count for our final optimal solution. Such MPC principle based operation is reasonable in reality because it considers the timeliness of every prediction solution. On the contrary, the rest parts of route in the solution become unreliable as they are far from the current position, the customer graph  $\mathcal{R}$  will be different when the vehicle reaches the next customer. Assuming that the disturbance on customer graph in the grid map is caused by the variation of a time-depending distribution  $\mathcal{N}^t$  when the current network customer graph  $\mathcal{R}$  is re-sampling from it. The following pseudo-code shows the working principle of the M.2 based dynamic routing model in the grid map case (GDM.2).

1	<b>M.2 based model in D-EVRP (Grid map): GDM.2</b>
2	<b>Input parameters:</b>
3	Total number of discrete time indexes: $N$ ,
4	Depot Position: $O$ ,
5	Random Customer Positions: $V$ ,
6	Customer Demands: $q$ ,
7	Maximum Payload: $Q$ ,
8	$\mathcal{R}^0 = MapInitialization([O, V])$
9	<b>Model parameters:</b>
10	Discrete time index: $t$ ,

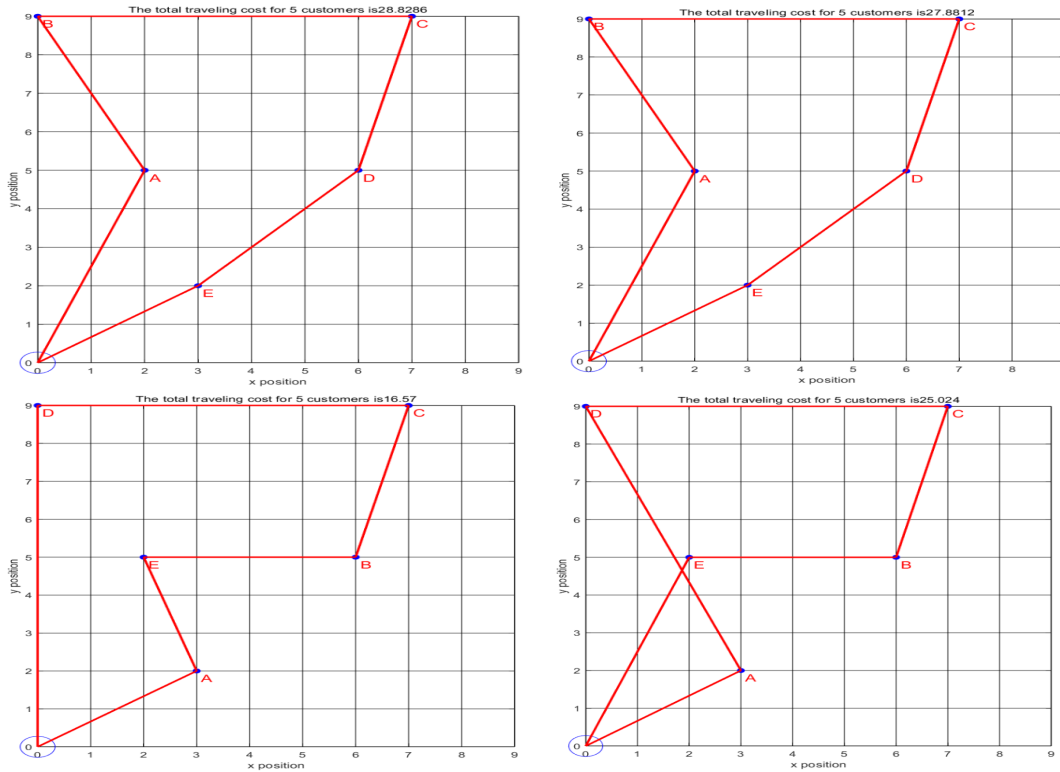
### 3. Grid Map

---

```
11     Total energy cost:  $E$ ,
12     Optimal visit order:  $R$ 
13
14 Initialize the time index  $t$  with 0,
15 Initialize the total energy cost  $E$  with 0,
16 Initialize the optimal visit order  $R$  with empty set
17
18 While  $t$  is less than  $N$  do:
19     Predictive order  $\leftarrow M.2(\mathcal{R}^t, [O, V])$ ,
20     Take the first element  $r$  in predictive order,
21      $R = R \cup r$ ,
22
23     Fixing the value of decision variable  $x_{ij}$  to 1
24     if the vehicle drove from  $i$  to  $j$ ,
25
26     Adding the corresponding trip cost:
27      $E = E + EnergyConsumption(i, j)$ ,
28
29     Time index evolving  $t = t + 1$ ,
30      $\mathcal{R}^t \leftarrow SampleFunction(CostDistribution)$ 
31 end While
32 Output:  $E$ ,  $R$ .
```

To ensure that the vehicle always starts from the depot (the origin in the pseudo code), in practical programming, the depot  $\mathbf{O}$  is an extra node that added as the first element in the customer vector. The cost that generates from the weight of payload as well as the body itself is not included in the objective function since in the grid map, the cost coefficients are calculated based on the direct distance between two nodes. Assuming that the total number of nodes (including the depot and customers) is  $N$ , hence the whole routing process will be divided into  $N$  time intervals. During every discrete time interval  $t$ , a temporary optimal solution is solved, but the vehicle picks only the first customer in the solution to be the next destination. During every iteration, the passed route is recorded and fixed, hence a smaller dimension problem is produced according to the adaptations from line 23 to 24. This process is repeated until all the customers have been visited and the vehicle returns to the depot. Then the optimal order of visiting customer is obtained and the given D-EVRP is solved. The total energy cost is obtained by summing up the local energy cost at each step.

We further test our GDM.2 model in disturbed dynamic case based on the same customers cluster in figure 3.6 with different proportional variance values  $\gamma$ . The result figures are shown in figure 3.8.



**Figure 3.8:** The result routing solutions of 10 customers under four variance cases:  $\gamma = 0$  (upper-left),  $\gamma = 0.05$  (upper-right),  $\gamma = 0.2$  (lower-left),  $\gamma = 0.4$  (lower-right)

When  $\gamma = 0.05$ , the routing result does not be changed because the disturbance level caused by the variance is too weak to affect the optimization process at each customer, so the same solution is obtained as the noise-free case where  $\gamma = 0$ . As the disturbance level goes up, when  $\gamma = 0.2$ , the disturbance will make some differences on the result since the amplitude of disturbance now is able to change some parts of the customer graph. This fact implies that to follow the original routing plan of the noise-free routes is not the optimal choice anymore. A larger variance coefficient  $\gamma$  will naturally generate a very different routing solution since the customer graph always changes significantly at each sampling time, this fact is proven as the last subplot shows when  $\gamma = 0.4$ . However, the optimality of the obtained solution at each time step is still be guaranteed by executing the M.2 model.

### 3.4 Experiments

To see how this model works in grid world under different variances, 20 unique customer sets (10 customers for each) are created, for each set the customer graph has four variance factor  $\gamma_g \in [0.05, 0.1, 0.15, 0.2]$ , implement GDM.2 on each case 5 times in parallel where 1. the truck keeps following the forecast route all the way (FD); 2. the truck updates the route if necessary (DR), and take the average of the total cost.

### 3. Grid Map

Case	DR	FD	$r$	Case	DR	FD	$r$
1	31.8355	32.1644	-1.02%	1	32.0709	31.3048	2.45%
2	33.7750	34.0503	-0.81%	2	31.4635	33.5202	-6.14%
3	36.6615	37.7066	-2.77%	3	34.1113	37.0796	-8.01%
4	35.8223	36.9385	-3.02%	4	34.5726	37.0298	-6.64%
5	30.4506	31.2140	-2.45%	5	29.8351	29.7652	0.23%
6	23.6688	24.0348	-1.52%	6	22.1127	23.8597	-7.32%
7	28.9988	28.6529	1.21%	7	29.4554	30.6240	-3.82%
8	28.9314	29.5952	-2.24%	8	27.5057	29.3650	-6.33%
9	32.3725	32.5803	-0.64%	9	31.3400	31.4643	-0.40%
10	33.0782	33.3039	-0.68%	10	32.3027	34.7416	-7.02%
11	32.4239	32.7265	-0.92%	11	31.3625	32.3381	-3.02%
12	29.8274	29.9736	-0.49%	12	28.4448	30.0789	-5.43%
13	32.6361	33.5453	-2.71%	13	31.8699	33.6336	-5.24%
14	30.5220	30.9767	-1.47%	14	29.1100	29.9570	-2.83%
15	31.6653	31.7702	-0.33%	15	29.0136	31.1067	-6.73%
16	35.2626	36.0185	-2.10%	16	33.9198	38.0685	-10.90%
17	29.6088	30.8193	-3.93%	17	28.2262	31.5182	-10.44%
18	32.8354	33.5437	-2.11%	18	30.0905	33.3772	-9.85%
19	31.5143	31.6544	-0.44%	19	30.5727	31.1473	-1.84%
20	27.7711	28.1678	-1.41%	20	27.4090	27.7432	-1.20%
AVG			-1.49%	AVG			-5.02%
MAX			1.21%	MAX			2.45%
MIN			-3.93%	MIN			-10.90%

(a) $\gamma_g = 0.05$				(b) $\gamma_g = 0.1$			
Case	DR	FD	$r$	Case	DR	FD	$r$
1	30.4610	33.3413	-8.64%	1	27.2026	27.7671	-2.03%
2	31.3370	33.1446	-5.45%	2	26.9804	29.7105	-9.19%
3	30.4290	38.6903	-21.35%	3	28.8434	40.3383	-28.50%
4	31.1470	35.8174	-13.04%	4	26.5213	36.3036	-26.95%
5	26.0255	30.7436	-15.35%	5	21.9543	30.5266	-28.08%
6	20.8324	25.2233	-17.41%	6	19.3566	26.0509	-25.70%
7	27.2947	27.7809	-1.75%	7	24.9496	29.6787	-15.93%
8	26.9691	27.3940	-1.55%	8	25.2114	28.5402	-11.66%
9	31.5727	30.3291	4.10%	9	26.7348	36.4629	-26.68%
10	30.1721	31.7625	-5.01%	10	27.6063	33.4467	-17.46%
11	29.5123	33.3584	-11.53%	11	25.6919	33.0603	-22.29%
12	25.8003	27.8479	-7.35%	12	25.9780	31.0705	-16.39%
13	29.4091	33.9102	-13.27%	13	28.5059	35.8233	-20.43%
14	24.9736	31.8899	-21.69%	14	22.7122	26.4453	-14.12%
15	27.6221	34.1713	-19.17%	15	23.1178	29.4154	-21.41%
16	29.4465	33.8120	-12.91%	16	27.9073	36.2187	-22.95%
17	28.5114	31.3738	-9.12%	17	22.0832	31.0061	-28.78%
18	27.9980	33.8793	-17.36%	18	26.0603	33.8561	-23.03%
19	28.0531	31.7574	-11.66%	19	22.6581	34.7870	-34.87%
20	25.1073	28.4219	-11.66%	20	24.3538	28.8965	-15.72%
AVG			-11.06%	AVG			-20.61%
MAX			4.10%	MAX			-2.03%
MIN			-21.69%	MIN			-34.87%

(c) $\gamma_g = 0.15$				(d) $\gamma_g = 0.2$			
Case	DR	FD	$r$	Case	DR	FD	$r$
1	30.4610	33.3413	-8.64%	1	27.2026	27.7671	-2.03%
2	31.3370	33.1446	-5.45%	2	26.9804	29.7105	-9.19%
3	30.4290	38.6903	-21.35%	3	28.8434	40.3383	-28.50%
4	31.1470	35.8174	-13.04%	4	26.5213	36.3036	-26.95%
5	26.0255	30.7436	-15.35%	5	21.9543	30.5266	-28.08%
6	20.8324	25.2233	-17.41%	6	19.3566	26.0509	-25.70%
7	27.2947	27.7809	-1.75%	7	24.9496	29.6787	-15.93%
8	26.9691	27.3940	-1.55%	8	25.2114	28.5402	-11.66%
9	31.5727	30.3291	4.10%	9	26.7348	36.4629	-26.68%
10	30.1721	31.7625	-5.01%	10	27.6063	33.4467	-17.46%
11	29.5123	33.3584	-11.53%	11	25.6919	33.0603	-22.29%
12	25.8003	27.8479	-7.35%	12	25.9780	31.0705	-16.39%
13	29.4091	33.9102	-13.27%	13	28.5059	35.8233	-20.43%
14	24.9736	31.8899	-21.69%	14	22.7122	26.4453	-14.12%
15	27.6221	34.1713	-19.17%	15	23.1178	29.4154	-21.41%
16	29.4465	33.8120	-12.91%	16	27.9073	36.2187	-22.95%
17	28.5114	31.3738	-9.12%	17	22.0832	31.0061	-28.78%
18	27.9980	33.8793	-17.36%	18	26.0603	33.8561	-23.03%
19	28.0531	31.7574	-11.66%	19	22.6581	34.7870	-34.87%
20	25.1073	28.4219	-11.66%	20	24.3538	28.8965	-15.72%
AVG			-11.06%	AVG			-20.61%
MAX			4.10%	MAX			-2.03%
MIN			-21.69%	MIN			-34.87%

**Figure 3.9:** Table: Average total energy consumption of each case by using DR ( $V_{DR}$ ) and FD ( $V_{FD}$ ) under four different variance factor.

In table 3.9,  $r = \frac{V_{DR}-V_{FD}}{V_{FD}}$  represents the amount (in proportion) of energy that GDM.2 has changed with respect to FD, negative  $r$  means the dynamic model has reduced the energy consumption, vice versa.

It is obvious that most of the cases have negative change rate, meaning that dynamic model does help to reduce energy consumption. Furthermore, as  $\gamma_g$  increases from

0.05 to 0.2 the absolute value of the costs increases as well in most cases, which means generally the dynamic model saves more energy when the customer graph becomes more unstable.

Some positive change rates exist because in some cases when the truck has visited 9 customers and only one left, there is no choice but go straight to the last customer and back to the depot, after the truck has reached the last customer the cost of the link to the depot might increase by a fairly large value, driving on this edge will cost a huge amount of energy even more than what it saved before. However, this sort of issue happens with a very small possibility and if it happens, the change rate would not be too high.

### 3. Grid Map

---

# 4

## Real Map

In this chapter, the electric routing problem within real world situations is discussed and analyzed. The grid map is replaced by Google Map and the road information comes from a traffic scenario for Luxembourg developed for SUMO [42], including but not limited to intersections, links, as well as the average speed, the required time for passing each link, etc. The cost is no more based on the distance but the electricity consumption. The process of the optimization now is divided into two steps. **Step one:** generate the customer graph; **Step two:** find the optimal order of the customers to obtain an optimal route.

Figure 4.1 shows the road network of Luxembourg City in Google Map [43]. Assuming the depot is located in the southwest of the city, labeled with *A*. The information of this network is provided by SUMO, including: 5779 intersections with location (longitude, latitude and altitude) and 11286 directed links connecting these intersections; each link is described by the inherent parameters:  $\alpha$ ,  $\beta$ , distance, speed,  $\sigma$ , and the passing time. The physical meanings of the letters will be explained in the following section.

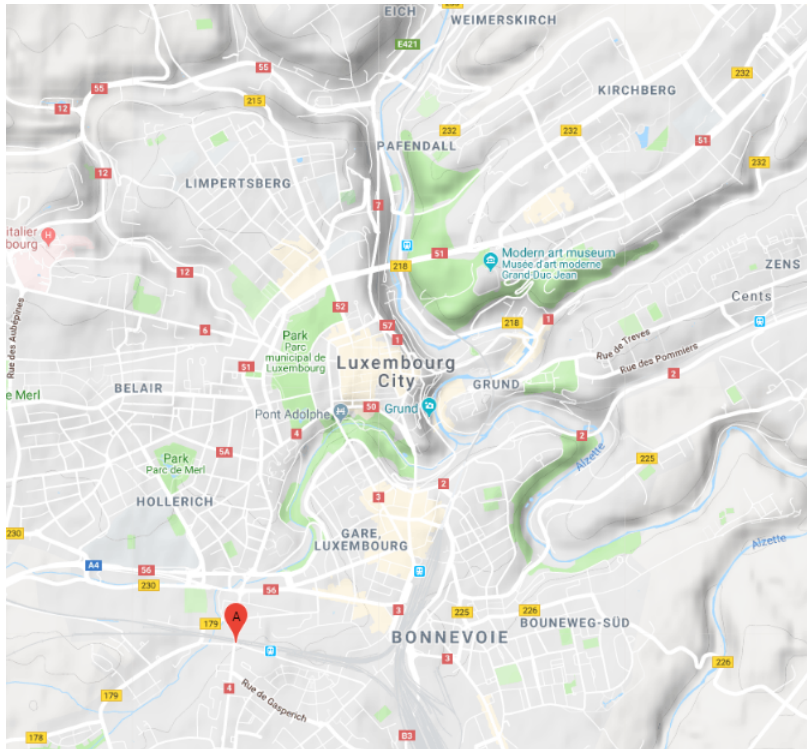


Figure 4.1: Map of Luxembourg City.

## 4.1 Routing without charging

### 4.1.1 Customer graph

Since the idea of creating a customer graph is to find the best path between each two nodes, and in the real map, one cannot do this by only taking direct links as the method did in the grid map. A series of consecutive edges must be considered here, while the summation of the energy costs of these edges should be the least. Firstly, One needs to figure out the energy cost of every edge.

According to equation (4.1), to estimate the energy consumption of link  $(i, j)$ ,  $\alpha_{ij}$  and  $\beta_{ij}$  need to be defined and the total weight  $m$  as well as the average passing velocity  $\bar{v}_{ij}$  should be known. Here  $m = W + \omega_{ij}$ , while  $W$  is the truck weight, and  $\omega_{ij}$  the payload.

$$e_{ij} = \alpha_{ij}(W + \omega_{ij}) + \beta_{ij}\bar{v}_{ij}^2 \quad (4.1)$$

However,  $\omega_{ij}$  depends on the order of the customers, and the only thing known beforehand is the demand of the customers. Here, let  $\omega_{ij} = \bar{\omega}_{ij}$ , which is the half value of the summation of the customers' demands.

$$\bar{\omega}_{ij} = \frac{1}{2} \sum_{i=1}^N q_i \quad (4.2)$$

Where  $q_i$  is the demand of customer  $i$ . After that, a variation of the Shortest Path algorithm is used to find the minimum cost path between each two nodes. Note that here the customer graph is built with approximate energy estimation, which will be used in the routing models, but the exact energy consumption will be calculated with the corresponding load weight.

### 4.1.2 Find the optimal route

The linear programming model introduced in the previous chapter is improved to be able to handle real map. Let  $x_{ij}$  be the selected link  $(i, j)$ , the cost function is the entire energy estimation and has the form:

$$f(x) = \sum_{i,j \in V} (\alpha_{ij}(W x_{ij} + \omega_{ij}) + \beta_{ij}\bar{v}_{ij}^2 x_{ij}) \quad (4.3)$$

Note that in real world there are numbers of factors that could affect the energy cost, such as traffic congestion, traffic light, driver's behavior, weather, etc. Numerically, they cause same result - increasing or decreasing the cost. In equation (4.1), it is simple to change the energy cost of link  $(i, j)$  by adding a factor  $\gamma_m$ , the new energy cost value is expressed as equation (4.4):

$$\begin{aligned} e_{ij}^* &= \gamma_m(\alpha_{ij}(W + \omega_{ij}) + \beta_{ij}\bar{v}_{ij}^2) \\ &= \gamma_m\alpha_{ij}(W + \omega_{ij}) + \gamma_m\beta_{ij}\bar{v}_{ij}^2 \end{aligned} \quad (4.4)$$

From (4.4), change the energy cost of  $(i, j)$  is equal to change the values of  $\alpha_{ij}$  and  $\beta_{ij}$ .

Note that the decision variable is no more  $x_{ij}$  but the payload  $\omega_{ij}$  as well. At the beginning, the payload equals to the summation of demands of all customers, when the truck reaches a customer  $i$ , the payload will be decreased by  $q_i$ , and so on. In the real-world case, the pseudo code of the M.2 based dynamic routing model in D-EVRP with the energy prediction model (RDM.2) is written as the following:

```

1 M.2 based dynamic routing model in D-EVRP with the energy predic-
2 tionmodel (Real map): RDM.2
3 Input parameters:
4   Total length of the discrete time index:  $N$ ,
5   Depot Position:  $O$ ,
6   Random Customer Positions:  $V$ ,
7   Customer Demands:  $q$ ,
8   Maximum Payload:  $Q$ ,
9   Mass related coefficient matrix:  $\alpha$ ,
10  Velocity related coefficient matrix:  $\beta$ ,
11  Initial customer graph:
12   $\mathcal{R}^0 = StageOneFunction([O, V], \alpha, \beta)$ ;
13 Model parameters:
14  Discrete time index:  $t$ ,
15  Total energy cost:  $E$ ,
16  Optimal visit order:  $Ord$ ,
17  Optimal detailed paths:  $P$ 
18 Initialize the time index  $t$  with 0,
19 Initialize the total energy cost  $E$  with 0,
20 Initialize the optimal visit order  $Ord$  with empty set
21 Initialize the optimal detailed paths  $P$  with empty set
22
23 While  $t$  is less than  $N$  do:
24
25   Predictive order,  $w \leftarrow M.2(\mathcal{R}^t, [O, V])$ ,
26
27   Pick the first element  $j$  in predictive order:
28    $R = R \cup j$ ;
29
30   Fixing the value of decision variable  $x_{ij}$  to 1
31   if the vehicle drove from  $i$  to  $j$ ;
32
33   Finding out the best path between node  $i$  and  $j$  from
34   the current customer graph  $\mathcal{R}^t$ :
35    $P(i, j) = FindDetailedPaths((i, j), \mathcal{R}^t)$ ;
36    $P = P \cup P(i, j)$ ;
37
38   Adding the corresponding trip cost:

```

```

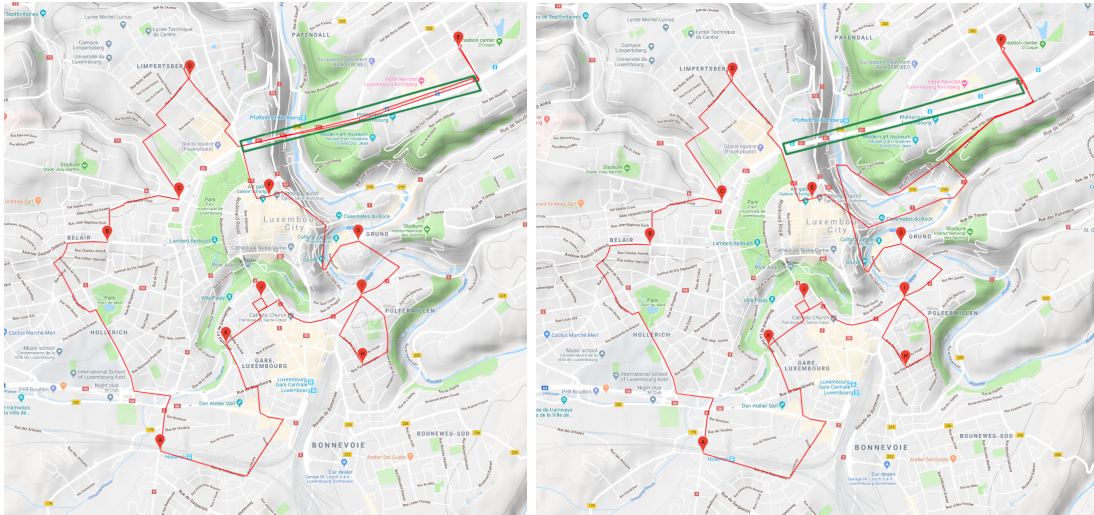
39    $E = E + \text{energyconsumption}(i, j);$ 
40   Discrete time index evolves:  $t = t + 1;$ 
41
42   if any incident is detected at time  $t$ , then: the cost
43   coefficients will change and the customer graph needs
44   to be re-calculated before the next time
45   re-optimization:
46    $\alpha = \alpha_{\text{incident}};$ 
47    $\beta = \beta_{\text{incident}};$ 
48    $\mathcal{R}^{t+1} = \text{StageOneFunction}([O V], \alpha, \beta);$ 
49   else:
50    $\mathcal{R}^{t+1} = \mathcal{R}^t;$ 
51   end if
52
53 end While
54 Return  $E, \text{Ord}, P;$ 

```

Similar to the pseudo codes of Grid map, for those selected paths in the solution, the value of the decision variables  $x_{ij}$  are set to 1 after each iteration. The corresponding payload variable  $\omega_{ij}$  will be calculated by the model automatically thereafter. In this dynamic case, the total energy cost of the whole trip is accumulated by the energy cost of every individual path  $(a, b)$  in each time re-optimization. Specifically, the energy cost from the current customer  $a$  to the next customer  $b$  is calculated based on (4.3):  $e_{ab} = \alpha_{ab}(Wx_{ab} + \omega_{ab}) + \beta_{ab}\bar{v}_{ab}^2x_{ab}$ . If any incident is detected on the road network ahead at time  $t$ , the corresponding congestion on those roads can be simulated manually by assigning the value of  $\gamma_m$  to make the product of  $\gamma_m\alpha_{ij}$  and  $\gamma_m\beta_{ij}$  larger. The new customer graph will be re-calculated based on the new cost coefficients, hence the optimal links of path around the incident area will probably be changed. The outcome from this dynamic algorithm consists of the total energy consumption, the visiting order of customers and the detailed link connections that used for visualization thereafter.

### 4.1.3 Experiments

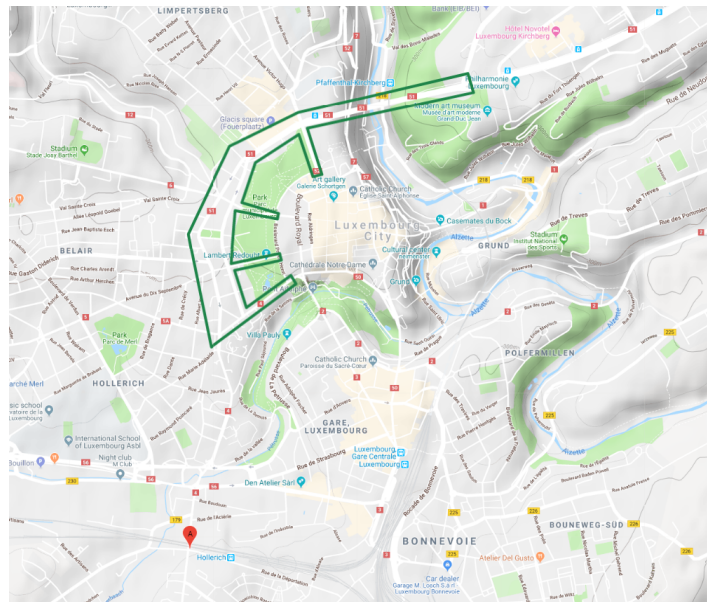
To see how this model works, given the map and a set of customers  $(B, C, D, E, F, G, H, I, J, K)$ , figure 4.2 (Left) shows the forecast route (the optimal route predicted before departure), the truck departures from depot  $A$ , and travels along the path  $(B - C - D - E - F - G - H - I - J - K - A)$ , and the entire energy cost is 20141  $Wh$ . Assume that when the truck has reached  $D$ , something happens to the area which is marked by the green window in figure 4.2 Left, leading to a large passing cost on all links within the area ( $\gamma_m = 20$ ). figure 4.2 Right shows the result from the improved M.2 based dynamic model RDM.2.



**Figure 4.2:** Left: Forecast route. Right: The route updated by the RDM.2 model.

From figure 4.2 (Right) it is obvious that the route after  $D$  has been changed and the order of the customers as well. The total energy consumption of the new route is  $24855 Wh$ . While if the vehicle does not change the routing plan and still follows the forecast route, the total energy cost would be  $38598 Wh$ . In comparison, the RDM.2 has saved 35.6% energy in this situation.

Now let us move on to general cases. 20 different cases are generated, in each case 10 customers are located randomly within the downtown area. For every case, there is an area ( $E_b$ ) that when the truck has arrived at the 5th node, all the links  $(i, j) \in E_b$  will be 'blocked' (figure 4.3) for a certain time period. Assume the battery is suffi-



**Figure 4.3:** 'Blocked' area in Luxembourg City.

cient for the entire trip. The table in figure 4.4 gives the total energy consumption with  $\gamma_m = [2.5, 5, 7.5, 10]$  respectively and under three circumstances: the total cost

#### 4. Real Map

of the forecast route with the initial customer graph (FI); the total cost of driving along the forecast route but considering the changed customer graph (FD); the total cost of following the route updated from the RDM.2. Note that our model can not give the energy consumption of FI directly since the customer graph has been changed, thus the energy cost after the 5th node is calculated by manually adding up the new costs link by link.

Case	FI	FC	DR	$r$	Case	FI	FC	DR	$r$
1	18947	19961	19678	-1.42%	1	18947	21651	19796	-8.57%
2	20966	21259	21259	0.00%	2	20966	21749	21749	0.00%
3	21489	23844	22973	-3.65%	3	21489	28076	24191	-13.84%
4	9909	14599	12300	-15.75%	4	9909	40257	14835	-63.15%
5	15303	16478	15463	-6.16%	5	15303	18436	15634	-15.20%
6	14345	15765	14782	-6.24%	6	14345	19508	15606	-20.00%
7	18101	21721	19521	-10.13%	7	18101	41816	20493	-50.99%
8	18294	18294	18294	0.00%	8	18294	18294	18294	0.00%
9	17634	18586	18279	-1.65%	9	17634	20172	18175	-9.90%
10	13015	17173	13974	-18.62%	10	13015	41360	14249	-65.55%
11	21973	26963	24869	-7.77%	11	21973	51847	27529	-46.90%
12	19828	23644	21248	-10.13%	12	19828	48336	23615	-51.14%
13	17218	17218	17218	0.00%	13	17218	17218	17218	0.00%
14	13693	18535	15530	-16.21%	14	13693	44239	17516	-60.41%
15	16759	20253	17653	-12.84%	15	16759	43934	17820	-59.44%
16	14015	18203	15202	-16.49%	16	14015	43454	16108	-62.93%
17	18548	18548	18548	0.00%	17	18548	18548	18548	0.00%
18	20116	21143	20173	-4.58%	18	20116	22854	20173	-11.73%
19	24692	28558	26699	-6.51%	19	24692	36153	26922	-25.53%
20	15370	16866	16866	-0.00%	20	15370	19664	18060	-8.16%
AVG				-6.91%	AVG				-28.67%
MAX				0.00%	MAX				0.00%
MIN				-18.62%	MIN				-65.55%

(a) $\gamma_m = 2.5$					(b) $\gamma_m = 5$				
Case	FI	FC	DR	$r$	Case	FI	FC	DR	$r$
1	18947	23342	19705	-15.58%	1	18947	25032	19404	-22.48%
2	20966	22238	22212	-0.12%	2	20966	22727	22212	-2.27%
3	21489	32693	24394	-25.38%	3	21489	37694	24435	-35.18%
4	9909	105273	17017	-83.83%	4	9909	229325	19109	-91.67%
5	15303	20395	15804	-22.51%	5	15303	22353	15975	-28.53%
6	14345	26283	16264	-38.12%	6	14345	37609	16921	-55.01%
7	18101	93069	21466	-76.94%	7	18101	191146	22385	-88.29%
8	18294	18294	18294	0.00%	8	18294	18294	18294	0.00%
9	17634	21757	17843	-17.99%	9	17634	23343	17282	-25.97%
10	13015	103788	14359	-86.17%	10	13015	223692	14242	-93.63%
11	21973	112037	31246	-72.11%	11	21973	224378	36223	-83.86%
12	19828	113469	23719	-79.10%	12	19828	239264	23821	-90.04%
13	17218	17218	17218	0.00%	13	17218	17218	17218	0.00%
14	13693	108843	19252	-82.31%	14	13693	231795	20036	-91.36%
15	16759	107006	17987	-83.19%	15	16759	229165	18153	-92.08%
16	14015	107985	16291	-84.91%	16	14015	230764	16474	-92.86%
17	18548	18548	18548	0.00%	17	18548	18548	18548	0.00%
18	20116	24566	20173	-17.88%	18	20116	26277	20173	-23.23%
19	24692	46454	26559	-42.83%	19	24692	60917	25984	-57.35%
20	15370	22845	18416	-19.39%	20	15370	26409	18214	-31.03%
AVG				-42.42%	AVG				-50.24%
MAX				0.00%	MAX				0.00%
MIN				-86.17%	MIN				-93.63%

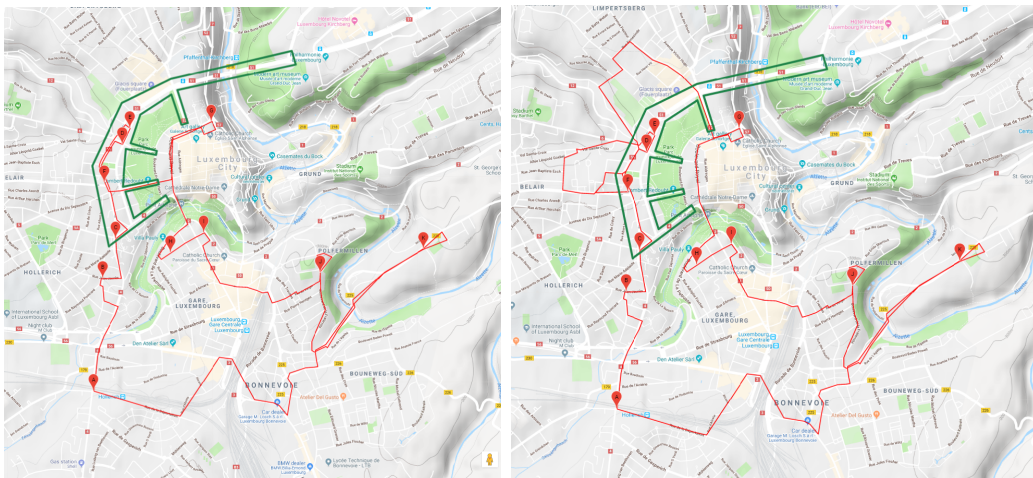
(c) $\gamma_m = 7.5$					(d) $\gamma_m = 10$				
Case	FI	FC	DR	$r$	Case	FI	FC	DR	$r$
1	18947	23342	19705	-15.58%	1	18947	25032	19404	-22.48%
2	20966	22238	22212	-0.12%	2	20966	22727	22212	-2.27%
3	21489	32693	24394	-25.38%	3	21489	37694	24435	-35.18%
4	9909	105273	17017	-83.83%	4	9909	229325	19109	-91.67%
5	15303	20395	15804	-22.51%	5	15303	22353	15975	-28.53%
6	14345	26283	16264	-38.12%	6	14345	37609	16921	-55.01%
7	18101	93069	21466	-76.94%	7	18101	191146	22385	-88.29%
8	18294	18294	18294	0.00%	8	18294	18294	18294	0.00%
9	17634	21757	17843	-17.99%	9	17634	23343	17282	-25.97%
10	13015	103788	14359	-86.17%	10	13015	223692	14242	-93.63%
11	21973	112037	31246	-72.11%	11	21973	224378	36223	-83.86%
12	19828	113469	23719	-79.10%	12	19828	239264	23821	-90.04%
13	17218	17218	17218	0.00%	13	17218	17218	17218	0.00%
14	13693	108843	19252	-82.31%	14	13693	231795	20036	-91.36%
15	16759	107006	17987	-83.19%	15	16759	229165	18153	-92.08%
16	14015	107985	16291	-84.91%	16	14015	230764	16474	-92.86%
17	18548	18548	18548	0.00%	17	18548	18548	18548	0.00%
18	20116	24566	20173	-17.88%	18	20116	26277	20173	-23.23%
19	24692	46454	26559	-42.83%	19	24692	60917	25984	-57.35%
20	15370	22845	18416	-19.39%	20	15370	26409	18214	-31.03%
AVG				-42.42%	AVG				-50.24%
MAX				0.00%	MAX				0.00%
MIN				-86.17%	MIN				-93.63%

**Figure 4.4:** The total cost ( $Wh$ ) in 20 different cases under three circumstances, the change rate  $r = \frac{V_{DR}-V_{FD}}{V_{FD}}$ .

The negative  $r$  represent the energy cost of the RDM.2 in % difference with respect

to the FD. As  $\gamma_m$  increases from 2.5 to 10, the saved energy amount in proportion increases from 6.91% to 50.24%. One implication is that the RDM.2 has changed the forecast route by changing the order of the customers or reforming the detailed path between them, or a combination of these. Case 2 and 20 indicate that sometimes RDM.2 does not change anything if  $\gamma_m$  is relatively low, in this case the cost of switching customers or planing a new detailed path is still larger than the forecast. When  $\gamma_m$  is greater than a certain number the RDM.2 will start to find another solution. Figure 4.5 shows the solutions of case 20. In the FI solution, after the vehicle has reached node  $E$ , it still drives within the 'blocked' area for a while, and leave the area after it has completed the service in  $F$ . In the FD solution, the truck tries to avoid the 'blocked' region once the customer graph has changed and only passes a single link within it before  $G$ , otherwise it would take a huge detour if the truck chooses another way without crossing the 'blocked' region, which would costs more energy.

The RDM.2 model does not make any change in case 8, 13 (figure 4.6), 17, the



**Figure 4.5:** Left: Forecast route (FI). Right: The route from the RDM.2 model (FD).

reason is that the forecast routes from  $E$  to the end in these three cases do not include any links which are inside the 'blocked' area, and thus the entire cost would not change regardless of how large the  $\gamma_m$  is.

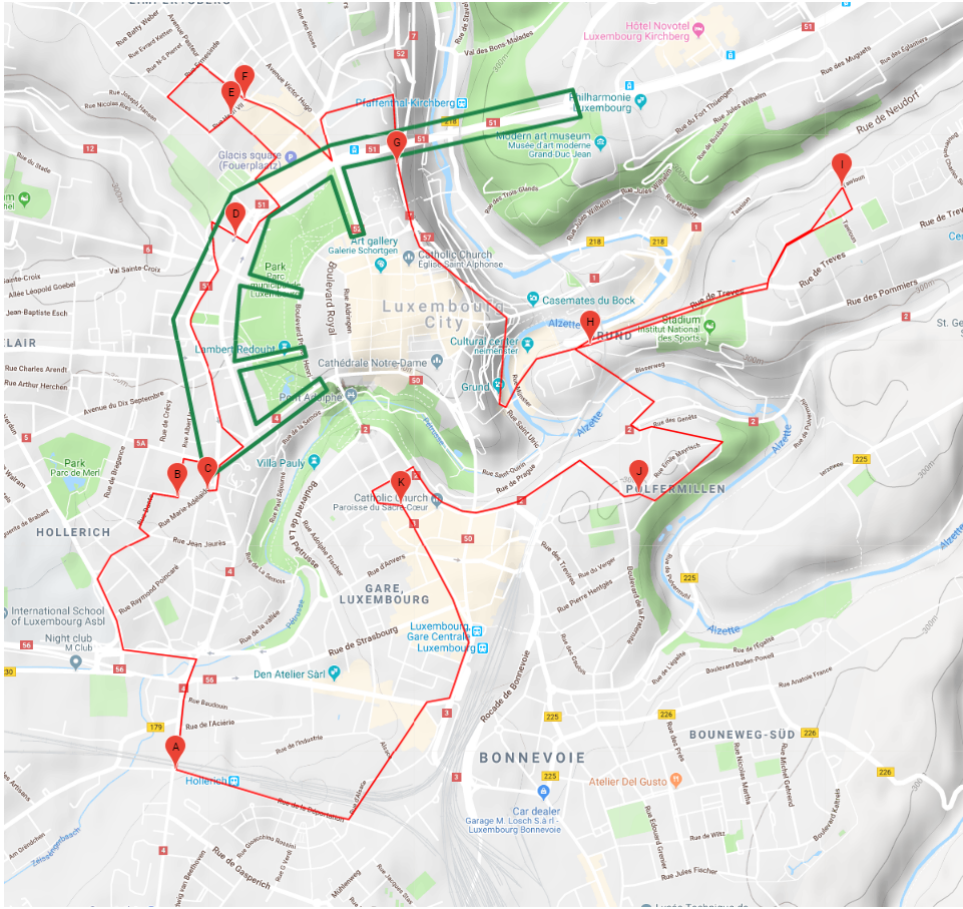


Figure 4.6: The route in case 13.

Since the scenario above is ideal where some extreme results exist, it can be made closer to reality by adding stochastic elements. Here  $\gamma_m$  is not fixed over the process but sampled from the distributions with the mean value  $\mu_m = [2.5, 5, 7.5, 10]$  and the standard variance  $\sigma^2 = 0.5$ .

## 4.2 Routing with charge stations

### 4.2.1 Model formulation

In real-time case, the current battery level of the electric vehicle depends on the energy depletion during the driving process. It is a necessary factor to concern because without enough charge, the electric vehicle can not go anywhere. Sometimes the vehicle has to go to the charging station before continuing the delivery tasks. Such consideration will put extra constraints on our previous mixed-integer model. Assuming that the variable set  $\mathbf{b} = \{b_i | i \in V\}$  represents the remaining battery level after reaching the certain node  $i$  and let  $V = S \cup C \cup O$ , where  $S$  represents the charging stations.  $C$  and  $O$  represent customers and depot respectively,  $B$  represents the capacity of the fully charged battery. The variable vector to be solved now consists of path decision variables  $x_{ij}$ , the weight variables  $\omega_{ij}$  and  $b_i$ . Therefore, by referring the model (30) in paper [10], the new mixed-integer linear programming

model with considering the recharging problem (M.3) can be written as:

$$\min \sum_{i,j \in V} (\alpha_{ij}(Wx_{ij} + \omega_{ij}) + \beta_{ij}\bar{v}_{ij}^2 x_{ij}) \quad (\text{M.3})$$

$$\text{s.t.} \quad \sum_{j=1}^n x_{ij} = 1, \quad \forall i \in C \quad (4.5)$$

$$\sum_{j=1}^n x_{ij} \leq 1, \quad \forall i \in S \quad (4.6)$$

$$\sum_{j=1}^n x_{ji} - \sum_{j=1}^n x_{ij} = 0, \quad \forall i \in \mathcal{V} \quad (4.7)$$

$$\sum_{j=1}^n \omega_{ij} - \sum_{j=1}^n \omega_{ji} = q_i, \quad \forall i \in \mathcal{V} \setminus O \quad (4.8)$$

$$q_j x_{ij} \leq \omega_{ij} \leq (Q - q_i) x_{ij}, \quad \forall (i, j) \in E \quad (4.9)$$

$$b_j \leq b_i - e_{ij} + B(1 - x_{ij}), \quad \forall i \in C, j \in \mathcal{V} \quad (4.10)$$

$$b_j \leq B - e_{ij}, \quad \forall i \in S, j \in \mathcal{V} \quad (4.11)$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in E \quad (4.12)$$

$$\omega_{ij} \geq 0, \quad \forall (i, j) \in E \quad (4.13)$$

Comparing with the situations that the previous M.2 models faced in the grid world and real map, introducing the charging stations implies that not all nodes are necessary to be visited in this case. The vehicle will go to the charging station if and only if the remaining charge level is not enough for the rest of tasks. The added constraints (4.6) limits that each recharging station will be visited no more than once. The principle behind constraint (4.7) is similar to (3.6) and (3.7), which maintain the incoming and outgoing relationship of each node. But it uses another way to expression because the vehicle might not go to the charging station during the dispatching process. Constraint (4.10) guarantees that the diminishing battery level between two customers will not exceed the current remaining charge level or the maximum capacity of the battery. (4.11) has similar meaning to (4.10), which indicates the remaining battery level after visiting the recharging station. Constraints (4.12) and (4.13) restrict the acceptable region of the decision variables.

### 4.2.2 Model adaption for dynamic situation

The energy estimation model is also incorporated with the model (M.3) to make it capable of solving the proposed D-EVRP, where the energy cost of links during working days are changeable. The pseudo code of the (M.3) based dynamic routing model the proposed D-EVRP with the energy prediction model and charging problem (CRDM.3) is written as the following:

1	<b>M.3 based dynamic routing model in D-EVRP with the energy prediction model and charging problem (Real map): CRDM.3</b>
2	
3	<b>Input parameters:</b>
4	Depot Position: $O$ ,

```

5   Charging station positions:  $S$ ,
6   Coordinates set of customers:  $C$ ,
7   Customer Demands:  $q$ ,
8   Maximum Payload:  $Q$ ,
9   Maximum battery capacity:  $B$ ,
10  The vector of real-time remaining battery level
11  at each node:  $\mathbf{b}_{real}$ ,
12  Mass related coefficient matrix:  $\alpha$ ,
13  Velocity related coefficient matrix:  $\beta$ ,
14  Initial customer graph:
15   $\mathcal{R}^0 = StageOneFunction([O, C, S], \alpha, \beta)$ .
16  Model parameters:
17  Discrete time index:  $t$ ,
18  The counter used to record the number of
19  times of reaching customers:  $count$ ,
20  Total energy cost:  $E$ ,
21  The vector of remaining battery level
22  at each node:  $\mathbf{b}$ ,
23  Optimal visit order:  $Ord$ ,
24  Optimal detailed paths:  $P$ .
25
26  Initialize the time index  $t$  with 0,
27  Initialize the total energy cost  $E$  with 0,
28  Initialize  $\mathbf{b}_{real}$  with empty set,
29  Initialize the optimal visit order  $Ord$  with empty set,
30  Initialize the optimal detailed paths  $P$  with empty set
31
32  While  $count$  is less than the number of customers in  $C$  do:
33
34      Predictive order,  $\mathbf{x}$ ,  $\mathbf{w}$ ,  $\mathbf{b} \leftarrow M.3(\mathcal{R}^t, [O, V]);$ 
35
36      Take out the first element  $j$  in predictive order:
37       $Ord = Ord \cup j$ ;
38
39      Fixing the value of decision variable  $x_{ij}$  to 1
40      if the vehicle drove from  $i$  to  $j$ ;
41
42      Finding out the best path between node  $i$  and  $j$  from
43      the current customer graph  $\mathcal{R}^t$ :
44       $P(i, j) = FindDetailedPaths((i, j), \mathcal{R}^t)$ ;
45       $P = P \cup P(i, j)$ ;
46
47      The value of discrete time index evolves:  $t = t + 1$ ;
48      if  $j \in C$ , then:
49      The value of counter evolves:  $count = count + 1$ ;
50      end if

```

---

```

51
52     Adding the corresponding trip cost during
53     every iteration :
54      $E = E + EnergyConsumption(i, j);$ 
55
56     Record the corresponding remaining battery
57     level  $b_j$ ,
58      $\mathbf{b}_{real} = \mathbf{b}_{real} \cup b_j;$ 
59
60     if any incident is detected at time  $t$ , then: the cost
61     coefficients will change and the customer graph needs
62     to be re-calculated before the next time
63     re-optimization :
64      $\alpha = \alpha_{incident};$ 
65      $\beta = \beta_{incident};$ 
66      $\mathcal{R}^{t+1} = StageOneFunction([O V], \alpha, \beta);$ 
67     else:
68      $\mathcal{R}^{t+1} = \mathcal{R}^t;$ 
69     end if
70
71 end While
72 Return  $E, Ord, P, \mathbf{b}_{real};$ 

```

Here, a new vector  $\mathbf{b}_{real}$  is introduced to store the value of the remaining charge at currently reached node. The value of later nodes' remaining charge in the vector that solved at this time is not reliable because the result will be different if the vehicle suddenly needs to detour to avoid incidents and reach the next destination. For those reached nodes, their remaining battery level variables will be calculated automatically in the later routing calculation if the corresponding decision variables  $x_{ij}$  of this node is set to 1.

### 4.2.3 Experiments

In this section, the numerical results of 10 dispatching cases based on different customer sets are presented. In each case, there will be 10 random customers and 2 fixed charging stations chosen from the real map. The incidental urban congestion is simulated by increasing the value of cost coefficients manually. Firstly, the optimal visiting order is calculated by using the static M.3 model, where the specific traveled paths and the total energy cost can also be obtained. Suppose that traffic congestion always happens randomly on several links of the path between the 3<sup>rd</sup> and the 4<sup>th</sup> customer, namely C and D. Here, based on the static routing plan, the congestion effect on those specific links that the vehicle will go through in the coming future is generated by multiplying an additional cost factor  $\gamma_\beta$  with the velocity related cost coefficient  $\beta$ . The maximum battery capacity is set to  $B = 26KWh$ , and the lowest acceptable battery level to 0. Meanwhile, for all 10 test cases, four types of energy cost under different situations are introduced here to clarify their physical meaning:

- (SO) It refers to the energy consumption of following the optimal visiting

order solution, which is calculated by using M.3 based on the static network customer graph at the beginning before the vehicle starts the day's work. In reality, this static solution offers the initial reference of routing for vehicle before in touch with the real-time information.

- (SC) It refers to the energy consumption of following the optimal visiting order solution, which is the same plan as in SO, regardless of whether any congestion exists on the a priori routing plan. This situation could happen in reality while the vehicle is driving en-route, and the current routing plan cannot be updated because the device on the vehicle lacks the support from the traffic information center. However, this routing plan is actually unfeasible because the total battery capacity is always not enough to do so.
- (DPC) It refers to the energy consumption of following the dynamic prediction visiting order solved by CRDM.3. The routing calculation takes into account the real-time traffic information, hence the customer graph is calculated based on the dynamic factors and it will evolve over the time. In this condition, the driver will need to update the current distribution order periodically according to the real-time routing solution given by the navigation device. In reality, such operation is very meaningful because the vehicle could always follow the energy-optimal routes and avoid emergency traffic incidents. In this case, Two assumptions are made: (1) only the traffic congestion is assumed to be considered as the dynamic factor, and (2) the congested links are known to the vehicle before it enters that area.
- (DPCS) It refers to the energy consumption of following the dynamic prediction visiting order solved by CRDM.3, but it assumes that the vehicle has already suffered from the congestion. This is a very common situation in reality when a severe traffic congestion happens: the routing system fail to receive the information of the congestion in time, which results in the vehicle has already been driven into the congested area without knowing the nearby accident. Then the driver will find itself getting stuck in the congestion and there are no other paths to get rid of it so the vehicle has to go through the congested roads directly.

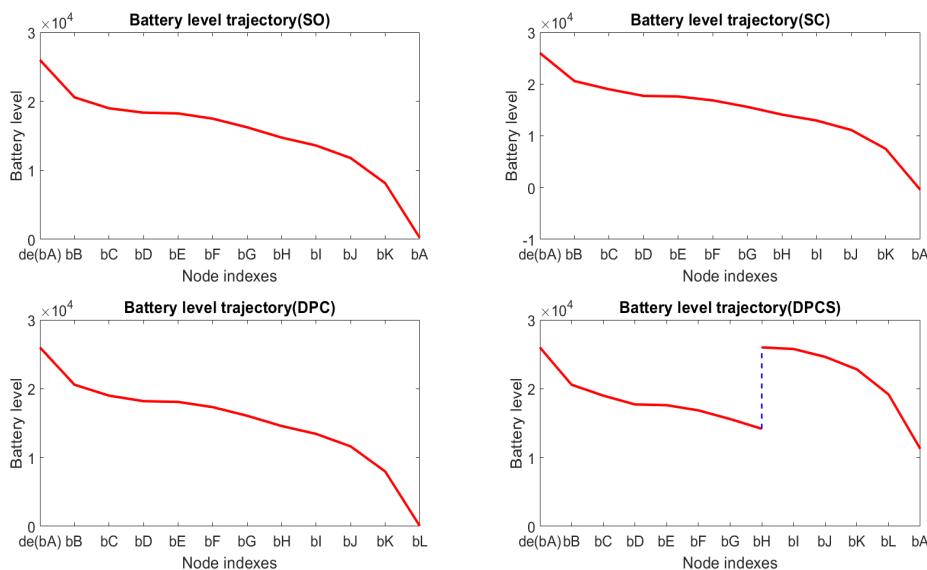
We assume the vehicle gets fully recharged after visiting charging stations and the congestion coefficient  $\gamma_\beta = 20$ . The results are shown in the table 4.1.

Case 1 and case 2 in the table indicate the same situation: the vehicle has adequate battery charge to serve all customers and back to depot even it will suffer the congestion. Hence SC and DPCS have the equal energy cost because they share the same customer graphs before the congested area and both of them spend extra energy on congestion roads. Note that DPC is 413.8Wh and 700.9Wh higher than SO in case 1 and case 2 respectively. This is because the model solves for another routing plan to avoid driving into the congested area. Such detour is named as the secondary routing plan, and such plan causes additional energy consumption due to the longer distance or time for driving. However, the numerical value reveals that the additional energy cost is still less than drive through the congested area directly. Cases 3,4,5 and 8 also indicate that the model successfully solves for a secondary routing plan to avoid driving into the congested area if such real-time traffic information could be known ahead of time. However, the DPCS columns of these cases

Cases	SO	SC	DPC	DPCS
1	23258.5	23941.5	23670.3	23941.5
2	23845.1	24608.1	24546.0	24608.1
3	25783.4	26427.1	25951.4	26581.9
4	25065.3	26529.3	25081.4	26832.2
5	25387.8	26129.1	25752.2	26864.5
6	25413.6	27426.2	26755.2	28161.7
7	28135.9	29011.4	28886.1	29011.4
8	25670.3	26349.8	25763.6	26637.8
9	27674.2	28531.0	27727.2	28531.0
10	25708.2	26364.5	26687.8	27099.9

**Table 4.1:** The energy consumption ( $Wh$ ) results from four routing strategies in 10 different customer cases.

show that the vehicle does not have sufficient energy storage to complete the rest distribution tasks without recharging at charging station. This fact can be deduced from the SC columns, where the total energy cost will be more than  $26KWh$  if passing through the congested roads is inevitable. Taking the case 3 as an example to illustrate the difference between four situations graphically. Figure 4.7 shows the remaining battery level trajectory during the dispatching case 3, where the recharging process can be clearly seen in the DPCS subplot. In figure 4.8, the routing plan of the vehicle in DPCS ( $c$ ) is same as in figure 4.8 SO ( $a$ ) except it goes to the charging station after visiting the sixth customer G, while the routing solution from our dynamic prediction model in figure 4.8 DPC ( $b$ ) chooses other detailed links between C and D to sidestep the congestion.



**Figure 4.7:** The remaining battery level trajectory of case 3.

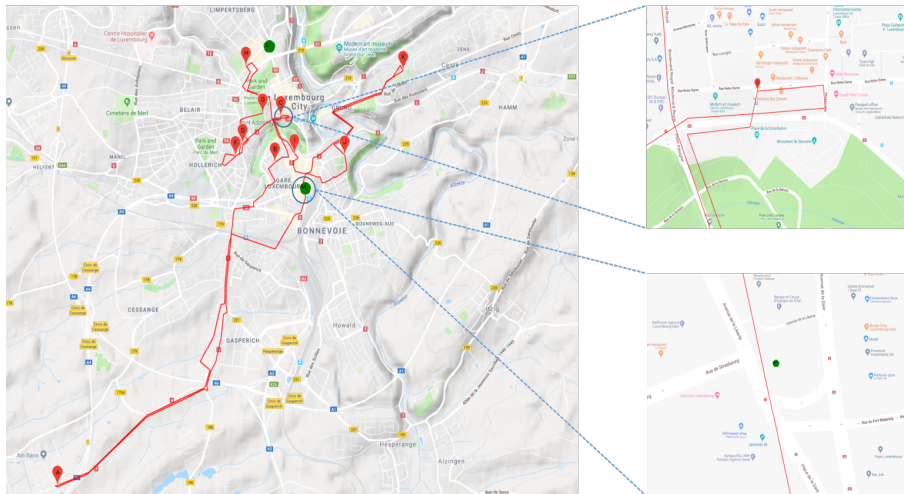
On the other hand, one can notice that the energy cost in DPCS of these customer cases is even larger than its values in SC. The reason is that in DPCS, the model changes the visiting order and force the vehicle to go the charging stations at some point after driving out from the congested roads. Hence additional energy cost is incurred on the way ahead to the charge stations.

Both DPC and DPCS in case 6 and 10 take the charging process into account, because the secondary routing plan of DPC solved by our dynamic model includes a far detour when congestion happens. Although such detour to charging station expends less energy than directly drives into the congested area, additional energy cost incurs when the vehicle drives on the way to charging stations. This fact explained why the value of DPC is larger than SC once the vehicle needs recharging in case 6 and 10.

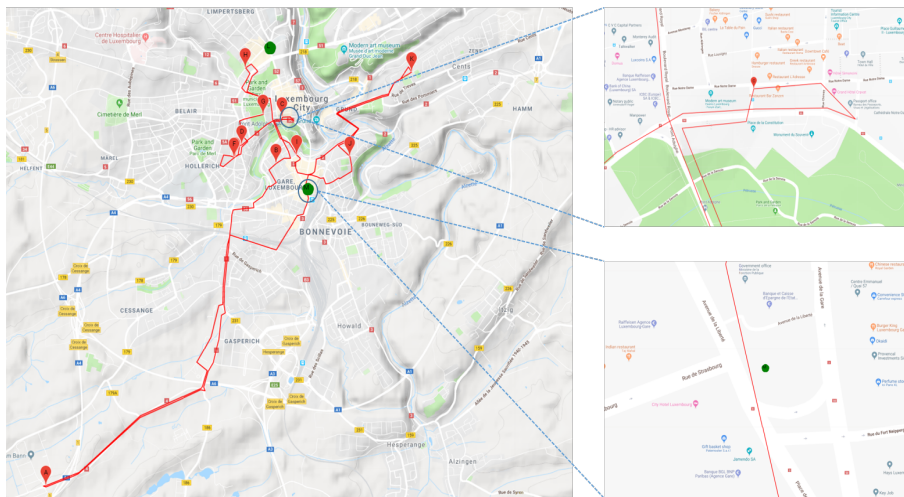
Figure 4.9 presents the graphic illustration of case 6, where one can see in both DPC (b) and DPCS (c), the predetermined routing plan is re-optimized and vehicle goes to the same charging station but in different order due to the effect of congestion. Figure 4.10 presents the corresponding remaining battery level trajectories of four types of energy consumption. One could notice that the remaining volume of charge increases in some points. This fact is caused by the regeneration property of battery when vehicle is braking or going downward. One notices that the given charging station at the lower right side is very close to the road that the vehicle will pass by. However, since the remaining battery charge is enough for the rest of trip when the vehicle comes to this point, the vehicle will not move into the charging station as the sub-figures show in the enlarged regional views of figure 4.8 and figure 4.9.

In case 7 and case 9, both the static and dynamic routing need to consider the recharging problem because of the geographically dispersed location of customers. These two cases are similar to cases 1 and 2 but have higher energy consumption and additional recharging process.

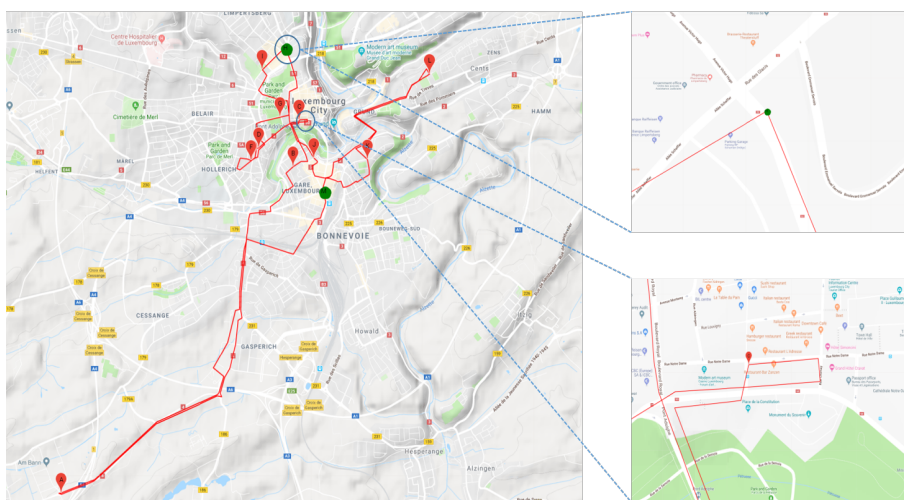
To summarize the table 4.1, the difference between SO and SC is the extra energy that consumed on the congested road in all cases, with The minimum  $0.6437KWh$  and maximum  $2.012KWh$ , which depend on the length of the congested road as well as the corresponding average driving speed on that road. The values of DPC are always larger than SO because of the necessary detour of distribution or recharging process. It is not very meaningful to compare the numerical value between DPC and DPCS because they follow different rules in dynamic circumstance. In DPCS, the vehicle is deliberately placed in a congested area, so the point that is worth to observe behind DPCS is how will the model make compensation for it if the remaining battery level is not enough to reach the rest of customers.



(a) SO of case 3



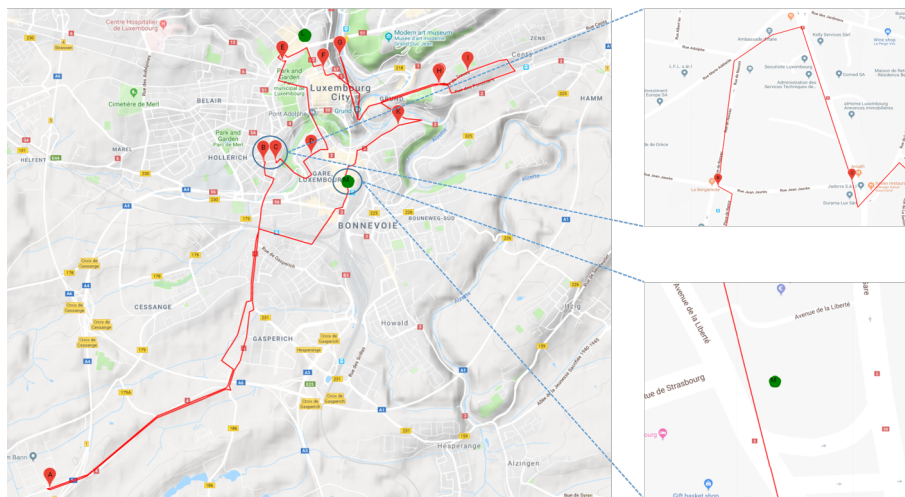
(b) DPC of case 3



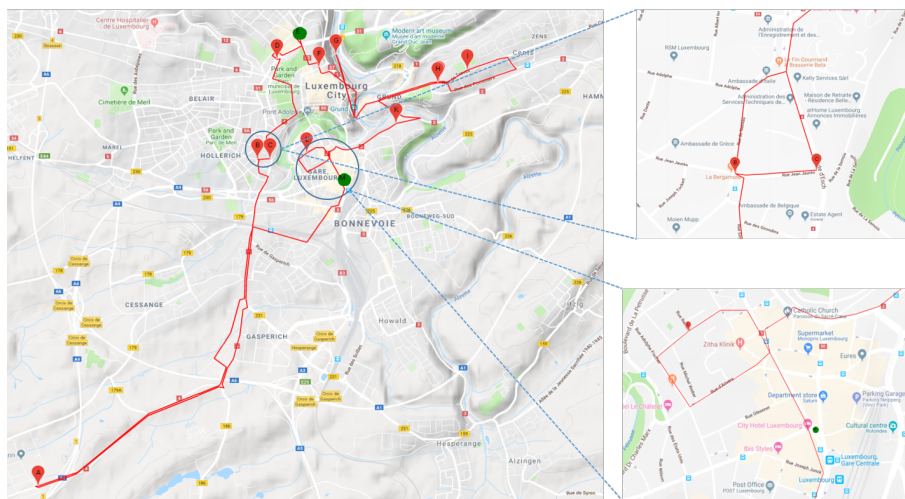
(c) DPSC of case 3

**Figure 4.8:** The routing solution of SO, DPC and DPSC respectively of case 3.

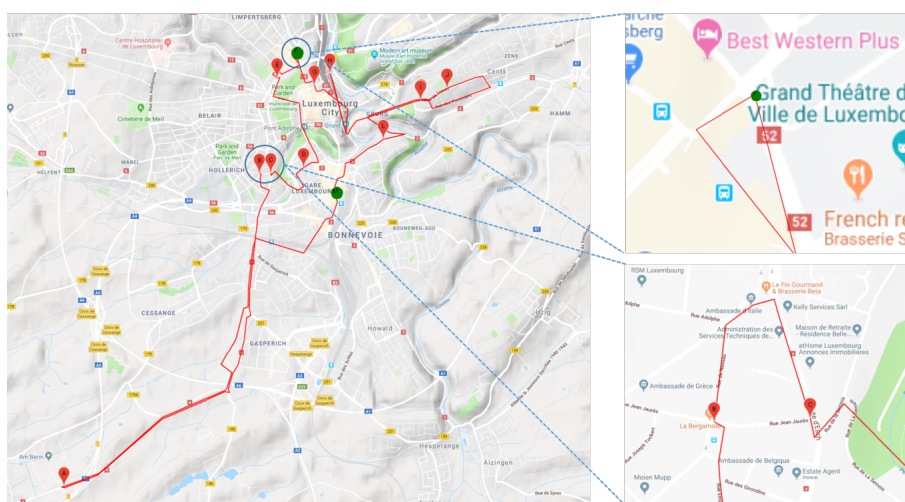
#### 4. Real Map



(a) SO of case 6



(b) DPC of case 6



(c) DPSC of case 6

**Figure 4.9:** The routing solution of SO, DPC and DPSC in case 6.

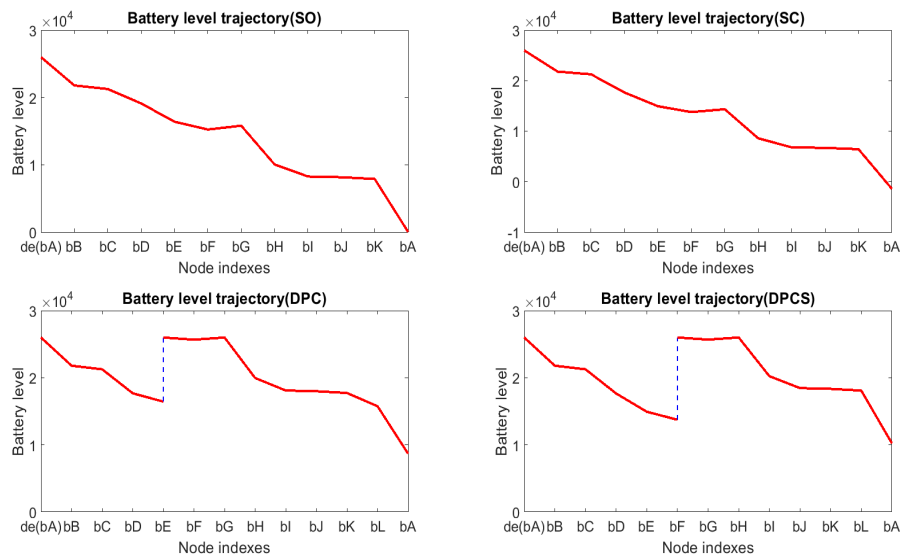


Figure 4.10: The remaining battery level trajectory of case 6.



# 5

## Conclusion

### 5.1 Work summary

In this thesis work, a series of VRP from the most traditional one for internal combustion vehicles to EVRP with capacity limitation and charging stations for electric vehicles have been studied. The adaptation works for transferring the static model into the dynamic one are carried out based on exact solution models such as M.1 and M.2. To solve the proposed routing problem dynamically, incorporating the static model with the look ahead theory is necessary. An energy estimation model is used to approximate the energy cost of links, which builds the foundation for the algorithm to find the energy-optimal path between nodes. The real-world traffic data during workdays are extracted from SUMO, which provides the data reference for the modified real-time algorithm to validate. The solution obtained, which includes the total energy cost as well as the route plot, is compared with the benchmark solution that is calculated from the static cases.

The main results of this thesis are:

- 1. In this report, a series of VRP and its variations have been studied, which provides the primary direction for solving the proposed real-time electric routing problem. Various methods from exact solution methods to metaheuristic methods also have been studied from the literature resources. General concepts of those methods are summarized and their advantages and disadvantages are presented and analyzed in the literature review chapter.
- 2. A combinatorial routing model is designed for the proposed real-time electric vehicle routing problem. The optimality of the static solution in each time period is guaranteed by the linear programming formulations and the timeliness of the dynamic solution is guaranteed by implementing the look ahead policy and the shrink horizon window principle.
- 3. The designed dynamic model could solve for a minimal energy cost routing plan within short time interval. Instead of consuming a large amount of energy on congested road, the model can effectively reduce the energy cost to some extent by re-routing when incidents happen. By introducing the battery limitation and charging stations, the model is able to have our vehicle recharged at charge stations whenever the remaining battery level is low.

To sum up in general, real-time routing technology is indispensable for EVs to

manage the energy consumption of the battery reasonably. A dynamic routing model which has more practicability in reality needs to take as many dynamic factors as possible into account in order to completely reproduce the real-time traffic scenarios and provide the optimal solution.

### 5.2 Future Works

Real-time electric vehicle routing is a considerable field that needs to integrate interdisciplinary subjects. To be able to design a capable dynamic model, a deeper understanding of the real time scenario in a mathematical way is required since it is used to transform the abstract circumstances messages to a certain mathematical model/formulation. The future works of this thesis report are listed as the following:

- This report presents an additional study based on Rafael Basso's research [10] that considers only static cases and situations, some uncertainties such as stochastic customers, accidents en-route and time window are not included, which exist commonly in reality. One could develop a more comprehensive routing model by considering those stochastic factors in the future model improvement.
- It could be interesting to implement metaheuristic methods such as GA and ACO on this optimization problem. Many optimal routing problems in reality could have a nonlinear objective function and complex constraints, where the linear models are no longer applicable and the global optimal solution might be impossible to be solved. In these cases, metaheuristic methods can find feasible solutions which are good enough in the practical application without having a heavy computational effort such as in exact optimization algorithms or iterative methods.

# Bibliography

- [1] Richard Eglese, Tolga Bakta. (2014) "Green Vehicle routing", in *Vehicle Routing Problems, Methods, and Applications Second Edition*. Society for Industrial and Applied Mathematics.
- [2] Jonathan D. Adler and Pitu B. Mirchandani. (2014) "Online routing and battery reservations for electric vehicles with swappable batteries", in *Transportation Research Part B: Methodological*, Vol. 70, pp. 285-302.
- [3] Gianpaolo Ghiani, Francesca Guerriero, Gilbert Laporte and Roberto Musmanno. (2003) "Real-time vehicle routing: Solution concepts, algorithms and parallel computing strategies", in *European Journal of Operational Research*, Vol. 151, pp. 1-11.
- [4] Michalis Mavrovouniotis and Shengxiang Yang. (2015) "Algorithm with immigrant schemes for the dynamic vehicle routing problem", in *Evolutionary Computation for Dynamic Optimization Problems*, pp. 317-341.
- [5] Paolo Toth and Daniele Vigo. (2002) "Models, relaxations and exact approaches for the capacitated vehicle routing problem", in *Discrete Applied Mathematics*, Vol. 123, pp. 487-512.
- [6] Chrysanthos E. Gounaris, Wolfram Wiesemann and Christodoulos A. Floudas. (2013) "The Robust Capacitated Vehicle Routing Problem", in *Operations Research*, Vol. 61, pp. iii-790.
- [7] David Mester and Olli Bräysy and Christodoulos A. Floudas. (2005) "Active-guided evolution strategies for large-scale capacitated vehicle routing problems", in *Computers & Operations Research*, Vol. 34, pp. 2964-2975.
- [8] Canhong Lin, K.L.Choy, G.T.S.Ho, S.H.Chung and H.Y.Lam. (2014) "Survey of Green Vehicle Routing Problem: Past and future trends", in *Expert Systems with Applications*, Vol. 41, pp. 1118-1138.
- [9] Jun Yang and Hao Sun. (2015) "Battery swap station location-routing problem with capacitated electric vehicles", in *Computers & Operations Research*, Vol. 55, pp. 217-232.

- [10] Rafael Basso, Balazs Kulcsar, Bo Egardt, Peter Lindroth, Ivan Sanchez-Diaz. (2018) "Energy consumption estimation integrated into the Electric Vehicle Routing Problem", in *Transportation Research Part D: Transport and Environment*, Vol. 69, pp. 141-167.
- [11] Rafael Basso. (2017) "Route planning and energy consumption estimation for electric commercial vehicles", Licentiate thesis, Chalmers University of Technology.
- [12] Larsen, Allan, Madsen and Oli B.G. (2000) "The dynamic vehicle routing problem", PhD thesis, Technical University of Denmark (DTU).
- [13] Victor Pillac, Michel Gendreau, Christelle Gueret, Andres L. Medaglia. (2013) "A review of dynamic vehicle routing problems", in *European Journal of Operational Research* Vol. 225, pp. 1-11.
- [14] Ulrike Ritzinger, Jakob Puchinger and Richard F.Hartl. (2016) "A survey on dynamic and stochastic vehicle routing problems", in *International Journal of Production Research*, Vol. 54, pp. 215-231.
- [15] Eiichi Taniguchi, Hiroshi Shimamoto. (2004) "Intelligent transportation system based dynamic vehicle routing and scheduling with variable travel times", in *Transportation Research Part C: Emerging Technologies*, Vol. 12, pp. 235-250.
- [16] Tolga Bektas, Panagiotis P.Repoussis and Christos D.Tarantilis. (2014) "Dynamic Vehicle Routing Problems" in *Vehicle Routing Problems, Methods, and Applications Second Edition*, Society for Industrial and Applied Mathematics, pp. 299-347
- [17] Michel Gendreau, Ola Jabali and Walter Rei. (2014) "Stochastic vehicle routing problems", in *Vehicle Routing Problems, Methods, and Applications Second Edition*, Society for Industrial and Applied Mathematics, pp. 213-239.
- [18] Ali Haghania and Soojung Jung. (2005) "A dynamic vehicle routing problem with time-dependent travel times", in *Computers & Operations Research*, Vol. 32, pp. 2959-2986.
- [19] Ali R.Guner, Alper Murat and Ratna Babu Chinnam. (2012) "Dynamic routing under recurrent and non-recurrent congestion using real-time ITS information", in *Computers & Operations Research*, Vol. 39, pp. 358-373.
- [20] Huey-Kuo Chen, Che-Fu Hsueh and Mei-Shiang Chang. (2006) "Real-time time-dependent vehicle routing problem", in *Transportation Research Part E: Logistics and Transportation Review*, Vol. 42, pp. 383-408.

- [21] Lianxi Hong. (2011) "An improved LNS algorithm for real-time vehicle routing problem with time windows", in *Computers & Operations Research* Vol. 39, pp. 151-163.
- [22] Raluca Necula, Mihaela Breaban and Madalina Raschip. (2017) "Tackling Dynamic Vehicle Routing Problem with Time Windows by means of Ant Colony System", in *IEEE Congress on Evolutionary Computation (CEC)*.
- [23] Sepideh Pourazarm, Christos G.Cassandras, Andreas Malikopoulos. (2014) "Optimal Routing of Electric Vehicles in Networks with Charging Nodes: A Dynamic Programming Approach", in *2014 IEEE International Electric Vehicle Conference (IEVC)*.
- [24] Michael Schneider, Andreas Stenger and Dominik Goeke. (2014) "The Electric Vehicle-Routing Problem with Time Windows and Recharging Stations", in *Transportation Science*, Vol. 48, no. 2, pp. 465-694.
- [25] Ryan G.Conrad, Miguel Andres Figliozzi. (2011) "The Recharging Vehicle Routing Problem", in *2011 Industrial Engineering Research Conference*.
- [26] Maximilian Schiffer and Grit Walther. (2017) "The electric location routing problem with time windows and partial recharging", in *European Journal of Operational Research*, Vol. 260, pp. 995-1013.
- [27] Kirtiwant P Ghadle, Yogesh M Muley. (2015) "Travelling salesman problem with MATLAB programming", in *International Journal of Advances in Applied Mathematics and Mechanics*, Vol. 2, no. 3, pp. 258-266.
- [28] Qiumei Liu and Yanmei Yang. (2018) "Interactive programming approach for solving multi-level multi-objective linear programming problem", in *Journal of Intelligent & Fuzzy Systems*, vol. 35, no. 1, pp. 55-61.
- [29] Taipei (2011) "Stochastic Routing" in *Routing Area Meeting IETF 82*.
- [30] Nabila Azi, Michel Gendreau and Jean-Yves Potvin. (2007) "An exact algorithm for a single-vehicle routing problem with time windows and multiple routes", in *European Journal of Operational Research*, Vol. 178, pp. 755-766.
- [31] Mostepha Khouadjia and Enrique Alba. (2013) "Metaheuristics for Dynamic Vehicle Routing", in *Metaheuristics for Dynamic Optimization*, pp. 265-289.
- [32] Anirudh Shekhawat, Pratik Poddar and Dinesh Boswal. (2009) "Ant colony Optimization Algorithms: Introduction and Beyond", in *Artificial Intelligence Seminar 2009*, Indian Institute of Technology Bombay.

- [33] Haitao Xu, Pan Pu, and Feng Duan. (2017) "Dynamic Vehicle Routing Problems with Enhanced Ant Colony Optimization", in *Discrete Dynamics in Nature and Society*, Vol. 2018, pp. 1-13.
- [34] Högskolan Dalarna. (2008) Solving the Vehicle Routing Problem with Genetic Algorithm and Simulated Annealing, Master thesis, Dalarna University, School of Technology and Business Studies, Computer Engineering.
- [35] Messaoud Elhassania, Boukachour Jaouad and Elhilali Alaoui Ahmed. (2015) "Solving the dynamic vehicle routing problem using genetic algorithms", in *2014 International Conference on Logistics Operations Management*.
- [36] Franklin T. Hanshar and Beatrice M. Ombuki-Berman. (2007) "Dynamic vehicle routing using genetic algorithms", in *Applied Intelligence*, Vol. 27, no. 1, pp. 89-99.
- [37] Pisinger David and Røpke Stefan. (2010) "Large Neighborhood Search", in *Handbook of Metaheuristics*, pp. 399-419.
- [38] Roman Lutz. (2014) "Adaptive Large Neighborhood Search", Bachelor thesis, Ulm University, Institute of Theoretical Computer Science.
- [39] Nabila Azi, Michel Gendreau, Jean-Yves Potvin. (2011) "A dynamic vehicle routing problem with multiple delivery routes", in *Annals of Operations Research*, Vol. 199, no. 1, pp. 103-112.
- [40] Warren B. Powell, Hugo P. Simao, Belgacem Bouzaiene-Ayari. (2012) "Approximate dynamic programming in transportation and logistics: a unified framework", *EURO Journal on Transportation and Logistics*, Vol. 1, no. 3, pp. 237-284.
- [41] Ulrich Pferschy and Rostislav Stanek. (2016) "Generating subtour elimination constraints for the TSP from pure integer solutions", in *Central European Journal of Operations Research*, Vol. 25, no. 1, pp. 237-260.
- [42] Lara Codeca, Raphael Frank, Thomas Engel. (2015) "Luxembourg SUMO Traffic (LuST) Scenario: 24 hours of mobility for vehicular networking research", in *2015 IEEE Vehicular Networking Conference (VNC)*.
- [43] Google Maps. (2019). "Luxembourg City". Retrieved from <https://goo.gl/maps/jcnUy1ZRqw9B6HM38>