Optimizing Electric Vehicle Routing and Charging/Discharging under Time-Variant Electricity Prices

by

Bo Lin

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Management Sciences

Waterloo, Ontario, Canada, 2020

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The integration of electric vehicles (EVs) and the power system has been becoming an increasingly important field of research, due to the rapid EV penetration and the evolvement in vehicle-to-grid (V2G) techniques in the past decade. Under appropriate management of EV charging and discharging, the current grid capacity can satisfy the energy requirements of a considerable number of EVs and EVs could help enhance grid reliability and stability through ancillary service provision.

In this thesis, we investigate the operational strategies of commercial EV fleets under the V2G context where energy price signals are utilized to incentivize EV owners to time-shift charging schedule and discharging EVs during peak hours. We propose and formulate a new EV routing problem with time windows under time-variant electricity prices (EVRPTW-TP), considering practical constraints of commercial EV fleets providing logistic services and optimizing over its overall electricity cost. In order to solve the EVRPTW-TP, we then formulate a Lagrangian relaxation as well as a variable neighborhood search and tabu search hybrid (VNS/TS) heuristic to approximate the optimal solution from below and above respectively. Our numerical experiments on small instances suggest that both algorithms are able to provide high quality bounds to the EVRPTW-TP. The VNS/TS heuristic outperforms CPLEX in terms of solution quality and efficiency on instances of 10 or more customers. In addition, we utilize the VNS/TS heuristic to study a use case of an EV fleet providing grocery delivery service in the Kitchener and Waterloo (KW) region in Ontario, Canada. Insights about the impacts of energy pricing scheme, service time slot design as well as fleet size are presented. Finally, as the first step towards implementing advanced machine learning techniques to solve the EVRPTW-TP, we develop a reinforcement learning (RL) model for the electric vehicle routing problem with time windows (EVRPTW) and provide computational results to assess the performance of the RL model.

Acknowledgements

First and foremost, I would like to give my sincerest thanks to my supervisor Professor Bissan Ghaddar. She has been supervising me since the third year of my undergraduate studies, and has been an extremely patient, thoughtful, dedicated, rigorous and insightful advisor. She has not only offered me great guidance on my thesis research, but also provided me with various other research opportunities. Without her support, this journey would not have been possible. I owe a great debt of gratitude to my supervisor Professor Jatin Nathwani for generously suggesting wonderful research ideas as well as funding support. I really appreciate my thesis readers Professor Houra Mahmoudzadeh and Professor Fatma Gzara for the valuable feedback and suggestions. I would also like to thank Dr. Ada Hurst with whom I worked on a very interesting project that indeed broadened my research horizon.

I am very grateful to my course instructors at the University of Waterloo Professor Houra Mahmoudzadeh and Professor Qiming He for delivering insightful courses that provided me with academic inspirations and motivated me to pursue further research in the field of operations research. I am extremely thankful to the good friends and colleagues at the University of Waterloo: Weilin Zhang, Affleck Chang, Haoran Wu, Esma Akgun, Zeynep Bulbul, and Nafise Niazi Shahraki. Especial thanks go to Weilin Zhang for being my amazing companion through the journey. The financial support of the Energy Council of Canada Energy Policy Research Fellowship was also highly appreciated.

Last but not least, I would like to express my sincere gratitude to my parents Tianyi Lin and Xiumei Hong for their unconditional support and love. They have offered me every opportunity that they never had and have always been great listeners, collaborators and my best friends.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

These days, sustainability is becoming an increasingly important factor to consider in industrial production. In the transportation sector, endeavors have been taken by regulators around the world to facilitate vehicle electrification for its ability to mitigate greenhouse gas (GHG) emission, promote sustainable ways of electricity generation, and reduce particulate matter pollution thus to benefit human health [14, 107, 124]. For instance, in 2019, Transport Canada announced an investment of 300 million Canadian dollars to a new federal zero-emission vehicle (ZEV) purchase incentive program [21]. At the company level, driven by the potential brand benefits of employing new green technologies, the growing demands for green products [69, 31], and the plummet of electric battery cost, market players, such as UPS, FedEx and Walmart, have deployed their own electric vehicle fleets to provide services [125]. As a consequence, the past decade has seen rapid expansion of electric vehicle adoption [54]. According to Statistics Canada [20], the number of new motor vehicle registration for battery electric vehicle (BEV), hybrid electric vehicle (HEV), and plug-in hybrid electric vehicle (PHEV) has increased from 25,163 in 2014 to 69,010 in 2018. As projected by Canada Energy Board [11], electric vehicles will account for over 60% of the new motor vehicle registration in Canada by 2040 assuming a global shift to low-carbon economy.

The penetration of electric vehicles (EV) has significant impacts on the power system. Previous studies have shown that, without proper management, the EVs could represent a sizable fraction of total demand [119, 34], increase the demand differences between off-peak and peak hours, and thus increase ramp requirements [120], which will potentially affect the power networks' stability and reliability. However, with controlled charging, the existing power system infrastructure can actually satisfy the energy requirements of a considerable number of EVs [95, 68, 78]. In doing so, the need for installing new capacity, which is

expensive, time-consuming and harmful to the environment [34], can be eliminated [120]. The authors of [60] take a further step as they develop the concept of vehicle-to-grid (V2G) proposing that electric vehicle can be potentially incorporated in to the power system as a reliable and cost-effective distributed power source and storage. Following studies, for example, [61] [62] and [111], reveal that, through time-shifting charging schedule and discharging to the grid during peak hours, EV fleet connected to the electric grid can assist to level out peaks in overall electricity consumption, thus alleviate the grid pressure. In addition, V2G can also support the utilization of intermittent renewable energy and generate profits for the grid and EV owners if certain business models are applied.

Although V2G represents an enticing idea with purported benefits, it nonetheless remains in the pilot stages of development [109]. Previous studies, see for example [49] and [106], mainly focus on the centralized architecture where a central controller is introduced to manage the ancillary service provision of a large group of EVs. They have shown that the aggregation is necessary because the battery capacity for a single EV is so small that its impact on the grid is negligible and communicating with each EV separately is expensive and of low efficiency. However, in practice, centralized control may reduce the profits earned by EV owners, requires a considerable amount of investment for constructing new communication infrastructures [94] and is lack of flexibility to incorporate EVs' other business operations such as routing and scheduling. In this scenario, especially in the infant stages of V2G deployment where there is a dearth of communication infrastructure, decentralized approaches that utilize price signals to incentivize EV owners to time-shift charging/discharging become a more realistic option [55, 102, 19]. In particular, cooperating with commercial EV fleet owned by logistics or e-commerce companies which naturally aggregate a large number of EVs might form an ideal first step towards V2G implementation. As key components in commercial EV fleet operation, efficient routing and charging/discharging scheduling techniques in V2G contexts are necessary for the successful integration of commercial EV fleets and the power system. However, only a limited number of studies, such as the ones presented in [2, 56, 128, 119, 114], have been conducted in this field. To the best of our knowledge, no previous study is able to jointly optimize the routing and charging/discharging for an EV fleet as a whole in bidirectional V2G context.

In this research, we consider a delivery service system that operates a fleet of EVs to service customers. We propose the electric vehicle routing problem with time window constraints under time-variant electricity prices (EVRPTW-TP), which 1) considers the practical constraints of commercial EV fleet and logistic services; 2) enables EVs to charge and discharge their batteries at time-of-use prices through the planning horizon and 3) directly optimizes over the monetary cost of operations. As an extension of the electric vechile routing problem with time windows (EVRPTW) proposed by [103] which is NP-

hard, the complexity of the proposed problem is even higher as an additional operation, discharge, is enabled, partial recharge/discharge is allowed, and time-variant electricity prices are taken into consideration. In order to efficiently solve the EVRPTW-TP, we develop a Tabu Search and Variable Neighborhood Search hybrid heuristic (TS/VNS) that can generate quality feasible solutions efficiently. Moreover, we formulate a Lagrangian Relaxation and a sub-gradient heuristic to obtain lower bounds to the EVRPTW-TP. The upper and lower bounds are evaluated based on the widely-used instances developed by [103] and are compared with the CPLEX over small instances. In addition, inspired by the recent applications of deep reinforcement learning in solving combinatorial problems, see for instance [8, 65, 72, 85], we solve the EVRTW employing a reinforcement learning framework which can be extended for EVRPTW-TP in the future. Finally, we apply the proposed model to a use case of an EV fleet performing grocery delivery in the Kitchener-Waterloo (KW) region in Ontario, Canada. Managerial insights are drawn from the case study about the reactions of EV fleet to the changes in electricity price, time slots design as well as fleet size.

The contributions of this research are three-fold. First, to the best of our knowledge, this is the first study investigating the joint optimization of routing and charge/discharge scheduling for multiple EVs under time-variant electricity price. The proposed model is able to provide operational decision making support to commercial EV fleet operators in order to lower overall charging costs, or even make profits through ancillary service provision in near-term bidirectional V2G context. Moreover, the proposed model offers some important implications for policy makers. The numerical experiments can assist the power system regulators to better understand the energy markets, predict and estimate the market reaction to energy price adjustment. The managerial insights extracted can help the policy makers create more efficient energy pricing schemes as well as better energy policies to maximize the environmental and economic benefits to different stakeholders in the transportation and energy markets. Further, this research joins two fields of studies, i.e. operations research and reinforcement learning, and leverages to help solve complex optimization problems.

The thesis is organized as follows. In Chapter 2, we review the related work, followed by the description of the proposed problem in Chapter 3. The Lagrangian relaxation of EVRPTW-TP and the sub-gradient heuristic for solving it are elaborated in Chapter 4. We introduce the VNS/TS heuristic for EVRPTW-TP in Chapter 5. Chapter 6 evaluates the lower and upper bounds generated in previous sections agianst the state of art solvers. We present the case study in Chapter 7, and introduce the deep reinforcement learning model for the EVRPTW in Chapter 8. Finally, we conclude the thesis with Chapter 9.

Chapter 2

Literature Review

This research primarily contributes to the energy and transportation literature on bidirectional V2G, and the operations research literature on EV routing and charging/discharging scheduling. In this chapter, we first review the V2G literature focusing on describing its concept and implementation architecture in Section 2.1. Since the EV routing problems are, in general, extensions of the vehicle routing problems, most formulations and solution methods for the former problem are originated from those for the latter problem, we review literature for the vehicle routing problem in Section 2.2 before we delve into electric vehicle routing problems in Section 2.3. Finally, we review the existing research work at the interface of electric vehicle routing and V2G in Section 2.4

2.1 Vehicle to Grid

Most of the literature in energy and transportation community on V2G focused on designing conceptual V2G implementation framework and performing simulations to investigate the feasibility, reliability, economics and social/environmental implications of the proposed system. Considerable research efforts were made on exploiting EVs to assist intermittent renewable energy storage and integration [81, 7], provide ancillary services center on peak shaving [123] and frequency regulation [51, 63]. In the meanwhile, topics related to EV battery degradation were intensively investigated [108, 10, 131], while there is a limited work from the perspectives of business model application and economic analysis [16, 47, 86]. In this literature, two types of V2G contexts are considered: 1) bi-directional V2G where charging and discharging are both enabled; 2) uni-directional V2G where EVs support grid operations only through time-shifting charging schedules and are not able to inject energy

back to the grid. For a comprehensive review about the landscape of the V2G research and the future research agenda, we refer readers to [55] and [109].

The vast majority of previous related research considers the centralized approaches where a central controller is able to manage the charging and discharging of all connected EVs so as to make sizable impacts on the grid and to perform efficient communication. The central controller here could be the grid operator that aims to minimize the overall system cost [50] or total energy consumption [101]. Yet in most previous studies, researchers consider introducing an intermediary, for example a garage or charging facility operator, to coordinate the charging and discharging of a large number of EVs to meet the service commitments to the grid while also achieving targeted charge levels for the contracted EVs [9]. As a large purchaser, the intermediate aggregator may be able to purchase energy, batteries and services at low prices [61, 62]. It, thus, has the capability to provide economic incentives to attract personal EV owners to participate in the program.

Nevertheless, relying on centralized control has several drawbacks. First, previous studies did not systematically explore the business models of V2G. The estimated economic benefits to EV owners vary significantly among different models [109]. Although several papers showed that V2G could bring major financial savings to EV owners under certain assumptions [66, 88], there were many more cases, for example [16, 122], where the incentive for drivers to participate is not sufficient. The authors of [94] point out that the aggregative architecture is able to provide more reliable frequency regulation service for the grid by contracting with a larger number of EVs than required, which, to some extent, reduces the revenue collected by individual EV owners. Second, current centralized frameworks generally assume that the travel patterns and future driving profiles of individual EVs are known in advance. However, in practice, this setting may be faced with privacy issues [55]. In addition, the central controller only takes care of the charging and discharging, jointly optimizing commercial fleet operations, such as routing, and fleet charging/discharging is very difficult under such architectures. Finally, the large-scale implementation of the centralized architecture requires communication infrastructure that is not readily available, making its application even harder in the near future [94]. In fact, investing in a centralized control technique may not be required until very high penetrations of EVs becomes a reality [96].

There is a recent but limited body of work on employing de-centralized approaches to realize the V2G concept. Hu et al. in [55] classify the de-centralized control strategies into two groups, transactive control and price control. Under transactive control, the grid operator and the EV owners iteratively update their price and power schedule until equilibriums are achieved. This way, the EV charging schedule is a result of the information exchange between the grid operator and the EVs. While under price control, EV schedule is

purely the decision of EV owners given the electricity price signals. To name a few examples considering transactive control. Karfopoulos et al. [59] develop a multi-agent system and perform simulation of a realistic urban distribution network to compare the performance of uncoupled and weak-coupled transactive control strategies and a centralized approach. The weak-coupled transactive control strategy is shown to outperform its uncoupled counterpart in terms of effectiveness of "valley filling", maximization of load factor and minimization of energy losses, and have comparable performance as the centralized strategy yet with significantly lower computational cost. Richardson et al. [96] propose a local control strategy that could be implemented by EV owners without global information about the grid so as to maximize their own charging rate with respect to some voltage and loading constraints. The simulation suggests that the proposed methods could drive similar amount of energy to the EVs within a certain time period compared to a centralized approach, but requires wider safety margins for the system parameters. Rotering and Ilic [97] construct a dynamic programming algorithm to control the charging and frequency regulation service provision under price control, they show that performing smart charging and providing grid support could substantially improve EV economics. Other research, such as [129] and [104] investigate the impact of time-of-use electricity prices on the distribution network with EV penetration and the design of electricity pricing schemes based on customer behaviors.

Our research differs from and extends this literature in a few substantive ways. First, the proposed model in this research considers the charging/discharging strategy for commercial EV fleets under price control in the context of bidirectional V2G, which has not been explored by previous studies. Moreover, in addition to scheduling charging/discharging for an EV fleet, we simultaneously optimize its routing for logistic services, making the proposed model of great practical values for commercial EV fleets.

2.2 Vehicle Routing Problem

2.2.1 Problem Variants

The vehicle routing problem (VRP) was first proposed by Dantzig and Ramsor [30] in 1959 as a generalization of the well-known travelling salesman problem (TSP), and were extensively studied in the subsequent 60 years. In general, given a set of geographically scattered customers each associated with a demand, a depot and the travelling distance between them, the VRP seeks to assign customers to the vehicles in such manner that the demand at each customer is satisfied while the total distance travelled by the fleet is minimized. The route of each vehicle should originate and terminate both at the depot. In

the gasoline distribution problem proposed by the original paper [30], the cargo capacity of a single vehicle is assumed to be much smaller than the total customer demands. Such a problem is later classified as the capacitated VRP (CVRP).

The VRP model were broadly applied in supply chain management and transportation. In order to take into account realistic constraints and objectives, numerous variants of the classical VRP were developed and investigated considering customer characteristics, service quality, system stochasticity, fleet heterogenity, environmental and energy issues etc. Among these variants, the one that is the most closely related to the proposed problem is the VRP with time windows (VRPTW) [98] where an additional set of constraints about the time intervals in which individual customers should be visited is introduced. Russell and Urban [100] further propose the VRP with soft time window (VRPSTW) by allowing the time window constraints to be violated at the price of some penalties. Of interest is also a more recent group of problems, the green VRP (GVRP) which deals with the environmental issues associated with the VRP. Due to the recent development of efficient optimization and computing techniques, researchers and practitioners placed greater attentions on the rich VRP (RVRP) that reflects most of attributes of a real-life vehicle-routing distribution system [18]. For comprehensive reviews about the evolution of the VRP and its major variants, we refer readers to the published survey papers [27, 35, 15, 73, 18, 79] and books [117, 46].

2.2.2 Solution Methods

The classic VRP and its variants were shown to be NP-hard. Considerable research efforts were made to solve these problems in the past decades. In general, the solution approaches could be classified into four categories: the exact methods, the classical heuristics, the meta-heuristics and reinforcement learning based approaches. Given the extremely large number of related papers, it is difficult to review these methods comprehensively. Instead, we outline the bigger picture through introducing the main existing ideas, analyzing their strengths and limitations, and discussing how they form the pathway towards the solution methods for the proposed problem.

Problem Formulation and Exact Methods

In terms of mathematical formulation, the classical VRP is originally formulated as a node pairing problem where a set of binary decision variables x_{ij} are used to represent if node i and node j are paired to form a route [30]. This formulation is later extended by Laporte

and Nobert [77], with an additional constraint on the fleet size, as the widely-implemented two-index vehicle flow formulation. In contrast, a limited body of literature, see [40] and [45] for instance, implement the three-index formulation utilizing binary variables x_{ijk} tracking if edge (i, j) is travelled by each vehicle separately. The three-index formulation, though in general employs a greater number of decision variables thus requires longer solution time, has led to several decomposition-based solution methods such as [58] and [41]. In addition, the set-partitioning formulation [6] has also been successfully applied to solve the VRP and its variations.

As an important variants of the classical VRP, VRPTW is in general easier to solve than VRP since the time windows offer tighter constraints for the problem. Acknowledging that the solution methods for VRPTW are in many aspects inherited from the work done for the TSP, Kallehauge [57] categorizes the formulations for VRPTW into: arc formulation, arc-node formulation, spanning tree formulation and path formulation. Among them, path formulation is usually utilized in the context of column generation, see for example [33], which were shown to have great performance in solving VRPTW. Moreover, with the help of Lagrangian relaxation and constrained shortest spanning tree as well as constrained shortest paths, lots of algorithms such as the ones presented in [39] and [71] were developed to obtain lower bounds to VRPTW to support branch-and-bound methods.

we refer to [76] and [57] for systematic reviews about the VRP and VRPTW formulations. Based on these formulations, exact methods, such as branch-and-bound algorithms supported by lower bound generators [24], and column generation algorithms based on the set-partitioning formulation [3], have been developed to solve the problems. However, due to the computational complexity, exact methods can only solve VRP instances of less than 100 customers within reasonable time, which is insufficient for real-world applications.

Classical Heuristic Methods

Consequently, there is a growing body of research on applying heuristics and meta-heuristics to approximate the optimal solution to the VRP. The classical heuristics could be categorized into constructive heuristics, improvement heuristics and two-phase heuristics. As their names suggest, the constructive heuristics often build routes from scratch, the improvement heuristics iteratively generate improved solution based on available ones, whereas the two-phase heuristics decompose the problem into two sub-problems, customer assignment and route generation, and solve them sequentially.

The most famous **construction heuristic** for the VRP is the saving heuristic first proposed by Clarke and Wright [25]. They initialize the solution assuming allocating

one vehicle to each customer is possible. In each subsequent iteration, two routes, one starts with node y and another ends with node z, are merged such that the biggest saving $d_{0,y} + d_{z,0} - d_{z,y}$ is achieved and the vehicle cargo capacity is not violated (note that node 0 is the depot and $d_{i,j}$ represents the distance between nodes i and j). The procedure is repeated until no more links are possible, followed by solving a TSP for each vehicle at final allocation to streamline the routes. Following studies make modifications to improve the original saving heuristic. For instance, Golden et al. [45] change the saving calculation formula to $d_{0,y} + d_{z,0} - \gamma d_{z,y}$. The parameter γ is employed to leverage the emphases on the newly added edge between vertices z and y and on their relative positions to the central depot. Altinkemer and Gavish [4] propose to perform the route merging through solving a maximum cost weighted matching problem. Solomon [105] extends the algorithm to the VRPTW, yet the solutions generated are far from the optimal.

Another well-known group of construction heuristics is the insertion-based algorithm put forward by Gillett and Miller [43]. In the sweep algorithm they develop, routes are sequentially constructed through inserting un-routed customers to the current routes without violating distance and cargo constraints. Solomon [105], again, applies this idea to the VRPTW considering both geographical and temporal information when determining the order and the position of insertion. However, the sequential approaches are performed in a greedy and myopic manner such that the last route generated is usually of poor quality. In order to overcome this intrinsic weakness, Potvin and Rousseau [92] design a parallel insertion heuristic to build routes simultaneously for the VRPTW. A regret measure is developed to select the next un-routed customer for insertion. Moreover, the work of Russel [99] also showcase the idea of parallel construction. The main difference between [99] and [92] lies in the processes of routes initialization, order of insertion and the post processing of un-routed customers.

The aforementioned construction algorithms could assist to find good feasible solutions quickly, whereas further improvement could still be realized through fine-tuning the current solution. To this end, a variety of **improvement heuristics** have been developed. The basic idea here is, given an available solution S (possibly generated from a construction heuristic), one explores its neighborhood, namely $N(S)$, defined by applying a neighborhood operator to S for a better solution. If a solution, say $S^{'}$, which is better than S was found in $N(S)$, we move to $S^{'}$ and, again, explore $N(S^{'})$. Such a procedure is repeated until no improvement could be made. The final solution is a locally, if not globally, optimal solution in the solution space defined by the neighborhood operator. One good example is shown in [92]. Based on the solution they obtain using the parallel insertion heuristic, they iteratively apply an interchange operator deleting $3 + L$ customers from and re-inserting them to the routes (here L is a pre-determined parameter). Other commonly implemented

operators include relocate operator, exchange operator, λ-opt operator etc., which are reviewed in [36, 76]. It is worth noting that, in practice, the choice of neighborhood operator should leverage the trade-off between computational tractability and the thoroughness of search. For instance, for the interchange operator used by [92], the size of neighborhood grows exponentially with L, thus they restrict L to $0, 1$ and 2.

In addition, Fisher and Jaikumar [40] consider the problem in a two-phase manner, i.e. first assign customers to vehicles and then optimize the route for each vehicle by solving a TSP separately. Although previous mentioned approaches, for example, the original saving algorithm proposed by Clarke and Wright [25], employ similar ideas, this is the first time the term **two-phase heuristic** has been used. The decomposition, to some extent, reduces the problem's complexity, yet, in the meanwhile, raises a new question – how to combine the two phases of the approach? Fisher and Jaikumar [40] construct a linear approximation for the optimal travel distance they could achieve in the second phase when solving the assignment problem in the first phase. They also note other plausible ways of approximation such as solving the problem iteratively with Benders decompostion. Solomon [105] exploits this cluster-first and route-second idea to design the time-oriented sweep heuristic for the VRPTW. This set of approaches is natural in that it follows the two phases often applied by human dispatchers, and could be potentially beneficial for future algorithmic improvement [76]. However, to the best of our knowledge, this research avenue has not been well pursued.

As mentioned earlier, the classical heuristics for VRP can be extended to VRPTW by taking into account time-related factors when constructing and improving the solutions. A more comprehensive review could be found in survey paper [17].

Meta-heuristic Methods

Recently, meta-heuristic, a top-level general strategy which guides other heuristics to search the feasible region, attracted attention for its capability of effectively finding quality solutions to hard combinatorial problems [74]. The meta-heuristics can be classified into local search and population search, yet a general tendency has been to move from algorithms based on a single paradigm to hybrid methods that draw on several principals [76].

The family of local search meta-heuristics that were successfully applied to solve the VRP includes, but not limited to, tabu search and simulated annealing search. The key differences between them lie in the neighborhood structure and mechanism for escaping from local optima.

Tabu search clearly stands out as the best meta-heuristic for the VRP [26]. The basic idea behind is to iteratively move to the best solution in the neighborhood even if it is worse than the current best one. Through the searching process, we maintain a tabu list that consists of edges that we temporarily exclude from considerations for a number of iterations in order to prevent cycling. To name a few successful applications to the VRP, the Taburoute by [42], the Taillard tabu search by [113], and the granular tabu search by [118] are some of the best examples. In [42], Gendreau et al. deign a generalized insertion routine to remove and re-insert customers in each tabu-search iteration. A continuous diversification term is added in the cost function to penalize the customers that are frequently removed. The algorithm consists of two searching phases, i.e. solution improvement and intensification. This is accomplished by applying the tabu-search process twice with different sets of parameters. Cordeau et al. [27] adapt this method for the VRPTW, yet with a post-optimization heuristic which is commonly adopted by tabu-search without intra-route neighborhood operator. It is acknowledged that one weakness of the tabu-search is that, in order to obtain high quality solutions, the searching process often need to be performed for thousands of iterations. In order to address this issue, Taillard [113] proposes two customer partition heuristics for the VRP. The routing problem for each cluster then could be solved in parallel. Toth and Vigo [118] tackle the problem by restricting the size of the neighborhood. They derive a granular neighborhood by discarding a large quantity of unpromising moves and actually exploring a small subset of the neighborhood containing the most promising ones. This method result in drastic reduction in computing time yet without considerably affecting solution quality.

The simulated annealing search (SA) explores the neighborhood in a different way. Instead of examing multiple neighbors, we randomly select one neighbor and move there if a better solution was found, otherwise we accept the deteriorated solution with a probability controlled by a temperature parameter that can be adjusted as the searching proceeds. Osman [90] first realizes this idea for the VRP. He first initializes a solution with a λ-interchange descent algorithm, and then conducts the SA enhanced by a tabu mechanism. The idea is extended by Chiang and Russell [23] for the VRPTW. Three neighborhood structure are developed and compared through numerical experiments. In order to accelerate the searching process, Czech et al. [28] develop a parallel computing scheme for the SA.

With regard to population search, instead of maintaining only the best solution so far as the local search algorithms do, it operates within a population of solutions. In general, it iteratively applies generic operators to generate new solutions based on the population and update the population periodically. To the best of our knowledge, all the population search algorithms that were successfully applied to the VRP are hybridized

with local search. Nagata and Braysy [84] propose a genetic algorithm for the VRP. At each iteration, children solutions are obtained by applying a crossover operator to parents selected from the solution population. They then perform modifications for each child for solution feasibility, followed by an improvement phase with local search. Various studies, for example [82] and [93] exploit the same framework as [84], while Thangiah et al. [115] hybridize the population search and local search in a different way. They employ an genetic algorithm to generate offspring solutions based on the "good" solutions they find during local search. These offspring solutions are then utilized as the starting points of the second-round local search.

Deep Reinforcement Learning Approaches

In addition to the aforementioned methods, there is an emerging group of literature on applying artificial intelligence techniques, more precisely, deep reinforcement learning (RL), to solve difficult combinatorial problems. Previous related research can be divided into two main branches by problem formulation: policy-based methods and value-based methods.

Policy-based methods seek to optimize over a policy model that can directly generate the optimal action given the current state. Its applications in combinatorial optimization originate from Vinyals et al. [121]. Considering that solutions to some combinatorial problems such as the TSP are permutations of the given nodes, they develop the pointer network (PN) based on the sequence to sequence model originally constructed for machine translation [112]. The PN consists of two separate long short-term memory newral networks (LSTM), one to encode the sequence of input nodes and another one to produce (decode) the output sequence. A content-based attention mechanism is introduced in between to enable the information to flow from the encoder to the decoder. They train the PN in a spervised manner with solutions generated by exact and heuristic algorithms. The model is able to efficiently find near-optimal solutions to small TSP instances. However, in theory, the model performance is constrained by the algorithms they use during training.

Bello et al. [8] then construct a RL framework employing the critic-arctic method to train the PN for solving the TSP. In the RL framework, they regard the PN as an agent making actions (determining the next node to visit) to solve the instances generated on the fly. Each action taken by the agent is assigned a reward based on which they update the parameters of PN so as to maximize the expected rewards the agent could obtain given any instances. This way, training the PN does not require any solutions generated by existing problem solvers. Therefore, it is possible for the model to outperform any other existing methods. Nazari et al. [85] argue that it is difficult to generalize the PN to combinatorial problems where system representations change over time. For example, in the VRP, every

time we visit a customer, the customer's demand will be zero afterwards. Since the LSTM encoder intakes input nodes sequentially, we have to feed the updated node information to the PN from scratch, which is computationally expensive. They point out that the order of input nodes does not provide any useful information. They thus replace the encoder LSTM with element-wise projections and apply the model to the VRP. More rencently, Kool, Hoof and Welling [72] propose a new model for the TSP and VRP based on multi-head attentions and train the model using RL with a greedy roll-out baseline.

On the other hand, value-based methods aim to find a value function that is able to evaluate the potential actions given a system state. Decisions could then be made based on the action evaluation. A successful application can be found in [65] where a graph-embedding network is introduced to parametrize the policy and is trained using the fitted Q-learning.

In general, though policy-based methods suffer from sample inefficiency thus generally requires longer training time in practice, they tend to more stably converge to good behaviors comparing with value-based methods.

2.3 Electric Vehicle Routing Problem

The electric vehicle routing problems (EVRP) could be regarded as a special variant of the green vehicle routing problem proposed by Erdogan and Miller-Hooks [37] in 2012 for alternative fuel vehicles (AFVs). To the best of our knowledge, it is the first research work that takes into account the opportunity to extend vehicles' distance limitation as a consequence of actions taken while en route, i.e. visiting an en-route station/satellite facility to replenish or unload. They formulate the problem as a classical VRP with additional constraints for en-route replenishment. Schneider et al. (2014) [103] extend the framework specifically for electric vehicles, propose the electric vehicle routing problem with time windows (EVRPTW). Instead of using constant replenishing time as in [37], they assume a linear charging time associated with the EVs' battery levels on arrival at the stations. While both [37] and [103] employ the assumption that an EV can only fully recharge its battery at the stations, Felipe et al. [38] consider partial charging stategies for EVs with multiple types of charger, each with a different charging speed and unit cost. Keskin and Catay [64] propose a similar study yet for EVRPTW. Starting from Lin et al. [80], researchers started to investigate the problem under more sophisticated modeling for EV battery. They develop a general framework for the EVRP considering the impact of EV load and speed on battery consumption, heterogeneous fleet, and topology of charging stations. Monotoya et al. [83] employ a non-linear function to model the battery charging process. Pelletier

et al. [91] incorporate the cost of battery degradation into the EVRP and formulate an at-depot charge scheduling problem with considerations regarding time-dependent energy prices, degradation cost and facility-related demand cost.

In terms of solution methods, due to the problem complexity, most of the existing studies implement meta-heuristics. As far as we know, the only paper that considers exact algorithm for the EVRP is the one done by Desaulniers et al. in 2016 [32]. A column generation algorithm is proposed for four variants of the EVPTW [103] with respect to full/partial recharge and the number of station visit(s) for a single route.

Given the intrinsic similarities between the VRP and EVRP, heuristic and meta-heuristic algorithms reviewed in Section 2.2.2 could be adapted for the EVRP with specific adjustments for the station nodes and charging schedule. For example, a construction heuristic based on the Clarke-and-Wright saving heuristic [25] is proposed in [37]. If the distance constraint is violated after merging two routes, a station will be inserted to the position that results in a minimal distance increase. This heuristic is employed by the following papers including [38] and [64]. When performing local search, Felipe et al. [38] design a new neighborhood operator, namely recharge relocation, to relocate the station, if exists, to its best location along the route without reordering the customers' sequence. Similar operators, namely station removal, station insertion [64] and StationReIn [103], are defined by other literature. For a hybrid electric vehicle routing problem, Abdallha [1] formulate three Lagrangian relaxations and a tabu search heuristic that can efficiently provide quality lower and upper bounds respectively. From the perspective of population search, Abdulaal et al. [2] exploit a route-first, schedule-second idea. They first employ a generic algorithm to generate customer sequences for vehicles, then, if needed, assign stations to routes with a Markov Decision Process. Finally, trust-region search is performed to optimize the EV's operational cost.

Summing up, since the proposed problem is much more complicated than the VRPs proposed by most of the existing research, implementing meta-heuristic or artificial intelligence methods are a more realistic option than using exact approaches. Specific attention should be placed on the design of station-related neighborhood operator. Further, the removal and insertion of station could have sizable impact on the time and battery level on arrival at the rest nodes along the route [103]. Fast evaluation of this impact is paramount for efficient implementation of a meta-heuristic.

2.4 Electric Vehicle Routing in V2G Context

Limited research efforts were made to investigate the joint optimization of EV routing and charging in V2G context. The authors in [119] and [128] study the problem for a single EV. Trivino-Cabrera et al. [119] formulate a mixed integer programming problem for a single EV providing energy delivery service between locations taking into account battery degradation, time-dependent energy prices and the interaction between the EV and the grid. The model is able to provide routing and charging/discharging schedule so as to maximize the EV owner's overall benefit. However, the EV is not responsible for any customer services in this research. Under time-variant electricity prices, Yang et al. [128] optimize over the monetary cost of one EV seeking to provide delivery/pickup services for a series of customers in the context of uni-directional V2G where discharging is not enabled.

For an EV fleet as a whole, Yu et al. in [56] propose an autonomous electric vehicle (AEV) logistics system for smart grid service provision in uni-directional V2G context. The system can generate optimal routes for the AEV fleet to satisfy all the logistic requests and strategically detour to store the power generated by intermittent sources. Three objectives are taken into consideration, i.e. minimizing total driving distance, maximizing the amount of energy charged from renewable sources, and minimizing the amount of time the AVs require for reaching final destinations. They formulate the problem as a quadratic-constrained mixed integer linear programming. To accelerate the solution process, they formulate a Lagrangian relaxation and a dual decomposition algorithm to enable distributed computation. Tang et al. [114] consider a set of EVs travelling from their sources to destinations without en-route customers. EVs could detour to two types of stations, one associated with renewable energy and located in suburb areas while the other type are normal stations located in urban areas, for en-route charging and discharging. An optimization problem is formulated to provide routing and charging/discharging decisions. From the perspective of an EV dispatcher, [2] investigates the routing and charging/discharging schedule for an EV fleet to provide services to customers with time window constraints. They consider continuous energy prices and optimize over a quadratic objective function simultaneously considering charging cost, discharging reward as well as battery degradation cost. Due to the complexity of the proposed model, it is difficult to coordinate the optimization across EVs. They employ the idea of cluster-first and route-second to enable parallel optimization of the routes for different EVs. However, they do not specify the methods for assigning customers to EVs.

Chapter 3

The EVRPTW-TP

3.1 Problem Description

We define the electric vehicle routing problem with time windows under dynamic electricity prices (EVRPTW-TP) on a complete directed graph $G(V_{c,s,od}, E)$ where $V_{c,s,od}$ denotes the set of all nodes, and E is the set of all edges. The nodes can be partitioned into three distinct categories: customer nodes, station nodes, and depot nodes. Let $V_c = \{1, 2, \ldots, N\}$ denote the set of N customers, each of them is associated with a demand d_i, a service time s_i, and a time window $[e_i, l_i]$ during which an EV should arrive at the node and start providing services. Let $V_s = \{N+1, \ldots, N+S\}$ be the set of S en-route stations. We introduce two nodes which make up the depot set $D = \{0, N+S+1\}$ to denote the same depot for formulation purposes. Nodes 0 and $N+S+1$ are the first and last nodes of every EV route respectively. In the following discussion, to indicate a set that includes the respective depot nodes, we subscript the set with "o" (origin) and/or "d" (destination). Similarly, we subscript the set with "c" and "s" to indicate the inclusion of customer and station nodes respectively. For instance, $V_{cd} = V_c \cup \{N+S+1\}$. Set E consists of edges among customer, station, and depot nodes. Each edge, say (i, j), is assigned a distance d_{ij}. Assuming constant consumption rate g and constant travel speed v, we can infer the travel time $t_{ij} = \frac{d_{ij}}{v}$ and energy consumption $c_{ij} = gd_{ij}$ for each edge. Note that, under the assumption of constant charging speed $\frac{1}{\alpha}$, the energy consumption here is represented as the time required to recharge the energy consumed along the edge, i.e. $f_{ij} = \alpha c_{ij}$.

The commercial EV fleet consists of K homogeneous EVs, each has a load capacity Q, and a battery capacity C. Similar to the time-energy transformation done for energy consumption along edges, we define $B = \alpha C$ which is the time required to fully recharge

the battery from 0. At the very beginning of the planning horizon, all the EVs are at the depot (node 0) with a full battery. They then are allowed to leave the depot to serve the customers during their time windows and visit en-route stations if scheduled. All EVs should return to the depot (node $N+S+1$) before the end of the planning horizon. During each station/depot visit, an EV can either pay to charge its battery or make profits by injecting energy back to the grid from its battery. The charging cost and discharging reward vary according to time. We assume that EVs are allowed to perform either charging or discharging during their station visits, but are not allowed to perform both of them during one single visit. We make this assumption to avoid frequent switches between charging and discharging which were shown to have detrimental effects on battery lifespan. We discretize the planning horizon into $|T|$ consecutive periods, each of length δ and is associated with a charging rate P^t_{re} and discharging rate P^t_{dis}, $\forall t \in T$. In order to avoid a non-linear formulation, we assume that once an EV starts charging/discharging its battery, it can only do so for the whole period. For example, suppose $\delta = 60$, an EV arrives at a station at time 110, it cannot start charging/discharging until time 120 when the 3^{rd} period starts. An EV can stop charging/discharging only at the end of the following periods. The only exception is that once the EV's battery is fully charged the process will be automatically terminated at anytime, yet the EV owner still need to pay for the whole charging period. In addition, EVs would be fully charged during the night at the minimum charging rate P_{night}. We seek to find K routes such that all the customer demands are satisfied and the total net electricity related cost is minimized without violating time, cargo capacity and battery capacity constraints. The notations are summarized in Table 3.1.

3.2 Mathematical Formulation

As discussed in Chapter 2, two-index, three-index and set partitioning formulations are commonly implemented by previous studies on VRP. As an extension of the VRP, EVRP can also be formulated in these manners. Set partitioning formulation is often associated with column generation, see for example [32], for solving the problem exactly. Two-index formulation, such as [103], generally requires less number of decision variables than three-index formulation, thus is favored by most previous research. Given the complexity of the proposed problem, it is difficult to solve it exactly. We formulate the problem in a three-index manner such that we can decompose the problem more conveniently using a way similar to [58] and [41], which will be explained in more detail in Chapter 4.

To formulate the problem we define the binary variable x_{ijk} that takes a value of 1 if edge(i, j) is travelled by vehicle k and a value of 0 otherwise. The real non-negative

Table 3.1: Summary of the Notation

	Definition
V_c	The set of customer nodes
V_s	The set of station nodes
V_{od}	The set of depot nodes
E	The set of all edges
T	The set of charging/discharging periods
N	The number of customers
S	The number of stations
K	The number of EVs
δ	The length of each charging/discharging period
Q	Cargo capacity of an EV
C	Battery capacity
B	The amount of time required to fully charge the battery
α	The reciprocal of the constant charging speed
g	Energy Consumption rate with respect to distance traveled
v	Constant traveling speed
d_{ij}	Length of edge (i, j)
t_{ij}	Travel time along edge (i, j)
c_{ij}	Energy consumption along edge (i, j)
f_{ij}	The amount of time required to charge the energy consumed along edge (i, j)
e_i	Earliest service start time at node i
l_i	Latest service start time at node i
s_i	Required service time at node i
q_i	The demand at node i
P_{re}^t	The cost for charging during the t^{th} period
P_{dis}^t	The reward for discharging during the t^{th} period
P_{night}	The unit cost of charging during the night

Table 3.2: Summary of the Decision Variables

Variable	Type	Definition
x_{ijk}	Binary	If edge (i, j) is travelled by vehicle k $(= 1)$ or not $(= 0)$
y_{ik}	Binary	Whether vehicle k charges $(= 1)$ or discharges at node i $(= 0)$
r_{itk}	Binary	If vehicle k charges at node i during period t $(= 1)$ or not $(= 0)$
d_{itk}	Binary	If vehicle k discharges at node i during period t $(= 1)$ or not $(= 0)$
τ_{ik}	Continuous	The arrival time of vehicle k at node i
b_{ik}	Continuous	The battery level of vehicle k on arrival at node i
u_{ik}	Continuous	The remaining cargo of vehicle k on arrival at node i

variables τ_{ik} refers to the time EV k arrives at node i, while b_{ik} and u_{ik} represents its battery level and remaining cargo on arrival at node i, respectively. Note that the battery level here indicates the amount of time required to charge the battery from 0 to the current level. The binary variables r_{itk} and d_{itk} indicate if EV k recharges and discharges its battery at node i during period t (=1) or not (=0) respectively. The binary variable y_{ik} takes a value of 1 if EV k recharges its battery at node i and takes a value of 0 otherwise. Note that the decision variables d_{itk}, r_{itk} and y_{ik} are only associated with station and depot nodes, while others are associated with all nodes. For the sake simplicity, we use subscript d to denote the depot node $N + S + 1$. The type of definition of the decision variables are summarized in Table 3.2.

We formulate the problem as a mixed-integer programming as follows and refer to it as EVRPTW-TP.

$$\min \quad \sum_{k=1}^{K} \sum_{i \in V_{s,od}} \sum_{t \in T} \delta[r_{itk} P^t_{re} - d_{itk} P^t_{dis}] + \sum_{k} P_{night}[B - b_{dk} - \sum_{t \in T} \delta(r_{dtk} - d_{dtk})] \tag{3.1}$$

$$\text{s.t.} \quad \sum_{k} \sum_{j \in V_{c,s,d}} x_{ijk} = 1, \quad \forall i \in V_c \tag{3.2}$$

$$\sum_{i \in V_{c,s,o}} x_{idk} = 1, \quad \forall k \in \{1, 2, \ldots, K\} \tag{3.3}$$

$$\sum_{j \in V_{c,s,o}} x_{jik} - \sum_{j \in V_{c,s,d}} x_{ijk} = 0, \quad \forall i \in V_{c,s}, k \in \{1, 2, \ldots, K\} \tag{3.4}$$

$$\tau_{ik} + (t_{ij} + s_i)x_{ijk} - M(1 - x_{ijk}) \leq \tau_{jk}, \quad \forall i \in V_{c,s,o}, j \in V_{c,s,d}, k \in \{1, 2, \ldots, K\} \tag{3.5}$$

$$t\delta(r_{itk} + d_{itk}) + t_{ij}x_{ijk} - M(1 - x_{ijk}) \leq \tau_{jk}, \quad \forall i \in V_{s,o}, j \in V_{c,s,d}, t \in T, k \in \{1, 2, \ldots, K\} \tag{3.6}$$

$$e_i \leq \tau_{ik} \leq l_i, \quad \forall i \in V_{c,s,od}, k \in \{1, 2, \ldots, K\} \tag{3.7}$$

$$b_{0k} = B, \quad \forall k \in \{1, 2, \ldots, K\} \tag{3.8}$$

$$b_{jk} \leq b_{ik} - f_{ij}x_{ijk} + M(1 - x_{ijk}), \quad \forall i \in V_c, \forall j \in V_{c,s,d}, k \in \{1, 2, \ldots, K\} \tag{3.9}$$

$$b_{jk} \leq b_{ik} + \sum_{t \in T} \delta r_{itk} - \sum_{t \in T} \delta d_{itk} - f_{ij}x_{ijk} + M(1 - x_{ijk}), \quad \forall i \in V_{s,o}, j \in V_{c,s,d}, k \in \{1, 2, \ldots, K\} \tag{3.10}$$

$$\sum_{t \in T} \delta r_{itk} \leq B - b_{ik}, \quad \forall i \in V_{s,od}, k \in \{1, 2, \ldots, K\} \tag{3.11}$$

$$\sum_{t \in T} \delta d_{itk} \leq b_{ik}, \quad \forall i \in V_{s,od}, k \in \{1, 2, \ldots, K\} \tag{3.12}$$

$$\sum_{t \in T} r_{itk} \leq |T| y_{ik}, \quad \forall i \in V_{s,od}, k \in \{1, 2, \ldots, K\} \tag{3.13}$$

$$\sum_{t \in T} d_{itk} \leq |T|(1 - y_{ik}), \quad \forall i \in V_{s,od}, k \in \{1, 2, \ldots, K\} \tag{3.14}$$

$$\tau_{ik} - (t - 1)\delta \leq M(1 - d_{itk} - r_{itk}), \quad \forall i \in V_{s,d}, t \in T, k \in \{1, 2, \ldots, K\} \tag{3.15}$$

$$u_{jk} \leq u_{ik} - q_i x_{ijk} + M(1 - x_{ijk}), \quad \forall i \in V_{c,s,o}, j \in V_{c,s,d}, k \in \{1, 2, \ldots, K\} \tag{3.16}$$

$$u_{0k} = Q, \quad k \in \{1, 2, \ldots, K\} \tag{3.17}$$

$$0 \leq h_{jk} \leq B \sum_{i \in V_{c,s,o}} x_{ijk}, \quad \forall j \in V_s, k \in \{1, 2, \ldots, K\} \tag{3.18}$$

$$u_{ik}, \quad \tau_{ik} \geq 0, \quad \forall i \in V_{c,s,od}, k \in \{1, 2, \ldots, K\} \tag{3.19}$$

$$x_{ijk} \in \{0, 1\}, \quad \forall i \in V_{c,s,o}, \forall j \in V_{c,s,d}, k \in \{1, 2, \ldots, K\} \tag{3.20}$$

$$y_{ik} \in \{0, 1\}, \quad \forall i \in V_s, k \in \{1, 2, \ldots, K\} \tag{3.21}$$

$$r_{itk}, \quad d_{itk} \in \{0, 1\}, \quad \forall i \in V_{c,od}, t \in T, k \in \{1, 2, \ldots, K\} \tag{3.22}$$

The objective function 3.1 minimizes the net cost, i.e. total charging cost minus total discharging reward. The first term corresponds to the net cost during the planning horizon, while the second term refers to the cost of fully charging all EVs at night. Constraints 3.2 ensure that every customer is served by exactly one EV. Constraints 3.3 enforce all the EVs to return to the depot by the end of the planning horizon. Constraints 3.4 guarantee that no route ends at a customer or a station node. Constraints 3.5 - 3.7 deal with the arrival time of EVs at a given node. Constraints 3.5 ensure the time feasibility of edges leaving customers and the depot, while Constraints 3.6 deal with the edges originated from charging stations. Note that $t\delta$ is the time when the t^{th} period ends. In Constraint 3.6, we assume an EV can leave the station right after its charging/discharging schedule at the station has completed. Constraints 3.7 ensure that the time windows of all the nodes are not violated. Constraints 3.8 - 3.10 consider EVs' battery levels on arrival at the nodes. Constraints 3.8 mean that every EV is fully charged before leaving the depot. Constraints 3.9 ensure battery feasibility along edges leaving customers and the depot, while Constraints 3.10 consider edges leaving stations. Constraints 3.11 - 3.15 deal with EVs' charging and discharging behaviors at stations and the depot. Constraints 3.11 mean that an EV battery cannot be recharged to a level that exceeds its capacity,

while Constraints 3.12 state that an EV battery cannot be discharged to a level below 0. Constraints 3.13 and 3.14 ensure an EV is allowed to discharge or recharge its battery at station and depot nodes but is not allowed to do both in a single visit. In Constraints 3.15, if $\tau_i - (t-1)\delta > 0$, it suggests that the EV has not arrived at station/depot i by the start of the t^{th} period. Therefore, $d_{i,t}$ and $r_{i,t}$ on the right-hand-side have to take a value of 0 implying that an EV is allowed to charge/discharge its battery only after its arrival at a station/depot. If $\tau_i - (t-1)\delta \leq 0$, we can set discharging/recharging schedule at node i for the t^{th} period, $d_{i,t}$ and $r_{i,t}$ thus can be either 0 or 1. Constraints 3.16 guarantee that the demands along an EV's route are all satisfied, whereas Constraints 3.17 state that all EVs are full at the start of the planning horizon. Constraints 3.18 - 3.22 define the ranges of decision variables. The "M" in the aforementioned inequalities is a very large constant number.

Chapter 4

Lagrangian Relexation for the EVRPTW-TP

The proposed EVRPTW-TP is an NP hard combinatorial problem. Thus, solving the problem with commercial solving tools, such as CPLEX, requires extensive computational efforts. Solving instances of over 10 customers using traditional MIP solvers is computationally expensive. Therefore, developing efficient solution algorithm is of great importance for its real-world application.

For the EVRPTW-TP, all the constraints other than Constraints 3.2 are associated with a sub-index k implying constraints for different EVs are separable. We can take advantage of this special structure by relaxing Constraints 3.2 and then decomposing the relaxed problem into a set of identical sub-problems (as the EV fleet is homogeneous) that can be solved relatively quickly using CPLEX. The relaxed problem can assist to provide lower bounds to the original problem which can help enhance the performance of branch-and-bound and allows to evaluate the solution quality of the heuristics that will be detailed in Chapter 5. In terms of the relaxation, previous studies, see for example [58] and [41], have successfully applied Lagrangian relaxation to various VRP problems and have shown it to be very effective and efficient. In this chapter, we exploit a Lagrangian relaxation to decompose the EVRPTW-TP and employ a sub-gradient heuristic to approximate the original problem.

4.1 Lagrangian Relaxation

We construct the Lagrangian relaxation, as shown in Equation 4.1, by relaxing Constraints 3.2 and introducing a set of penalty parameters λ's to penalize the violations of Constraints 3.2 in the objective function. The feasible region of the EVRPTW-TP is a subset of that of Problem 4.1, enabling Problem 4.1 to provide lower bounds to the original problem. Through fine-tuning the values of λ's, we can obtain "the best" lower bound whose distance from the original optima, in theory, could be zero. The corresponding maximization problem is the Lagrangian master problem as in Equation 4.2 where λ is a vector of penalty parameters. As mentioned, the Lagrangian relaxation problem can be decomposed into a set of identical sub-problems, each associated with an EV. The sub-problem is presented in Equation 4.3. We note that the constraints as well as the decision variables of the sub-problem are in the same formats as the original EVRPTW-TP yet without sub-index k and Constraints 3.2.

Lagrangian Relaxation Problem

$$\begin{aligned} Z_{LR}(\lambda) = \min_{s.t(3.3)-(3.22)} \quad & \sum_{k} \sum_{i \in V_{s,od}} \sum_{t \in T} \delta[r_{itk} P^t_{re} - d_{itk} P^t_{dis}] \\ & + \sum_{k} P_{night}[B - b_{dk} - \sum_{t \in T} \delta(r_{dtk} - d_{dtk})] \\ & + \sum_{i \in V_c} \lambda_i (1 - \sum_{k} \sum_{j \in V_{c,s,d}} x_{ijk}) \end{aligned} \tag{4.1}$$

Lagrangian Master Problem

$$\begin{aligned} Z_{LM} &= \max_{\lambda \in \mathbb{R}} \quad Z_{LR}(\lambda) \\ &= \max_{\lambda \in \mathbb{R}} \quad K Z_{SP}(\lambda) + \sum_{i \in V_c} \lambda_i \end{aligned} \tag{4.2}$$

Lagrangian Sub-problem

$$\begin{aligned} Z_{SP}(\lambda) = \min_{s.t.(3.3)-(3.22)} \quad & \sum_{i \in V_{s,od}} \sum_{t \in T} \delta[r_{it} P^t_{re} - d_{it} P^t_{dis}] + P_{night}[B - b_d - \sum_{t \in T} \delta(r_{dt} - d_{dt})] \\ & - \sum_{i \in V_c} \lambda_i \sum_{j \in V_{c,s,d}} x_{ij} \end{aligned} \tag{4.3}$$

In the sub-problem defined above, penalty parameters λ's only appear in the objective function, implying that the feasible region defined by the constraints is fixed regardless of the values of λ's. Each feasible solution to the sub-problem corresponds to a route and a schedule that an EV could take. We define several parameters for each feasible solution $h \in H$ where H is the feasible region of the Lagrangian sub-problem, as shown in Equations 4.4 - 4.7. We superscript the decision variables with h indicating their association with solution h. The value of c^h represents the net cost of the schedule implied by h, a_i^h is the number of times node i is visited along the route in h. Additionally, w^h is a N dimensional vector whose i^{th} entry w_i^h indicates the negative number of times node i is visited more than once by the whole EV fleet.

$$c^h = \sum_{i \in V_{s,od}} \sum_{t \in T} \delta[r_{it}^h P_{re}^t - d_{it}^h P_{dis}^t] + P_{night}[B - b_d^h - \sum_{t \in T} \delta(r_{dt}^h - d_{dt}^h)] \tag{4.4}$$

$$a_i^h = \sum_{j \in V_{c,s,d}} x_{ij}^h \tag{4.5}$$

$$w^h = (w_1^h, w_2^h, \ldots, w_N^h) \tag{4.6}$$

$$w_i^h = 1 - K a_i^h \tag{4.7}$$

With these parameters, we can re-write the Lagrangian sub-problem and the Lagrangian relaxation problem as in Equations 4.8 and 4.9 respectively. Further, we re-write the Lagrangian master problem as in Equation 4.10.

$$Z_{SP}(\lambda) = \min_h c^h - \sum_{i \in V_c} \lambda_i a_i^h \tag{4.8}$$

$$\begin{aligned} Z_{LR}(\lambda) &= \min_h K c^h + \sum_{i \in V_c} \lambda_i (1 - K a_i^h) \\ &= \min_h K c^h + \sum_{i \in V_c} \lambda_i w_i^h \end{aligned} \tag{4.9}$$

$$\begin{aligned} Z_{LM} = &\max_{\lambda, \theta \in \mathbb{R}} \quad \theta \\ \text{s.t.} \quad &\theta \leq K c_h + \sum_{i \in V_c} \lambda_i w_i^h, \quad \forall h \in H \end{aligned} \tag{4.10}$$

4.2 Sub-gradient Heuristic

Solving the Lagrangian master problem defined in 4.10 is non-trivial as enumerating all feasible solutions of the sub-problem requires an extensive amount of computations. To overcome this issue, we utilize a sub-gradient heuristic to solve the Lagrangian master problem iteratively. The basic idea is maintaining only a subset of all feasible solutions and accumulate feasible solutions one after another as we move from one λ vector to another one along its sub-gradient direction. The feasible solutions accumulated are then exploited to approximate the sub-gradient at the subsequent λ vectors. This type of sub-gradient heuristics is initiated by Held and Karp in [52] and redefined and experimentally studied in [53] where a way for choosing step size is presented. However, the choice of step size in [53] largely relies on the choice of a parameter λ_j (λ_j here is a constant parameter set for iteration j in the original paper, and is different from the vector λ we defined in this thesis), thus is relatively difficult to implement. Instead, we employ the method developed by Kallehauge et al. [58] that incorporates the trust region defined by Griffith and Stewart [48] in a non-linear programming context. In doing so, the width of the trust region, equivalently the step size, is dynamically adjusted through the search process based on the quality of sub-gradient approximation at a given iteration without parameter engineering. More specifically, the sub-gradient heuristic is introduced as follows.

At the initialization phase, we randomly generate a vector λ^1 and set the search starting point μ^1 to be equal to λ^1. We set a tolerance *tol* as a termination criteria, an initial trust region width ω^1, and an iteration counter i is set to 1. We then solve the Lagrangian sub-problem $Z_{SP}(\mu^1)$ and $Z_{LR}(\mu^1)$, and c_1 and w_1 are set based on its optimal solution.

At iteration i, we first solve the master problem presented in 4.11 which considers the solutions we found up to iteration i and optimize the penalty factors μ's within the trust region defined by the current penalty factors λ's and width ω^i. The local optimal solution to the master problem is denoted as μ^{i+1} and its corresponding objective value $\hat{Z^i_{LM}}$ is an estimation of $Z_{LR}(\mu^{i+1})$. We thus can estimate the improvement in the lower bound $\Delta_{est} = \hat{Z^i_{LM}} - Z_{LR}(\lambda^i)$. If the estimated improvement is smaller than the tolerance *tol*, we terminate the search process as the lower bound is already "good enough". In addition, it is possible that the current best lower bound $Z_{LR}(\lambda^i)$ is greater than a pre-defined large value M_{LR}, then we claim that the instance is infeasible. For EVRPTW-TP, we set M_{LR} as three times the cost for fully charging the whole EV fleet at the highest charging rate, i.e. $M_{LR} = 3KP^{max}_{re}\frac{B}{\delta}$ where P^{max}_{re} indicates the highest charging rate throughout the planning horizon, $\frac{B}{\delta}$ is the number of periods required to fully charge an EV.

$$
\begin{aligned}
\hat{Z^i_{LM}} = \max_{\mu,\theta} \quad & \theta \\
s.t. \quad & \theta \leq Z_{LR}(\mu_q) + < w_q, \mu - \mu^q >, \quad \forall q = 1, 2, \ldots, i \\
& \mu \leq \lambda^i + \omega^i \\
& \mu \geq \lambda^i - \omega^i
\end{aligned} \tag{4.11}
$$

If the search process is still "alive", we solve the sub-problem $Z_{SP}(\mu^{i+1})$ and compute the corresponding $Z_{LR}(\mu^{i+1})$ as well as c_{i+1} and w_{i+1}. Then we evaluate the quality of the trust region by calculating a metric $\rho = \frac{Z_{LR}(\mu^{i+1}) - Z_{LR}(\lambda^i)}{\Delta_{est}}$. We enlarge the region width by 50% if $\rho > 1$, divide the width by 1.1 if ρ is negative, and make no adjustments otherwise. Further, if $\rho > 0$, we move to a new penalty vector by setting $\lambda^{i+1} = \mu^{i+1}$ and take a non-step, i.e. setting $\lambda^{i+1} = \lambda^i$, otherwise. The search procedure is repeated until one of the termination criteria is met, and the latest $Z_{LR}(\lambda)$ at termination is the best lower bound to the original EVRPTW-TP. In particular, the pseudo code of the sub-gradient heuristic is summarized in Algorithm 1.

Algorithm 1 Sub-gradient Heuristic for Solving the Lagrangian Dual

$tol, \omega^1, \lambda^1 \leftarrow Initialization()$
$\mu^1 \leftarrow \lambda^1$
$i \leftarrow 1$
solve $Z_{SP}(\mu^1)$, and compute $Z_{LR}(\mu^1)$, c_1 and w_1 based on its optima
while True **do**
    $\mu^{i+1}, \hat{Z^i_{LM}} \leftarrow Master(\omega^i, \mu^1, \ldots, \mu^i, w_1, \ldots, w_i)$
    $\Delta_{est} \leftarrow \hat{Z^i_{LM}} - Z_{LR}(\lambda^i)$ (estimate improvement)
    if $\Delta_{est} < tol$ or $Z_{LR}(\lambda^i) > M_{LR}$ **then**
        break
    else
        solve $Z_{SP}(\mu^{i+1})$, and calculate $Z_{LR}(\mu^{i+1})$, c_{i+1}, and w_{i+1}
        $\rho \leftarrow \frac{Z_{LR}(\mu^{i+1}) - Z_{LR}(\lambda^i)}{\Delta_{est}}$
        if $\rho = 1$ **then** $\omega^{i+1} = \omega^i \times 1.5$
        else if $\rho < 0$ **then** $\omega^{i+1} = \omega^i / 1.1$
        else $\omega^{i+1} = \omega^i$
    if $\rho > 0$ **then**, move forward: $\lambda^{i+1} \leftarrow \mu^{i+1}$
    else $\lambda^{i+1} \leftarrow \lambda^i$
    $i \leftarrow i + 1$
$LB \leftarrow Z_D(\lambda^i)$

Chapter 5

VNS/TS Hybrid Heuristic for EVRPTW-TP

The Lagrangian relaxation introduced in the last chapter assists us to obtain quality lower bounds to the original problem. However, we still do not have a feasible solution that can be used to dispatch EVs in real-world operations. In this chapter, we focus on implementing a meta-heuristic to generate feasible solutions of high quality.

As mentioned earlier, meta-heuristics for EV routing problems can be classified into three main categories: annealing search, tabu search and population search. Due to the presence of charging stations, constructing feasible offspring solutions from parent solutions is not straight forward making population search algorithms, such as generic algorithms, less desirable. Following the framework presented in [103] for the EVRPTW, we develop a variable neighborhood search and tabu search hybrid (VNS/TS) meta-heuristic with an annealing mechanism to solve the EVRPTW-TP.

5.1 The Framework

The overall framework of the VNS/TS heuristic is shown in Algorithm 2. The heuristic consists of two components: a variable neighborhood search (VNS) components serves to diversify the search process, and a tabu search (TS) component for local intensification. Each function and component mentioned here are elaborated in the following sections. First, we initialize a set of routes that visit each customer with a positive demand exactly once. We then search the region defined by the EVRPTW-TP for at most η_{vns} iterations.

Specifically, in each VNS iteration, we first move to a random solution $S^{'}$ in the neighborhood of the current best solution S. After that, we implement tabu search to find the local optima $S^{''}$ starting from $S^{'}$ for at most η_{tabu} steps. If the local optima satisfies the acceptance criteria, we move from the current solution S to $S^{''}$. If we do not find any solutions better than the current best with respect to the generalized cost function for η_{early} consecutive VNS iterations, the early stop mechanism will be triggered, otherwise we terminate after η_{VNS} iterations.

Algorithm 2 VNS/TS Heuristic For EVRPTW-TP

$S = initialization()$
$counter \leftarrow 0$
for $i = 1, 2, \ldots, \eta_{vns}$ **do**
    $S^{'} = Move2Neighbor(S)$
    for $j = 1, 2, \ldots, \eta_{tabu}$ **do**
        $S^{'} = Tabu(S^{'})$
    $S^{''} \leftarrow S'$
    if $f_{gen}(S) > f_{gen}(S^{''})$ **then**
        $counter \leftarrow 0$
    else
        $counter \leftarrow counter + 1$
    if $counter \geq \eta_{early}$ **then**
        Break
    else if $Accept(S, S^{''})$ **then** $S \leftarrow S^{''}$

5.2 Initialization

We construct the initial solution with a sweep heuristic similar to [103]. The pseudo code for the heuristic is presented in Algorithm 3. First, we sort the customers in an increasing order of the angle between the depot, a randomly selected point and the customer. Then, starting from the customer with the smallest angle, we insert the customers to the active route at the position resulting in minimal increase in the traveling distance of the active route. Once the battery or cargo constraints of the active route are violated, we activate a new route if the number of routes used so far has not exceeded the given EV fleet size. Otherwise, all the remaining customers are inserted into the last route. Finally, all the constructed routes form the initial solution $S = (R^1, R^2, \ldots, R^K)$

Algorithm 3 Initialization

1: $V_c \leftarrow SortByAngle(V_c)$
2: $i \leftarrow 1$
3: $R^i \leftarrow [0, N+S]$
4: **for** c in V_c **do**
5: $\quad R^i \leftarrow insert(R^i, c)$
6: $\quad$ **if** $BattOrCargoVio(R^i)$ and $i < K$ **then**
7: $\quad\quad i \leftarrow i+1$
8: $\quad\quad R^i \leftarrow [0, N+S]$

5.3 VNS Component

The neighborhood structure is defined by a cyclic-exchange operator. We first select N_r routes from the current solution S to form an exchange cycle. For each of the selected routes, namely R^i, we generate a random number υ_i indicating the number of consecutive nodes in R^i that form the exchange block, and then randomly pick the starting nodes of the block. After that, the blocks are reversed and then transferred among the selected routes forming a new solution $S^{'}$, feasible or infeasible, in the neighborhood of the current solution. Figure 5.1 shows an example of the cyclic exchange operator where $N_r = 3$, the selected routes are R^i, R^j and R^k with $\upsilon_i = 2$, $\upsilon_j = 3$, $\upsilon_k = 2$. The three dash blocks on the left form an exchange cycle. The blocks are reversed and transferred forming the three new routes $R^{i'}$, $R^{j'}$ and $R^{k'}$ on the right.

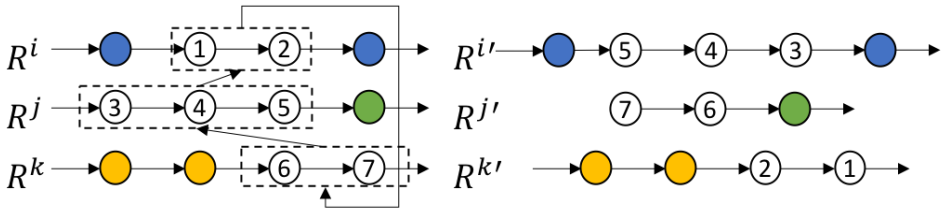

Figure 5.1: An Example of the Cyclic Exchange Operator

Based on S', we then use the tabu search which is detailed in the next section to find the local optima $S^{''}$. Instead of accepting only $S^{''}$ that is better than the current solution S, we also accept solution worse than the current solution with a probability of $\exp\left[\frac{f_{gen}(S)-f_{gen}(S^{''})}{Temp}\right]$ to diversify the searching process. Here $Temp$ is the temperature

that is initially set to $Temp_0$ such that a solution with cost $f_{gen}(S'')$ that is κ worse than the current best solution will be accepted with a probability of 50%. The temperature will decrease by a factor after each VNS ietration so that in the last 20% iterations, the temperature is below 0.0001.

The VNS component will be performed at most η_{VNS} times. The best solution we found before the termination of the VNS component is the final solution we get for the EVRPTW-TP.

5.4 Tabu Component

We consider four widely-used operators for the Tabu search. The operators are visualized in Figure 5.2 where nodes in the same row are travelled by an EV before the operator is applied, the dash arrows are edges to be removed, the stripped and shadowed nodes are the nodes selected by the algorithm. New routes after the operator is applied are highlighted in blue and orange. In particular, we introduce the four operators as follows.

- **2-opt*:** Select two routes and remove one edge from each of them. Connect the first part of the first route with the second part of the second route and vice versa.
- **Exchange:** Exchange the positions of two nodes. The two nodes could either be in the same route or in different routes.
- **Relocate:** Select one route, remove one node from this route and reinsert this node to another position. The new position could either be in the same route or in another route.
- **StationInRe:** Perform insertion or removal of a station node.

In each tabu iteration, we exam all the possibilities of applying the aforementioned operators to different parts of S', and take the action that results in maximal decrease in the generalized cost. We prohibit the reinsertion of removed edges for a specified number of tabu iterations called tabu tenure. The tabu tenure for each deleted edge is randomly selected from an interval $[v_{min}, v_{max}]$. We lift all the tabu restrictions once we find a new best solution. The procedures are repeated until no improvement could be made or for at most η_{tabu} iterations and the best solution we find before termination would be S''.

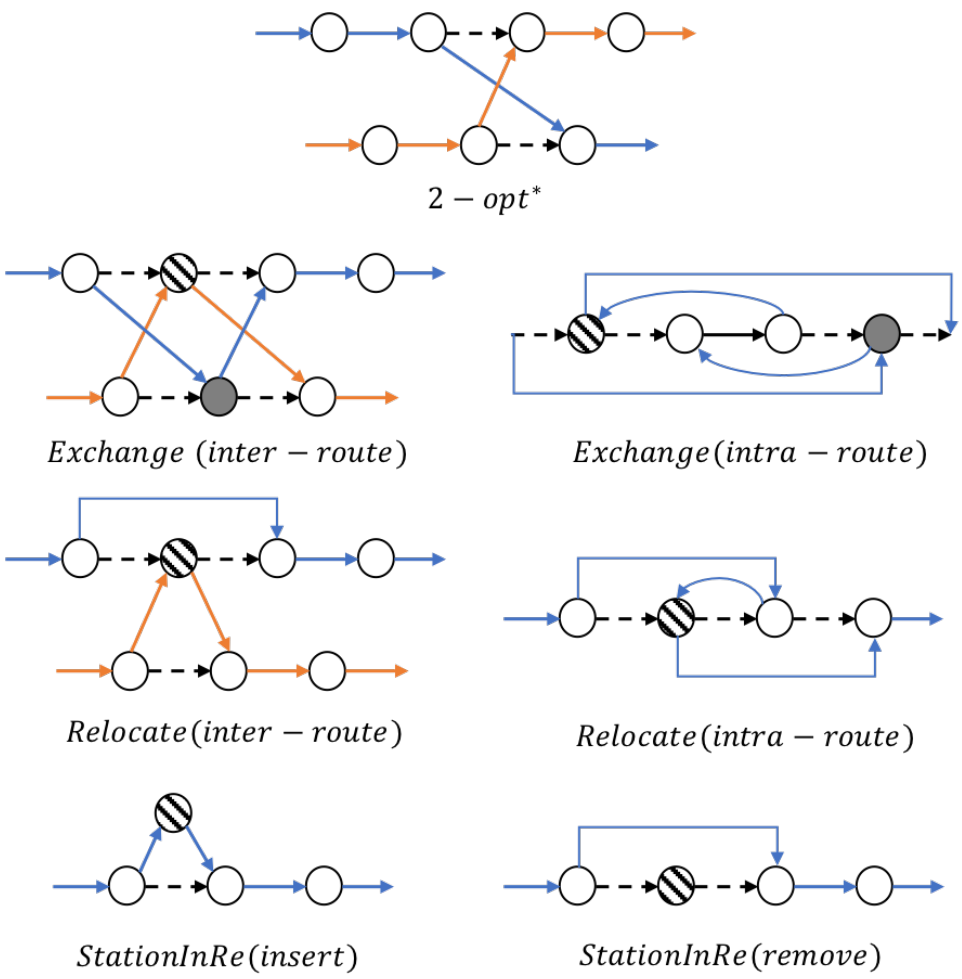

Figure 5.2: Tabu Search Operators

5.5 Generalized Cost Function

The generalized cost function, which is used to evaluate the solution quality, is defined as follows:

$$f_{gen}(S) = f_{elec}(S) + \beta_{tw}\Phi_{tw}(S) + \beta_{batt}\Phi_{batt}(S) + \beta_{cargo}\Phi_{cargo}(S) \tag{5.1}$$

Where S is a solution consisting of K routes $(R^1, R^2, \ldots, R^K)$; Each route can be represented by a sequence of nodes with the start and end being the two depot nodes

respectively; $f_{elec}(S)$ is the net electricity related cost (mentioned as electricity cost for simplicity) of the routes, i.e. charging cost minus discharging reward; $\Phi_{tw}(S)$, $\Phi_{batt}(S)$ and $\Phi_{cargo}(S)$ are the violations of the time window, battery and cargo constraints, β_{tw}, β_{batt} and β_{cargo} are their corresponding penalty factors.

5.5.1 Violation Evaluation

In order to evaluate the violations and the electricity cost, we define the following variables in Equations 5.2 - 5.6 for each node along a given route R of length n. Since the generalization cost of each route can be evaluated separately, we ignore the index differentiating the routes for different EVs. We use r_i to denote the i^{th} node along route R. We write the travelling time t_{ij} and energy consumption f_{ij} along edge (i, j) in the forms of $t[i, j]$ and $f[i, j]$ respectively for clear presentation.

$$T_i^E = \begin{cases} 0, \quad i = 1 \\ \max\left\{\min\left(T_{i-1}^E, l_{r_{i-1}}\right) + s_{r_{i-1}} + t[r_{i-1}, r_i], e_{r_i}\right\}, \quad \forall i = 2, 3, \ldots, n \end{cases} \tag{5.2}$$

$$T_i^L = \begin{cases} \min\left\{T_{i+1}^L - t\left[r_i, r_{i+1}\right], l_{r_i} + s_{r_i}\right\}, \quad \forall i = 1, 2, \ldots, n-1 \\ |T|\delta, \quad i = n \end{cases} \tag{5.3}$$

$$T_i^{FS} = \begin{cases} 0, \quad i = 0 \\ T_{i-1}^{FS} + \max\left\{e_{r_i} - \left(T_{i-1}^E + s_{r_{i-1}} + t[r_{i-1}, r_i]\right), 0\right\}, \quad \text{if } r_{i-1} \in V_c \\ \max\left\{e_{r_i} - \left(T_{i-1}^E + s_{r_{i-1}} + t\left[r_{i-1}, r_i\right]\right), 0\right\}, \quad \text{otherwise} \end{cases} \tag{5.4}$$

$$T_i^{BS} = \begin{cases} 0, \quad i = n \\ T_{i+1}^{BS} + \max\left\{T_{i+1}^L - t[r_i, r_{i+1}] - l_{r_i} - s_{r_i}, 0\right\}, \quad \text{if } r_{i+1} \in V_c \\ \max\left\{T_{i+1}^L - t[r_i, r_{i+1}] - l_{r_i} - s_{r_i}, 0\right\}, \quad \text{otherwise} \end{cases} \tag{5.5}$$

$$F_i = \begin{cases} 0, \quad \text{if } i = 0 \\ F_{i-1} + f\left[r_{i-1}, r_i\right], \text{ if } r_{i-1} \in V_c \\ f\left[r_{i-1}, r_i\right], \text{otherwise} \end{cases} \tag{5.6}$$

The value of T_i^E is the earliest service start time at node r_i without violating any time window constraints before it. Similarly, T_i^L is the latest departure time from r_i that will not result in any time window violations after it. Note that, for the calculation of T_i^E, we assume the arrival time at previous nodes is no later than their corresponding latest service start time for better evaluation of time window violations which is explained in detail later; T_i^{FS} is the forward cumulative slack time, i.e. the difference between earliest arrival time and earliest service start time, from the last station/depot to r_i; T_i^{BS} is the backward cumulative slack time, i.e. the difference between the latest departure time and latest end service time, from r_i to the first station/depot afterwards; F_i is the energy consumption (in time of recharge) from the last station/depot to r_i.

With these variables, we evaluate the violations and electricity cost as follows:

Cargo Capacity Violation

The cargo capacity violations of a route R and a solution S are computed as in Equations 5.7 and 5.8 respectively.

$$\Phi_{cargo}(R) = \max\left\{\sum_{i=1}^{n} q_{r_i} - Q, 0\right\} \tag{5.7}$$

$$\Phi_{cargo}(S) = \sum_{R \in S} \Phi_{cargo}(R) \tag{5.8}$$

Time Window Violation

We calculate the time window violation for route R as in Equation 5.9. As mentioned, we assume the arrival time at nodes before r_i is no later than their latest service start time for the calculation of time window violation at node r_i. In doing so, we can avoid penalizing a good customer sequence only because they occur after a time window violation [103]. The time window violation for a solution S is computed as in Equation 5.10.

$$\Phi_{tw}(R) = \sum_{i=1}^{n} \max\left\{T_i^E - l_{r_i}, 0\right\} \tag{5.9}$$

$$\Phi_{tw}(S) = \sum_{R \in S} \Phi_{tw}(R) \tag{5.10}$$

Battery Capacity Violation

The battery violations for a route R and a solution S are defined in Equations 5.11 and 5.12, respectively.

$$\Phi_{batt}(R) = \sum_{r \in R \cap V_s} \max\{F_r - B, 0\} \tag{5.11}$$

$$\Phi_{batt}(S) = \sum_{R \in S} \Phi_{batt}(R) \tag{5.12}$$

5.5.2 Net Electricity Related Cost

In order to calculate the electricity cost of a given route, we need to determine the optimal charging/discharging schedule associated with the route. To this end, we first infer the possible charging/discharging time at each station and depot node along the route. And then, since the schedule at a station/depot node can affect the possible staying time at its subsequent nodes thus influence the subsequent schedules, we infer the interactions of the schedules at different station and depot nodes. Finally, we construct an optimization model based on the extracted information along with other problem constraints and solve it to determine the final schedule.

The staying time at node $r_i \in R$ can be easily represented as $\left[T_i^E, T_i^L\right]$. However, under the discretized charging/discharging assumption, an EV can not start charging/discharging at r_i until time $\delta\lceil\frac{T_i^E}{\delta}\rceil$, and it is not allowed to charge/discharge at node r_i after time $\delta\lfloor\frac{T_i^L}{\delta}\rfloor$. Therefore, the possible charging/discharging time at node $r_i \in R$ is $\left[\delta\lceil\frac{T_i^E}{\delta}\rceil, \delta\lfloor\frac{T_i^L}{\delta}\rfloor\right]$. Accordingly, we define the set of **connected periods** at r_i as $T_i = \left\{\lceil\frac{T_i^E}{\delta}\rceil, \lceil\frac{T_i^E}{\delta}\rceil + 1, \ldots, \lfloor\frac{T_i^L}{\delta}\rfloor\right\}$.

To model the interaction between consecutive station/depot nodes, we define the mutual exclusiveness of charging/discharging periods at different nodes. Suppose r_i and r_j ($i < j$) are two consecutive station nodes along route R, their connected periods are $T_i = \{t_i^1, t_i^2, \ldots, t_i^{m_i}\}$ and $T_j = \left\{t_j^1, t_j^2, \ldots, t_j^{m_j}\right\}$ respectively. If the charging/discharging during period $t_i^p \in T_i$ at node r_i will makes it impossible for the EV to arrive at node r_j before the start of period $r_j^q \in T_j$, we say periods t_i^p for node r_i and t_j^q for node r_j are **mutually exclusive**. All such t_j^q form the **mutually exclusive set** for t_i^p at node r_i, namely M_i^p.

More specifically, we use the forward slack time T_i^{FS} to infer the mutually exclusive set for $t_i^p \in T_i$ as in Equation 5.13. Recall the definition of the forward slack time, we claim

that as long as the EV leaves node r_i before $T_i^E + T_j^{FS}$, the schedule at r_j will not be affected by the schedule at r_i. Hence, the mutually exclusive set is empty. On the other hand, if the EV charges/discharges during period t_i^p such that $\delta t_i^p > T_i^E + T_j^{FS}$, the arrival time at node r_j will be postponed by $\delta t_i^p - \left(T_i^E + T_j^{FS}\right)$. Consequently, charging/discharging at r_j during periods that starts before $T_j^E + \delta t_i^p - \left(T_i^E + T_j^{FS}\right)$ will be impossible.

$$M_i^p = \begin{cases} \emptyset, \text{ if } \delta t_i^p \leq T_i^E + T_j^{FS} \\ \left\{ t_j^q : \delta(t_j^q - 1) < T_j^E + \delta t_i^p - \left(T_i^E + T_j^{FS}\right) \right\}, \text{ otherwise} \end{cases} \tag{5.13}$$

As simultaneously determine charge/discharging schedule at multiple nodes is very complicated, we make the following assumptions which might sacrifice solution quality but drastically enhance solution efficiency.

- **Limited Station Visits:** we assume there are at most two station nodes along a route. This assumption is realistic because the main focus of commercial EV fleets is satisfying logistic requests, whereas ancillary service provision is just a sideline. Visiting stations over twice per day might affect an EV's regular businesses. Moreover, frequent charging/discharging can accelerate battery degradation which will result in high battery replacement costs later.

- **Simplified Schedule:** each EV is allowed to perform either discharge or charge at en-route station(s), but is prohibited to do both during a single visit. In addition, if an EV visits stations twice and requires en-route charging, we only consider performing minimal charging ensuring the EV can successfully return to the depot. This assumption excludes the possibility of over-charging at one station during off-peak hours and discharging at another station later, but can largely accelerate the solving process. Further, as mentioned, this assumption assists to avoid frequent charging/discharging, which is beneficial to battery lifespan.

- **Hierarchical Decision Making:** for the case of two en-route station visits, instead of jointly optimizing en-route and at-depot schedule, we solve these two problems sequentially in the spirit of divide-and-conquer. We first schedule en-route charging/discharging and then determine at-depot schedule based on it. We calculate the cost of the routes as the sum of en-route and at-depot costs as shown in Equation 5.14. Under this assumption, the problem complexity is largely reduced, and we are still able to find solutions of high quality according to preliminary manual investigations.

$$f_{elec}(r) = f_{depot}(r) + f_{en-route}(r) \tag{5.14}$$

We define the electricity related cost of a solution S in Equation 5.15.

$$f_{elec}(S) = \sum_{R \in S} f_{elec}(R) \tag{5.15}$$

In particular, for each of the routes, we consider the following three cases with respect to the number of station nodes along the route.

No Station Node

If there is no station node along R, $f_{en-route}(R) = 0$, we only consider the at-depot schedule. Since the EV's battery is fully charged at the beginning of the planning horizon and the night charging price is the cheapest throughout a day, it is unreasonable to charge the EV at the depot during the planning horizon. The pseudo code for scheduling at-depot discharge is presented in Algorithm 4.

As mentioned earlier, given a route R of length n, we can infer the connected periods at the two depot nodes (the first and the last node along R) $T_1 = \{t_1^1, \ldots, t_1^{m_1}\}$ and $T_n = \{t_n^1, \ldots, t_n^{m_n}\}$. For each period $t_1^i \in T_1$, we infer its mutually exclusive set M_1^i. Moreover, we calculate the total number of periods we can discharge without violating battery constraints $\Omega = \lfloor \frac{B - F_n}{\delta} \rfloor$.

We then construct the following MIP problem named NoStat:

$$\Upsilon^* = \underset{d}{Max} \sum_{t \in T_1} P_{dis}^t d_1^t + \sum_{k \in T_n} P_{dis}^k d_n^k \tag{5.16}$$

$$\text{s.t.} \quad \sum_{t \in T_1} d_1^t + \sum_{k \in T_n} d_n^k \leq \Omega \tag{5.17}$$

$$d_1^t + d_n^k \leq 1, \quad \forall k \in M_1^t, \forall t \in T_1 \tag{5.18}$$

Where $d_1^t, \forall t \in T_1$ and $d_n^k, \forall k \in T_n$ are binary decision variables indicating if the EV discharges during the associated time period at the two depot nodes ($= 1$) or not ($= 0$) respectively. P_{dis}^t is the discharging reward rate at time period t. The objective function 5.16 seeks to maximize the discharging reward, Constraint 5.17 makes sure that the battery

level is always positive, while Constraints 5.18 serve to describe the mutual exclusiveness among periods.

Let d represents a vector of all decision variables in NoStat Problem. Since $d = 0$ is a feasible solution to the NoStat Problem and the feasible region of the problem is bounded, there must exist an optimal solution, namely d^*. We denote the corresponding optimal value of the objective function as Υ^*. The route's electricity cost is then calculated as the sum of discharge reward and the cost for fully recharge battery during the night as in Equation 5.19 where $sum(d^*)$ means the summation of all entries in d^*.

$$f_{elec}(R) = -\Upsilon^* + (\frac{F_n}{\delta} + sum(d^*)) \times P_{re}^{night} \tag{5.19}$$

Algorithm 4 Discharging Schedule for a Route with No Station Node

```
T_1, T_n ← ConnectedPeriod(T_1^E, T_1^L, T_n^E, T_n^L)
Ω ← ⌊(B − F_n)/δ⌋
if Ω > 0 then
    for t_1^i in T_1 do
        infer mutually exclusive set M_1^i ⊂ T_n
    Υ*, d* ← NoStat()
    f_elec(R) = −Υ* + (F_n/δ + sum(d*)) × P_re^night
else
    f_elec(R) = (F_n/δ) × P_re^night
```

Solving the NoStat problem with traditional MIP solvers requires exponential time. In order to accelerate the solving process, we reformulate the problem as a 0-1 knapsack problem with mutually exclusive items (KPMEI). We regard each discharging period as an item with the weight of 1 and the value of the associated discharging reward. Given a knapsack of capacity Ω, we seek to fill the knapsack with a subset of the items such that the total weight does not exceed the knapsack capacity and the total value is maximized. According to the mutual exclusiveness described by Equations 5.18, some pairs of items can not be included in the collection simultaneously.

As shown in Algorithm 5, we decompose this problem into a sequence of classical 0-1 knapsack problems, i.e. 0-1 knapsack problem without mutually exclusive items, by enumerating the last period that the EV could performs discharge at node r_1. In particular, suppose the last period at node r_1 is t_1^l, the possible discharging period at node r_n is $T_n \setminus M_1^l$ (it is not hard to find that the mutually exclusive sets for periods before t_1^l at node r_1 are

subsets of M_1^l). Then, the problem become selecting Ω items from $\{t_1^1, \ldots, t_1^l\} \cup \left(T_n \setminus M_1^l\right)$, which can be solved in polynomial time using the dynamic programming approach proposed in [116]. We solve the classical knapsack problem for at most $|T_1|$ times, then select the highest value achieved as the optima of the original problem. In this way, the KPMEI can be solved in very efficiently. We denote the highest reward achieved through solving the classical 0-1 knapsack problems and the set of periods that the EV discharges with Υ^* and dis^* respectively, the electricity cost can be computed as in Equation 5.20.

$$f_{depot}(R) = -\Upsilon^* + (\frac{F_n}{\delta} + |dis^*|) \times P_{rch}^{night} \tag{5.20}$$

Algorithm 5 Knapsack Approach for No Station Visit

$\Upsilon^*, dis^* \leftarrow Knapsack(T_n, \Omega)$ (No item selected from T_1)
for t_1^l in T_1 **do**
    $T^{'} \leftarrow \{t_1^1, \ldots, t_1^l\} \cup \left(T_n \setminus M_1^l\right)$
    $val, dis \leftarrow Knapsack(T^{'}, \Omega)$
    if $val > \Upsilon^*$ **then**
        $\Upsilon^*, dis^* \leftarrow val, dis$
$f_{depot}(R) = -\Upsilon^* + (\frac{F_n}{\delta} + |dis^*|) \times P_{rch}^{night}$

Here we provide an example. Suppose $T_0^E = 0$, $T_1^L = 122$, $T_n^E = 1070$, $T_n^L = 1140$, $T_n^{FS} = 100$, $F_n = 130$, and battery capacity $B = 270$, the number of the planning periods $|T| = 19$, length of each period $\delta = 60$, the planning horizon is $[0, 1140]$.

The EV can discharge the battery for $\Omega = \lfloor \frac{B - f_n}{\delta} \rfloor = 2$ periods. As $\lfloor \frac{T_1^L}{\delta} \rfloor = 2$, the connected period at node r_1 is $T_1 = \{1, 2\}$. Similarly, as $\lceil \frac{T_n^L}{\delta} \rceil = 18$, the connected period at node r_n is $T_n = \{18, 19\}$. For period 1 in T_1, since $\delta < T_n^{FS}$, $M_1^1 = \emptyset$. If we discharge during period 2 at node r_1, the earliest arrival time at node r_n will be postponed by $2\delta - T_n^{FS} = 20$. Given that $(18-1)\delta < T_n^E + 20 = 1090$ and $(19-1)\delta > 1090$, the mutually exclusive set for period 2 at node r_1 is $M_1^2 = \{18\}$. We then construct the following NoStat Problem:

$$\begin{aligned} \Upsilon^* = & \max_d P_{dis}^1 d_1^1 + P_{dis}^2 d_1^2 + P_{dis}^{18} d_n^{18} + P_{dis}^{19} d_2^{19} \\ \text{s.t.} \quad & d_1^1 + d_1^2 + d_n^{18} + d_n^{19} \leq 2 \\ & d_1^2 + d_n^{18} \leq 1 \end{aligned} \tag{5.21}$$

The schedule can be easily determined by solving the following two sub-problems.

- If period 2 is the last period at r_1, select 2 periods from $\{1, 2, 19\}$
- If period 2 is not last period at r_1, select 2 periods from $\{1, 18, 19\}$

One Station Node

The pseudo code for the case of one station node is shown in Algorithm 6.

Algorithm 6 Charge/Discharge Schedule for a Route with One Station Node

$k \leftarrow$ the position of the station in the route, i.e. $r_k \in V_s$
$\Delta \leftarrow F_k + F_n - B$
$T_1, T_k, T_n \leftarrow$ ConnectedPeriod$(T_0^E, T_1^L, T_k^E, T_k^L, T_n^E, T_n^L)$
Check mutually exclusive periods for $\forall t_1^i \in T_1$ and $\forall t_k^j \in T_k$
$\Upsilon^*, d^*, r^* \leftarrow$ SolveMIP(Δ)
if feasible **then**
    $f_{elec}(r) \leftarrow \Upsilon^* + (\frac{F_k+F_n}{\delta} + sum(d^*) - sum(r^*)) \times P_{re}^{night}$
else
    $f_{elec}(r) \leftarrow LargeNumber$

Suppose the station is the k^{th} node along the route R. We first calculate $\Delta = F_k + F_n - B$, and Ω as in Equation 5.22. If $\Delta > 0$, we have to recharge at the station to make sure the EV have enough energy to complete the trip, otherwise we can either discharge or charge at the station. Then, similar to the analysis done for "No Station Visit" we infer the connected periods at the two depots node and the station, i.e. T_1, T_n and T_k. In addition, for each period $t_1^i \in T_1$ and $t_k^j \in T_k$ we infer their mutually exclusive sets M_1^i at node r_k and M_k^j at node r_n respectively.

$$\Omega = \begin{cases} \lceil \frac{\Delta}{\delta} \rceil, \text{ if } \Delta \geq 0 \\ \lfloor \frac{-\Delta}{\delta} \rfloor, \text{ otherwise} \end{cases} \tag{5.22}$$

We then construct the OneStat Problem with respect to the value of Δ.

If $\Delta < 0$, the problem is defined by Equations 5.23 - 5.30 where $d's$, $r's$ and y are binary decision variables. d_j^i indicates if the EV discharges at node r_j during period i $(= 1)$ or not $(= 0)$. $r's$ are defined in a similar way yet for recharging. y takes a value of 1 if the EV discharges at node r_k and takes 0 otherwise.

$$\Upsilon^* = \min_{d,r} \sum_{j \in T_k} P^j_{re} r^j_k - \sum_{i \in T_1} P^i_{dis} d^i_1 - \sum_{j \in T_k} P^j_{dis} d^j_k - \sum_{t \in T_n} P^t_{dis} d^t_n \tag{5.23}$$

$$s.t. \quad \sum_{i \in T_1} d^i_1 + \sum_{j \in T_k} d^j_k + \sum_{t \in T_n} d^t_n - \sum_{j \in T_k} r^j_k \leq \Omega \tag{5.24}$$

$$\delta \sum_{j \in T_k} r^j_k \leq F_k + \delta \sum_{i \in T_1} d^i_1 \tag{5.25}$$

$$\delta \left(\sum_{i \in T_1} d^i_1 + \sum_{j \in T_k} d^j_k \right) + F_k + F_n \leq B + \delta \sum_{j \in T_k} r^j_k \tag{5.26}$$

$$\sum_{j \in T_k} d^j_k \leq |T_k| y \tag{5.27}$$

$$\sum_{j \in T_k} r^j_k \leq |T_k|(1 - y) \tag{5.28}$$

$$d^i_1 + r^j_k + d^j_k \leq 1, \quad \forall j \in M^i_1, \forall i \in T_1 \tag{5.29}$$

$$r^j_k + d^j_k + d^t_n \leq 1, \quad \forall t \in M^j_k, \forall j \in T_k \tag{5.30}$$

The objective function 5.23 minimizes the electricity cost during the planning horizon. Constraints 5.24 and 5.25 serve to make sure that the battery level will not be below zero or exceed its capacity. Constraints 5.26 guarantee the EV has enough energy to complete the trip. Constraints 5.27 and 5.28 ensure the EV will not perform both charge and discharge during a single station visit. Constraints 5.29 and 5.30 describe the mutual exclusiveness among periods.

If $\Delta > 0$, we replace Constraint 5.24 by Constraint 5.31 to make sure the EV have enough energy to complete the trip. In addition, we force $d^j_k = 0, \forall j \in T_k$ because en-route recharging is necessary in this case. We note that for a similar reason described in the case of "No Station Visit", the optimization model is feasible when $\Delta < 0$. When $\Delta > 0$, it is possible for the OneStat Problem to be infeasible. If so, we assign a large cost to this route as a penalty. Otherwise, the route's cost is computed as in Equation 5.32 where Υ^* is the minima of the OneStat Problem, d^* (a vector of decision variables $d's$) and r^* (a vector of

decision variables $r's$) are optimal solution to the problem, sum() is a function intakes a vector and returns the summation of all of its entries.

$$\sum_{j \in T_k} r_k^j - (\sum_{i \in T_1} d_1^i + \sum_{j \in T_k} d_k^j + \sum_{t \in T_n} d_n^t) = \Omega \tag{5.31}$$

$$f_{elec}(R) = \Upsilon^* + (\frac{F_k + F_n}{\delta} + sum(d^*) - sum(r^*)) \times P_{re}^{night} \tag{5.32}$$

Two Station Nodes

The pseudo code for the case of two station nodes are shown in Algorithm 7.

Algorithm 7 Charge/Discharge Schedule for a Route with Two Station Nodes

$k_1, k_2 \leftarrow$ the positions of the en-route stations, i.e. $r_{k_1}, r_{k_2} \in V_s$
$\Delta_1 \leftarrow F_{k_1} + F_{k_2} + F_n - B$
Calculate Ω based on Δ_1
$T_{k_1}, T_{k_2} \leftarrow$ ConnectedPeriod$(T_{k_1}^E, T_{k_1}^L, T_{k_2}^E, T_{k_2}^L)$
Infer mutually exclusive sets
if $\Delta_1 \geq 0$ **then**
    $\Upsilon^*, r^* \leftarrow SolveMIP(\Delta_1)$
    $f_{elec}(R) = \Upsilon^* + \left(\frac{F_{k_1}+F_{k_2}+F_{k_n}}{\delta} - sum(r^*)\right) \times P_{re}^{night}$
else if $\Delta_1 < 0$ **then**
    $\Upsilon^*, d^* \leftarrow SolveMIP(\Delta_1)$
    $\Delta_2 \leftarrow \Omega - sum(d^*)$
    Update $\hat{T_l^0}$, $\hat{T_e^n}$ and $t_{sf}^{\hat{n}}$
    $\Upsilon^{**}, d^{**} \leftarrow$ AtDepotSchedule(Δ_2)
    $f_{elec}(R) = \Upsilon^* + \Upsilon^{**} + \left(\frac{F_{k_1}+F_{k_2}+F_{k_n}}{\delta} + sum(d^*) + sum(d^{**})\right) \times P_{re}^{night}$

Suppose the positions of the two station visits on the route are k_1 and k_2. We first calculate $\Delta_1 = F_{k_1} + F_{k_2} + F_n - B$ and Ω as in Equation 5.22. If $\Delta_1 > 0$, the EV has to recharge at en-route stations, otherwise we schedule discharge. Then, similar to what we have done for the previous two cases, we infer the connected periods at the two stations, T_{k_1} and T_{k_2}, and check if these periods are mutually exclusive or not.

We then construct the TwoStat Problem with respect to the value of Δ_1.

If $\Delta_1 > 0$, the TwoStat problem is defined by Equations 5.33 - 5.37 where r^i_j are binary decision variable specifying if the EV charges its battery at node r_{k_j} during the period i $(= 1)$ or not $(= 1)$. We note that, under the simplified schedule assumption, we only consider performing minimum charge at the two stations such that the EV has enough energy to complete the trip. No discharge will be scheduled at any station/depot nodes.

$$\Upsilon^* = \min_r \sum_{i \in T_{k_1}} P^i_{re} r^i_{k_1} + \sum_{j \in T_{k_2}} P^j_{re} r^j_{k_2} \tag{5.33}$$

$$s.t. \quad \sum_{i \in T_{k_1}} r^i_{k_1} + \sum_{j \in T_{k_2}} r^j_{k_2} = \Omega \tag{5.34}$$

$$F_{k_1} + F_{k_2} - B \leq \delta \sum_{i \in T_{k_1}} r^i_{k_1} \leq F_{k_1} \tag{5.35}$$

$$F_{k_2} + F_n - B \leq \delta \sum_{j \in T_{k_2}} r^j_{k_2} \leq F_{k_1} + F_{k_2} - \delta \sum_{i \in T_{k_1}} r^i_{k_1} \tag{5.36}$$

$$r^i_{k_1} + r^j_{k_2} \leq 1, \quad \forall j \in M^i_{k_1}, \forall i \in T_{k_1} \tag{5.37}$$

The objective function 5.33 seeks to minimize the en-route electricity cost. Constraints 5.34 make sure the EV has enough energy to complete the trip. Constraints 5.35 ensure that the amount of energy we recharge at the first station won't let the battery level exceeds its capacity and will allow the EV to reach the next station. Constraints 5.36 serve in the same way for the second station. Constraints 5.37 describe if the periods at the two stations are mutually exclusive or not.

When $\Delta_1 > 0$, it is possible that the OneStat Problem is infeasible. In this case, we assign a large number to the route's cost as a penalty. Otherwise, suppose r^* is the optimal solution, we calculate the route's cost as in Equation 5.38.

$$f_{elec}(R) = \Upsilon^* + \left(\frac{F_{k_1} + F_{k_2} + F_n}{\delta} - sum(r^*) \right) \times P^{night}_{re} \tag{5.38}$$

Similar to the analysis done for the case of "No Station Node", the TwoStat Problem defined above can be regarded as a variant of the classical 0-1 knapsack problem as well. Since the knapsack problem is usually constructed in a maximization manner, we re-write

the objective function 5.33 as Equation 5.39 where $\hat{P^i_{re}} = 100 - P^i_{re} > 0$. Then the problem defined by Equations 5.39 and 5.34 - 5.37 can be regarded as selecting Ω items from sets T_{k_1} and T_{k_2}, each associated with a weight of 1 and a value of $\hat{P^t_{re}}, \forall t \in T_{k_1} \cup T_{k_2}$, so as to maximize the total value selected. Among them, at least $L_{k_1} = \lceil \frac{F_{k_1}+F_{k_2}-B}{\delta} \rceil$and at most $U_{k_1} = \lfloor \frac{F_{k_1}}{\delta} \rfloor$ items should be from T_{k_1}, at least $L_{k_1} = \lceil \frac{F_{k_2}+F_n-B}{\delta} \rceil$ items should be from T_{k_2}. Moreover, some items in the two sets are mutually exclusive.

$$\hat{\Upsilon} = \max_r \sum_{i \in T_{k_1}} \hat{P^i_{re}} r^i_{k_1} + \sum_{j \in T_{k_2}} \hat{P^j_{re}} r^j_{k_2} \tag{5.39}$$

The pseudo code for solving this problem is presented in Algorithm 8. Again, we decompose the problem into a sequence of sub-problems by enumerating the last period the EV can charge at station r_{k_1}.

Algorithm 8 Knapsack Approach for Two Station Visit (Recharge)

$\hat{\Upsilon}, re^* \leftarrow 0, \emptyset$
$L_{k_1}, L_{k_2}, U_{k_1} \leftarrow \lceil \frac{F_{k_1}+F_{k_2}-B}{\delta} \rceil, \lceil \frac{F_{k_2}+F_n-B}{\delta} \rceil, \lfloor \frac{F_{k_1}}{\delta} \rfloor$
for $t^l_{k_1}$ in T_{k_1}, **do**
    infer the mutually exclusive set at r_{k_2}, i.e. $M^l_{k_1}$
    $T^{'}_{k_1}, T^{'}_{k_2} \leftarrow \{t^1_{k_1}, \ldots, t^l_{k_1}\}, T_{k_2} \setminus M^l_{k_1}$
    if $|T^{'}_{k_1}| \geq L_{k_1}$ & $|T^{'}_{k_2}| \geq \max(L_{k_2}, \Omega - U_{k_1})$ **then**
        $val_1, re_1 \leftarrow Knapsack(T^{'}_{k_1}, L_{k_1})$
        $val_2, re_2 \leftarrow Knapsack\left(T^{'}_{k_2}, \max(L_{k_2}, \Omega - U_{k_1})\right)$
        $val_3, re_3 \leftarrow Knapsack\left((T^{'}_{k_1} \setminus re_1) \cup (T^{'}_{k_2} \setminus re_2), \Omega - |re_1| - |re_2|\right)$
        if $val_1 + val_2 + val_3 > \hat{\Upsilon}^*$ & $|re_1| + |re_2| + |re_3| = \Omega$ **then**
            $\hat{\Upsilon}^*, re^* \leftarrow val_1 + val_2 + val_3, re_1 \cup re_2 \cup re_3$
if $\hat{\Upsilon} > 0$ **then**
    $\Upsilon^* \leftarrow 100|re^*| - \hat{\Upsilon}$
    $f_{elec}(R) = \Upsilon^* + \left(\frac{F_{k_1}+F_{k_2}+F_n}{\delta} - |re^*|\right) \times P^{night}_{re}$
else
    $f_{elec}(R) \leftarrow LargeNumber$

In each iteration, we first infer the possible charging period at nodes r_{k_1} and r_{k_2}, i.e. $T^{'}_{k_1}$ and $T^{'}_{k_2}$, based on the mutually exclusive relationship. If $|T^{'}_{k_1}| < L_{k_1}$ we will not be able to select L_{k_1} items from T_{k_1}, which renders the problem infeasible. For a similar reason, the cardinality of set $T^{'}_{k_2}$ should be greater than $\max(L_{k_2}, \Omega - U_{k_1})$. And then, we select

L_{k_1} and $\max(L_{k_2}, \Omega - U_{k_1})$ items from T'_{k_1} and T'_{k_2} respectively by solving two classical knapsack problems. After that, we solve another classical knapsack problem to select $\Omega - |re_1| - |re_2|$ from the remaining periods where re_1 and re_2 are the sets of periods we selected from T'_{k_1} and T'_{k_2} in the previous steps respectively. We note that, when applying the dynamic programming approach to solve the knapsack problem, the algorithm might select less than the required number of items. Hence, we have to check if exactly Ω items are selected before we update the best schedule. Finally, after completing the search, the set re, if not empty, is the optimal schedule for the EV, and we calculate the optimal value of the original problem as $\Upsilon^* \leftarrow +100|re^*| - \hat{\Upsilon}^*$. The electricity cost of the route is calculated as in Equation 5.40.

$$f_{elec}(R) = \Upsilon^* + \left(\frac{F_{k_1} + F_{k_2} + F_n}{\delta} - |re^*| \right) \times P_{re}^{night} \tag{5.40}$$

If $\Delta_1 < 0$, we only consider discharging at the two station nodes and the two depot nodes. Using the hierarchical decision making assumption, we first determine the schedule at stations based on which we set the at-depot schedule later. The TwoStat problem for at-station schedule is defined as in Equations 5.41 - 5.43 where d_i^j here are decision variables suggesting if the EV discharges its battery during period j at node r_i $(= 1)$ or not $(= 0)$. The objective function 5.41 seeks to maximize the total discharging reward. Constraints 5.42 guarantee the battery level of the EV will not go below zero while the last set of Constraints 5.43 describe the mutual exclusiveness among periods.

$$\Upsilon^* = \underset{d}{Max} \sum_{i \in T_{k_1}} P_{dis}^i d_{k_1}^i + \sum_{j \in T_{k_2}} P_{dis}^j d_{k_2}^j \tag{5.41}$$

$$s.t. \quad \sum_{i \in T_{k_1}} d_{k_1}^i + \sum_{j \in T_{k_2}} d_{k_2}^j \leq \Omega \tag{5.42}$$

$$d_{k_1}^i + d_{k_2}^j \leq 1, \quad \forall j \in M_{k_1}^i, \forall i \in T_{k_1} \tag{5.43}$$

It is obvious that the TwoStat Problem here has the same structure as the NoStat Problem, thus can be solved efficiently using Algorithm 5. Suppose dis_1^* and dis_2^* are the at-station schedules at nodes r_{k_1} and r_{k_2} we obtain through solving the problem, it is possible that $|dis_1^*| + |dis_2^*| < \Omega$, suggesting that the EV could make additional profits by discharging remaining energy at the depot. We then set the at-depot schedule by solving the NoStat Problem again, yet some adjustments should be made. We assume the two

stations along the route are all customers with 0 demand, and update the earliest arrival time at node r_n, latest departure time at node r_1, and forward slack time at node r_n as in Equations 5.44 - 5.46 respectively.

$$\hat{T_n^E} = \begin{cases} T_n^E + \max\left[\max(dis_2^*)\delta - T_{k_2}^E - T_n^{FS}, 0\right], & \text{if } |dis_2^*| > 0 \\ T_n^E + \max\left[\max(dis_1^*)\delta - T_{k_1}^E - T_{k_2}^{FS} - T_n^{FS}, 0\right], & \text{otherwise} \end{cases} \tag{5.44}$$

$$\hat{T_0^L} = \begin{cases} T_0^L - \max\left[T_{k_1}^L - (\min(dis_1^*) - 1)\delta - T_0^{BS}, 0\right], & \text{if } |dis_1^*| > 0 \\ T_0^L - \max\left[T_{k_2}^L - (\min(dis_2^*) - 1)\delta - t_0^{BS} - t_{k_1}^{BS}, 0\right], & \text{otherwise} \end{cases} \tag{5.45}$$

$$\hat{T_n^{FS}} = t_{k_1}^{FS} + t_{k_2}^{FS} + t_{k_n}^{FS} - \max\left[0, \max(dis_1^*)\delta - T_{k_1}^E\right] - \max\left[0, \max(dis_2^*)\delta - T_{k_2}^E\right] \tag{5.46}$$

where $\max(dis_i^*)$, and $\min(dis_i^*), i \in \{1, 2\}$ are the latest and earliest periods in set dis_i^*. If $dis_i^* = \emptyset$, we set $\max(dis_i^*) = \min(dis_i^*) = 0$. We then use the updated variables to solve the NoStat Problem. We denote the optimal at-depot schedule as dis^{**} and its cost as Υ'. Then, we calculate the cost for route R as in Equation 5.47.

$$f_{elec}(R) = \Upsilon^* + \Upsilon' + \left(\frac{F_{k_1} + F_{k_2} + F_n}{\delta} + |dis_1^*| + |dis_2^*| + |dis^{**}|\right) \times P_{re}^{night} \tag{5.47}$$

Chapter 6

Computational Results

In this chapter, we evaluate the lower and upper bounds generated using the algorithms we developed in the previous two chapters.

6.1 Test Instances

As the EVRPTW-TP has not been investigated by previous research, there is no benchmark data set readily available. Therefore, we construct test instances based on the ones developed by [103] for the EVRPTW. The EVRPTW benchmark data set was constructed based on the instances proposed by [105], it consists of one set of large instances, each with 100 customers and 21 stations, and one set of small instances, each with 5, 10, or 15 customers and no more than 6 stations. Given the complexity of the EVRPTW-TP, we only consider the small instances in this research. In the subsequent discussion, we refer to the small instances in EVRPTW data set as the Schneider instances for simplicity.

The Schneider instances can be classified into 3 categories by geographical distribution of the customer nodes. In the first column of Table 6.2, the instances start with "R" are random instances where customers are uniformly distributed, those start with "C" are clustered instances in which customers are clustered into small groups, the customer distribution of the "RC" instances is the mixture of random and clustered distributions. The locations of the stations are randomly selected by [103]. We directly use the geographical coordinates and demand information provided by the Schneider instances and calculate distances, travel time and energy consumption among nodes based on them. EV specifications also come from the Schneider instances. The EV fleet is homogeneous, each with a

cargo capacity $Q = 200$, a battery capacity $B = 270$, and a constant energy consumption rate $g = 1$. EVs' travelling speed v is set to a constant value of 1, and EVs are charged at a constant speed such that $\alpha = 3.39$. For the sake of simplicity, we assume the discharging speed is the same as the charging speed.

For time-related parameters, adjustments are made to fit the EVRPTW-TP settings. Since fully charging an EV requires $B = 270$ minutes (4.5 hours), we set the planning horizon as $5am - 12am$, i.e. $[0, 1140]$ in minutes, such that EV operators have enough time to recharge their fleets at night. The length of planning horizon in the Schneider instances ranges from 230 to 3390. For instances with a planning horizon of length $L > 1140$, we scale the time windows by multiplying them with a factor $\frac{L}{1140}$. Moreover, the time windows in the Schneider instances are relatively tight, and even tighter after scaling. The EVs, thus, do not have much flexibility detouring to perform "unnecessary" charge and discharge, which is what we want to incentivize the EV owners to do in this research. To this end, we relax the time windows to three periods of time: morning $(5am - 12pm)$, afternoon $(12pm - 6pm)$, and evening $(6pm - 12am)$. If a time window covers a subset of the period, then EV could visit the corresponding node during the whole period. For example, a time window $[320, 500]$ is relaxed to $[0, 780]$ because it covers subsets of both the morning ($[0, 420]$) and afternoon periods ($[420, 780]$). In addition, we set service time at each node to be 0 to further enhance the EVs' flexibility. This can be easily generalized in the use case that we will introduce in Chapter 7.

For charging and discharging, we set the length of each period as one hour, i.e. $\delta = 60$. According to the data presented in [126], less than 10% of the charging sessions at public EV stations in urban areas are shorter than 60 minutes. Hence, we assume that setting the minimum charging/discharging time to 60 minutes here will not influence the EV's operational flexibility in a significant way. The charging cost and discharging reward rates associated are based on the real time-of-use hydro rate in Ontario, Canada [13] in effect between May 1, 2019 and October 31, 2019 as shown in column "Charging Cost" in Table 6.1. The unit is cents per KWh. Since the charging power is not specified in the Schneider instances, we set the cost of charging for a whole period as the per KWh rate without a unit, which is proportional to the actual cost under the assumption of constant charging power. Similarly, we set the discharging reward rates with no unit. We note that the reward rate presented in Table 6.1 are not the actual rates as they are chosen such that the EVs could make profits by discharging at peak hours (11:00 AM - 5:00 PM) and recharge the battery later. In addition, the reward rates also economically benefit the grid because the discharging reward they pay is lower than the corresponding market price. The time-of-use prices we set here is just for evaluation purposes, a more comprehensive numerical study as well as insights about the electricity pricing scheme are presented in Chapter 7.

Table 6.1: Time-of-Use Electricity Prices

From	To	Charging Cost	Discharging Reward
12:00 AM	7:00 AM	6.5	6.5
7:00 AM	11:00 AM	9.4	8.0
11:00 AM	5:00 PM	13.4	10.0
5:00 PM	7:00 PM	9.4	8.0
7:00 PM	12:00 AM	6.5	6.5

Finally, we note that it is possible for the proposed EVRPTW-TP to be infeasible on the instances we construct. The reason is that the minimal staying time at each en-route station is 60 minutes due to the discrete charging/discharging assumption, while EVs are allowed to charge for however long time in the EVRPTW. If an instance is found to be infeasible, we increase the number of EVs by one until the instance becomes feasible. In particular, we add one EV for instances "C103-5", "C206-5", "RC108-5", "C202-10", "R102-10", "R203-10", "RC201-10" and "RC202-15".

6.2 Experimental Setting

All the tests are performed on a workstation running Ubuntu 18.04.4, equipped with 40 CPU processors clocked at 2.20 GHZ and 64 GB of RAM. For CPLEX, we use version 12.8.0.0 and we parallelize the computation in 32 threads. We set the time limit as 7200 seconds and (after-compression) memory limit of 40 GB. The time limit is set to 7200 seconds because, in practice, we need to solve this operational problem at night in order to operate the fleet in the following day. We assume that solving the problem within 2 hours allows the operator to schedule relevant issues before the start of the planning horizon. In addition, since the value of big "M" has significant impact on the performance of CPLEX, we should set it to a value that makes the constraints as tight as possible. Based on the given instances, we set the value of "M" to 1141 for all the time-related constraints, to 271 for all the battery-level-related constraints, and to 201 for all the cargo-related constraints.

With regards to the Lagrangian relaxation, we set the initial penalty factors λ's as 1.0, the width of trust region $\omega^1 = 3$ at the very beginning. We chose the tolerance $tol = 0.5$. Similarly, we parrallelize all the sub-problems in 32 threads but set no time and memory limit because sub-problems are supposed to be solved to optimality, otherwise the lower bound we obtain might be unreasonable.

For the VNS/TS hybrid heuristic, we set all the penalty parameters β_{tw}, β_{batt}, β_{cargo} to 10, the number of tabu iterations per round $\eta_{tabu} = 30$, and vary the number of VNS iterations η_{vns} and early stop criteria η_{early} with respect to the number of customers included. For instances of 5, 10 and 15 customers, we set η_{vns} to 10, 20 and 30 respectively. We set $\eta_{early} = 10$ for instances of over 10 customers and $\eta_{early} = 5$ otherwise. For the cyclic operator, we set N_r equal to 2 when the fleet consists of 3 EVs or less and set N_r to 3 otherwise. The length of each exchange block is randomly selected from $\{1, 2, 3\}$. The upper and lower bounds of the tabu tenure are $v_{min} = 5$ and $v_{max} = 15$. The κ for the annealing mechanism is set to 0.5.

6.3 Model Performance

The performance of CPLEX, the Lagrangian relaxation and the VNS/TS heuristic is presented in Table 6.2. The bounds achieved by the three algorithms are presented in the "UB" and "LB" columns, while the "Time" columns document the solving time in seconds. In the "Gap" columns, we present the gap between the upper bounds obtained implementing the corresponding algorithm and the lower bound generated by the Lagrangian relaxation. We do not use the gap reported by CPLEX since those gaps are generally greater than the gap we reported. For the heuristic, we have an additional column "Best Iter" to record the VNS iteration during which the best solution was found.

For instances with 5 customers, all three algorithms can solve the problem in reasonable time. CPLEX outperforms the VNS/TS heuristic and the Lagrangian relaxation in terms of solving time for most of these instances. The efficiency of the heuristic is comparable with that of CPLEX, while the Lagrangian relaxation is obviously more time-consuming. With regard to bound quality, the Lagrangian relaxation achieves a 0.00% gap for every instance except "RC105-5" where the gap is 0.21%. The VNS/TS heuristic achieves a zero gap for 8 out of 12 instances. All the gaps of the heuristic are below 5%. Both algorithms are shown to be very effective.

When it comes to instances with 10 customers, the performance of the CPLEX worsen significantly. It can solve only 6 out of the 12 instances to optimality within the time and memory limits. Memory usage becomes a big concern as the memory limit is violated in 6 instances. For instances that the CPLEX is able to find the optimal solutions, the solving time on average is 3789.11 seconds, which is significantly higher than the average solving time of the heuristic on the same set of instances of 91.41 seconds. The VNS/TS heuristic outperforms CPLEX in all of the instances here in terms of both quality and efficiency. The Lagrangian relaxation is still able to generate bounds of high quality. In instances solvable

Table 6.2: Performance of CPLEX, Lagrangian relaxation, and VNS/TS heuristic

Instance		Cplex			Lagrangian		Heuristic			
Name	K	UB	Time	Gap	LB	Time	UB	Time	best iter	Gap
C101-5	2	82.61	3.27	0.00%	82.61	4.63	82.61	13.34	2	0.00%
C103-5	2	50.49	1.26	0.00%	50.49	3.06	50.49	9.36	3	0.00%
C206-5	2	87.19	76.41	0.00%	87.19	91.47	87.19	22.64	5	0.00%
C208-5	1	63.26	1.03	0.00%	63.26	19.25	63.26	4.34	1	0.00%
R104-5	2	40.35	64.06	0.00%	40.35	30.50	40.35	13.23	3	0.00%
R105-5	2	45.93	33.81	0.00%	45.93	44.89	45.93	15.07	3	0.00%
R202-5	1	52.86	3.58	0.00%	52.86	35.45	53.09	5.67	1	0.44%
R203-5	1	71.56	6.05	0.00%	71.56	87.17	72.47	7.11	2	1.27%
RC105-5	2	82.22	121.62	0.21%	82.05	99.07	84.21	25.07	3	2.64%
RC108-5	2	101.96	38.18	0.00%	101.96	8.75	101.96	18.56	1	0.00%
RC204-5	1	70.61	7.41	0.00%	70.61	96.11	73.51	8.12	1	4.11%
RC208-5	1	69.55	1.80	0.00%	69.55	9.88	69.55	5.94	2	0.00%
C101-10	3	$164.03^{(*)}$	3527.81	28.74%	127.41	5641.83	131.15	144.73	2	2.94%
C104-10	2	$113.90^{(-)}$	7200.00	0.00%	113.90	27800.64	113.90	113.95	15	0.00%
C202-10	2	$89.65^{(*)}$	3529.64	2.35%	87.59	14071.45	89.65	89.30	2	2.35%
C205-10	2	90.03	4693.57	0.00%	90.03	503.84	90.03	82.63	1	0.00%
R102-10	4	$81.22^{(*)}$	2910.64	-	-	-	57.66	110.57	2	-
R103-10	2	$54.75^{(-)}$	7200.00	3.66%	52.82	23548.13	54.75	124.55	12	3.66%
R201-10	1	105.02	2741.33	-	-	-	105.02	18.56	9	-
R203-10	2	82.75	1515.83	-	-	-	83.49	163.54	7	-
RC102-10	4	$165.67^{(*)}$	2383.08	14.68%	144.47	3684.50	144.47	168.12	4	0.00%
RC108-10	3	$153.99^{(*)}$	2776.38	8.63%	141.76	1708.13	142.48	113.62	1	0.51%
RC201-10	2	$125.40^{(*)}$	5295.07	0.00%	125.40	3362.22	125.40	98.32	6	0.00%
RC205-10	2	161.15	1288.87	0.00%	161.15	814.86	162.36	71.43	2	0.75%
C103-15	3	*	1443.57	-	-	-	108.22	367.32	1	-
C106-15	3	$111.94^{(*)}$	2638.52	-	-	-	96.21	465.78	12	-
C202-15	2	-	7200.00	-	-	-	151.83	652.23	21	-
C208-15	2	-	7200.00	-	-	-	118.58	277.66	2	-
R102-15	5	-	7200.00	-	-	-	65.17	172.19	2	-
R105-15	4	$90.28^{(-)}$	7200.00	-	-	-	75.63	769.23	22	-
R202-15	2	*	1096.87	-	-	-	145.99	337.82	3	-
RC103-15	4	*	1755.29	-	-	-	126.20	699.84	11	-
RC108-15	3	*	1953.75	-	-	-	147.52	768.09	23	-
RC202-15	3	*	1824.96	-	-	-	143.80	471.38	3	-
RC204-15	2	*	934.20	-	-	-	142.20	332.50	2	-

Instances that violate the memory and time limits are labeled with $*$ and $-$ respectively

using the Lagrangian relaxation, the gap between the lower bound obtained and the best solution found by the heuristic is no more than 3.66%. However, as the sub-problems become difficult to solve in instances with 10 customers even when there is only one EV, we are not able to obtain reasonable lower bound for 3 out of the 12 instances.

For instances with 15 customers, both CPLEX and the Lagrangian relaxation suffer in terms of memory and computational time. CPLEX finds feasible solutions that are far

away from the optima for only 2 out of the 11 instances, while the VNS/TS heuristic is able to solve all the instances within 483.09 seconds on average.

The results clearly show the VNS/TS heuristic's ability to find "high quality" solutions within reasonable time for these instances. In the real-world settings where logistic demands and price information are given one day ahead, the heuristic has the capability to solve instances of larger sizes. We further discuss the heuristic's performance on a realistic use case in Chapter 7.

Chapter 7

Use Case and Analysis

In this study, with the solution methods we developed in the previous chapters, we apply the EVRPTW-TP model to a realistic use case that we construct for an EV fleet providing online grocery delivery services, which attracted great attention amid the COVID-19 pandemic, in the Kitchener-Waterloo (KW) region in Ontario, Canada. Through computational experiment, we investigate the impact of electricity pricing schemes, time windows, and fleet size on the fleet's routing and charging/discharging behaviors based on which we provide some critical managerial insights.

7.1 Use Case Settings

Online Grocery Delivery

We consider a local grocery store that uses EVs to provide online grocery delivery services. This business model has been increasingly popular in recent years [110], especially during the COVID-19 pandemic when people are performing social distancing and avoiding going to public areas such as supermarkets and local stores. In the KW region, retail giants, such as Walmart, Zehrs, and T&T Supermarket, as well as local small businesses are all providing such services. Customers could browse available items and place orders online, service providers would then perform touch-free delivery during the time slot that customers booked. Due to the considerable demand during the COVID-19 crisis, orders usually have to be placed at least one day ahead for free delivery. Customers could, of course, pay additional fees for same-day delivery, but this is not the choice that the majority of people would make. In this study, we only take into account regular orders, i.e. the

orders made one day before their deliveries, such that routing and charging/discharging could be scheduled before the start of the planning horizon.

EV Fleet Specifications

Due to the complexity of the EVRPTW-TP, we only consider a small fleet consists of 3 EVs in the majority of our discussion and vary the fleet size to investigate that impact on fleet operations in Section 7.4. This fleet can provide delivery service for around 30 customers in the designated area, which reflects the real situations that local small businesses are faced with. The EVs in the fleet are homogeneous, i.e. with identical range, cargo capacity etc. We consider employing an economic EV model with a range of 150 kilometers, a 32.4 kWh battery, and a cargo capacity of 200. In practice, especially in urban areas, the travel time between locations are less affected by EV horsepower than by external factors such as traffic conditions, weather and so on. Therefore, we directly use the travel time along the shortest path between locations estimated by Google Maps. The planning horizon is set as $5am$ - $12am$.

Customers, Stations and the Depot

The geographical locations of the customers, stations and the depot are presented as blue, red and green dots as shown in Figure 7.1 respectively. All the customers are real condos/apartments randomly selected. In the majority of our discussion, we consider the 3-periods setting, i.e. partitioning the planning horizon into morning ($5am$ - $12pm$), afternoon ($12pm$ - $6pm$) and evening ($6pm$ - $12am$). We also consider the 2-hours setting where the planning horizon is divided into consecutive time slots lasting 2 hours each and the no time window setting. We make comparisons among different time window settings in Section 7.3. No matter which setting we apply, the time slot for each customer is randomly generated. The service time at each customer is set as a constant 10 minutes which is short because there is no waiting time for touch-free delivery.

Regarding the stations, according to the data presented by ChargeHub [22], there are over 130 EV charging stations in the designated area among which approximately 89% are level-2 stations and others are level-3 stations. The supply power of a normal level-2 charger is $240VAC/30A$. One hour of charge using level-2 stations add 30 kilometers of range for the EV model we employ, i.e. $B = 270$, $\alpha = 1.8$, while level-3 stations use a 480 Volt system and can add over 100 kilometers of range per hour [89]. In this study, we only take into account the level-2 stations for its economics and compatibility with most EV

models. We only select 7 from the existing level-2 stations for which reasons are two-fold. First, including too many stations will make our proposed model unsolvable with currently available tools. However, in order to reflect the geograophical distribution of the level-2 stations, the 7 stations are selected such that EVs could easily access charging/discharging services from anywhere in the area. Second, to the best of our knowledge, no station in the designated area is enabled for EV discharge. It is more realistic to deploy only a small number of stations with such capabilities at the pilot stage. We set a station located at a large parking lot in uptown Waterloo as the depot. All of the stations and the depot are open 24/7.

Electricity Rate

The Ontario Energy Board [12] sets two types of electricity rates: 1) time-of-use (TOU) rates where the rate depends on when customers use electricity and 2) tired rates where customers use a certain amount of electricity at a lower rate and pay a higher rate when the amount exceeds the limit. The TOU scheme is applied to most residential and small business customers while the tired scheme is adopted by a very small number of customers. In this study, we apply the TOU rates in Ontario, Canada for the majority of the analysis and compare different electricity pricing schemes in Section 7.2. With regard to discharging reward, currently, the province of Ontario does not have such a discharging reward plan. We set the reward rate in a way to ensure that it is profitable for the EV owners to provide ancillary services for the grid. In most of the senarios that we consider, the reward rates are set to be slightly lower than the electricity rates during the same time period. We compare different electricity pricing schemes in detail in Section 7.2.

7.2 Electricity Pricing Scheme

As shown in Table 7.1, we consider four electricity pricing schemes including two dynamic schemes (A and B), one static scheme (C) in bidirectional V2G context and one dynamic scheme (D) in unidirectional V2G context. The electricity price periods are defined by the Ontario Energy Board [12] as shown in Figure 7.2.

In scheme A, the charging rates are the real hydro rates in effect between May 1, 2018 and Nov 1, 2019. The reward rates are set between the charging rate in the same period and the the charging rate in a period when the grid pressure is relatively lower. For instance, given the on-peak charging rate 13.40¢/kWh and the mid-peak charging rate 9.40¢/kWh, the on-peak reward rate is set as 10.0¢/kWh. This way, the EVs could

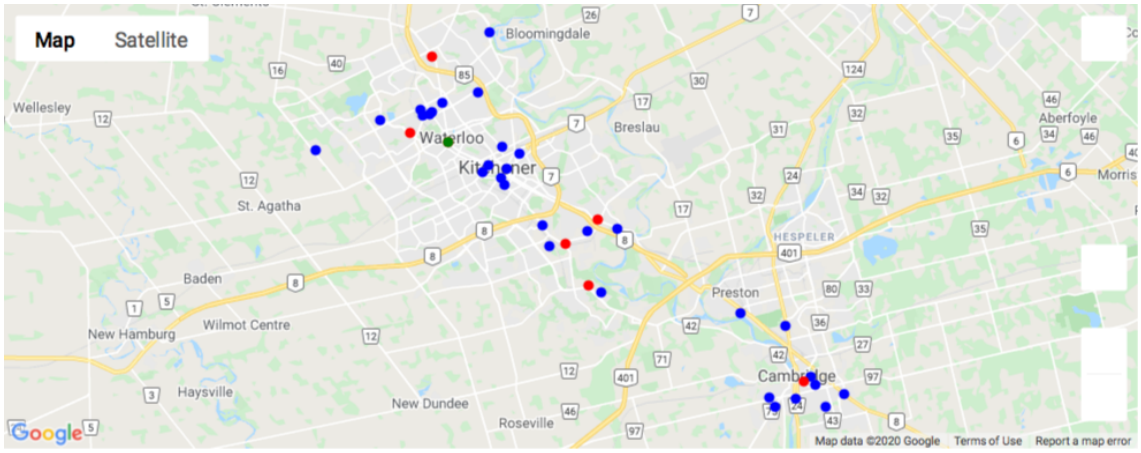

Figure 7.1: Selected Customers and Stations

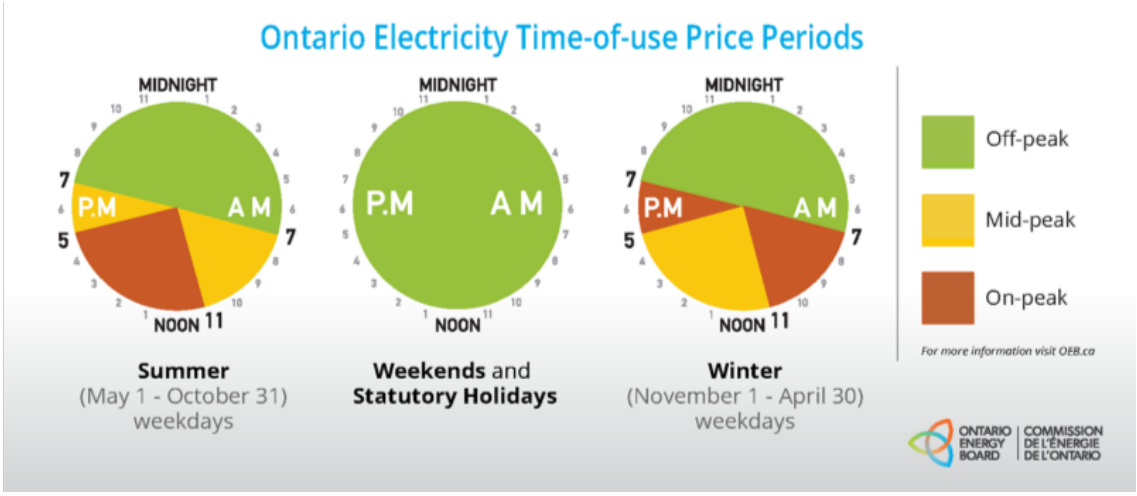

Figure 7.2: Time-of-Use Electricity Price Periods [12]

Table 7.1: Electricity Prices

	On-Peak (¢/kWh)		Mid-Peak (¢/kWh)		Off-Peak (¢/kWh)	
Scheme	Charge	Discharge	Charge	Discharge	Charge	Discharge
A	13.40	10.00	9.40	8.00	6.50	6.50
B	13.40	13.40	9.40	9.40	6.50	6.50
C	9.40	8.00	9.40	8.00	6.50	6.50
D	13.40	0.00	9.40	0.00	6.50	0.00

make profits by discharging in a on-peak (mid-peak) period and re-charging its battery in mid-peak (off-peak) periods. On the other hand, in a competitive energy market [130], we believe, the original electricity price can well represent the overall cost of energy generation and grid operation. Setting the reward rate slightly lower than the same-period charging rate makes the scheme also economically attractive to the grid operator. Scheme B where reward rates are equal to the same-period hydro rate is the dynamic electricity pricing scheme commonly implemented by previous research, see, for example, [61], [87] and [66]. Scheme C is a static pricing scheme where charging and discharging rates are constant values throughout the planning horizon. This is the case in Ontario starting from May 30, 2020 as one of the policy efforts made to support the gradual reopening of the local economy. We note that, instead of setting constant charging and discharging rates for the whole day, we set them to a low value during off-peak hours so that it's profitable for the EVs to provide the ancillary services. Scheme D is the case without discharging reward.

The results of the numerical studies are shown in Table 7.2. The first two columns present the daily total distance traveled and the total electricity cost of the fleet, while the last two columns document the number of hours that the fleet performs charge and discharge during the planning horizon. Note that for the "Cost" column, a negative number means the fleet make positive profits during the day. For the dynamic schemes, we consider both the summer and winter rates, while, for schemes C and D, the TOU prices are identical in summer and winter.

The first insight is that the bidirectional V2G programs are economically attractive to EV owners. In the unidirectional V2G setting (scheme D), the grocery store is supposed to pay $2.02 each day for charging which is 52% higher than the cost they have to pay when discharging is enabled and priced in a static manner (scheme C). If the dynamic prices are applied (schemes A, B) the fleet does not need to pay for its energy consumption anymore. Further, in the most optimistic case (scheme B in winter), the fleet could earn around $2.80 per day for providing ancillary services for the grid. The potential gains for

Table 7.2: Overall Electricity Cost and Charging/Discharging Hours under Pricing Schemes

Scheme	Distance (km)	Electricity Cost (¢)	Charging (hr)	Discharging (hr)
A (summer)	154.89	-9.36	2	9
A (winter)	148.60	-18.14	2	9
B (summer)	187.29	-233.86	3	10
B (winter)	180.75	-279.72	5	12
C	148.24	132.55	0	8
D	143.88	202.03	0	0

the fleet might look negligible in a daily basis, however, the annual difference could be up to $1205 assuming 250 business days in a year. The potential savings and profits could be even higher for companies who own EV fleets of larger sizes.

In addition, though not surprising at all, we observe that electricity prices have great impacts on fleet operations. In particular, since the reward rate in scheme B is much higher than that in scheme A, EVs are more willing to over-discharge amid peak hours and detour to recharge their batteries later. The travelling distance under scheme B is, on average, 32 kilometers longer than that under scheme A. EVs spend 1 - 3 more hours to perform discharge under scheme B than under scheme A. Moreover, the overall electricity cost varies significantly between these two cases. An important observation here is that price is a very powerful information signal that can assist to conduct effective communication between commercial EV fleets and the grid. Though this is beyond the scope of this research, we could definitely incorporate the analytical framework we develop in this thesis with the domain knowledge in energy and logistic systems to investigate the electricity pricing strategies in bidirectional V2G context.

Further, for schemes A and B, the total electricity cost is lower in winter than in summer. Taking scheme B as an example, in Figure 7.3, each blue line depicts one EV's battery level through a day while its charging/discharging behaviors are highlighted in red. The green dash lines in Figure 7.3 present the discharging reward rates through a day. In summer, in order to perform discharging during peak hours ($11am$ - $5pm$), at least one EV has to detour at the middle of the day because, some customers should be served before $12pm$, i.e. the end of the morning period. However, in the winter setting, on-peak periods are $7am$ - $11am$ and $5pm$ - $7pm$. All the EVs could stay at the depot before $11am$ to perform discharge and leave for service provision after that. As a consequence, the traveling distance in winter is shorter than in summer for both schemes. One takeaway for EV operators here is that they could also adjust their time slot settings with respect to

the season. For instance, in this case, cancelling the morning service period could, in some way, decrease the overall electricity cost. The real adjustments in practice may not be this extreme, but one should definitely take into account the price and TOU period changes between seasons when designing service time slots.

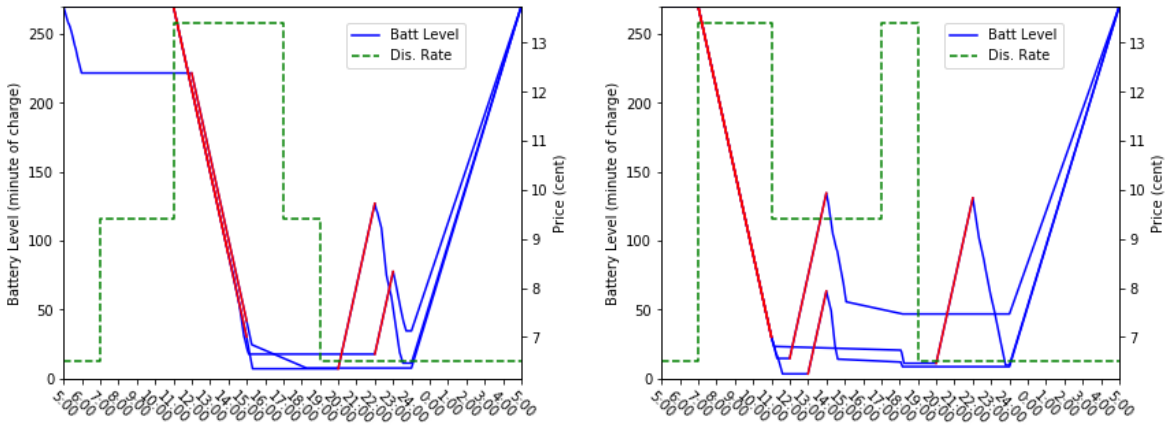

Figure 7.3: EVs' Battery Levels through a Day ($K = 3$, left: summer, right: winter)

7.3 The Impact of Time Window

As mentioned earlier, we take into account 3 time window settings which, from the tightest to the loosest, are the 2-hours setting, 3-periods setting and the case of no time window. Among them, the 2-hours setting is currently employed by most grocery delivery service providers. Its relatively restrictive time slots allow customers to know the time that they would be served more precisely, therefore, reduce their waiting times and enhance user's experience. However, under the 2-hours setting, the EV fleet's flexibility in routing and scheduling would be lower compared with that under the other two settings.

In particular, the results presented in Table 7.3 and Figure 7.4 reflect the trade-off between timely service and operational flexibility. Under the 2-hours setting, customers who are geographically close might book time slots that are very far away from each other. In this case, an EV cannot always wait at a customer's location until the beginning of the time slot for the nearest customer. Instead, it would travel to a customer who is relatively distant but with a time slot starting earlier, leaving the nearest customer to another EV. However, when 3-period time window is applies, with a higher probability, the EV would be

Table 7.3: Overall Electricity Cost under Time Window Settings

Season	TW Setting	Distance (km)	Electricity Cost (¢)	Charging (hr)	Discharging (hr)
Summer	2-hours	229.19	184.90	0	6
	3-periods	154.89	-9.36	2	9
	no-tw	131.81	-117.36	4	12
Winter	2-hours	212.78	164.09	2	7
	3-periods	148.60	-18.14	2	9
	no-tw	135.95	-111.53	4	12

able to serve the customers in an order based on their geographical locations. As presented in Figure 7.4, two EVs have to travel to Cambridge under the 2-periods setting, while only one has to do so under the other two settings. Thanks to the increased flexibility, the total travelling distance is reduced from over 210 kilometers for the 2-hours setting to around 150 kilometers for the 3-period setting and could even be as low as 131.81 kilometers for the case of no time window constraints. The reduction in travelling distance from "3-periods" to "no-tw" is marginal comparing with that from "2-hours" to "3-periods".

With regards to ancillary service provision, the fleet, of course, have the greatest flexibility in scheduling discharging under the "no-tw" setting. In both winter and summer cases, it is able to discharge for 12 hours in total during the planning horizon. Under the "3-period" setting, the fleet could perform 9 hours of discharging, while this number decreases to 6 in summer and 7 in winter when the 2-hours setting is applies. As a consequence, as presented in Table 7.3, the fleet is supposed to pay around $1.70 each day under the 2-hours setting, while the fleet could break even in the case of "3-periods" and make a daily profits over $1.10 under "no-tw". The $2.80 difference could partially, if not fully, represent the price of "timely service".

7.4 The Impact of EV Fleet Size

In this section, we consider the impact of EV fleet size on overall electricity cost under two restrictive time window settings, i.e. the 3-periods time window and the 2-hours time window as defined in Section 7.3. According to instance setup, the customers in our case could not be fully satisfied by less than 3 EVs which renders the problem infeasible. Therefore, starting from the fleet size of 3, we increase the number of EVs by 1 at a time and document the corresponding electricity cost in Table 7.4. Intuitively, increasing fleet

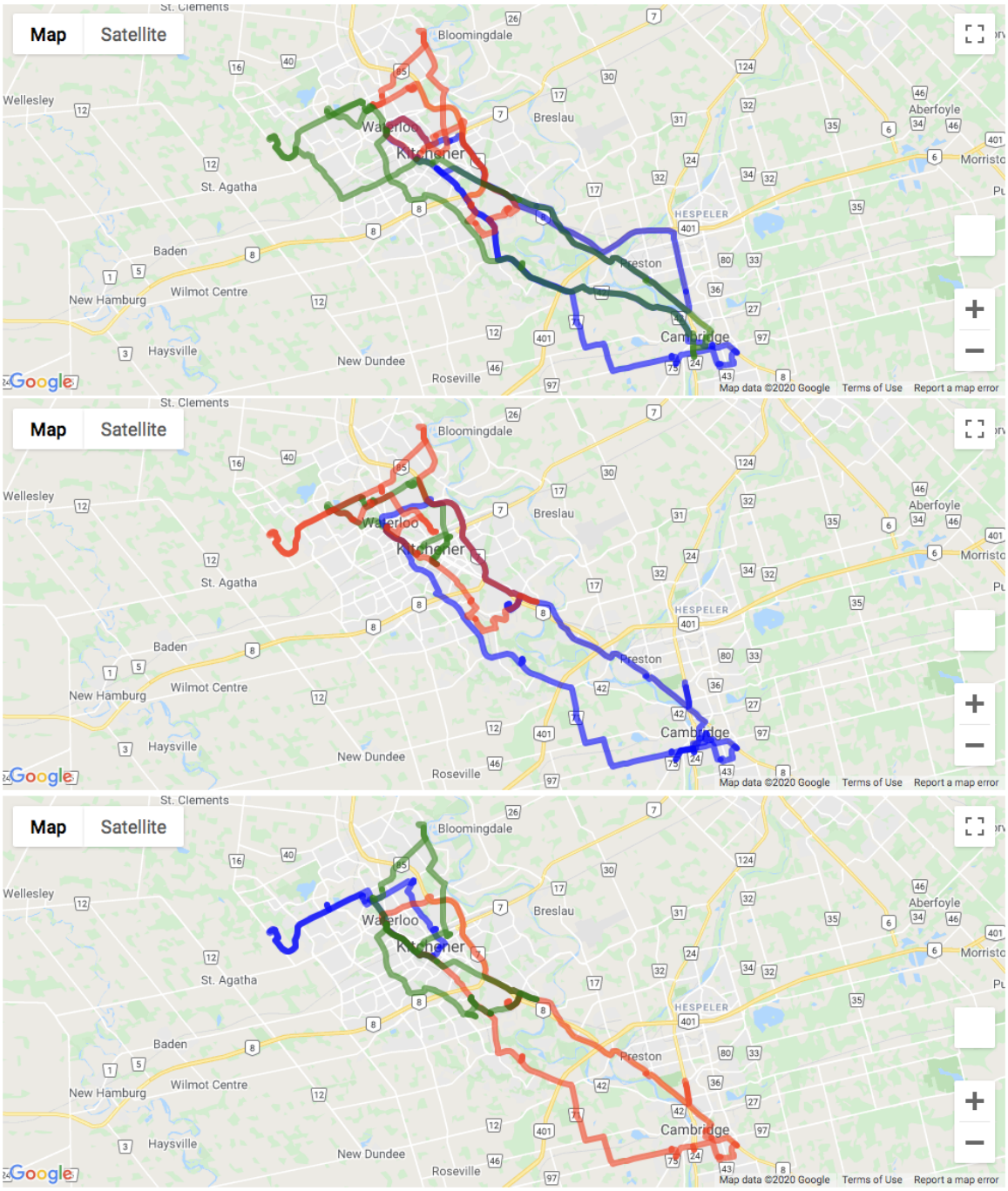

Figure 7.4: EV Routes (from top to bottom: 2-hours, 3-periods and no-tw)

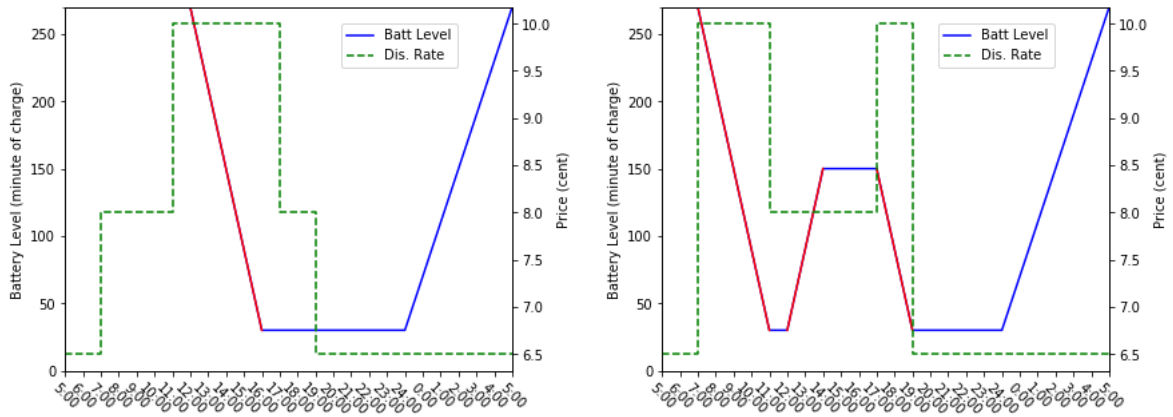

Figure 7.5: Battery Level of an EV without Logistic Service (left: summer, right: winter)

size will bring additional operational flexibility yet higher fixed cost in the meanwhile. As a consequence, routes could be streamlined so as to reduce the overall travelling distance and spare more time as well as battery capacity for ancillary service provision. However, given the high acquisition cost of the EVs, deploying an EV solely for ancillary service provision is obviously not economical for the fleet owner. In particular, Figure 7.5 depicts the battery levels of such an EV in summer and winter respectively. In summer, the EV discharges its battery under the on-peak rate for 4 hours and recharges under the off-peak rate at night, bringing a profit of 100.8¢. Similarly in winter, the EV makes a profit of 109.44¢ by discharging for 6 hours under on-peak rates and recharging for 2 and 4 hours under mid-peak and off-peak rates respectively. If adding one EV brings the fleet a profit no greater than 100.8¢ in summer or 109.44¢ in winter, it suggests that the newly added EV is working as a static battery storage. In this case, we should not consider adding any EVs to the fleet anymore.

The overall electricity cost (in cents) under different fleet sizes and time window settings are shown in Table 7.4 and visualized in Figure 7.6. Under the 2-hours time window setting (the red lines), increasing fleet size from 3 to 4 leads to over twice as large as the cost reduction results from the increase from 4 to 5 which, in fact, satisfies the termination condition for both summer and winter cases. The daily profits associated with adding one EV to the original fleet (from 3 to 4) are \$2.53 and \$2.55 for summer and winter cases respectively. As for the 3-periods cases (blue lines), the termination condition is met when the fleet size comes to 6 in summer and 5 in winter. However, the profits linked to the additional EVs are all very marginal. As clearly shown in Figure 7.6, the blue dash line segment from 4 to 6 appear to be nearly straight, suggesting the corresponding cost

Table 7.4: Overall Electricity Cost of EV fleets of Different Sizes

TW	Season	Number of EVs			
		3	4	5	6
2-hours	Summer	184.90	-67.75	-168.55	-
	Winter	164.09	-91.15	-199.30	-
3-periods	Summer	-9.36	-150.05	-265.97	-366.77
	Winter	-18.14	-186.98	-296.42	-

reduction is very close to the termination thresholds as shown by the line segment from 5 to 6.

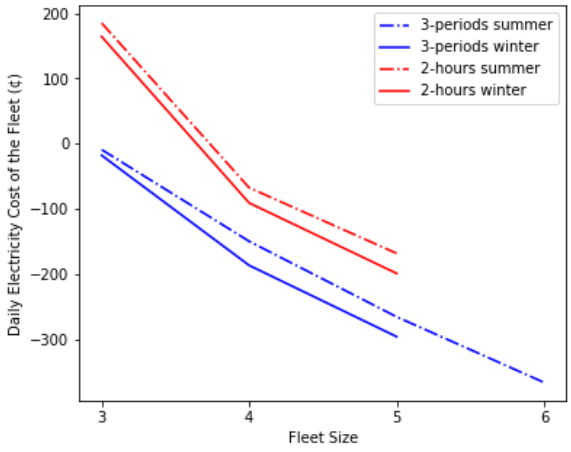

Figure 7.6: Electricity Cost under Different Fleet Sizes

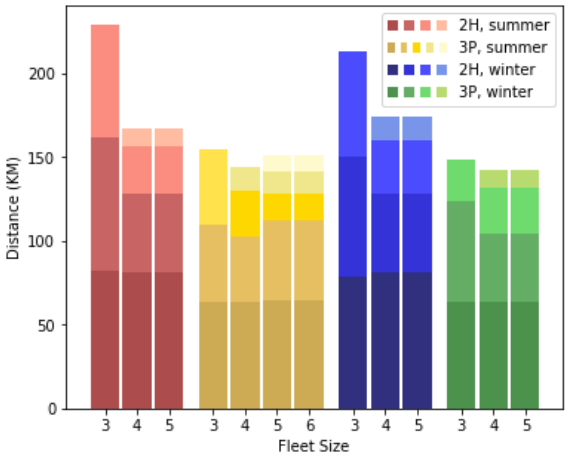

Figure 7.7: Overall Travelling Distance under Different Fleet Sizes

We also visualize the travelling distance for the fleet of different sizes and for each EV in Figure 7.7. Each bar consists of several sub-bars highlighted in different colors representing the travelling distance of each EV in the fleet. The sub-bars from the bottom to the top correspond to the longest and shortest distances travelled by a single EV in the fleet. As clearly shown, for all the four cases, when the fleet size is 3, the fleets' overall travelling distance is the highest, and the travelling distance is almost evenly allocated to EVs in a fleet. As fleet size increases, the overall travelling distance decreases and the variations in the distance travelled by a single EV increases. For the two "2-hours" time window cases (red and blue bars), increasing fleet size from 3 to 4 results in relatively large reduction

in overall travelling distance, while for other cases, the distance reductions are marginal. This observation suggests that the newly added EVs assist to reduce overall electricity cost mainly through grid ancillary service provision instead of through streamlining the routes.

The analysis done in this section answers the question if we need to enlarge the EV fleet or not and If so, how many additional EVs we need. Under the 3-periods setting, it is obvious that the additional EVs will play a role very similar, if not identical, to a battery storage without mobility. Hence, it is not economical to expand the EV fleet in this case. As for the 2-hours time window setting, one additional EV will bring the fleet operator more flexibility to deal with the relatively restrictive time windows. Nevertheless, the daily profits of around \$2.54, which is equivalent to an annual profits of \$635 assuming 250 business days through a year, is clearly insufficient to cover the acquisition cost of the EV. Therefore, the best strategy is to maintain the current fleet size.

Chapter 8

Reinforcement Learning for the EVRPTW

Motivated by the recent successful applications of deep reinforcement learning (RL) in solving combinatorial optimization problems such as the travelling salesman problem (TSP) and the vehicle routing problem (VRP), we conduct some preliminary analysis to explore the possibility of employing deep reinforcement learning to solve EVRPTW-TP. In this chapter, as the first step towards solving EVRPTW-TP, we follow the a framework similar to that developed by [85] to solve the EVRPTW where discharging and partial charging is not allowed and the objective is to minimize the total distance travelled by the fleet.

8.1 Reinforcement Learning Model

In a general RL framework, we regard the model we construct for solving the target problem as an agent. Based on the given problem, we define an environment where the agent takes a sequence of actions to tackle the problem. Each time an action is made, the agent receives a reward signal specifying the quality of the action it just made and the state of the environment might change as a consequence of the action. Through solving a considerable number of instances of the target problem with the agent, we adjust the parameters in the proposed model (train the agent) based on its performance such that the agent is able to receive high overall reward (generate high quality solution) when faced with instances of the target problem.

8.1.1 The MDP Formulation

We implement a Markov Decision Process (MDP) formulation based on the framework developed by Nazari et al [85]. In general, given a combinatorial problem whose solutions are permutations of the nodes defined by the problem, the agent applies a stochastic policy π to generate (decode) a solution to the problem sequentially, i.e. one node at a time.

We parameterize the policy with a deep learning model. At each decoding step, the model intakes information about the current system state and output a pointer indicating the next node to add to the solution. Some transition rules are then applied to update the state of the system based on the current system state as well as the newest pointer. This procedure is repeated until certain termination criteria is met. In particular, for the EVRPTW where each customer, station and the depot represents a node in the system, we define the environment and the actions as follows.

We use the local information at each node along with global information to describe the state of the whole environment. In particular, we assign an information vector X_i^t to each node i representing the local information associated with node i at the decoding step t. Vector X_i^t is set to $X_i^t = (x_i, z_i, e_i, l_i, q_i^t)$ where x_i and z_i represent the geographical coordinate of node i, e_i and l_i represent the corresponding time window, and q_i^t is the remaining demand at node i at decoding step t. The first four entries are static through the decoding process, while the last one is initialized as the demand at node i, i.e. q_i as defined in Chapter 3, and is set to 0 once node i has been visited by an EV. Information vectors for all of the nodes form a set $X^t = \{X_i^t | i = 0, 1, \ldots, N + S\}$. We note that the service time at each node is not taken into consideration here because we regard it as a constant to simplify the problem in this chapter.

Besides, these nodes share a global vector $G^t = \{\tau^t, b^t, ev^t\}$ that consists of global information of the whole system. The values of τ^t, b^t and ev^t indicating the time, battery level of the active EV and the number of EV(s) available at the start of decoding step t respectively. We note that the remaining cargo of the active EV is also an important global variable, however, we do not consider it as one of the model inputs because the presence of the masking scheme that is detailed in Subsection 8.1.2.

Moreover, we denote the pointer that generated at step t with y^t, the set of pointers up to step t with $Y^t = \{y^i | i = 1, 2, \ldots, t\}$, and the decoding step at termination with t'. In EVRPTW, y^t is the node our EV is going to visit at step t, and Y^t is the trajectory of the fleet up to step t. The routes for different EVs are separated by depot visit(s). For example, given a trajectory $Y^{t'} = \{0, 3, 2, 0, 4, 1, 0\}$, we need two EVs to complete the trip, one travels along $0 \to 3 \to 2 \to 0$, the other travels along $0 \to 4 \to 1 \to 0$.

At each decoding step t, we exploit a stochastic policy π to determine y^{t+1} (next node to visit) based on X^t, G^t (current system condition) and Y_t (the EV trajectory up to step t). In particular, we estimate the probability $P\left(y^{t+1} = i|X^t, G^t, Y^t\right), \forall i \in \{0, 1, \ldots, N+S\}$. And then, y^{t+1} could be selected in either a greedy or stochastic manner, or, alternatively, using the beam search. After that, the transition functions defined in 8.1 - 8.4 are applied to update the system state.

$$\tau^{t+1} = \begin{cases} \max(\tau^t, e_{y^t}) + s_{y^t} + t[y^t, y^{t+1}] \text{ , if } y^t \text{ is a customer} \\ \tau^t + t[y_t, y_{t+1}] + re(b^t) \text{ , if } y^t \text{ is a station} \\ t[y^t, y^{t+1}] \text{ , otherwise} \end{cases} \tag{8.1}$$

where $t[y^t, y^{t+1}]$ represents the travel time from y^t to y^{t+1}, $re(b^t)$ is the time required to fully charge the EV battery from the battery level b^t, and s_{y^t} is the service time at node y^t which is a constant in this thesis.

$$b^{t+1} = \begin{cases} b^t - f[y^t, y^{t+1}] \text{ , if } y^t \text{ is a customer} \\ B - f[y^t, y^{t+1}] \text{ , otherwise} \end{cases} \tag{8.2}$$

where $f[y^t, y^{t+1}]$ is the energy consumption from y^t to y^{t+1}, B is the battery capacity.

$$ev^{t+1} = \begin{cases} ev^t - 1 \text{ , if } y^t \text{ is the depot} \\ ev^t \text{ , otherwise} \end{cases} \tag{8.3}$$

$$d_i^{t+1} = \begin{cases} 0 \text{ , } y^t = \text{i} \\ d_i^t \text{ , otherwise} \end{cases} \tag{8.4}$$

8.1.2 Proposed Model

We use a neural network to parameterize the policy π, i.e. to estimate $P\left(y^{t+1} = i|X^t, G^t, Y^t\right)$, $\forall i \in \{0, 1, \ldots, N+S\}$. The model structure is illustrated in Figure 8.1. The proposed model consists of three components: the embedding, the attention mechanism and the LSTM decoder, which is very similar to the framework proposed in [85]. The differences mainly lie in the embedding components where a set of global variables τ^t, b^t along with ev^t and an additional graph embedding layer are employed.

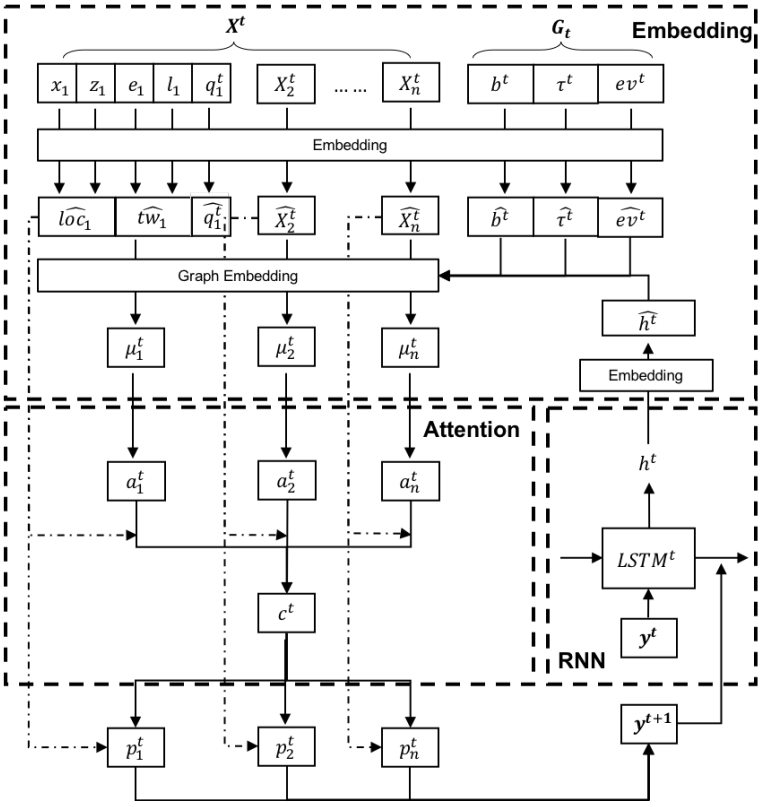

Figure 8.1: The Proposed Model Structure

Embedding

At decoding step t, the problem inputs X^t and G^t are mapped into a high dimensional vector space. We denote the embedded model inputs with $\hat{X}^t$ and $\hat{G}^t$. More specifically, for node i, its geographical coordinate (x_i, z_i), time window (e_i, l_i) and demand q_i^t are embedded to $\hat{loc}_i$, $\hat{tw}_i$ and $\hat{q}_i^t$ respectively. The embedded vectors are of the same dimension, say ξ. The embedding layers for location, time window and demand are different but are shared among nodes. Besides, we have three embedding layers for global variables τ^t, b^t and ev^t, and one additional layer for the hidden state of the LSTM decoder h^t, mapping them to ξ-dimensional vectors $\hat{\tau}_t$, $\hat{b}_t$, $\hat{ev}_t$ and $\hat{h}_t$ respectively.

In the classical VRP that [85] investigates, only local information at each node, i.e. location and remaining demand, are of interest. Given that the problem is defined on a complete graph where inter-nodes relationship is of lower importance, directly feeding these

embedded vectors to the attention model could generate decoding sequence of high quality. However, in the EVRPTW, global information as previously mentioned has significant influences on solution quality and feasibility. Therefore, combining the local information at different nodes and the global information is highly significant.

To this end, we employ the Structure2Vec tool [29], which has been successfully applied by [65] to tackle combinatorial problems over graphs. The graph embedding layer intakes all the aforementioned embedded vectors and computes a ξ-dimensional vector μ_t^i for each node i. In particular, we initialize a vector for node i as $\mu_i^{(0)} = \hat{loc}_i$, and then update $\mu_i^{(k)}, \forall k = 1, 2, \ldots, p$ recursively using Equation 8.5.

$$\begin{aligned}\mu_i^{(k)} = relu\{&\theta_1 \hat{b}^t + \theta_2 \hat{\tau}^t + \theta_3 \hat{ev}^t + \theta_4 \hat{h}^t + \theta_5 \hat{q}_i^t + \theta_6 \hat{tw}_i + \\ &\theta_7 \sum_{j \in V_{c,s,od}} \mu_j^{(k-1)} + \theta_8 \sum_{j \in V_{c,s,od}} relu\left[\theta_9 w(i,j)\right]\}\end{aligned} \tag{8.5}$$

where $V_{c,s,od}$ is the set of all nodes, $w(i,j)$ represents the weight on edge (i,j). For EVRPTW, $w(i,j)$ is defined as the distance among edge $(i,j), \forall i,j$.

After p steps of recursion, the network will generate a ξ-dimensional vector $\mu_i^{(p)}$ for node i and we let μ_i^t equal to $\mu_i^{(p)}$. At each recursion step, the global information and location information are aggregated by the first 6 terms of Equation 8.5. Moreover, the information at different nodes and edges propagates among each other via the last two summation terms. The final embedded vectors μ_i^t contains both local and global information, thus could better represent the complicated context of the graph.

The Attention Mechanism

Similar to [85], we utilize the context-based attention mechanism proposed by [5]. We first calculate a context vector c^t specifying the state of the whole graph as a weighted sum of $\mu_0^t, \mu_2^t, \ldots, \mu_{M+N}^t$:

$$c^t = \sum_{i=0}^{M+N} a_i^t \mu_i^t, \tag{8.6}$$

where the weight of node i is computed as

$$a_i^t = softmax\left(v^t\right) \tag{8.7}$$

$$v_i^t = W_v tanh\left(W_u \mu_i^t\right) \tag{8.8}$$

Then, based on the context vector c^t and embedded embedded vectors of all the nodes, we estimate the probability of visiting each node at the next step as

$$p_i^t = P(y^{t+1} = i | X^t, G^t, Y^t) = softmax(g^t) \tag{8.9}$$

$$g_i^t = W_g tanh\left(W_c[\mu_i^t; c_t]\right). \tag{8.10}$$

In 8.6 - 8.10, W_v, W_u, W_g and W_c are trainable variables, and [;] means concatenating the two vectors on the left and right of the ; symbol.

In order to accelerate the training process and ensure solution feasibility, we design several masking schemes to exclude infeasible routes. In particular, suppose that the EV is currently at node i at decoding step t, if node $j, \forall j \neq i$ satisfies one of the following conditions, we assign a very large negative number to the corresponding v_j^t and g_j^t such that the calculated weight a_j^t and probability p_j^t will be very close to 0.

- There is no remaining demand at customer node j, or the remaining demand is greater than the cargo capacity available.
- Current battery level b^t can not support the EV to travel to node j, and to travel back to the depot from node j.
- The earliest arrival time at node j is later than l_j.
- Given the current time τ^t, the travelling time from node i to node j and the recharging time at node j, if node j is a station, will result in violations of the planning horizon constraint, i.e. fail to return to the depot before the end of planning horizon.
- We mask all the nodes except the depot node if the EV is currently at the depot and there is no remaining cargo at any customer nodes.

The LSTM Decoder

Similar to [85] and previous literature including [5] and [112], we use the Recurrent Neural Network (RNN), more specifically LSTM, to model the decoder network. At decoding step t, The LSTM intakes the embedded location vector of the EV's current position $\hat{loc}_{y^t}$ as

well as the hidden state from the preceding decoding step h^{t-1} and output a hidden state h^t maintaining information about the trajectory up to step t, i.e. Y^t. The hidden state h^t is then embedded to $\hat{h^t}$ which is fed to the attention model as introduced earlier in this section.

8.2 Training Method

We implement a policy gradient method similar to [85] and [8] to train the proposed model. To this end, we use θ to denote all the trainable parameters in the model, and define a loss function as in Equation 8.11 representing the expected total reward of the trajectory sampled using the policy pamaterized by θ, i.e. π^θ. We minimize $L(\theta)$ using the REINFORCE gradient estimator with greedy rollout baseline BL similar to [72], as shown in Equation 8.12. At each training step, given a batch of instances generated on the fly, we can estimate the gradient by sampling trajectories for the instances with π^θ and taking average of the terms inside the expectation symbol in Equation 8.12.

$$L(\theta) = E_{Y\sim P_\theta(Y)}\left[r(Y)\right] \tag{8.11}$$

$$\nabla_\theta L = E_{Y\sim P_\theta(Y|X^0)}\left\{\left[r(Y) - BL(X^0)\right]\nabla_\theta log P_\theta(Y|X^0)\right\}, \tag{8.12}$$

where X^0 is a given instance, Y is the trajectory sampled using policy π^θ, $r(Y)$ is the total reward of trajectory Y, $BL(X^0)$ is the reward achieved by the baseline policy π^{BL} on instance X^0, and $P_\theta(Y|X^0)$ is the probability that trajectory Y is generated by policy π^θ given instance X^0.

We define the reward of trajectory $Y = \{y^0, y^1, \ldots, y^{t'}\}$ as given in Equation 8.13. The first term represents the negative total distance travelled by the EV fleet. In the training process, it is possible that the model will generate trajectory that requires more than the given EVs, we use the second term to penalize the excessive usage of EVs. Moreover, if the depot is located very close to a station, the model might achieve low travelling distance by constantly moving between this station and the depot. In order to prevent this "trick", we introduce the third term to penalize every station visit, which is plausible because we only visit charging station when necessary under the EVRPTW setting.

$$r(Y) = \sum_{t=1}^{t'} \gamma(y^{t-1}, y^t) + \beta_{ev}\max\{-ev^{t'}, 0\} + \beta_{stat}Stat(Y), \tag{8.13}$$

where $\gamma(y^{t-1}, y^t)$ is the negative distance between nodes y^{t-1} and y^t; $ev^{t'}$ is the number of EV(s) remained (note that this could be of a negative value); $Stat(Y)$ is the number of station visit(s) along trajectory Y; β_{ev} and β_{stat} are two pre-determined constants.

With regard to $P_\theta(Y|X^0)$, we use the probability chain rule proposed in [112] to decompose the probability as shown in Equation 8.14. Terms $P(y^{t+1}|X^t, G^t, Y^t)$ on the right hand side could be obtained from the model at each decoding step.

$$P(Y|X^0) = \prod_{t=0}^{t'-1} P(y^{t+1}|X^t, G^t, Y^t) \tag{8.14}$$

Regarding the baseline policy, for the first Λ training steps, we simply use the mean reward of the whole batch, after which we use the rollout baseline proposed by [72]. More specifically, we initialize the baseline policy π^{BL} as the policy we have at the beginning of the Λ^{th} steps. And then, after the Λ^{th} step, we compare the current model with the baseline model every ζ training steps. If the current model is significantly greater than the baseline model on a separated test set according to a single-side paired t-test ($\alpha = 5\%$), then we update the baseline and generate a new test set to avoid overfitting, otherwise we keep the baseline and the test set.

The pseudo code of the training algorithm is summarized in Algorithm 9. At each training step, we generate $\mathcal{N}$ random instances and perform simulation on them to approximate the gradient as in Equation 8.12. In each instance, the nodes are uniformly distributed among a region $[0, 1] \times [0, 1]$. We use a way similar to [105] to generate the time window for each customer. The center of a time window is uniformly distributed among $[0, 1]$ while the length is normally distributed with mean 0.2 and standard deviation 0.05. We use the Adam optimizer [67] to update the parameters θ and update the baseline policy if certain conditions are met. For the EVRPTW, the termination condition is that the EV has come back to the depot and there is no remaining demand at any customer nodes.

8.3 Numerical Experiment

8.3.1 Experimental Settings

We perform all the tests in this section using a Macbook Pro (2018) running Mac OS 10.13.6 with 4 CPU processors at 2.3 GHZ and 16 GB of RAM. The RL model is realized using the Tensorflow 2.2.0. The codes are implemented in Python.

Algorithm 9 REINFORCE with Rollout Baseline

initialize the actor network with random weights θ, test instance set $\mathcal{S}$.
for $i = 1, 2, ...$ **do**
  Generate $\mathcal{N}$ random instances $\mathcal{X} = \{\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_{\mathcal{N}}\}$
  for $n = 1, \ldots, \mathcal{N}$ **do**
    initialize decoding step counter $t_n \leftarrow 0$
    repeat
      choose $y_n^{t_n+1}$ according to the distribution $P_\theta(y_n^{t_n+1}|\mathcal{X}_n^{t_n}, G_n^{t_n}, Y_n^{t_n})$
      observe new state $\mathcal{X}_n^{t_n+1}, G_n^{t_n+1}, Y_n^{t_n+1}$
      $t_n \leftarrow t_n + 1$
    until termination condition is satisfied
    compute reward $r(Y_n^{t_n})$
  if $i \leq \Lambda$ **then** $BL(\mathcal{X}_n) \leftarrow avg\left[r(Y_1^{t_1}), r(Y_2^{t_2}), \ldots, r(Y_N^{t_N})\right]$
  else
    $BL(\mathcal{X}_n) \leftarrow \pi^{BL}(\mathcal{X}_n)$
  $\mathrm{d}\theta \leftarrow \frac{1}{\mathcal{N}} \sum_{n=1}^{\mathcal{N}} \sum_{t=1}^{t_n-1} \left[r(Y_n^{t_n}) - BL(\mathcal{X}_n)\right] \nabla_\theta \log P(Y_n^{t_n}|\mathcal{X}_n, \theta)$
  $\theta \leftarrow Adam(\theta, \mathrm{d}\theta)$
  if $i = \Lambda$ **then** Initialize baseline policy π^{BL}
  else if $i \mod \zeta = 0$ **then**
    if $OneSideTTest(\pi^\theta(\mathcal{S}), \pi^{BL}(\mathcal{S})) < \alpha$ **then**
      $\pi^{BL} \leftarrow \pi^\theta$
      Update test set $\mathcal{S}$

For the RL model, we adapt most hyper-parameters from the work done by [85]. We use seven 1-dimensional convolution layers for the embedding of location, time window, remaining demand, LSTM hidden state, current time, battery level and the number of available EVs respectively. All this information is embedded to a 128-dimensional vector space. We utilize a LSTM with a state size of 128. For the Adam optimizer, the initial step size is set to 0.001, and the batch size is $\mathcal{N} = 128$. To stablize the training, we clip the gradients such that their norms are no more than 2.0. With regard to the rollout baseline, we set Λ to 1000 and evaluate the baseline model every 100 training steps after that. In the reward function, the penalty factors for depot and station visits are set to 1.0 and 0.5 respectively. All the trainable variables are initialized with the Xavier initialization [44]. We train the model for 10000 steps which takes approximately 90 hours.

When training the model, we sample the trajectories in a stochastic manner to diversify the possible circumstances encountered by the RL model. In particular, at each training

step, the model outputs the probabilities of visiting different nodes and we select the next node to visit according to this probability distribution. When testing, we consider three different ways to generate solutions. First, we sample the route for a given instance in a stochastic manner for several times and select the best solution. Second, we select the nodes in a greedy manner, i.e. to select the node with the highest probability at each step. Finally, we use the beam search (BS) which simultaneously maintain more than one routes and seek to find a route with the highest overall probability.

8.3.2 Computational Results

The performance of the RL model, CPLEX and the VNS/TS heuristic developed by Schneider et al. [103] is presented in Tables 8.1 and 8.2. We apply these approaches to six different scenarios whose names indicate the numbers of customers, stations, and available EVs. For example, "C5-S2-EV2" means the task of 5 customers, 2 stations and 2 EVs. For each scenario, we solve 100 instances created in the same way as we produce the RL's training data and report the mean total distance travelled by the EV fleet and the gap from the minimal distance achieved by these algorithms in Table 8.1. The average solution time over the test set in seconds is recorded in Table 8.2. We only report the results for algorithms that can successfully solve an instance in 15 minutes. For stochastic implementation of the RL, we perform simulations for 100 times, while for the beam search, we maintain 3 channels simultaneously.

As shown, the stochastic implementation, though more time-consuming, achieves the lowest travelling distance among the three implementation methods for the RL. However, its optimality gaps are 7.29% and 11.93% for scenarios "C5-S2-EV2" and "C10-S3-EV3" respectively, while the performance of the VNS/TS heuristic is considered the state of the art for both scenarios. When it comes to scenarios with 20 or more customers, similar to the results reported in [103], CPLEX is not able to solve the problem within reasonable time. The VNS/TS heuristic outperforms the RL model in terms of solution quality on scenarios "C20-S3-EV3" and "C30-S4-EV4", yet spends 7-10 times the solution time utilized by the RL model. The gaps between the distances accomplished by the VNS/TS heuristic and the RL model are 17.16% and 26.19% respectively for these two scenarios. Regarding scenarios with 40 or more customers, the RL model is the only algorithm that is able to solve the EVRPTW within 15 minutes. In fact, even the most time-consuming RL implementation requires less than 3 minutes.

To conclude the chapter, we note that the RL model developed in this thesis is able to solve the EVRPTW fairly efficiently. Although it takes over 3 days to train the model, we

Table 8.1: Comparisons of Average Total Travel Distance of the 5 Approaches

	CPLEX		VNS/TS		RL(stochastic)		RL(Greedy)		RL(Beam)	
Instance	Distance	Gap	Distance	Gap	Distance	Gap	Distance	Gap	Distance	Gap
C5-S2-EV2	**2.47**	0.00%	2.48	0.40%	2.65	7.29%	2.85	15.38%	2.95	19.43%
C10-S3-EV3	**3.52**	0.00%	3.55	0.85%	3.94	11.93%	4.46	26.70%	4.63	31.53%
C20-S3-EV3	-	-	**5.42**	0.00%	6.35	17.16%	7.47	37.82%	7.43	37.08%
C30-S4-EV4	-	-	**6.91**	0.00%	8.72	26.19%	9.74	40.96%	9.84	42.40%
C40-S5-EV5	-	-	-	-	**11.36**	0.00%	12.76	12.32%	12.49	9.95%.
C50-S6-EV6	-	-	-	-	**13.98**	0.00%	15.34	9.73%	14.94	6.87%

Table 8.2: Comparisons of Average Solution Time of the 5 Approaches

Instance	CPLEX	VNS/TS	RL(stochastic)	RL(Greedy)	RL(Beam)
C5-S2-EV2	0.08	1.76	4.81	0.46	0.53
C10-S3-EV3	67.65	14.03	11.08	0.91	0.94
C20-S3-EV3	-	212.26	30.09	1.76	2.03
C30-S4-EV4	-	633.52	59.59	2.50	2.85
C40-S5-EV5	-	-	87.48	2.84	3.77
C50-S6-EV6	-	-	166.41	4.58	5.7

can utilize the model to solve any instance once the training is completed. With regard to solution quality, the performance of the proposed model is still relatively far away from the state of the art. At this stage, we regard it as a relatively good feasible solution generator rather than an optimal solution finder, which is helpful especially for instances of very large sizes.

Chapter 9

Conclusion and Future Work

In this thesis, we develop an optimization approach for operating a commercial EV fleet under bidirectional V2G context. In particular, we propose a new optimization problem, namely electric vehicle routing problem under time-variant electricity prices (EVRPTW-TVEP), considering practical constraints including customer service time window as well as vehicle cargo and battery capacities. The model jointly optimizes the fleet's route and charge/discharge schedule over the associated monetary cost, which reflects a realistic operational objective.

However, the proposed problem is so complicated that even powerful commercial MIP solvers, such as CPLEX, can only solve instances of 10 customers to optimality within 2 hours. To solve relatively larger instances, we formulate a Lagrangian relaxation and a variable neighborhood search and tabu search hybrid (VNS/TS) heuristic to generate lower and upper bounds to the optimal solution respectively. Our computational experiments show that both algorithms are able to provide good quality bounds for instances of less than 10 customers. The Lagrangian relaxation is relatively computationally expensive because the derived sub-problem is still too complex to solve with CPLEX. On the other hand, the VNS/TS heuristic is fairly powerful. It is able to generate close-to-optimal solutions for 10-customers instances with less than 5% of the time spent by CPLEX. Further, the VNS/TS heuristic is able to solve instances of 15 customers, which cannot be solved with any readily available tools, in less than 13 minutes.

In addition, we utilize the VNS/TS heuristic to study a case of a local grocery store providing online grocery delivery in the KW region in Ontario, Canada with a small EV fleet. Implications about the economical benefits for the V2G implementation, the electricity pricing, service time slot design and fleet size are drawn from the case study. First,

participating in the proposed V2G program can bring the fleet an annual benefit per EV of $\$279 - \$402$ comparing with a constant electricity price and no discharge setting. The potential profit is able to cover the fleet's energy consumption cost even in the worst case. Moreover, our numerical results show that electricity prices have significant impact on fleet operations. Adjusting service hours with respect to the changes in time-of-use period between summer and winter might be able to provide additional profits for the fleet. Additionally, the design of service time slot influences the operational flexibility in a substantial way. Under the tight setting where each time slot lasts 2 hours, the fleet can only discharge for $6-7$ hours per day while that number can go up to 12 when no time window is applied. Besides, the travelling distance of the first case is $57\% - 75\%$ higher than the latter one. Regarding the fleet size, we find that, in general, employing additional EV(s) enhances the fleet's operational flexibility. However, it is not worthwhile to enlarge the fleet in this case considering the high acquisition cost of EVs.

Furthermore, as the first step towards utilizing advanced machine learning algorithms to solve the proposed problem, we develop a deep reinforcement learning (RL) model to solve the EVRPTW. The RL model manages to obtain feasible solutions fairly efficiently. However, there is still rooms for improvement in solution quality.

We highlight a few potential research directions for future work. Firstly, previous research has shown that frequent charging and discharging has negative impacts on EV battery's lifespan. However, we do not take into account the cost associated with battery degradation through the thesis. To provide more reliable and realistic operational strategies, future research could incorporate the model with the battery wear cost. Secondly, to reduce the model complexity, we employ a linear charging/discharging function and a discretized charging/discharging schedule. Research efforts could be made to adopt more sophisticated modeling of charging/discharging processes and continuous energy prices. Additionally, it is fruitful to extend the proposed model to a robust optimization version to incorporate the uncertainties in energy prices, and customer demands. Nevertheless, more powerful solution algorithms should be developed before we delve into robust optimization. To be more specific, improvements could be made on the charge/discharge optimization given a route, especially for the case of one station visit along a route. Further, advanced techniques such as deep reinforcement learning is worth exploring as well. The Lagrangian relaxation and the VNS/TS heuristic introduced in this thesis could be employed in a branch-and-price approach to solve the proposed problem exactly. Finally, it is also interesting to investigate the variants of the proposed problem considering a heterogeneous fleet, time-dependant travelling cost, periodic routing and so on.

References

[1] Tarek Abdallah. The plug-in hybrid electric vehicle routing problem with time windows. Master's thesis, University of Waterloo, 2013.

[2] Ahmed Abdulaal, Mehmet H Cintuglu, Shihab Asfour, and Osama A Mohammed. Solving the multivariant ev routing problem incorporating v2g and g2v options. *IEEE Transactions on Transportation Electrification*, 3(1):238–248, 2016.

[3] Yogesh Agarwal, Kamlesh Mathur, and Harvey M Salkin. A set-partitioning-based exact algorithm for the vehicle routing problem. *Networks*, 19(7):731–749, 1989.

[4] Kemal Altinkemer and Bezalel Gavish. Parallel savings based heuristics for the delivery problem. *Operations Research*, 39(3):456–469, 1991.

[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[6] Michel L Balinski and Richard E Quandt. On an integer program for a delivery problem. *Operations research*, 12(2):300–304, 1964.

[7] C Battistelli, L Baringo, and AJ Conejo. Optimal energy management of small electric energy systems including v2g facilities and renewable energy sources. *Electric Power Systems Research*, 92:50–59, 2012.

[8] Irwan Bello, Hieu Pham, Quoc V Le, Mohammad Norouzi, and Samy Bengio. Neural combinatorial optimization with reinforcement learning. *arXiv preprint arXiv:1611.09940*, 2016.

[9] Ricardo J Bessa and Manuel A Matos. Economic and technical management of an aggregation agent for electric vehicles: a literature survey. *European transactions on electrical power*, 22(3):334–350, 2012.

[10] Justin DK Bishop, Colin J Axon, David Bonilla, Martino Tran, David Banister, and Malcolm D McCulloch. Evaluating the impact of v2g services on the degradation of batteries in phev and ev. *Applied energy*, 111:206–218, 2013.

[11] National Energy Board. Canada's energy future. Technical report, National Energy Board, 2018.

[12] Ontario Energy Board. Electricity rates. Available at: https://www.oeb.ca/rates-and-your-bill/electricity-rates. Accessed: 2020-06-02.

[13] Ontario Energy Board. Historical electricity rates. https://www.oeb.ca/rates-and-your-bill/electricity-rates/historical-electricity-rates. Accessed: 2019-06-12.

[14] Albert G Boulanger, Andrew C Chu, Suzanne Maxx, and David L Waltz. Vehicle electrification: Status and issues. *Proceedings of the IEEE*, 99(6):1116–1138, 2011.

[15] Kris Braekers, Katrien Ramaekers, and Inneke Van Nieuwenhuyse. The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*, 99:300–313, 2016.

[16] Tobias Brandt, Sebastian Wagner, and Dirk Neumann. Evaluating a business model for vehicle-grid integration: Evidence from germany. *Transportation Research Part D: Transport and Environment*, 50:488–504, 2017.

[17] Olli Bräysy and Michel Gendreau. Vehicle routing problem with time windows, part i: Route construction and local search algorithms. *Transportation science*, 39(1):104–118, 2005.

[18] Jose Caceres-Cruz, Pol Arias, Daniel Guimarans, Daniel Riera, and Angel A Juan. Rich vehicle routing problem: Survey. *ACM Computing Surveys (CSUR)*, 47(2):1–28, 2014.

[19] Duncan S Callaway and Ian A Hiskens. Achieving controllability of electric loads. *Proceedings of the IEEE*, 99(1):184–199, 2010.

[20] Statistics Canada. Table 20-10-0021-01 new motor vehicle registrations. DOI: https://doi.org/10.25318/2010002101-eng.

[21] Transport Canada. Zero-emission vehicles. Available at: https://www.tc.gc.ca/en/services/road/innovative-technologies/zero-emission-vehicles.html.

[22] ChargeHub. Charge your ev in kitchener. Available at: https://chargehub.com/en/countries/canada/ontario/kitchener.html?city_id=2397. Access on 2020-06-01.

[23] Wen-Chyuan Chiang and Robert A Russell. Simulated annealing metaheuristics for the vehicle routing problem with time windows. *Annals of Operations Research*, 63(1):3–27, 1996.

[24] Nicos Christofides, Aristide Mingozzi, and Paolo Toth. Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. *Mathematical programming*, 20(1):255–282, 1981.

[25] Geoff Clarke and John W Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, 12(4):568–581, 1964.

[26] Jean-Francois Cordeau, Michel Gendreau, Gilbert Laporte, Jean-Yves Potvin, and Frédéric Semet. A guide to vehicle routing heuristics. *Journal of the Operational Research society*, 53(5):512–522, 2002.

[27] Jean-François Cordeau, Gilbert Laporte, Martin WP Savelsbergh, and Daniele Vigo. Vehicle routing. *Handbooks in operations research and management science*, 14:367–428, 2007.

[28] Zbigniew J Czech and Piotr Czarnas. Parallel simulated annealing for the vehicle routing problem with time windows. In *Proceedings 10th Euromicro workshop on parallel, distributed and network-based processing*, pages 376–383. IEEE, 2002.

[29] Hanjun Dai, Bo Dai, and Le Song. Discriminative embeddings of latent variable models for structured data. In *International conference on machine learning*, pages 2702–2711, 2016.

[30] George B Dantzig and John H Ramser. The truck dispatching problem. *Management science*, 6(1):80–91, 1959.

[31] Rommert Dekker, Jacqueline Bloemhof, and Ioannis Mallidis. Operations research for green logistics–an overview of aspects, issues, contributions and challenges. *European Journal of Operational Research*, 219(3):671–679, 2012.

[32] Guy Desaulniers, Fausto Errico, Stefan Irnich, and Michael Schneider. Exact algorithms for electric vehicle-routing problems with time windows. *Operations Research*, 64(6):1388–1405, 2016.

[33] Martin Desrochers, Jacques Desrosiers, and Marius Solomon. A new optimization algorithm for the vehicle routing problem with time windows. *Operations research*, 40(2):342–354, 1992.

[34] Kevin J Dyke, Nigel Schofield, and Mike Barnes. The impact of transport electrification on electrical networks. *IEEE Transactions on Industrial Electronics*, 57(12):3917–3926, 2010.

[35] Burak Eksioglu, Arif Volkan Vural, and Arnold Reisman. The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering*, 57(4):1472–1483, 2009.

[36] Nasser A El-Sherbeny. Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods. *Journal of King Saud University-Science*, 22(3):123–131, 2010.

[37] Sevgi Erdoğan and Elise Miller-Hooks. A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 48(1):100–114, 2012.

[38] Ángel Felipe, M Teresa Ortuño, Giovanni Righini, and Gregorio Tirado. A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges. *Transportation Research Part E: Logistics and Transportation Review*, 71:111–128, 2014.

[39] Marshall L Fisher. Optimal solution of vehicle routing problems using minimum k-trees. *Operations research*, 42(4):626–642, 1994.

[40] Marshall L Fisher and Ramchandran Jaikumar. A generalized assignment heuristic for vehicle routing. *Networks*, 11(2):109–124, 1981.

[41] Marshall L Fisher, Kurt O Jörnsten, and Oli BG Madsen. Vehicle routing with time windows: Two optimization algorithms. *Operations research*, 45(3):488–492, 1997.

[42] Michel Gendreau, Alain Hertz, and Gilbert Laporte. A tabu search heuristic for the vehicle routing problem. *Management science*, 40(10):1276–1290, 1994.

[43] Billy E Gillett and Leland R Miller. A heuristic algorithm for the vehicle-dispatch problem. *Operations research*, 22(2):340–349, 1974.

[44] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[45] Bruce L Golden, Thomas L Magnanti, and Hien Q Nguyen. Implementing vehicle routing algorithms. Technical report, MASSACHUSETTS INST OF TECH CAMBRIDGE OPERATIONS RESEARCH CENTER, 1975.

[46] Bruce L Golden, Subramanian Raghavan, and Edward A Wasil. *The vehicle routing problem: latest advances and new challenges*, volume 43. Springer Science & Business Media, 2008.

[47] Rebecca Gough, Charles Dickerson, Paul Rowley, and Chris Walsh. Vehicle-to-grid feasibility: A techno-economic analysis of ev-based energy storage. *Applied energy*, 192:12–23, 2017.

[48] Ro E Griffith and RA Stewart. A nonlinear programming technique for the optimization of continuous processing systems. *Management science*, 7(4):379–392, 1961.

[49] Christophe Guille and George Gross. A conceptual framework for the vehicle-to-grid (v2g) implementation. *Energy policy*, 37(11):4379–4390, 2009.

[50] G Haddadian, N Khalili, M Khodayar, and M Shahidehpour. Optimal scheduling of distributed battery storage for enhancing the security and the economics of electric power systems with emission constraints. *Electric Power Systems Research*, 124:152–159, 2015.

[51] Sekyung Han, Soohee Han, and Kaoru Sezaki. Estimation of achievable power capacity from plug-in electric vehicles for v2g frequency regulation: Case studies for market participation. *IEEE Transactions on Smart Grid*, 2(4):632–641, 2011.

[52] Michael Held and Richard M Karp. The traveling-salesman problem and minimum spanning trees. *Operations Research*, 18(6):1138–1162, 1970.

[53] Michael Held, Philip Wolfe, and Harlan P Crowder. Validation of subgradient optimization. *Mathematical programming*, 6(1):62–88, 1974.

[54] Patrick Hertzke, Nicolai Müller, Patrick Schaufuss, Stephanie Schenk, and Ting Wu. Expanding electric-vehicle adoption despite early growing pains. Available at: https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/expanding-electric-vehicle-adoption-despite-early-growing-pains.

[55] Junjie Hu, Hugo Morais, Tiago Sousa, and Morten Lind. Electric vehicle fleet management in smart grids: A review of services, optimization and control aspects. *Renewable and Sustainable Energy Reviews*, 56:1207–1226, 2016.

[56] JQ James and Albert YS Lam. Autonomous vehicle logistic system: Joint routing and charging strategy. *IEEE Transactions on Intelligent Transportation Systems*, 19(7):2175–2187, 2017.

[57] Brian Kallehauge. Formulations and exact algorithms for the vehicle routing problem with time windows. *Computers & Operations Research*, 35(7):2307–2330, 2008.

[58] Brian Kallehauge, Jesper Larsen, and Oli BG Madsen. Lagrangian duality applied to the vehicle routing problem with time windows. *Computers & Operations Research*, 33(5):1464–1487, 2006.

[59] Evangelos L Karfopoulos and Nikos D Hatziargyriou. A multi-agent system for controlled charging of a large population of electric vehicles. *IEEE Transactions on Power Systems*, 28(2):1196–1204, 2012.

[60] Willett Kempton and Steven E Letendre. Electric vehicles as a new power source for electric utilities. *Transportation Research Part D: Transport and Environment*, 2(3):157–175, 1997.

[61] Willett Kempton and Jasna Tomić. Vehicle-to-grid power fundamentals: Calculating capacity and net revenue. *Journal of power sources*, 144(1):268–279, 2005.

[62] Willett Kempton and Jasna Tomić. Vehicle-to-grid power implementation: From stabilizing the grid to supporting large-scale renewable energy. *Journal of power sources*, 144(1):280–294, 2005.

[63] Willett Kempton, Victor Udo, Ken Huber, Kevin Komara, Steve Letendre, Scott Baker, Doug Brunner, and Nat Pearre. A test of vehicle-to-grid (v2g) for energy storage and frequency regulation in the pjm system. *Results from an Industry-University Research Partnership*, 32, 2008.

[64] Merve Keskin and Bülent Çatay. Partial recharge strategies for the electric vehicle routing problem with time windows. *Transportation Research Part C: Emerging Technologies*, 65:111–127, 2016.

[65] Elias Khalil, Hanjun Dai, Yuyu Zhang, Bistra Dilkina, and Le Song. Learning combinatorial optimization algorithms over graphs. In *Advances in Neural Information Processing Systems*, pages 6348–6358, 2017.

[66] Mahdi Kiaee, Andrew Cruden, and Suleiman Sharkh. Estimation of cost savings from participation of electric vehicles in vehicle to grid (v2g) schemes. *Journal of Modern Power Systems and Clean Energy*, 3(2):249–258, 2015.

[67] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[68] Michael Kintner-Meyer, Kevin Schneider, and Robert Pratt. Impacts assessment of plug-in hybrid vehicles on electric utilities and regional us power grids, part 1: Technical analysis. *Pacific Northwest National Laboratory*, 1, 2007.

[69] Paul R Kleindorfer, Kalyan Singhal, and Luk N Van Wassenhove. Sustainable operations management. *Production and operations management*, 14(4):482–492, 2005.

[70] Niklas Kohl and Oli BG Madsen. An optimization algorithm for the vehicle routing problem with time windows based on lagrangian relaxation. *Operations research*, 45(3):395–406, 1997.

[71] Antoon WJ Kolen, AHG Rinnooy Kan, and Harry WJM Trienekens. Vehicle routing with time windows. *Operations Research*, 35(2):266–273, 1987.

[72] Wouter Kool, Herke Van Hoof, and Max Welling. Attention, learn to solve routing problems! *arXiv preprint arXiv:1803.08475*, 2018.

[73] Suresh Nanda Kumar and Ramasamy Panneerselvam. A survey on the vehicle routing problem and its variants. *Intelligent Information Management*, 4(3):66–74, 2012.

[74] Jari Kytöjoki, Teemu Nuortio, Olli Bräysy, and Michel Gendreau. An efficient variable neighborhood search heuristic for very large scale vehicle routing problems. *Computers & operations research*, 34(9):2743–2757, 2007.

[75] Gilbert Laporte. The vehicle routing problem: An overview of exact and approximate algorithms. *European journal of operational research*, 59(3):345–358, 1992.

[76] Gilbert Laporte. Fifty years of vehicle routing. *Transportation science*, 43(4):408–416, 2009.

[77] Gilbert Laporte and Yves Nobert. A branch and bound algorithm for the capacitated vehicle routing problem. *Operations-Research-Spektrum*, 5(2):77–85, 1983.

[78] Steven Letendre, Richard Watts, Michael Cross, et al. Plug-in hybrid vehicles and the vermont grid: a scoping analysis. Technical report, University of Vermont. Transportation Research Center, 2008.

[79] Canhong Lin, King Lun Choy, George TS Ho, Sai Ho Chung, and HY Lam. Survey of green vehicle routing problem: past and future trends. *Expert systems with applications*, 41(4):1118–1138, 2014.

[80] Jane Lin, Wei Zhou, and Ouri Wolfson. Electric vehicle routing problem. *Transportation Research Procedia*, 12(Supplement C):508–521, 2016.

[81] Henrik Lund and Willett Kempton. Integration of renewable energy into the transport and electricity sectors through v2g. *Energy policy*, 36(9):3578–3587, 2008.

[82] David Mester and Olli Bräysy. Active guided evolution strategies for large-scale vehicle routing problems with time windows. *Computers & Operations Research*, 32(6):1593–1614, 2005.

[83] Alejandro Montoya, Christelle Guéret, Jorge E Mendoza, and Juan G Villegas. The electric vehicle routing problem with nonlinear charging function. *Transportation Research Part B: Methodological*, 103:87–110, 2017.

[84] Yuichi Nagata and Olli Bräysy. Edge assembly-based memetic algorithm for the capacitated vehicle routing problem. *Networks: An International Journal*, 54(4):205–215, 2009.

[85] Mohammadreza Nazari, Afshin Oroojlooy, Lawrence Snyder, and Martin Takác. Reinforcement learning for solving the vehicle routing problem. In *Advances in Neural Information Processing Systems*, pages 9839–9849, 2018.

[86] Eva Niesten and Floortje Alkemade. How is value created and captured in smart grids? a review of the literature and an analysis of pilot projects. *Renewable and Sustainable Energy Reviews*, 53:629–638, 2016.

[87] Lance Noel and Regina McCormack. A cost benefit analysis of a v2g-capable electric school bus compared to a traditional diesel school bus. *Applied Energy*, 126:246–255, 2014.

[88] Mehdi Noori, Yang Zhao, Nuri C Onat, Stephanie Gardner, and Omer Tatari. Light-duty electric vehicles to improve the integrity of the electricity grid through vehicle-to-grid technology: Analysis of regional net revenue and emissions savings. *Applied Energy*, 168:146–158, 2016.

[89] Ontario Ministry of Transportation. Charging electric vehicles. Available at: http://www.mto.gov.on.ca/english/vehicles/electric/charging-electric-vehicle.shtml.

[90] Ibrahim Hassan Osman. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of operations research*, 41(4):421–451, 1993.

[91] Samuel Pelletier, Ola Jabali, and Gilbert Laporte. Charge scheduling for electric freight vehicles. *Transportation Research Part B: Methodological*, 115:246–269, 2018.

[92] Jean-Yves Potvin and Jean-Marc Rousseau. A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. *European Journal of Operational Research*, 66(3):331–340, 1993.

[93] Christian Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, 31(12):1985–2002, 2004.

[94] Casey Quinn, Daniel Zimmerle, and Thomas H Bradley. The effect of communication architecture on the availability, reliability, and economics of plug-in hybrid electric vehicle-to-grid ancillary services. *Journal of Power Sources*, 195(5):1500–1509, 2010.

[95] Ghazal Razeghi and Scott Samuelsen. Impacts of plug-in electric vehicles in a balancing area. *Applied Energy*, 183:1142–1156, 2016.

[96] Peter Richardson, Damian Flynn, and Andrew Keane. Local versus centralized charging strategies for electric vehicles in low voltage distribution systems. *IEEE Transactions on Smart Grid*, 3(2):1020–1028, 2012.

[97] Niklas Rotering and Marija Ilic. Optimal charge control of plug-in hybrid electric vehicles in deregulated electricity markets. *IEEE Transactions on Power Systems*, 26(3):1021–1029, 2010.

[98] Robert A Russell. An effective heuristic for the m-tour traveling salesman problem with some side conditions. *Operations Research*, 25(3):517–524, 1977.

[99] Robert A Russell. Hybrid heuristics for the vehicle routing problem with time windows. *Transportation science*, 29(2):156–166, 1995.

[100] Robert A Russell and Timothy L Urban. Vehicle routing with soft time windows and erlang travel times. *Journal of the Operational Research Society*, 59(9):1220–1228, 2008.

[101] Jyri Salpakari, Topi Rasku, Juuso Lindgren, and Peter D Lund. Flexibility of electric vehicles and space heating in net zero energy houses: an optimal control model with thermal dynamics and battery degradation. *Applied energy*, 190:800–812, 2017.

[102] Tomás Gómez San Román, Ilan Momber, Michel Rivier Abbad, and Alvaro Sánchez Miralles. Regulatory framework and business models for charging plug-in electric vehicles: Infrastructure, agents, and commercial relationships. *Energy policy*, 39(10):6360–6375, 2011.

[103] Michael Schneider, Andreas Stenger, and Dominik Goeke. The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science*, 48(4):500–520, 2014.

[104] Shengnan Shao, Tianshu Zhang, Manisa Pipattanasomporn, and Saifur Rahman. Impact of tou rates on distribution load shapes in a smart grid with phev penetration. In *IEEE PES T&D 2010*, pages 1–6. IEEE, 2010.

[105] Marius M Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*, 35(2):254–265, 1987.

[106] Eric Sortomme and Mohamed A El-Sharkawi. Optimal charging strategies for unidirectional vehicle-to-grid. *IEEE Transactions on Smart Grid*, 2(1):131–138, 2010.

[107] Benjamin K Sovacool. A transition to plug-in hybrid electric vehicles (phevs): why public health professionals must care, 2010.

[108] Benjamin K Sovacool and Richard F Hirsh. Beyond batteries: An examination of the benefits and barriers to plug-in hybrid electric vehicles (phevs) and a vehicle-to-grid (v2g) transition. *Energy Policy*, 37(3):1095–1103, 2009.

[109] Benjamin K Sovacool, Lance Noel, Jonn Axsen, and Willett Kempton. The neglected social dimensions to a vehicle-to-grid (v2g) transition: a critical and systematic review. *Environmental Research Letters*, 13(1):013001, 2018.

[110] Sabah Mikha Steven Begley, Eric Marohn and Aaron Rettaliata. Digital disruption at the grocery store. Available at: https://www.mckinsey.com/industries/retail/our-insights/digital-disruption-at-the-grocery-store.

[111] Olle Sundstrom and Carl Binding. Flexible charging optimization for electric vehicles considering distribution grid constraints. *IEEE Transactions on Smart Grid*, 3(1):26–37, 2011.

[112] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[113] Éric Taillard. Parallel iterative search methods for vehicle routing problems. *Networks*, 23(8):661–673, 1993.

[114] Wanrong Tang, Suzhi Bi, Ying Jun Zhang, and Xiaojun Yuan. Joint routing and charging scheduling optimizations for smart-grid enabled electric vehicle networks. In *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, pages 1–5. IEEE, 2017.

[115] Sam R Thangiah, Ibrahim H Osman, and Tong Sun. Hybrid genetic algorithm, simulated annealing and tabu search methods for vehicle routing problems with time windows. *Computer Science Department, Slippery Rock University, Technical Report SRU CpSc-TR-94-27*, 69, 1994.

[116] Paolo Toth. Dynamic programming algorithms for the zero-one knapsack problem. *Computing*, 25(1):29–45, 1980.

[117] Paolo Toth and Daniele Vigo. *The vehicle routing problem*. SIAM, 2002.

[118] Paolo Toth and Daniele Vigo. The granular tabu search and its application to the vehicle-routing problem. *Informs Journal on computing*, 15(4):333–346, 2003.

[119] Alicia Triviño-Cabrera, José A Aguado, and Sebastián de la Torre. Joint routing and scheduling for electric vehicles in smart grids with v2g. *Energy*, 175:113–122, 2019.

[120] José Villar, Cristian A Díaz, Jose Arnau, and Fco Alberto Campos. Impact of plug-in-electric vehicles penetration on electricity demand, prices and thermal generation dispatch. In *2012 9th International Conference on the European Energy Market*, pages 1–8. IEEE, 2012.

[121] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in neural information processing systems*, pages 2692–2700, 2015.

[122] Lu Wang, Suleiman Sharkh, and Andy Chipperfield. Optimal coordination of vehicle-to-grid batteries and renewable generators in a distribution system. *Energy*, 113:1250–1264, 2016.

[123] Zhenpo Wang and Shuo Wang. Grid power peak shaving and valley filling using vehicle-to-grid systems. *IEEE Transactions on power delivery*, 28(3):1822–1829, 2013.

[124] Rashid A Waraich, Matthias D Galus, Christoph Dobler, Michael Balmer, Göran Andersson, and Kay W Axhausen. Plug-in hybrid electric vehicles and smart grids: Investigations based on a microsimulation. *Transportation Research Part C: Emerging Technologies*, 28:74–86, 2013.

[125] Andrew Winston. Inside ups's electric vehicle strategy. Available at: https://www.ups.com/us/es/services/knowledge-center/article.page?kid=ac91f520.

[126] Rick Wolbertus, Maarten Kroesen, Robert van den Hoed, and Caspar Chorus. Fully charged: An empirical study into the factors that influence connection times at ev-charging stations. *Energy Policy*, 123:1–7, 2018.

[127] Ye Wu, Zhengdong Yang, Bohong Lin, Huan Liu, Renjie Wang, Boya Zhou, and Jiming Hao. Energy consumption and co2 emission impacts of vehicle electrification in three developed regions of china. *Energy Policy*, 48:537–550, 2012.

[128] Hongming Yang, Songping Yang, Yan Xu, Erbao Cao, Mingyong Lai, and Zhaoyang Dong. Electric vehicle route optimization considering time-of-use electricity price by learnable partheno-genetic algorithm. *IEEE Transactions on smart grid*, 6(2):657–666, 2015.

[129] Rongshan Yu, Wenxian Yang, and Susanto Rahardja. A statistical demand-price model with its application in optimal real-time price. *IEEE Transactions on Smart Grid*, 3(4):1734–1742, 2012.

[130] Hamidreza Zareipour, Claudio A Cañizares, and Kankar Bhattacharya. The operation of ontario's competitive electricity market: overview, experiences, and lessons. *IEEE Transactions on Power Systems*, 22(4):1782–1793, 2007.

[131] Chengke Zhou, Kejun Qian, Malcolm Allan, and Wenjun Zhou. Modeling of the cost of ev battery wear due to v2g application in power systems. *IEEE Transactions on Energy Conversion*, 26(4):1041–1050, 2011.