Efficient Path Planning and Battery Management for Electric Vehicles

by

Seyedmohammad Kashefi Mofrad

A THESIS
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE FACULTY OF GRADUATE STUDIES
OF LAKEHEAD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

2023
Lakehead University
Thunder Bay, Ontario, Canada

Examining Committee Membership

The thesis of Seyedmohammad Kashefi Mofrad, titled Efficient Path Planning and Battery Management for Electric Vehicles, is approved by:

Supervisor:

Dr. Dariush Ebrahimi
Professor, Department of Physics & Computer Science
Wilfrid Laurier University, Waterloo, Ontario, Canada

Co-supervisor:

Prof. Sabah Mohammed
Professor, Department of Computer Science
Lakehead University, Thunder Bay, Ontario, Canada

Internal Examiner:

Dr. Thiago Eustaquio Alves de Oliveira
Professor, Department of Computer Science
Lakehead University, Thunder Bay, Ontario, Canada

External Examiner:

Dr. Salimur Choudhury
Professor, Queen's School of Computing
Queen's University, Kingston, Ontario, Canada

ABSTRACT

The rapid advancement in battery technology has brought electric vehicles (EVs) into reality, and the increasing adoption of autonomous electric vehicles (AEVs) has presented significant challenges. Existing research in the realm of IoT has extensively explored EV transportation systems, focusing on aspects like routing, energy management, and grid system equilibrium. In this context, this thesis readdresses the challenge of determining the fastest route for AEVs considering the battery charging time.

Diverging from the current state-of-the-art, our work delves into the prospect of not only minimizing travel time but also maximizing battery life for the optimal utilization of electric vehicles. We commence by formalizing the problem of "Efficient Path Planning and Battery Management for Electric Vehicles" as a mixed integer linear programming (MILP) model, thereby deriving its optimal solutions mathematically. Given the inherent complexity of the optimization model, we introduce a range of heuristic algorithms designed to address the problem at scale. Furthermore, this problem is similar to the traveling salesman problem(TSL), which means it has an NP-hard nature.

Recognizing the dynamic environment within which EVs operate, we transform the problem into a Markov Decision Process (MDP) and propose a Q-Learning-based reinforcement learning (RL) algorithm to solve it. Our thorough analysis and evaluations, spanning small and large node networks, underscore the efficacy of our proposed methodologies, while also identifying the superior approach in practical scenarios.

This study signifies a significant stride towards unlocking the full potential of Autonomous EVs, optimizing both travel time and battery life. Through this research, we aim to provide valuable insights into the efficient utilization of AEVs, thereby contributing to the advancement of sustainable and intelligent transportation systems.

ACKNOWLEDGEMENTS

I am deeply grateful to my supervisor, Dr. Dariush Ebrahimi, for his unwavering support and guidance throughout my master's program. His expertise and patience have been invaluable to me and have played a crucial role in the success of this thesis. I would also like to sincerely thank my co-supervisor, Prof. Sabah Mohammed, and Dr. Fadi Alzhouri, for their invaluable guidance and support throughout my master's program. I am grateful to the Computer science faculty of Lakehead University for providing me with the opportunity to conduct my research and for all of the resources and support they provided. I would also like to thank my family especially my sister for their love and support during this process. Without them, this journey would not have been possible.

PUBLICATIONS

- "Fastest Route and Charging Optimization of an Electric Vehicle With Battery's Life Consideration", manuscript submitted to the 26th International Conference on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM2023). (Accepted)
- "Route and Charging Optimization of an Electric Vehicle With Battery's Life Consideration: A Reinforcement Learning Approach", manuscript prepared and aimed to be submitted to IEEE Internet of Things Journal.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Most people believe that the future is for electric vehicles, but all indicators show that this era is the era of electric vehicles (EVs) and the future is for autonomous electric vehicles (AEVs). In recent years, the demand for EVs has dramatically increased due to their low operational cost, Government policies and laws, emission-free, new battery technologies, and improving vehicle range [1]. In this context, establishing and finding smart urban environments and systems play a crucial role in obtaining the maximum benefits of AEVs. This imposes cooperation between engineers from different fields on one hand, and developers of computer systems, networks, and applications on the other hand. While researchers have worked to provide solutions to many challenges in AEV, such as safety, security, batteries, autonomous driving, routing, energy, and grid systems among others, approaches based on both autonomous charging and driving are modest and do not take optimal battery usage into account.

Motivated by the aforementioned information, this thesis proposes a practical approach that finds the fastest route from the source to the destination of an EV, and leverages the optimum battery energy levels, considering the distribution of charging stations. Currently, not only deciding on the right time and place for charging is an important process but also the amount of charging required to reach the destination is essential, especially since electric charging is still time-consuming compared to refueling with gasoline [2]. Therefore the aim is to automate and systematize the battery charging level, routing, and setting the right times and charging stations, to reach the destination in the shortest time and the optimal level of utilizing the battery as well. In particular, we outline the thesis's contributions as follows:

- We mathematically formulate the problem of EV routing, maintaining charging levels for efficient battery life, finding appropriate charging stations on the route, and charging the battery based on demand, to minimize the traveling time from the source to the destination, and obtain optimal solutions for different graphs.

- Due to the complexity of the optimization model, we propose a time-efficient routing and charging (TERC) heuristic that decomposes the problem into two sub-problems: (i) selection of charging stations on the way to the destination, and (ii) determination of the optimal paths between the source, charging stations, and the destination. The TERC method applies a dynamic programming approach to solve the second sub-problem and uses a greedy algorithm to solve the first one.

- To improve the performance of our heuristic algorithm, we propose a newer version of the TERC method, called TERC2, which incorporates a more sophisticated charging station selection procedure based on an ant colony optimization (ACO) algorithm. The ACO algorithm assigns a pheromone value to each charging station based on its proximity to the vehicle's current location and the predicted energy consumption until the next charging stop. The vehicle then selects the charging station with the highest pheromone value, and the procedure is repeated until the vehicle reaches its destination.

- For comparison purposes, we update the well-known state-of-the-art K-shortest path (KSP) algorithm to fit our problem by considering all required constraints and propose a new method called the K-fastest path (KFP) method. This method selects a path (from K paths) with available charging stations on the route that allows the vehicle to reach its destination with the minimum number of recharging stops and time, thus minimizing the total travel time.

- We evaluate and validate the proposed methods in small and large environments by varying the number of road intersections and the distribution of charging stations, and using traveling time, battery usage, distance, and computation time as the performance metrics.

- In an intriguing development, we extend our investigation by augmenting the environment with dynamic features. We subsequently establish that the problem, even in this more complex setting, retains its NP-hard status—a kinship with the classic Traveling Salesman Problem (TSP). This finding strengthens the inherent complexity of the problem and underscores its significance.

- Furthermore, we extend the proposed heuristic method to adapt to dynamic changes in the environment over time. We devise a Reinforcement Learning (RL) approach using Q-learning to determine the optimal path in each time unit based on evolving environmental conditions. We evaluate this updated heuristic against the original heuristic and the newly introduced RL approach. This comparative analysis serves to highlight the advancements achieved through the integration of reinforcement learning techniques.

The remainder of this thesis is organized as follows:

- In the following chapter (Chapter 2), we review the literature on related work and characterize the uniqueness of the proposed approaches.
- In Chapter 3, we present the system model that underpins our study and then we provide a comprehensive description of the core problem at hand. This section lays the foundation for our analysis in subsequent chapters.
- In chapter 4, we present the mathematical formulation of the problem. Moreover, we introduce multiple heuristic algorithms and rigorously evaluate their performance across varying scales of networks, spanning both small-scale and expansive scenarios.
- Furthermore, in Chapter 5, we extend the scope by incorporating dynamic features into the environment and enhancing our earlier heuristic method to operate within this dynamic context. Furthermore, we introduce a Reinforcement Learning (RL) approach to address the problem, and meticulously compare its outcomes with those of the original and adapted heuristics. This chapter showcases our innovative strategies for adapting to evolving real-world conditions.
- Finally, we conclude the thesis in Chapter 6 and summarize the findings, and suggest the potential for future research.

Chapter 2

literature Review

The popularity of electric vehicles has grown over the past few years as a result of their affordability and environmental friendliness. Unfortunately, route planning is significantly hampered by the short driving range and lengthy charging times [3]. For this reason, various academics have proposed different strategies to determine the fastest route for EVs considering the location of charging stations along the route. In [4], the authors proposed a nonlinear programming model that takes into account EV speed and loads to reduce energy consumption. In their studies speed and loads have been considered as key factors in reducing energy consumption rather than distance as suggested in [5]. The latter assumed the scarcity of charging stations and the habits of full charging in their studies, which contradict the increase of charging stations nowadays, and that time will be wasted in charging a vehicle to its fullest. Similarly in [6], the authors modeled the problem as a Mixed Integer Linear Programming (MILP) by adding more factors to serve a fleet of EVs in a certain area. Whereas, in [7] the authors adopted the tabu search heuristic method and used the full charge strategy to solve the routing problem using time windows for a fleet of EVs. A machine learning technique was used by [8] to predict energy consumption along road segments for energy-efficient routing. In this context, the authors in [9] extended the energy consumption prediction model introduced in [10] to include the temperature and internal resistance of the battery. In their strategy, after predicting the EV energy consumption on all roads, by using one of the shortest-path algorithms, they tried to calculate and find the best energy-efficient route. It should be noted that the optimal energy-efficient route is different from the fastest route although the two objectives may be met in special cases. In addition, no battery recharging has been considered on the route of the EV from the source to the destination. There are very few research studies that consider recharging EVs. For example, the research study introduced in [11] coordinates the allocation of charging stations to a flow of EVs. However, this solution does not serve an individual vehicle in all situations. To extend battery life, the authors in [12] focused on the state of battery charging along the EV routing problem, and proposed four models to charge an EV battery, such as partial or full charging. Their goal is to reduce the delivery time while taking into account the battery life. They also mentioned that their suggested model did not work with large instances due to the increased complexity of the proposed model.

In 2020, Ming Meng and Yun Ma introduced an optimal problem concerning electric vehicle routes with time windows [13]. Their work incorporated two charging methods and featured a mathematical model with early and late arrival penalties as the objective function, aimed at minimizing transportation, vehicle usage, power supply, and penalty costs. While their model utilized an ant colony algorithm to solve the problem, it did not account for the dynamic nature of the environment, including changes in road speed and battery degradation rates. In 2019, Wenjuan Zhou [14] addressed route planning in dynamic transportation networks where travel times were treated as random variables. Their goal was to minimize total energy consumption and travel time. They presented a method to transform the multi-objective optimization problem into a single-objective problem. However, their approach focused on adapting routes based on environmental changes and did not consider scenarios where the vehicle needed to charge to reach its destination, potentially altering the routing plan. In 2023, Alberto Ponso, conducted research aimed at solving the EV routing problem while considering the state of health (SoH) of the battery and temperature conditions [15]. Their innovative route planning method factored in SoH, temperature, and driving style while selecting charging stations along the planned route that could be reached with the available battery energy. Although this approach could extend the EV's battery life, it prioritized SoH over finding the fastest route. In 2021, a research [16] employed an off-policy model-free reinforcement learning approach to generate energy-feasible paths for EVs from source to target. They tested their algorithm on a Swiss road network but treated the charging process as a discrete variable, not suggesting specific battery charge amounts at each station. Instead, they charged the battery to its maximum capacity. In 2022, Norbert Lech and Piotr Nikończuk published a research paper proposing a solution to the EV routing problem using genetic algorithms [17]. Their method took into account factors such as the EV's range, charging station locations, and slow/fast charging station parameters. The primary objective of their algorithm was to minimize travel distance, which might result in routes that are not necessarily the fastest, as road speed can vary based on laws and traffic conditions.

In this thesis, we present a solution to the problem of routing and charging an electric vehicle to reach its destination as fast as possible, considering the extension of battery life. In other words, unlike [17], our main objective is to find the route with minimum traveling time rather than the one with the least distance. Furthermore, In contrast to [12], our proposed solutions can be applied to large instances.

Chapter 3

System Model and Problem Definition

The system model, as shown in Figure 3.1, is presented with a weighted graph $G = (N, E)$, where N is a set of nodes (intersections and charging stations) in the scope of the system, and E is a set of edges (road segments) connecting any two nodes, and weighted based on the distance, traveling time, and battery level. The source and destination nodes are shown in the figure with black nodes, respectively with notations S and D. Whereas, the intersections and charging stations are respectively shown with blue and red nodes. The dotted arrow illustrates the EV traveling direction on the road visiting a set of nodes on its way from the source to the destination. We let the energy depletion and charging of the EV per time unit be $E_{Depletion}$ and $E_{Charging}$, respectively. For longevity of battery life, the maximum and minimum battery level percentages are kept respectively between B_{min} and B_{max}. We also let $I_{Station}$ represent the set of charging stations. For simplicity, we assume the EV takes a fixed time $T^{Edge}_{(i,j)}$ to traverse edge (i, j) or any edge. Consequently, the arrival time and battery level percentage of the vehicle on each visited node can be calculated, and they are highlighted in the graph respectively with T_i and B_i. The red arrow illustrates the direction to the nearest charging station, which is estimated to consume the energy of $E_{toStation}$.

Hence, the problem is to find the optimum route on the road, traversing a set of nodes from the source to the destination, such that the battery level maintains within B_{min} and B_{max}, choosing and charging EV at minimum charging station points as needed, which eventually minimizes the traveling time. In addition, the battery level of the EV at the destination should be at least enough to reach the nearest charging station, and still, its battery level is above B_{min}. In other words, the battery level of the EV at the destination should be $B_D = B_{min} + E_{toStation}$. The EV first collects all necessary information about the environment with the help of a roadside unit or base station and then solves the routing

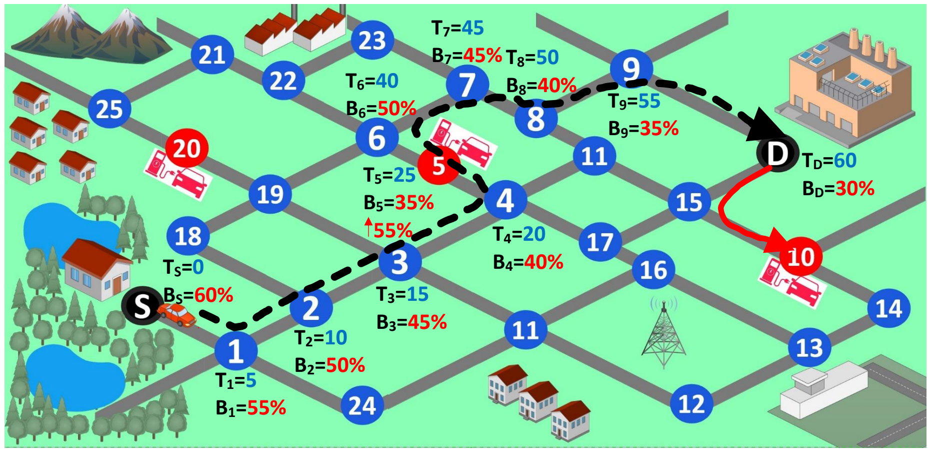


Figure 3.1: Illustration of the system model; the intersections are shown with blue nodes, charging stations with red nodes, the source and destination nodes respectively with black S and D nodes, and roads connecting nodes with links. The dotted black arrow shows the EV traveling direction on the road with arriving time (T_i) and battery level percentage (B_i) at each node.

and charging problem. The figure shows an example where the battery level of the EV at the source is $B_S = 60\%$, and since the EV has not yet started, the traveling time is $T_S = 0$. Next, when the EV reaches node 1, if we assume $T^{Edge}_{(1,2)} = 5$ and $B_{Depletion} = 5$, then $T_1 = 5$ and $B_1 = 55\%$. Notice that the EV did not take the shortest path to the destination because the battery level could be below $B_{min} + E_{toStation}$. Therefore, the EV had to change its course at node 4, to reach a charging station at node 5, to charge with enough energy to let it get to the destination satisfying the battery level constraint. The EV arrives at node 5 at $T_5 = 25$ and with a battery level of $B_5 = 35\%$. It charged with 20% only within 5 unit time to get to 55% battery level and then continued its course to node 6. When arrived at that node, as shown in the figure, the battery as expected was $B_6 = 50\%$ (i.e., $B_6 = B_5 - E_{Depletion}$). Finally, the EV arrives at the destination in $T_D = 60$ with a battery level of $B_D = 30\%$. Note that the EV has enough battery energy to get to the nearest charging station (node 10) and still its battery level will be equal or above B_{min} if we assume $B_{min} = 20\%$.

Chapter 4

Fastest Route and Charging Optimization of an Electric Vehicle With Battery's Life Consideration

Electric vehicles (EVs) are widely applied in logistics companies' urban logistics distribution, as fuel prices increase and environmental awareness grows. The tremendous development in battery technology has made the use of electric vehicles (EVs) a reality and the growing usage of autonomous electric vehicles (AEVs) over the past couple of years has posed many serious challenges. To date, much IoT research exists on EV transportation systems in general, particularly in the routing, energy, and grid system balance. In this context, throughout this chapter, we revisit the task of finding the fastest path for AEVs. In contrast to the state-of-the-art, we explore the capabilities of minimizing the traveling time and considering the battery life for the effective utilization of electric vehicles. We first model the problem mathematically to obtain optimal solutions for small instances, and then owing to the complexity of the optimization model, we propose several heuristic algorithms to solve the problem on large instances. Our analysis and evaluations, which are based on small and large graphs, demonstrate the effectiveness of our proposed approaches and which one is superior in practice. We believe that this study provides a very strong step towards finding the optimal usage of AEVs in terms of time and battery life.

4.1 Introduction

The prevailing consensus points toward an imminent era dominated by EVs, particularly AEVs. Recent times have witnessed a remarkable surge in the demand for EVs, driven by factors such as their economical operation, environmentally friendly attributes, supportive governmental policies, advancements in battery technologies, and increasing travel range [1].

Amid this context, the establishment of intelligent urban systems assumes a pivotal role in maximizing the benefits harnessed from AEVs. This endeavor necessitates a harmonious collaboration between engineers spanning diverse disciplines, and computer systems, network, and application developers. While research efforts have strived to address multifaceted challenges in AEV technology encompassing safety, security, battery innovations, autonomous navigation, routing, energy dynamics, and grid infrastructure, the integration of optimized battery utilization and optimal autonomous charging and driving approaches remains relatively modest.

Motivated by these insights, this chapter advances a pragmatic methodology aimed at identifying the swiftest route between an EV's origin and destination while optimizing battery energy levels and capitalizing on strategically placed charging stations. In the contemporary landscape, making decisions not only about when and where to charge but also about the quantum of charge needed for the journey assumes paramount significance, given the persistently extended duration associated with electric charging when compared to conventional gasoline refueling [2]. Therefore, our objective entails the automated orchestration of battery charging levels, routing, and the meticulous scheduling of charging instances and locations, all geared toward achieving the dual objective of minimizing travel time and maximizing battery utilization. Furthermore, it is worth mentioning that it is the vehicle (EV) that employs these algorithms in order to find its route to the destination.

The core contributions of this work encompass the following facets:

- **Mathematical Formulation**: We rigorously model the intricate EV routing problem, encompassing battery management and optimal charging station selection, to minimize travel time from source to destination. Our approach is devised to yield optimal solutions across diverse graph scenarios.

- **TERC Heuristic**: To surmount the intricacies inherent in the optimization model, we introduce a time-efficient routing and charging (TERC) heuristic. This heuristic bifurcates the problem into two sub-problems: (i) optimal charging station selection en route to the destination, and (ii) determination of the optimal paths between source, charging stations, and destination. TERC employs dynamic programming to solve the second sub-problem and employs a greedy algorithm to tackle the first.

- **TERC2 Method**: Enhancing the performance of our heuristic, we present an evolved variant termed TERC2. This approach incorporates a sophisticated charging station selection mechanism founded on an ant colony optimization (ACO) algorithm. By attributing a pheromone value to each charging station based on proximity to the vehicle's current location and anticipated energy consumption until the subsequent charging stop, the ACO-driven TERC2 makes informed decisions.

- **KFP Method**: To facilitate comparison, we adapt the renowned K-shortest path (KSP) algorithm to our scenario, leading to the introduction of the K-fastest path (KFP) method. This technique selects a path from a set of K paths, integrating available charging stations along the route. This selection minimizes recharging stops and overall travel time, aligning with the overarching goal of time optimization.

- **Evaluation and Validation**: We comprehensively assess and validate the proposed methods across diverse settings, spanning both small-scale and expansive environments. The evaluation metrics encompass travel time, battery utilization, distance, and computation time, all contributing to a comprehensive understanding of each method's performance characteristics.

In summation, this chapter embodies a crucial step forward in realizing the potential of EVs, and expediting travel times while optimizing battery deployment strategies. The seamless synergy of technological innovation and strategic planning resonates with the broader vision of sustainable, efficient electric mobility.

4.2 Problem Formulation

In this section, we mathematically formulate the problem as a mixed integer linear programming (MILP). The used notations are listed in Table 4.1. Let T_D indicate the time required for the EV to reach its destination. The objective of the optimization model is to minimize the arrival time of the EV. It can be mathematically formulated as follows:

$$Min \qquad T_D \tag{4.1}$$

subject to: (4.2) - (4.9), where these constraints are derived in detail in subsections 4.2.1 to 4.2.4.

4.2.1 Routing Constraints

In this subsection, the route for the EV from its source (starting point, S) to its destination (D) is constructed. Let $R_{(i,j)} \in \{0, 1\}$ indicate whether link (i, j) is set on the path of the EV from the source to the destination (i.e., $R_{(i,j)} = 1$, if link (i, j) is on the route of the EV and zero otherwise). The route construction is mathematically formulated as shown in eq. (4.2).

Table 4.1: Notations used in the problem formulation

Parameters		
N	Set of nodes.	
E	Set of edges (links).	
S	Starting point (source) of the EV.	
D	Destination of the EV.	
L	Large constant (larger than any possible system time).	
$I_{Station}$	Set of charging stations.	
$T^{Edge}_{(i,j)}$	Time to traverse edge (i,j).	
B_{Max}	Maximum battery level.	
B_{Min}	Minimum battery level.	
$E_{toStation}$	Required energy to get to the nearest station.	
$E_{Depletion}$	Energy depletion per time unit.	
$E_{Charging}$	Energy charging per time unit.	
Variables		
T_D	≥ 0	Arrival time at destination.
$R_{(i,j)}$	$\in \{0,1\}$	Indicate whether link (i,j) is set on the route of the EV.
B_i	≥ 0	EV's battery level at node i.
T_i	≥ 0	EV's arrival time at node i.
C_i	$\in \{0,1\}$	Indicate whether the EV has been charged at node i.
G_i	≥ 0	Required time to charge EV at station i.

$$\sum_{j:(i,j)\in E} R_{(i,j)} - \sum_{j:(j,i)\in E} R_{(j,i)} = \begin{cases} 1, & i = S; \\ -1, & i = D; \\ 0, & otherwise. \end{cases} \tag{4.2}$$

The above constraint obtains the differences between the number of outgoing links to incoming links on node i for route construction. If node i is the source of the EV, obviously the difference between the number of outgoing links to incoming links is equal to 1 and equal to -1 when node i is the destination node since the total number of assigned incoming links to node i is one and the total number of assigned outgoing links is zero, hence the subtraction result is -1. Consequently, when node i is a relay node or neutral (none of the above), the difference is zero; that is, if there is an incoming link to a relay node (for route construction), there should be an outgoing link, and if there is no incoming link (not on a route), there must be no outgoing link as well; so the subtraction in both cases is equal to zero. To avoid a loop, the following constraint restricts the edge to be active in one direction but not in both directions at the same time.

$$R_{(i,j)} + R_{(j,i)} \leq 1; \qquad\qquad \forall (i,j) \in E \tag{4.3}$$

4.2.2 Travel Time Constraints

The EV's presence at any node except the source means reaching that node has taken a while and the time increases as the EV travels to another node. To evaluate the arrival time of the EV to any visited node, let T_i be the arrival time of the EV to node i. Constraint (4.4) obtains the arrival time of the EV at node i (i.e., T_i) after traversing link (h,i). If node h is a charging station and the EV has charged its battery there, then the charging time G_h is added to the traversed time of the EV on edge (h,i). It is noted that when edge (h,i) is not on the route of the EV (i.e., $R_{(h,i)} = 0$), then the constraint is always satisfied since the left-hand-side of the inequality is very large (i.e., equal to the large constant L).

It is to be noted that T_D is equal to T_i obtained in constraint (4.4) when the EV reaches the destination node (i.e., $D = i$).

$$\begin{aligned} T_i + L(1 - R_{(h,i)}) \geq T_h + T^{Edge}_{(h,i)} + G_h, \\ \forall (h,i), \forall i \in N : i \neq S \end{aligned} \tag{4.4}$$

4.2.3 Charging Constraints

In this work, to restrict charging only to charging stations when needed, nodes belonging to subset $I_{Station} \subseteq N$ of set N are considered to act as charging stations. And since charging can only occur at some charging stations depending on the energy level of the battery, $C_i \in \{0,1\}$ indicates whether node i is a charging station or not as shown in (4.5).

$$C_i = \begin{cases} 0, & i \notin I_{Station}; \\ 1, & i \in I_{Station}, \end{cases} \tag{4.5}$$

This constraint assures that when a node i is not a charging station, then no EV will be charged at that node, and therefore, $C_i = 0$, while $C_i = 1$ indicates that node i is a charging station. Therefore, it is normal to use constraint (4.6) to make sure that the charging time G_i on the non-charging station node, where $C_i = 0$, is zero and it is positive value $G_i > 0$ when EV is charged.

$$G_i \leq L * C_i \qquad\qquad \forall i \in N \tag{4.6}$$

4.2.4 Battery Constraints

To ensure that the EV arrives at the specified destination without any problems related to running out of energy, and to maintain the quality of battery work for longevity, the battery energy level must be maintained within the recommended minimum (B_{min}) and maximum (B_{max}) values. Let B_i be the battery level of the EV at node i. Indeed, constraint (4.7) ensures that the battery level is within the range at any node and time from source S to destination D. Whereas, constraint (4.8) makes sure that the battery level is above the minimum level plus enough energy (i.e., $E_{toStation}$) to reach the nearest charging station. Thus, in our study, we raised the advantage of maintaining enough energy not only to safely finish the current trip but also to launch the next journey. Finally, constraint (4.9) calculates the battery level of the EV at each node it traverses from the source to the destination. In this constraint, when the EV arrives at node i after traversing link (h, i), the battery level of the EV at node i is calculated by subtracting the energy needed to traverse link (h, i) (that is the time requires to traverse the link multiplying the depletion energy $E_{Depletion}$ per unit time) from the EV battery level at node h. Obviously, if node h is a charging station and the EV has been charged for G_h time, then the charging energy $E_{Charging}$ per unit time is added to the battery level. Also, it is to be noted that the constraint is always satisfied when the EV does not traverse edge (h, i) as explained for constraint (4.4).

$$B_{min} \leq B_i \geq B_{max} \qquad \forall i \in N \tag{4.7}$$

$$B_D \geq B_{Min} + E_{toStation} \tag{4.8}$$

$$\begin{aligned} B_i \leq B_h - E_{Depletion} * T^{Edge}_{(h,i)} \\ + G_h * E_{Charging} + L(1 - R_{(h,i)}) \\ \forall (h, i), \forall i \in N : i \neq S \end{aligned} \tag{4.9}$$

4.3 Proposed Heuristic Approaches

From the fact that the optimization model presented in Section 4.2 is very complex and cannot solve large environments (graphs). Hence, in this section, we propose three heuristic methods.

Algorithm 1: TERC

Data: Graph $G(N, E)$, B_s, S, D.

Result: P_T: The fastest path by considering all constraints from S to D.

1 $T_D = 0$;
2 **Function** `FindPath`(S, D, B_S)**:**
3     $P_T \longleftarrow Dijkstra(S, D)$;
4     $T_P \longleftarrow TimeOnPath(P_T)$;
5     $B_{used} \longleftarrow BatteryUsedOnPath(P_T)$;
6   **if** $B_i - B_{used} >= B_{min} + E_{toStation}$ **then**
7         $T_D = T_D + T_P$;
8   **return** P_T
9   **else**
10         $T_P = L$;
11   **for** *each* $i \in \mathcal{I}_{Station}$ **do**
12             $P_{temp} \longleftarrow Dijkstra(S, i)$;
13             $T_{temp} \longleftarrow TimeOnPath(P_{temp})$;
14   **if** $T_P > T_{temp}$ **then**
15                 $T_P = T_{temp}$;
16                 $S_{new} \longleftarrow i$;
17         $B_{new} = B_{max}$;
18         $T_D = T_D + T_P + G_i$;
19   **return** $P_{temp} + FindPath(S_{new}, D, B_{new})$

4.3.1 TERC (Time Efficient Routing & Charging)

The TERC method is given in Algorithm 1, where it takes as an input; graph $G(N, E)$, the initial battery level B_s, the source node S, and the destination node D, and returns the fastest path from the source to the destination by considering all the constraints necessary for the longevity of the battery's life. The algorithm works in a recursive function. First, it finds the fastest path (P_T) from S to D using the $Dijkstra$ algorithm in line 3, and finds respectively the traveling time (T_P) and battery consumption (B_{used}) using $TimeOnPath$ and $BatteryUsedOnPath$ functions in lines 4 and 5. Then, the algorithm checks whether the battery percentage after traversing the fastest path P_T and arriving at the destination node is above the required minimum battery level (i.e., B_{Min}) plus the energy needed to reach the nearest charging station (i.e., $E_{toStation}$); mathematically, we can write it as $B_d - B_{used} >= B_{min} + E_{toStation}$. It is noted that the battery level B_d in the if condition in line 6 is written as B_i because of the recursive function $FindPath$. If the condition is met, then the algorithm terminates and returns the fastest path P_T, where the travel time T_D is the result of the TERC method. Else, the algorithm, in lines 11-16, finds the closest charging station to the source node, and obtains the fastest route and battery consumption, to be added later to the path from the current charging station to the destination node, in a recursive fashion. Finally, after calculating the total traveling time in line 18 (that is time to traverse the path from the source to the current charging station and the time to charge the battery), the algorithm returns the concatenation path resulting from the current path plus the path that will be obtained from the recursive function $FindPath$, which will find the fastest route from the current charging station to the destination node (line 19).

Furthermore, as our proposed method goal is to find the best route for EVs from a source to a destination, it is clear that there should be at least one possible route that does not violate any of our constraints. Because of this, in order to make sure that the algorithm will never face a dead-end or stuck in a loop, the environment is designed in a way to make sure that there is at least one route from any source to any destination with the consideration of minimum battery level at the source. Since the goal of the algorithm is to assist EVs in finding the best route from a source to a destination, when the EV is reviewing all possible charging stations in order to find the right one (line 11), it is clear that there is at least one charging station within its current state range of travel that can be reached and hence, the algorithm continues finding a path until it directs the EV to the destination to terminate. In other words, the algorithm will always find a result to terminate and thus this proves its correctness.

The algorithm, in the best case, when no charging is required, finds the path from the source to the destination in $O(E^2)$, where this is the time taken to run Dijkstra's algorithm. However, in the worst case, when it is required to charge the EV's battery, the time complexity of the algorithm is $O(IE^2 logI)$; the algorithm recursively finds the path from the source to charging stations and eventually to the destination, which in the worst case, checks all the charging stations I in $O(logI)$ iterations, and in each iteration and for each charging station, should run Dijkstra's algorithm.

4.3.2 TERC2 (the updated TERC algorithm)

The details of the TERC2 method are given in Algorithm 2. It follows the same steps as in Algorithm 1 for the TERC method, except for the step to find the charging station in case the battery level constraint is not satisfied. As shown in lines 12-16 of Algorithm 2, the closest charging station to the source node which is also closest to the destination node is chosen. This step optimizes the process of finding the nearest charging station to the source and destination nodes. Line 16 calculates T_{temp}, the time required to reach a charging station from the source node (i.e., T_{temp1}), plus the time to get to the destination node (i.e., T_{temp2}). Lines 16 to 18 find the charging station through which the distance from the source to the destination can be traversed in the least possible time. The advantage of the TERC2 method is that it can find a charging station on a route from the source to the destination without deviating much from the shortest path. In other words, it prevents

Algorithm 2: TERC2

Data: Graph $G(N, E)$, B_s, S, D.
Result: P_T: The fastest path by considering all constraints from S to D.

$T_D = 0$;
Function `FindPath`(S, D, B_S):
  $P_T \longleftarrow Dijkstra(S, D)$;
  $T_P \longleftarrow TimeOnPath(P_T)$;
  $B_{used} \longleftarrow BatteryUsedOnPath(P_T)$;
  if $B_i - B_{used} >= B_{min} + E_{toStation}$ **then**
    $T_D \leftarrow T_D + T_P$;
    return P_T
  else
    $T_P = L$;
    for *each* $i \in \mathcal{I}_{Station}$ **do**
      $P_{temp1} \leftarrow Dijkstra(S, i)$;
      $P_{temp2} \leftarrow Dijkstra(i, D)$;
      $T_{temp1} \longleftarrow TimeOnPath(P_{temp1})$;
      $T_{temp2} \longleftarrow TimeOnPath(P_{temp2})$;
      $T_{temp} = T_{temp1} + T_{temp2}$;
      if $T_P > T_{temp}$ **then**
        $T_P = T_{temp}$;
        $S_{new} \leftarrow i$;
    $B_{new} = B_{max}$;
    $T_D = T_D + T_{temp1} + G_i$;
    return $P_{temp1} + FindPath(S_{new}, D, B_{new})$

the EV from charging at a charging station which is located far from the shortest path between the source and destination although it is the closest to the source node. The time complexity of this algorithm is similar to the TERC algorithm.

4.3.3 KFP (K-Fastest Path)

The KFP approach presented in this subsection is the updated method of the K-shortest path that finds the fastest path between the source and destination nodes considering the EV battery level for the longevity of battery life. It begins by finding the K number of fastest paths and selects the best one that satisfies all battery constraints. The details of this method are given in Algorithm 3. In line 2 of the algorithm, the $KNearestPaths$ function is similar to finding the K shortest paths, it determines the K fastest paths between S and D, and stores all candidate paths in table P_{Kpaths}. The for loop in line 3 checks all the paths one by one from the fastest to the next fastest until it finds the one that satisfies all battery conditions. Line 4 measures the energy required to traverse the current path P. If

Algorithm 3: KFP

Data: Graph $G(N, E), S, D, B_s$

Result: $P =$ The fastest path from S to D

1 **Function** `FindPath`(S, D, B_S)**:**

2 $\quad P_{Kpaths} \longleftarrow KNearestPaths(S, D)$;

3 $\quad$ **for** *each* $P \in \mathcal{P}_{Kpaths}$ **do**

4 $\quad\quad B_{used} \leftarrow BatteryUsedOnPath(P)$;

5 $\quad\quad$ **if** $B_i - B_{used} > B_{min} + E_{toStation}$ **then**

6 $\quad\quad\quad$ Return(P)

7 $\quad\quad$ **else**

8 $\quad\quad\quad B_m = B_{used} - B_s$;

9 $\quad\quad\quad$ **for** *each* $i \in \mathcal{P}$ **do**

10 $\quad\quad\quad\quad B_{toi+1} \leftarrow BatteryUsed(i, i+1)$;

11 $\quad\quad\quad\quad B_{i+1} = B_i - B_{toi+1}$;

12 $\quad\quad\quad\quad$ **if** $B_{i+1} < B_{min} + E_{toStation}$ **then**

13 $\quad\quad\quad\quad\quad$ Jump to the next P;

14 $\quad\quad\quad\quad$ **if** $(i+1)$ *in* $I_{station}$ **then**

15 $\quad\quad\quad\quad\quad B_n = B_{max} - B_{i+1}$;

16 $\quad\quad\quad\quad\quad$ **if** $B_m > B_n$ **then**

17 $\quad\quad\quad\quad\quad\quad B_{i+1} = B_{max}$;

18 $\quad\quad\quad\quad\quad\quad B_m = B_m - B_n$;

19 $\quad\quad\quad\quad\quad$ **else**

20 $\quad\quad\quad\quad\quad\quad B_{i+1} = B_{i+1} + (B_m - B_n)$;

21 $\quad\quad\quad\quad\quad\quad$ Return(P)

the current energy level is sufficient to reach the destination and the remaining battery level is still above $B_{min} + E_{toStation}$, then the algorithm returns the current path and time and terminates. Otherwise, in line 8, the algorithm calculates the amount of energy shortage B_m required to reach the destination. The for loop in line 9 traverses all edges on the current path P, where at each node the remaining battery level is monitored (lines 10 and 11). Whenever, after traversing any node, the required energy which is $B_{min} + E_{toStation}$ is unsatisfied, then the current path is dropped, and the next shortest path from table P_{Kpaths} is retrieved (lines 12 and 13). When the next node after traversing an edge is a charging station, although the battery can be charged to the maximum allowable limit which is B_{Max}, it is not required to do that if there is enough battery to reach the destination and satisfied the battery level constraint.

Since the KFP method unlike the TERC and TERC2 methods does not find the next node step by step (node-by-node) and rather finds the entire path in advance, therefore it is not required to charge the EV to the maximum limit. Consequently, the required charging

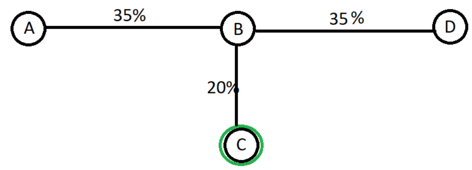


Figure 4.1: Example of the KFP issue.

level at each charging station on path P can be measured to save time and not waste time on extra charging; lines 16-18 check if it requires more than B_{Max} to reach the destination, then charge the EV to its maximum battery level. Otherwise, line 20 calculates the amount of energy required to charge to reach the destination and satisfy the battery constraint. Eventually, the algorithm terminates in line 21 when the battery energy level is sufficient to reach the destination.

Furthermore, as the K-Shortest-Paths algorithm will find all the possible routes from S to D, and based on the fact that there is always at least one correct route from S to D that does not violate our constraints, it is safe to say that the KFP algorithm will always find a route and hence its correctness is proven. The time complexity of the KFP algorithm is $O(KN(E + NlogN))$ similar to the K-Shortest-Path.

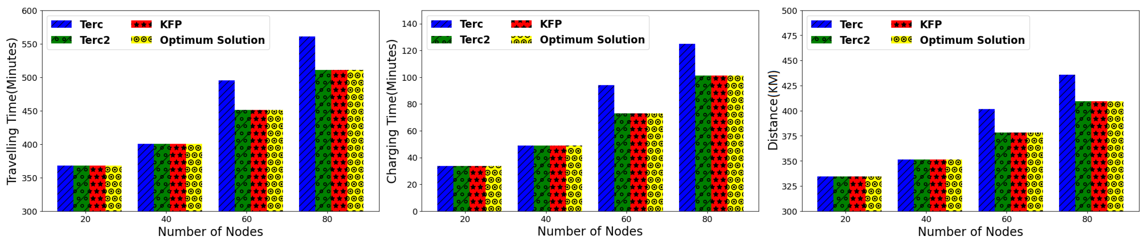
It is worth noting that the KFP method finds simple paths without revisiting any node twice. Therefore the direct implementation of this algorithm suffers a deficit because it conflicts with some scenarios where the EV has to visit an intersection more than once during its journey to the destination node. Figure 4.1 explains such a scenario. Assume that EV is at node A, node D is the destination node, and node C is a charging station. Let us also assume that the EV is required to charge at node C. Then, the KFP method can not find a path such that the EV goes from node A to the charging station (node C), and then from node C to the destination node (node D). That is because the EV has to traverse node B twice (once to reach node C from node A, and once when going to node D from node C). Therefore, the KFP can not solve such a case.

4.4 Performance Evaluation

To evaluate the performance of our heuristic approaches (TERC, TERC2, and KFP), in this section, we compare them first with the optimal solutions obtained from the optimization model (OPL) on small networks, and then we evaluate their performance on big instances with respect to the total traveling time, distance, charging time, and computation time. The inputs for our algorithm are derived from the specifications of the Tesla Model 3. For our evaluation, we consider a travel range of 545km per full battery, which reflects the distance that the Tesla Model 3 can cover under combined-mild weather usage. Furthermore, the average traveling speed is considered 60km per hour. All available charging stations are considered AC charging stations. This choice is based on the prevailing trend of AC charging being the most commonly used method for EV charging at the present time. While other charging technologies exist, such as DC fast charging, focusing on AC charging allows us to assess the algorithm's performance under typical charging infrastructure conditions. To calculate the charging process time, we made an assumption regarding the time required to charge the battery from 0 to 100 percent. In our evaluation, as the Tesla Model 3 can be fully charged in 8 hours and 15 minutes, To calculate the time required for charging each percent of the battery, we divide the total charging time by 100. While this assumption does not account for variations in battery charging speed due to factors such as different charging speeds based on battery level and temperature, it provides a reasonable approximation for evaluating the algorithm's performance within the scope of this study. To validate our proposed approaches based on battery life, empirical evaluations have been performed with the maximum and minimum battery level of $B_{max} = 80\%$ and $B_{min} = 20\%$ respectively. The initial charging level of the EV at the starting point is set to 80%.

4.4.1 Graph Generation

To construct graphs for performance evaluation of our proposed methods (i.e., creating nodes and weighted edges), we used Erdős–Rényi model [18] to generate random graphs with different sizes (number of nodes and edges). The number of edges is obtained by taking an appropriate probability of the existence of an edge between a pair of nodes. It is very crucial to select a good probability for edge creation so that the graph is not disconnected. Furthermore, the amount of battery energy consumption in percentage and time to traverse an edge is considered as the fixed weight of any edge. For simplicity, we consider the edge distance and vehicle speed to be fixed so that we can calculate the traveling time on each edge easily. After generating the graph and making sure that it is connected, we add several charging stations to the graph. First, we choose a random node in the graph as the first charging station and add it to the list $I_{Station}$. Then, for the next charging station, we select the farthest node in the radius of the first chosen one that can be reached with the maximum battery charged B_{Max}. Recursively, we repeat the same strategy from all charging stations in the $I_{Station}$ list until the entire graph has been covered and there is no part that cannot be reached with B_{Max}. Finally, we randomly choose a source node and a destination node on the graph with a specific range distance for the EV. In addition, to avoid having a scenario where the KFP method cannot find a route from the source to the destination nodes as explained earlier, we make sure that each charging



((a)) ((b)) ((c))

Figure 4.2: Comparing different heuristic methods(TERC, TERC2, and KFP) to Optimum Solutions by varying the size of the network by considering different performance metrics: (a) traveling time, (b) charging time, and (c) traveling distance.

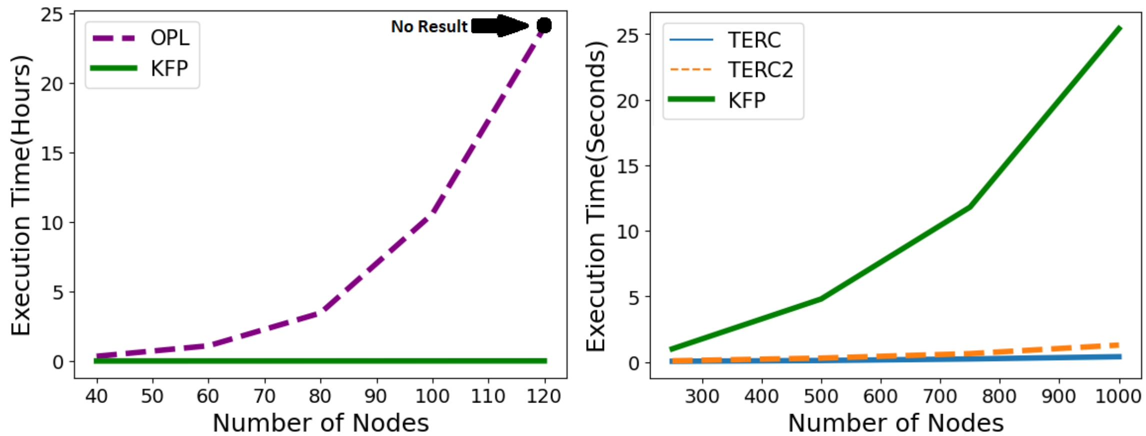
station has no dead-end path to avoid returning to the same visited node twice.

4.4.2 Evaluation over small Graphs

In this subsection, we evaluate the performance of our heuristic methods by comparing them to the optimization model (OPL). Figure 4.2 illustrates the gap between the three heuristic methods (TERC, TERC2, and KFP) to the OPL in terms of total travel time, distance, and charging time. The results are averaged over ten runs, where in each run, the source and destination have been chosen randomly in a range of 90 to 120 kilometers.

As shown in the three sub-figures for travel time, distance, and charging time, the performance of different methods is almost identical with the same variation as the size of the graph increases from 20 to 80 nodes. This is because as the EV takes longer routes, obviously more battery energy is consumed. It is also observable that both TERC2 and KFP reach the optimal solutions in tested small graphs. However, the TERC method performs a near-optimal solution with a maximum gap of 7.4%, and its results deviate slightly from OPL as the number of nodes increases.

These results are in the interest of proving the validity of our proposed methods, where in the case of large graphs, we expect that the heuristic methods will perform very close to optimal solutions. However, because of the complexity of the optimization method, we cannot obtain results for large instances as shown in Figure 4.3(a). The figure shows that the computation time of the optimization model (OPL) increases exponentially, starting with a few minutes for 40 nodes, increasing to more than 10 hours for 100 nodes, and eventually no results (failed to obtain results) for a slightly large graph of 120 nodes due to complexity of the optimization model and lack of memory. In contrast, one of the heuristic methods, like the KFP method, obtains results in less than a minute for the same size graph. Figure 4.3(b) illustrates the computation time for all proposed heuristic methods. In the figure, as the number of nodes increases, the computation time of the TERC and



((a)) ((b))

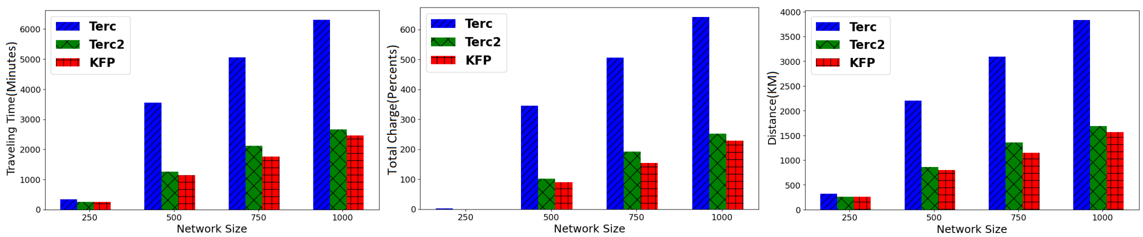
Figure 4.3: Comparing the execution time of (a) the optimization model (OPL) versus one of the heuristic methods (KFP) in small networks, and (b) the heuristic methods (TERC, TERC2, and KFP) in large networks.

TERC2 methods increases slightly (from a very few milliseconds for a graph with 300 nodes to a maximum of 14 seconds for a graph of size 1000 nodes). Whereas, the KFP method increases exponentially for large graphs (7.5 seconds to 76.5 seconds respectively for graphs of size 300 to 1000 nodes). In the next subsection, we will compare the performance of the heuristic methods in terms of total EV travel time, distance, and battery consumption.

4.4.3 Evaluation Over Large Graphs

For performance evaluation of heuristic methods on large graphs, the size is varied from 250 nodes to 1000 nodes. Figure 4.4 illustrates the total traveling time, distance, and battery charge of the three methods: TERC, TERC2, and KFP. It is to be noted that the total traveling time covers both traversing the path and charging the EV.

From both sub-figures (a and b), we can observe that as expected the total travailing time and battery consumption of the EV increase gradually with the size of the graph. The KFP outperforms the TERC2 very slightly (a maximum of 4.5% and 8% gap respectively for traveling time, battery charged, and distance when considering 1000 nodes and 750 nodes, respectively). However, the performance of the TERC is the worst among other methods in which its performance worsens as the size of the graph is enlarged. Luckily, the battery consumption of the TERC is leveled with other methods when evaluated with a graph of size 250 nodes. Similarly, the TERC2 method in terms of total traveling time leveled with the KFP method in a graph of 250 nodes. The simulation results in this section show that although the performance of the TERC2 method is slightly behind the KFP method in some cases, it is more reliable, efficient, and effective in solving the routing and charging



((a)) ((b)) ((c))

Figure 4.4: Comparing different heuristic methods(TERC, TERC2, and KFP) by varying the size of the network considering different performance metrics: (a) traveling time, (b) battery consumption, and (c) traveling distance.

problem for all graphs. That is because the KFP method fails to solve the routing and charging problem for some graphs as explained in Section 4.3.3, has a higher computation time, and it cannot be used for a dynamic environment since it requires finding the entire path in advance.

4.5 Summary and Future Directions

This chapter investigated the problem of routing and charging an electric vehicle to travel from a source to a destination in the shortest time, considering the maximum and minimum battery level for the longevity of its life and the energy required to reach the nearest charging station for the next trip. The problem has been formulated mathematically to obtain optimal solutions. However, due to the complexity of the optimization model, three heuristic approaches have been proposed: TERC, TERC2, and KFP. To evaluate the performance of the proposed methods, several performance metrics such as travel time, travel distance, battery consumption, and execution time, have been considered by varying the size of the graph. The numerical results showed that both TERC2 and KFP methods perform very close to the optimal solution. Furthermore, the KFP method in a few cases outperforms the TERC2 method in terms of traveling time and battery consumption. However, the former requires much more computation time and fails to find a path in some rare cases. Therefore, the latter is more reliable, effective, and efficient for solving the problem. In the next chapter, we use an AI-based reinforcement learning approach to integrate dynamic situations into the EV charging and route planning process. We also conduct extensive simulations to provide valuable insights into the strengths and limitations of our proposed approaches.

Chapter 5

Fastest Route and Charging Optimization of an Electric Vehicle With Battery's Life Consideration in a Dynamic Environment

In the preceding chapter, we explored the application of heuristic methods for determining optimal routes for Electric Vehicles (EVs) with a focus on charging considerations. The results obtained from the heuristic approach, particularly the TERC2 algorithm introduced in Chapter 4, marked a significant advancement in EV route planning. However, as we delve deeper into the intricacies of real-world EV operations, it becomes apparent that the environment in which these vehicles operate is far from static. Factors such as road speeds, traffic, and battery consumption exhibit variability that is contingent on the time of day, introducing a dynamic aspect to the problem.

In this chapter, we embark on an exploration of the dynamic nature of the environment in EV routing and charging optimization. Recognizing that the performance of routing algorithms can be greatly affected by these temporal fluctuations, we extend our investigation to adapt our methodologies to such dynamic scenarios. We commence by providing a concise recapitulation of the heuristic methods employed in Chapter 4 as a foundation for the subsequent advancements. It is crucial to underscore the fundamental challenge posed by the dynamic nature of the EV routing problem. This chapter offers an in-depth analysis that demonstrates the inherent complexity of the issue. We rigorously establish that the dynamic EV routing and charging optimization problem is NP-hard, further emphasizing the need for innovative techniques capable of addressing these intricacies effectively.

A primary contribution of this chapter lies in the integration of reinforcement learning techniques to tackle the dynamic EV routing problem. Recognizing that traditional heuristic methods might fall short in capturing the real-time adaptability required for dynamic environments, we delve into the principles of reinforcement learning. By leveraging reinforcement learning algorithms, we propose an approach that enables EVs to make intelligent decisions on the fly, accounting for varying road conditions and battery consumption patterns. Moreover, we enhance our existing heuristic approach, TERC2, by imbuing it with the capability to dynamically adjust its decision-making process. We detail the modifications and extensions made to TERC2, showcasing how it now incorporates real-time information to make more informed routing and charging decisions. This updated version, termed TERC2-Dynamic, stands as a testament to our commitment to refining and adapting existing methodologies to align with the ever-changing nature of the problem.

To provide a comprehensive assessment of the innovations presented in this chapter, we conduct a thorough comparative analysis. We contrast the performance of our reinforcement learning-based approach, TERC2-Dynamic, with not only the baseline random method but also the original TERC2 algorithm and its performance in dynamic settings. By doing so, we present a holistic view of the advancements made and the improvements achieved. Nevertheless, the new heuristic approaches still exhibit a higher likelihood of finding a worse path or encountering dead-end situations compared to the reinforcement learning approach. Moreover, our study demonstrates that the Q-learning-based solution outperforms both versions of the heuristic method, offering a potential path toward more robust and adaptable route planning for electric vehicles. These insights contribute to the ongoing effort to optimize electric vehicle routing strategies.

5.1 Problem Description

In our context, the environment of the EV routing problem has a dynamic nature characterized by constant changes. Among these changes are several critical factors, including but not limited to the waiting time required at charging stations, the velocity of traffic flow along different road segments, and battery consumption rate. It's imperative to acknowledge that these variables evolve over time, thereby shaping a fluctuating landscape in which our problem operates.

Given this dynamic setting, in this chapter, our objective is centered on identifying the optimal route for an Electric Vehicle (EV) in a dynamic environment. The central challenge lies in adeptly addressing these ever-shifting elements to craft a path that takes into account the variations in the environment. Furthermore, this problem's complexity is further underscored by its classification within the realm of NP-hard problems. To prove this, we draw a connection to a well-known NP-hard problem that shares similarities with our situation. Imagine the "Traveling Salesman Problem" (TSP), where you aim to find the shortest path that visits all cities and returns to the start. Our EV problem is somewhat akin to TSP but with a crucial difference. Instead of needing to visit every charging station, the EV only stops at these stations when it requires more charge to reach its destination. This flexibility introduces a distinct challenge, setting it apart from the classical TSP. Our demonstration lies in showcasing that the complexity inherent to TSP can be mirrored in our problem. While not identical, the intricacies of determining the optimal path considering state-dependent charging align with the NP-hardness of TSP-like problems. This connection highlights the computational difficulties involved in discovering the optimal route for an EV in a shifting environment. This has far-reaching implications for our approach to solving this problem. While it isn't a pure TSP, the shared complexity indicates the need for advanced strategies and creative techniques to devise efficient solutions. In the subsequent sections, We aim to develop practical solutions tailored to the nuances of dynamic EV routing with state-dependent charging, thereby tackling a problem that finds its roots in NP-hard domains.

5.2 Dynamic TERC2

To cope with the changes and dynamic environment, in this section, we present the dynamic TERC2 method, which is the updated method of the TERC2 introduced in Chapter 4 and detailed in Algorithm 2. The dynamic TERC2 approach's specifics are outlined in Algorithm 4. It mirrors the steps of the TERC2 method, with an exception when pinpointing the subsequent charging station in scenarios where battery capacity isn't met. This distinction is evident in line 4 of the algorithm, where information concerning links (like traffic speed and battery consumption rate) gets refreshed. This responsive update equips the algorithm to adapt to environmental changes and revise its decisions, all geared towards optimizing route choices to the destination or the upcoming charging station. This alteration notably streamlines the process of locating the nearest charging station from both the source and destination nodes.

To detail the updates of the dynamic TERC2, line 17 calculates T_{temp}, which signifies the time required to reach a charging station from the source node (T_{temp1}), combined with the time needed to reach the destination node (T_{temp2}). Lines 17 to 19 collectively identify the charging station that enables the shortest possible time for traversing the distance between the source and destination points. The notable advantage of this algorithm lies in its ability to steer the EV away from charging stations or destinations that were previously efficient to access at the outset of the journey but have now become time-consuming due to alterations in traffic conditions.

Algorithm 4: Dynamic TERC2

Data: Graph $G(N, E)$, B_s, S, D.

Result: P_T: The fastest path in Dynamic Environment by considering all constraints from S to D.

$T_D = 0$;
Function `FindPath`(S, D, B_S)**:**
$\bar{E} \longleftarrow E_{state}$;
$P_T \longleftarrow Dijkstra(S, D)$;
$T_P \longleftarrow TimeOnPath(P_T)$;
$B_{used} \longleftarrow BatteryUsedOnPath(P_T)$;
if $B_i - B_{used} >= B_{min} + E_{toStation}$ **then**
    $T_D \leftarrow T_D + T_P$;
    return P_T
else
    $T_P = L$;
    for *each* $i \in \mathcal{I}_{Station}$ **do**
        $P_{temp1} \leftarrow Dijkstra(S, i)$;
        $P_{temp2} \leftarrow Dijkstra(i, D)$;
        $T_{temp1} \longleftarrow TimeOnPath(P_{temp1})$;
        $T_{temp2} \longleftarrow TimeOnPath(P_{temp2})$;
        $T_{temp} = T_{temp1} + T_{temp2}$;
        if $T_P > T_{temp}$ **then**
            $T_P = T_{temp}$;
            $S_{new} \leftarrow i$;
    $B_{new} = B_{max}$;
    $T_D = T_D + T_{temp1} + G_i$;
    return $P_{temp1} + FindPath(S_{new}, D, B_{new})$

5.3 The Reinforcement Learning Approach

Reinforcement learning encompasses an agent's interaction with an environment to acquire a strategy that maximizes cumulative rewards over time [19]. In our approach, we conceptualize our predicament as a Markov Decision Process (MDP). This MDP is characterized by a tuple (S, A, γ, P, R), where: S represents a finite set of states, denoted as s_t at time t; A denotes the set of possible actions in state s_t, with $a_t \in A$; γ, a value between 0 and 1, reflects the discount factor, governing the influence of future rewards in the decision-making procedure; P signifies the Markovian transition model, denoted as $P(s_{t+1} \| s_t, a_t)$, quantifying the likelihood of transitioning from state s_t to s_{t+1} upon taking action a_t; R signifies the reward distribution, noted as $P(r_t \| s_t, a_t)$, offering an immediate reward $r_t \in R$ when action a_t is executed in state s_t at time t. The functions for state, action, and reward under the MDP components are as follows:

- Agent: In this context, the agent corresponds to the Electric Vehicle (EV) itself. The EV is responsible for making decisions at each step of its journey, selecting actions that will navigate it through the dynamic environment while aiming to minimize travel time.

- State: The state space encompasses the different situations or conditions in which the EV can find itself. Specifically, a state is defined by four key attributes: the EV's location within the environment and the road structure (represented by nodes in the road network), Ev's current battery level, and Ev's current traveling time. Furthermore, the destination is considered the goal of the search problem to determine whether the EV reached the terminal state or not. These factors collectively provide the necessary information for the agent to make informed choices regarding its next steps.

- Actions: The actions available to the agent involve the EV's movement within the environment. The EV can travel to neighboring nodes. Additionally, when the EV is situated at a charging station, there's an extra action available: the option to charge its battery.

- Reward: The reward function is designed to align with the overarching goal of minimizing travel time. As such, the reward is defined as the negative of the time expended. The objective is to encourage the agent to take actions that lead to the shortest travel time. Time, in this context, can be thought of as the sum of two components: travel time, which is the time taken to move between nodes, and charging time at stations. Furthermore, there is a big penalty (negative reward) for times when the EV's battery level goes below 20 or above 80 percent of its capacity.

By structuring the problem within the MDP framework, the EV, as the agent, learns the policy on how to strategically navigate the environment to minimize its travel time by iteratively taking actions that lead to more favorable states and consequently accumulating higher rewards.

The key parameters for our Q-learning algorithm are as follows:

- Number of Epochs: 2500

- Learning Rate: 0.1

- Discount Factor: 0.8

- Epsilon (Exploration Rate): 0.05

Algorithm 5: Proposed Q Learning

Data: Graph $G(N, E), B_s, S, D$.

Result: P_T: The fastest path in Dynamic Environment by considering all constraints from S to D.

$T_D = 0$;
$Q(s, t) \leftarrow \emptyset$;
for $k \leftarrow 1 : K$ **do**
  $N_{current} \leftarrow S$;
  $P = \leftarrow N_{Current}$;
  while *true* **do**
    Update($G(N, E)$);
    Observe the environment;
    Choose an action(a) from a list of possible actions using an ϵ Greedy policy, allocate and choose the next node (N_{next});
    $P \leftarrow P + N_{next}$;
    $R \leftarrow R - T_a$;
    Update $Q(s, a)$¸$Q(s, a) + \alpha[r + \gamma \, max \, a'Q(s + 1, a + 1) - Q(s, a)]$;
    $N_{current} \longleftarrow N_{next}$;
    if $N_{current} = D$ **then**
      Break;
Execute trained Q-Learning agent on environment interface.
Return route from S to D based on Q Table

The details of the Q-Learning algorithm are given in Algorithm 5, and the interaction between the agent and the environment is illustrated in Figure 5.3. The algorithm begins by initializing the Q-values for each state-action pair. The state space comprises the EV's location within the environment and its current battery level. Actions involve moving to neighboring nodes, reaching charging stations, and charging when available. The Q-values represent the estimated rewards associated with taking specific actions in different states.

At each epoch, the EV, acting as the agent, selects an action based on an exploration-exploitation trade-off. The exploration rate, epsilon, determines the probability of taking a random action, encouraging the agent to explore new possibilities. As the algorithm progresses, the EV refines its strategy through a learning process that involves updating the Q-values according to the observed rewards and the optimal future Q-values.

One distinctive feature of our approach is the frequent updating of the entire environment after each action. This dynamic update ensures that the EV learns to adapt to the ever-changing conditions, grasping patterns of changes that impact the environment's state transitions and rewards. By considering the historical changes and their effects, the agent can better discern the most favorable route to the destination.

The algorithm's input consists of the following components: the source node (S), the

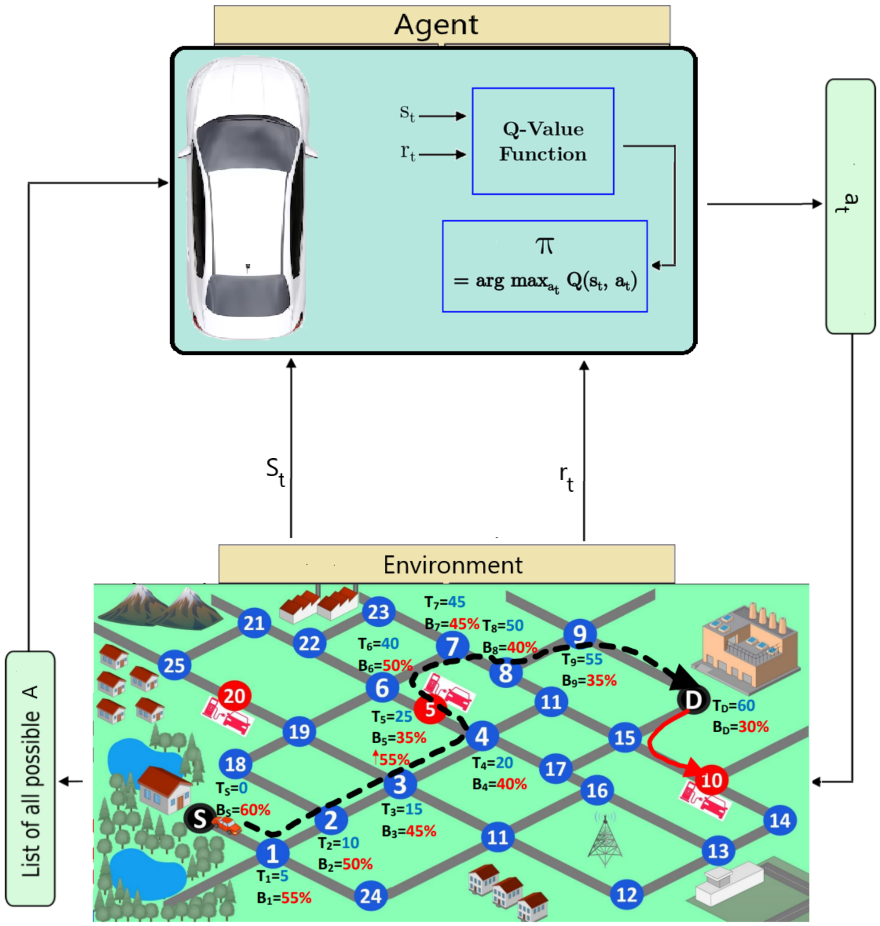


Figure 5.1: The interaction between the Agent and the Environment

destination node (D), the set of nodes (N), and the links between them (E). The output of the algorithm is the selected path, complete with the necessary charging information, effectively guiding the EV through the dynamic environment while minimizing travel time.

Through the integration of Q-learning, our approach empowers the EV to make informed decisions based on learned patterns, leading to an adaptive and efficient route selection that accounts for the intricate interplay between node choices, battery management, and changing environment dynamics.

5.4 Performance Evaluation

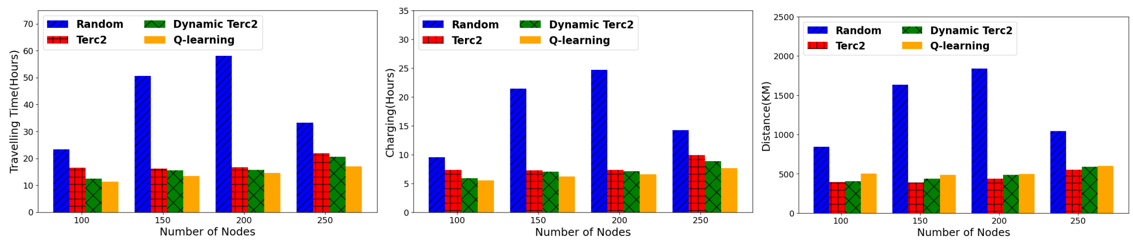
As it is extremely difficult to obtain optimal solutions using the optimization model for large graphs due to the lack of memory and computation time, to evaluate the performance of our RL and the Dynamic TERC2 method, in this section, we compare them with the static TERC2 algorithm. We evaluate their performance in dynamic environments with respect to the total traveling time, distance, and battery consumption, by varying the size of the graph, the number of charging stations, and the distance between the source and the destination. To validate our proposed approaches based on battery life, empirical evaluations have been performed with the maximum and minimum battery level of $B_{max} = 80\%$ and $B_{min} = 20\%$ respectively. The initial charging level of the EV at the starting point is set to 80%.

5.4.1 Graph Generation

To construct graphs for performance evaluation of our proposed methods (i.e., creating nodes and weighted edges), we used a real map to generate a graph with different sizes (number of nodes and edges). To have a specific number of nodes, we used a partial of the generated graph that contained the wanted number of nodes. Nowadays, with the EV battery range, an EV can travel within a city from any single source to a destination without the need to charge the battery. Hence, for our experiment, we had to multiply the distance between edges to be able to test our algorithms' performances in finding charging stations within the way. Furthermore, we assigned three different pairs of route traffic speed and battery consumption for each edge based on their distance. By doing this, we can simulate the dynamic nature of traffic for routes and change the speed and battery consumption of the EV at each time sluts. Then we considered some of the nodes as charging stations in a way that all nodes have at least one charging accessible station in the worst scenario (when the battery consumption of that route is the highest) to charge if it is needed. By doing this, we make sure there is no dead end for all tested algorithms. To add the charging stations, first, we choose a random node in the graph as the first charging station and add it to the list $I_{Station}$. Then, for the next charging station, we select the farthest node in terms of battery consumption in the radius of the first chosen one that can be reached with the maximum battery charged B_{Max}. Recursively, we repeat the same strategy from all charging stations in the $I_{Station}$ list until the entire graph has been covered and there is no part that cannot be reached with B_{Max}. Finally, we randomly choose a source node and a destination node on the graph with a specific range distance for the EV. For simplicity, we considered three different traffic statuses for each road.

5.4.2 Performance evaluation over different network sizes

In this subsection, we evaluate the performance of our methods by comparing them to the fastest path (TERC2) evaluated in Chapter 4 for a static environment. Figure 5.2 illustrates the gap between the two heuristic methods (TERC2 and dynamic TERC2) to the Q-earning algorithm in terms of total travel time and charge time. As we can see, though Q-Learning takes a longer path compared to the other two algorithms, it reaches the destination faster and with less battery consumption because of choosing the path based on the traffic status.



((a)) ((b)) ((c))

Figure 5.2: Traveling time (a), Charging Time (b), and distance (c) as comparison metrics for different methods (TERC2, Dynamic TERC2, Q-Learning, and Random) by varying the size of the network.

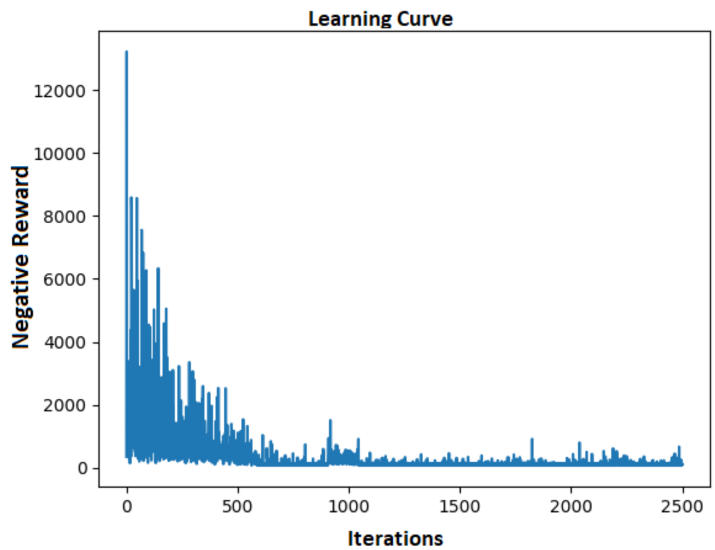
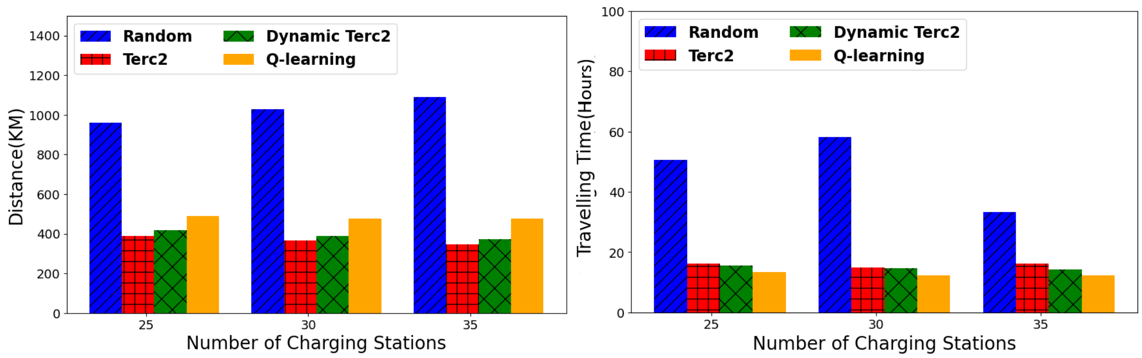


Figure 5.3: The learning rate of the Q-Learning algorithm to find the best path from a source to a destination.

As shown in all three sub-figures for travel distance and battery charge time, although the dynamic TERC2 performs better than its static version, it performs worse than the Q-Learning algorithm in terms of traveling time. Furthermore, as we can observe from the figures, in a dynamic environment, the shortest path is not necessarily the fastest path due to the several traffic statuses.

In Figure 5.2, we explore how different methods—TERC2, Dynamic TERC2, Q-learning, and Random—perform across a range of charging station counts within the road network. These comparisons help us understand which methods fare better in terms of both the distance of paths and the time it takes to travel those paths.

Just like our earlier comparisons based on the number of nodes, we can observe that the Random methods consistently give the worst results in both distance and traveling time. On



((a)) ((b))

Figure 5.4: Traveling time and distance as comparison metrics for different methods (TERC2, Dynamic TERC2, Q-Learning, and Random) by varying the number of charging stations.

the other hand, TERC2 and Dynamic TERC2 methods, in that order, offer shorter paths. However, there's a trade-off – their traveling times are longer compared to the outcomes from the Q-learning approach.

Interestingly, as we increase the number of charging stations within the same network, we notice a growing difference between Q-learning results and the results from the other methods in terms of traveling time. This can be explained by the interplay between the available paths and how the charging stations are spread out. When there are only a few charging stations, the possible routes are limited, leading to algorithmic results that are more similar to each other. However as the number of charging stations increases, the flexibility in choosing paths becomes greater. This means that TERC2's paths become shorter and closer to the shortest route (like Dijkstra's algorithm), but this doesn't necessarily mean that travel times improve. Additionally, with more charging stations, the trained Q-learning approach excels in finding quicker paths as it has more travel options to choose from.

In a nutshell, these comparisons offer insights into how different methods react to varying numbers of charging stations, showcasing the strengths and trade-offs of each approach in terms of both distance and traveling time metrics.

These results are in the interest of proving the validity of our Q-Learning method, where in the case of a dynamic environment, the heuristic methods lack learning from the environment changes and consider the future changes before they happen. The learning curve of the reinforcement learning technique is depicted in Figure 5.3. On the x-axis, you can observe the episode numbers, while the y-axis showcases the progression of negative rewards. The experiment was conducted on a road network encompassing 250 nodes. In the initial episode, the methods commence with approximately 14,000 negative rewards, which progressively decrease as they engage with the environment and accumulate insights. Given the rapid fluctuations in the environment after each action, the methods exhibit swift learning dynamics, swiftly adapting to changing circumstances and achieving reduced negative rewards in the initial episodes. As the graph illustrates, the methods exhibit a convergence of rewards, notably becoming apparent from epoch 12,000, indicating the culmination of training efforts.

5.4.3 Performance evaluation Over a different number of charging stations

For performance evaluation of our methods on the same network with different numbers of charging stations, first, we create a graph with the minimum number of possible charging stations (i.e., 25 charging stations). Then we add 5 and then 10 extra charging stations randomly in the map to observe the changes in the behavior of algorithms. In this comparison, the size of the graph nodes (i.e., 250 nodes) and the number and placement of edges are fixed.

Figure 5.4 illustrates the total traveling time and distance of the four methods: TERC2, dynamic TERC2, Q-Learning, and random. From both sub-figures, we can observe that as expected there is no useful intel in the random approach as it will be different every time it runs and all other algorithms outperform it by a high margin. Furthermore, as we can see, as we increase the number of charging stations, the heuristic methods' performance start to get better as increasing the number of charging stations will give more possible paths to take, and eventually they can have shorter distance. On the other hand, though the Q-Learning algorithm performs slightly better than when it runs with fewer charging stations, its performance is more stable and there are no significant changes when the number of charging stations is between 25 and 30. The simulation results in this section show that although the performance of the TERC2 and dynamic TERC2 can improve by adding more charging stations, they still underperform the Q-Learning algorithm.

5.4.4 Performance evaluation over rural and urban areas

In this section, we generated two networks to observe the performance of different algorithms in rural and urban areas. The rural network was characterized by longer distances between nodes, spanning 50 to 150 kilometers. This elongated nature of the highways demanded a higher frequency of charging stations to accommodate the extended travel between waypoints. In contrast, the urban network comprised shorter distances, ranging from 0 to 50 kilometers, reflecting the intricacies of urban travel.

As is shown in Figure 5.5, the discrepancy in charging station density between these two environments was a direct reflection of the inherent differences in travel requirements.

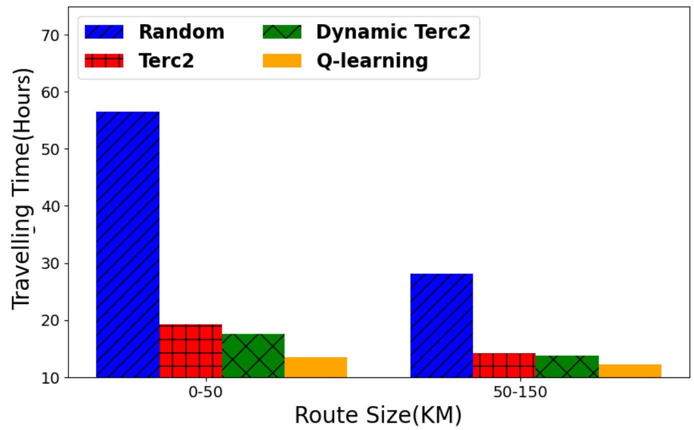


Figure 5.5: Comparison of all algorithms (TERC2, Dynamic TERC2, Q-Learning, and Random) in both rural and urban areas in terms of traveling time

The highway network necessitated more frequent charging opportunities due to the substantial distances covered between nodes, aligning with the greater energy demands of EVs traversing extended highway segments. Conversely, in the city, the network exhibited comparatively lower charging station frequency, reflecting the relatively shorter distances between waypoints and the potential for regenerative braking within urban settings.

The comparative analysis revealed intriguing insights into the performance of the heuristic methods and the Q-Learning approach across these two distinct driving scenarios. As anticipated, the heuristic methods exhibited performances that were relatively closer to each other on rural highways. This outcome was aligned with the constrained nature of highway driving, where fewer alternative routes and lower variability in driving speeds left less room for algorithmic differentiation. However, even within the confines of the rural environment, the heuristic methods still demonstrated a discernible performance gap when compared to the Q-Learning approach.

The findings from this evaluation underscore the robustness of the heuristic methods in highway scenarios, while also emphasizing the superior performance achieved by the Q-Learning approach. Despite the inherently constrained nature of highway travel, where algorithmic freedom is curtailed, the Q-Learning approach exhibited a notable advantage in optimizing EV routing. This suggests the potential of reinforcement learning techniques to adapt and excel in real-world scenarios where dynamic changes and intricate decision-making are paramount. In summary, the evaluation across rural and urban areas sheds light on the adaptive prowess of the proposed algorithms, reaffirming the overarching goal of optimizing electric vehicle routing in diverse driving environments. The subsequent sections delve into the nuanced analysis of these findings and provide deeper insights into the mechanisms underlying the observed performance disparities.

5.5 Summary and Future Directions

In this chapter, we delved into the dynamic realm of EV routing and charging optimization. We aimed to address the challenges posed by fluctuating factors like changing road speeds and battery consumption rates. To tackle this, we harnessed reinforcement learning techniques to ensure real-time adaptability in EV routing.

Building upon our earlier heuristic methods, we embraced the complexity of dynamic EV routing, establishing its NP-hard nature. This propelled us to integrate reinforcement learning principles, empowering EVs to make agile decisions in response to evolving conditions, resulting in enhanced routes and charging strategies. We also upgraded our heuristic approach, yielding TERC2-Dynamic, which effectively incorporated real-time data for improved decision-making. This enhancement underscored our commitment to refining methodologies to match the evolving EV landscape. Comparing our innovations, TERC2-Dynamic emerged as a front-runner, showcasing superior performance in dynamically shifting scenarios. Notably, the Q-learning-based approach outperformed both heuristic versions, suggesting a path towards agile and responsive EV route planning.

These findings promise a more adaptive future for electric mobility by optimizing routing strategies in dynamic environments.

Chapter 6

Conclusion

In the pages of this thesis, our mission centered on the pursuit of the optimal routes for Electric Vehicles (EVs) with the dual considerations of swiftness and charging logistics. As the automotive landscape evolves, the integration of electric mobility becomes increasingly vital, and our work sought to address the practical challenges intertwined with this transformation.

Our exploration commenced with the ambitious aim of identifying the fastest paths for EVs to traverse from their origins to destinations, accounting for the intricate nuances of charging requirements. The objective extended beyond mere traversal times; we ensured that each EV harbored the minimum requisite charge upon reaching a destination, guaranteeing its viability to proceed to the next charging station for impending journeys.

Mathematical formalization emerged as a cornerstone of our methodology, paving the way to an optimal solution. Yet, the complexity embedded within this challenge unveiled its formidable nature—resulting in protracted computation times and limitations in scalability, especially concerning expansive road networks. To address these obstacles, we introduced three distinct heuristic methods, each representing a strategic approach to solving the pathfinding conundrum. Notably, the TERC2 method emerged as a frontrunner, closely approximating the optimal solution while exhibiting remarkably accelerated execution times.

Acknowledging the dynamic nature of real-world environments, we transcended the confines of static models. Our endeavor encompassed the infusion of dynamic attributes into our environment—attributes susceptible to temporal shifts, such as traffic speeds and battery consumption rates. However, this dynamic environment brought forth a fresh challenge: the TERC2 algorithm's performance faltered as it failed to adapt to updated inputs. To address this, we innovated a solution that factored in changes in the road map, enhancing the algorithm's responsiveness to evolving scenarios.

In our exploration of computational complexities, we revealed the problem's classification as NP-hard—a hallmark of its intricate nature. Leveraging a reinforcement learning approach, specifically Q-learning, we tackled the intricate dynamics intrinsic to EV routing. Our simulations, conducted across highway and city road maps, enabled a comprehensive assessment of these algorithms under diverse parameters.

As the results unfolded, a clear pattern emerged. The updated TERC2 variant exhibited marked improvements over its predecessor, showcasing heightened prowess within our dynamic environment. Yet, the spotlight remained firmly fixed on the Q-learning approach, showcasing superior performance across all examined scenarios.

In closing, our convictions are steadfast: Electric Vehicles stand as the vanguard of future mobility. As we tread further into this transformative era, the imperative to uncover optimal solutions becomes increasingly pronounced, essential for minimizing waiting times at charging stations and extending battery life—the pinnacle of an EV's longevity. With this in mind, we believe our contributions mark a significant stride towards charting the most advantageous routes, resonating with the broader vision of sustainable and efficient electric mobility.

Appendix A

List of Abbreviations

Appendix B

Examiner's Comments and Responses

Bibliography

[1] J. Morris, D. Hone, M. Haigh, A. Sokolov, and S. Paltsev, “Future energy: in search of a scenario reflecting current and future pressures and trends,” *Environmental Economics and Policy Studies*, vol. 25, 2022.

[2] M. N. Boukoberine, T. Donateo, and M. Benbouzid, “Optimized energy management strategy for hybrid fuel cell powered drones in persistent missions using real flight test data,” *IEEE Transactions on Energy Conversion*, vol. 37, no. 3, pp. 2080–2091, 2022.

[3] A. Mirheli and L. Hajibabai, “Charging network design and service pricing for electric vehicles with user-equilibrium decisions,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, pp. 2888–2902, 2023.

[4] S. Zhang, Y. Gajpal, S. Appadoo, and M. Abdulkader, “Electric vehicle routing problem with recharging stations for minimizing energy consumption,” *International Journal of Production Economics*, vol. 203, pp. 404–413, 2018.

[5] G. Hiermann, J. Puchinger, S. Ropke, and R. F. Hartl, “The electric fleet size and mix vehicle routing problem with time windows and recharging stations,” *European Journal of Operational Research*, vol. 252, pp. 995–1018, 2016.

[6] M. Bruglieri, M. Paolucci, and O. Pisacane, “A matheuristic for the electric vehicle routing problem with time windows and a realistic energy consumption model,” *Computers & Operations Research*, vol. 157, 2023.

[7] M. Schneider, A. Stenger, and D. Goeke, “The electric vehicle-routing problem with time windows and recharging stations,” *Transportation Science*, vol. 48, pp. 500–520, 2014.

[8] M. Masikos, K. Demestichas, E. Adamopoulou, and M. Theologou, “Machine-learning methodology for energy efficient routing,” *IET Intelligent Transport Systems*, vol. 8, pp. 255–265, 2014.

[9] C. De Cauwer, W. Verbeke, J. Van Mierlo, and T. Coosemans, "A model for range estimation and energy-efficient routing of electric vehicles in real-world conditions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, pp. 2787–2800, 2020.

[10] C. De Cauwer, W. Verbeke, T. Coosemans, S. Faid, and J. Van Mierlo, "A data-driven method for energy consumption prediction and energy-efficient routing of electric vehicles in real-world conditions," *Energies*, vol. 10, 2017.

[11] X. Bi, A. J. Chipperfield, and W. K. S. Tang, "Coordinating electric vehicle flow distribution and charger allocation by joint optimization," *IEEE Transactions on Industrial Informatics*, vol. 17, pp. 8112–8121, 2021.

[12] C. Cataldo-Díaz, R. Linfati, and J. W. Escobar, "Mathematical model for the electric vehicle routing problem considering the state of charge of the batteries," *Sustainability*, vol. 14, 2022.

[13] M. M. Y. Ma, "Route optimization of electric vehicle considering soft time windows and two ways of power replenishment," 2020.

[14] W. Z. L. Wang, "The energy-efficient dynamic route planning for electric vehicles," 2019.

[15] A. B. Alberto Ponso and G. Belingardi, "Route planning for electric vehicles including driving style, hvac, payload and battery health," 2023.

[16] N. W. Marina Dorokhova, Christophe Ballif, "Routing of electric vehicles with intermediary charging stations: A reinforcement learning approach," vol. 4, 2021.

[17] N. Lech and P. Nikończuk, "The method of route optimization of electric vehicle," *Procedia Computer Science*, vol. 207, 2022.

[18] C. Seshadhri, T. G. Kolda, and A. Pinar, "Community structure and scale-free collections of erdős-rényi graphs," *Physical Review E*, vol. 85, no. 5, p. 056109, 2012.

[19] W. Zaher and et al., "Omnidirectional-wheel conveyor path planning and sorting using reinforcement learning algorithms," *IEEE Access*, vol. 10, pp. 27 945–27 959, 2022.