POLITECNICO DI TORINO

Master's Degree in Automotive Engineering

Master's Degree Thesis

Cluster - Based Approach for Route Planning Optimization of Electric Vehicles

Supervisors

Prof. Angelo BONFITTO

Alberto PONSO

Candidate

LUCA ARGIOLAS

October 2023

Summary

This Master's Thesis presents a comprehensive exploration of an innovative project aimed at enhancing minimum travel time path research algorithms for electric vehicles. The algorithm used for defining the shortest path is Dijkstra's algorithm, which is implemented on a cost matrix constructed by assigning a specific recharging station within a map to each column and row. The value contained in each cell is the sum of the travel time taken from the station connected to the column to the station connected to the row, to which is added the recharging time required to restore the battery to its initial charge level.

The first part of this project focuses on constructing a vehicle and battery recharging model to ensure compliance with the vehicle's range limits, as well as accurately modeling the battery recharging profile for better evaluation of recharging time. This involves defining a vehicle model that includes: a more accurate assessment of the power required at the wheels by incorporating an inertial coefficient to account for the rotational inertia of driveline components; constructing an efficiency map of the electric motor based on its nominal parameters; defining a regeneration coefficient to modulate the regenerative capacity of the vehicle based on its driving speed, and specifying a non-constant recharging profile using the CP-CV charging protocol.

The second part of this project focuses on reducing the processing time involved with the construction of the cost matrix, which will subsequently be used in the Dijkstra's algorithm. This reduction is primarily achieved by clustering the recharging stations based on their geographic density using the DBSCAN algorithm, thereby reducing the number of recharging stations to be considered within the cost matrix in areas with high density of recharging stations. Additionally, to the cost matrix obtained with clustered recharging stations, a series of pruning techniques were further applied. The objective of these techniques is to reduce the number of recharging stations considered to only those that are essential for defining the desired route. The primary function of these pruning techniques is to deem as non-essential, for defining the minimum travel path, all recharging stations whose geographic coordinates fall outside a defined straight corridor with a specified width, beginning and ending at the departure and arrival coordinates of

the desired journey.
Overall, the project's innovations significantly improved the original program, with potential applications in artificial intelligence and route planning, combining computational approaches with cognitive principles to enhance efficiency and reduce processing times. This research marks a significant advancement in the domain of electric vehicle route planning, offering valuable insights for future developments in sustainable transportation and artificial intelligence.

Acknowledgements

"I would like to express my deep gratitude to the people who have supported me in achieving this milestone, both those who have been by my side from the beginning and those who have joined me more recently.
I want to start by thanking my little brother, who has encouraged me to be the best version of myself since my earliest childhood, in order to be a worthy example for him. A special thanks goes to my parents, who have educated and guided me throughout my life, constantly enlightening me on the various aspects of life.
In addition to my family members, I want to express my gratitude to Professor Angelo Bonfitto and Alberto Ponso, who with meticulous dedication have guided me through every phase of this project, making it possible.
Finally, I wish myself a future filled with successes, believing that this achievement is just the beginning of an even brighter journey."

Table of Contents

List of Tables

List of Figures

Acronyms

EV

Electric Vehicle

EM

Electric Motor

BEV

Battery-Powered Electric Vehicle

SoC

State of Charge

SoH

State of Health

CC-CV

Constant Current Constant Voltage

CP-CV

Constant Power Constant Voltage

IM

Induction Motor

BLDC

Brushless DC Motor

HVAC

Heating, Ventilation, and Air Conditioning

CSP

Constrained Shortest Path

NP

Non-Deterministic Polynomial

DBSCAN

Density-Based Spatial Clustering of Applications with Noise

OSM

OpenStreetMap

Chapter 1

Introduction

For more than a century, the world has been under the dominance of cars propelled by combustion engines. Nevertheless, the widespread proliferation of these vehicles has led to various public health and environmental issues, necessitating the imperative formulation and implementation of stringent environmental policies.

The main issue is that ICE vehicles burn fossil fuels like gasoline and diesel, releasing a mix of harmful substances into the air [1]. This encompasses particulate matter (PM), nitrogen oxides (NOx), carbon monoxide (CO), sulfur dioxide (SO2) and greenhouse gases, including carbon dioxide (CO2).

These emissions create deleterious consequences for public health [1]. For instance, PM represents a particularly menacing threat due to its capacity to infiltrate the pulmonary recesses, thereby instigating an array of respiratory maladies. NOx irritate the respiratory tract, weakening the body's innate immunological defenses against pathogenic incursions. CO, a lethal gas, obstructs the essential oxygenation of vital organs. Furthermore, the release of benzene, a noxious air pollutant, has been correlated with carcinogenic outcomes.

The atmospheric contamination within urban agglomerations, affects a substantial fraction of the population since it lives in areas afflicted by sub-optimal air quality indices [2]. Such areas manifest high concentrations of ozone, PM and emissions leading to the formation of atmospheric smog. The health problems stemming from this pollution mean higher healthcare costs [1], from treating respiratory illnesses to more severe conditions. Furthermore, the problem of people dying early due to long exposure to bad air puts a heavy load on healthcare system.

In response to these imperatives, national governments and international regulatory organizations have introduced strict environmental policies [3], with the express aim of reducing emissions from ICE vehicles and promote cleaner ways of getting around. One big solution is transitioning to electric vehicles (EVs). Unlike traditional cars, EVs produce zero emissions at the tailpipe, making them much cleaner for the environment. However, the efficacy of this transition is linked to the migration of power generation from fossil fuels to renewable and sustainable sources, including wind, solar, and hydroelectric power.

In the late 19th century, EVs gained popularity alongside steam and gasoline vehicles. Steam had a slow startup and limited range, while gasoline cars were challenging to operate, noisy and smelly. EVs were quiet, easy to drive especially appealing to urban residents and women. However, Henry Ford's affordable Model T and the electric starter's invention in 1912 favored gasoline cars. By the 1920s, better roads, affordable gas and limited rural electricity led to the decline of EVs. EVs saw little progress for about 30 years due to cheap gasoline and improved internal combustion engines. In the late 1960s and early 1970s, soaring oil prices and gas shortages, particularly the 1973 Arab Oil Embargo [4], drove interest in domestic fuel sources. Congress passed the Electric and Hybrid Vehicle Research Development and Demonstration Act of 1976 [5], enabling Energy Department support for electric and hybrid vehicle research and development. During this period, various automakers explored alternative fuel vehicles, including EVs. General Motors displayed an urban EV prototype in 1973 [6], and the American Motor Company tested electric delivery jeeps with the U.S. Postal Service in 1975 [7]. However, 1970s EVs had limitations, with limited performance and a 40-mile range before recharging.
EVs' true revival came in the early 21st century. Two key events drove this shift. The first was Toyota's introduction of the Prius in 1997 [8], a mass-produced hybrid EVs that gained global success due to rising gas prices and eco-concerns. Tesla Motors' 2006 announcement of a luxury electric sports car with more than 200 miles range marked the second pivotal moment [9]. Tesla's rapid rise led other carmakers to accelerate EVs development.

Despite the increasing prevalence of EVs, several challenges must be addressed before they can become a viable option for all. One primary concern is range anxiety [10] [11], the fear of running out of power before finding a charging station. Efforts to combat this have included advancements in battery efficiency and improved charging infrastructure. Nowadays, many EVs offer a range of up to 500 km on a single charge [12], typically sufficient for most drivers. However, access to reliable charging infrastructure while on the road remains crucial. Although charging stations are becoming more common, there are still areas lacking them, resulting in potential charging difficulties. The UK's Climate Change Committee [6] estimates that by 2030, 1170 charging stations will be required for every 100 km. Given the current growth rate, only a quarter of the anticipated total public charging stations will be in place by 2032. This delay is partially attributed to a global shortage of vital EV charger components and precious metals like lithium [13]. An apparent solution against range anxiety is to expand battery size and capacity. However, this isn't sustainable due to the high battery manufacturing costs [14] [15]. In addition, the cost of electric cars remains a significant hurdle [16]. While prices have dropped considerably in recent years due to battery technology advancements, EVs still tend to be pricier than traditional ICE vehicles. This cost barrier makes it challenging for many individuals to switch from ICE vehicles to EVs, as the initial expenses are often prohibitive.

A more captivating, feasible and cost-effective alternative lies in strategies that can extend the driving range of electric vehicles through the provision of driving range predictions. To offer insight and effectiveness, a driving range prediction strategy must consider the power-train model of the electric vehicle, as well as the characteristics of the road transportation network. Inaccurate estimations of energy consumption often result in both overly conservative and excessively optimistic driving range estimates. The former may lead to unnecessary delays during a journey, while the latter can result in the battery being completely depleted, requiring immediate recharging. Such circumstances are highly counterproductive and can further intensify range anxiety [10] [11]. In order to solve this issue, many EV manufactures have developed their own route planner software that are able to select the best route according to the battery SoC and the availability of charging stations. However, most of them do not take into account the battery SoH, ambient temperature conditions and the related energy consumption caused by Heating Ventilation and Air Conditioning (HVAC) activation.

The present Master's Thesis takes advantage of an innovative model [17] which encompasses a scheduler incorporating battery's SoH and external temperature parameters. This advanced scheduler selectively determines charging stations by evaluating an adjusted range value, subsequently adapting the planned route accordingly. The outcomes of these simulations underscore the inherent risks associated with disregarding range degradation when planning a route, particularly highlighting the potential danger of battery depletion during the journey.
Despite the notable enhancements incorporated, the planner presented by [17] exhibits certain limitations that influence the final outcome. Primarily, the vehicle model employed for energy consumption calculations is characterized by excessive simplification, while the recharging profile consistently operates at a non-real constant power level. Furthermore, the algorithm employed to identify optimal charging stations lacks optimization in terms of travel time minimization. Lastly, the algorithm requires prolonged processing times, exceeding the standards typically observed in contemporary route planning software.

The objective of this Master's Thesis is to address and solve the identified weaknesses of the existing framework. Specifically, a refined vehicle model is introduced, incorporating efficiency maps of the electric motor and modeling the recharging protocol. These enhancements are seamlessly integrated into a revisited algorithm for recharging station selection, which incorporates optimization techniques to minimize travel time, leveraging the well-established Dijkstra's algorithm. Furthermore, this Master's Thesis proposes an innovative clustering system for charging stations, utilizing their geographical positioning, in order to reduce processing time to the greatest extent possible. The findings and insights gained from this research contribute to the advancement of the field and offer potential solutions to the existing limitations, thereby offering promising avenues for further research and development.

Chapter 2

State of Art

As previously indicated in Section 1, the primary objective of this Master's Thesis is to address and resolve the identified weaknesses within the project presented by [17], whose objective was to enhance the state-of-the-art in route planning software utilized within the field of EVs.
The structure of route planning software, though not inherently intricate, is based on three core components: the vehicle's energy consumption model, crucial for continuous monitoring of energy requirements from the battery; the battery charging protocol, essential for tracking battery recharge times and consistently offering the optimal solution in terms of minimizing total travel time; and the route planning algorithm, which combines the two aforementioned key elements, ensuring mathematical optimization to delineate the route that minimizes overall travel time. This optimization also accounts for potential battery recharging, thereby ensuring the practicality of the journey in terms of energy consumption.

2.1 Vehicle Consumption Model

The current state of the art of vehicle consumption model involves the estimation of energy consumption with varying degrees of complexity. Among the simpler methodologies, the work presented by [18] proposed the calculation of energy consumption based exclusively on driving speed. This involves multiplying the difference between the actual driving speed and an optimal reference speed by a constant energy consumption coefficient expressed in kWh/km. On the other hand, [19] maintain this foundational structure, but introduce increased complexity in the energy consumption parameter, moving from a constant value to a higher-degree polynomial function, thus affording greater flexibility in approximating real-world consumption patterns. Conversely, [20] and [21] deviate from the conventional approach, which relies solely on vehicle speed and its associated energy consumption parameters, offering a more physically accurate representation of the problem by calculating the tractive force required at the wheels at each moment. From this calculation, they derive the power and energy demands on the battery. However, despite these models providing a more precise determination of power requirements from a physical standpoint, they still exhibit certain fundamental limitations. These limitations include the need for an improved representation of electric motor and battery efficiency, as well as a more sophisticated modeling of energy recovery during regenerative braking.

It is noteworthy that regenerative braking is a topic that often receives only cursory attention in the existing literature. Frequently, the ability of electric vehicles to recover and store energy is approximated using a constant regeneration factor [22]. Another approach, presented by [23], introduces a speed-dependent regenerative braking factor that represents the percentage of braking energy recoverable by the motor. Additionally, [24] propose a power based energy consumption model, which computes regenerative energy efficiency based on instantaneous deceleration. However, all these models assume a constant value for either the electric motor's efficiency and the energy regeneration factor.

Considering these characteristics and limitations of the actual state-of-art of vehicle consumption models, [25] focuses on developing and validating a computationally efficient battery EV simulation model able to estimate the energy consumption, based on a high-level representation of its main components, while employing a dynamic approach that combines vehicle speed and motor torque limitations in order to simulate the energy recovery capability of the EV. A significant contribution of the proposed model lies in its utilization of universally accessible technical specifications as input parameters. These generic data sources, like vehicle information leaflets and manufacturer websites, eliminate the need for confidential information or field measurements. Following this approach enhances transparency in the thesis methodology, avoiding reliance on proprietary data.

As a starting point, [25] adopts a fundamental physics based model for estimating the tractor effort necessary to overcome resistance forces and to accelerate the vehicle. Specifically, the traction effort, as described in previous literature [26], encompasses the combined effects of aerodynamic drag, rolling resistance, hill climbing force, linear acceleration and the inertia force resulting from rotating vehicle components.

The construction of part-load efficiency curves is conducted through the utilization of rational and linear fitting techniques. To determine the part-load efficiency curve for the motor, it is assumed that the minimum and maximum efficiency values can be inferred based on the known efficiency ranges of IMs and BLDC motors [27]. By utilizing these characteristic points and considering the efficiency range specific to each motor type, a piece wise function is employed to approximate the load-efficiency curve.

The relationship between motor size, specifically rated power and efficiency is derived from the minimum efficiency requirements for electric motors [28]. The average efficiency of each rated power class is calculated and normalized by the maximum average efficiency. This normalization factor serves as a multiplier, adjusting the motor's efficiency to account for its size in the determination process. In [25], it is postulated that the regenerative braking system is ineffective at low vehicle speeds due to limited available torque. To simulate this behavior, the model proposed in this thesis incorporates a speed-dependent regeneration factor that represents the proportion of available braking power recoverable based on the vehicle's speed.

In order to verify the accuracy of the simulation model created in its study, [25] compared its results to those obtained from the FASTSim, a well-established simulation tool that utilizes real-world data from operational vehicles. The validation results demonstrate that the average absolute error of cumulative energy consumption remains below 45 Wh, approximately equal to an average absolute percentage error of 4%, after conducting the primary work cycles commonly employed for validating vehicle fuel consumption and emissions. These cycles include the LA92-Unified, LA92-Short, HWFET, JP 10 Mode, JP 15 Mode, UDDS, NYCC, SFTP and NEDC. Moreover, with a similar hardware adopted in this thesis, [25] results in a computational time to complete each driving cycle on the scale of tens of milliseconds. These findings demonstrate that the developed model successfully achieves a favorable equilibrium between representation effectiveness and computational efficiency.

2.2 Battery Recharge Model

As previously highlighted, the modeling of battery charging profiles stands as a fundamental component within route planning software, complementing the vehicle's energy consumption model and the algorithm employed for the actual route planning. The importance of accurately defining the battery charging model is emphasized by the substantial time investment required for charging, coupled with the restricted travel range attainable before necessitating a recharge. This limitation translates to an increased frequency of stops during extended journeys, underscoring the value of making more precise predictions regarding trip duration. This, in turn, affords drivers greater autonomy and informed decision-making when selecting the optimal route to minimize overall travel time.

Nevertheless, it is imperative to acknowledge that, in practice, charging power profiles are often approximated as constant [29]. This approximation does not account for the fact that, in reality, the charging power of lithium-ion batteries tends to decrease significantly once the battery's SoC reaches approximately 80% [30]. Within the context of this Master's Thesis project, a portion of the research conducted by [18] will be incorporated. This research delineates two distinct battery charging protocols, both of which encompass a charging process divided into two phases. The point of separation between these two phases is typically situated on the battery's SoC level at 80%. These two charging protocols are:

- Constant Current - Constant Voltage (CC-CV);
- Constant Power - Constant Voltage (CP-CV);

It is worth mentioning that in the study conducted by [18], the two battery's recharging protocols have been validated against measurement data from charging an electric vehicle [31]. During such comparison, CP-CV protocol achieves a charging time match within ±2%, while using the CC-CV protocol yields a relative error exceeding 10%.

Constant Current - Constant Voltage (CC-CV)

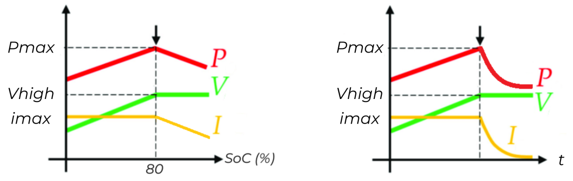
The first phase of this protocol is characterized by a constant current and a voltage that increases linearly with respect to the battery's SoC, resulting in a charging power that increases linearly with the battery's SoC. The second phase, on the other hand, is characterized by a constant voltage and a current that decreases with respect to the battery's SoC, resulting in a charging power that linearly decreases. The mathematical model adopted by [18] to represent the CC-CV recharging protocol is shown in the following from Eq.2.1 to Eq.2.3:

$$P_{CC-CV}(SoC) = V(SoC) \cdot i(SoC) \tag{2.1}$$

$$i(SoC) = \begin{cases} i_{max} & SoC < 80\% \\ \frac{1-SoC}{0.2} \cdot i_{max} & SoC \geq 80\% \end{cases} \tag{2.2}$$

$$V(SoC) = \begin{cases} V_{low} + \frac{SoC}{0.8} \cdot (V_{high} - V_{low}) & SoC < 80\% \\ V_{high} & SoC \geq 80\% \end{cases} \tag{2.3}$$

where $i_{max} = P_{max}/V_{high}$ is the maximum current, P_{max} is the maximum charging power of the charging station, V_{low} is the voltage defined at $SoC = 0\%$, while V_{high} is the voltage defined at $SoC = 80 - 100\%$.
Figure 2.1 presents the formulas previously illustrated for the CC-CV charging protocol, emphasizing the values of voltage, current and charging power in relation to the battery's SoC and charging time duration.



(a) Function of SoC **(b)** Function of time

Figure 2.1: CC-CV charging protocol as function of SoC and time

Constant Power - Constant Voltage (CP-CV)

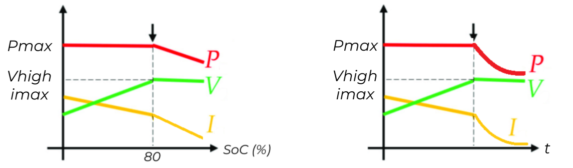
The first phase of this protocol is characterized by a current that decreases and a voltage that increases linearly with respect to the battery's SoC, resulting in a constant charging power. The second phase, on the other hand, is characterized by a constant voltage and a current that decreases with respect to the battery's SoC, resulting in a charging power that decreases linearly.
The mathematical model adopted by [18] to represent the CC-CV recharging protocol is shown in the following from Eq.2.4 to Eq.2.6 :

$$P_{CP-CV}(SoC) = V(SoC) \cdot i(SoC) \tag{2.4}$$

$$i(SoC) = \begin{cases} \frac{P_{max}}{V(SoC)} & SoC < 80\% \\ \frac{1-SoC}{0.2} \cdot i_{max} & SoC \geq 80\% \end{cases} \tag{2.5}$$

$$V(SoC) = \begin{cases} V_{low} + \frac{SoC}{0.8} \cdot (V_{high} - V_{low}) & SoC < 80\% \\ V_{high} & SoC \geq 80\% \end{cases} \tag{2.6}$$

adopting this protocol the recharging power is recalculated after each second of charging and terminates when $SoC = 99\%$.
Figure 2.2 presents the formulas previously illustrated for the CP-CV charging protocol, emphasizing the values of voltage, current and charging power in relation to the battery's SoC and charging time duration.



(a) Function of SoC **(b)** Function of time

Figure 2.2: CP-CV charging protocol as function of SoC and time

2.3 Route Planning Algorithms

The research for the optimal route is currently carried out through the utilization of algorithms designed with the objective of defining a sequence of actions that, when executed, serve to minimize a designated cost. To elucidate further, within the context of route planning, this sequence of actions corresponds to the specific road segments to be travelled, thereby minimizing the aforementioned cost. When the cost is related to the length of individual road segments, the output of these algorithms will be the shortest route. Conversely, when the cost is associated with travel time, the algorithms yield a route that minimizes the overall travel time.

The contemporary landscape of route planning algorithms encompasses a diverse array of techniques, each applied to achieve results that are, in most cases, inversely proportional to the processing time required to obtain them. For instance, the bidirectional search techniques, adopted by [17], undertake simultaneous exploration of routes originating from both the source and destination, with their convergence at an intermediate point facilitating the discovery of the shortest path. Conversely, solutions such as the A* algorithm [32] exemplify heuristic-based approaches that effectively search for the shortest path. These methods consider not only the cost from the origin node but also an estimated cost to reach the destination node. Additionally, certain algorithms belong to the category of time-dependent and dynamic routing [33]. These algorithms adapt to real-time fluctuations in travel times, encompassing factors like traffic conditions. They excel in finding optimal routes amidst dynamically evolving scenarios.

In the scope of this section, particular emphasis is put on Dijkstra's algorithm [34], since it assumes a key role in the context of this project. Discussion extends to the Bellman-Ford algorithm [35] and two techniques adopted to enable the application of the Dijkstra's algorithm within the context of electric vehicles. In this regard, we introduce the Johnson's Shifting Technique [36], alongside a technique that significantly truncates the processing time of the Dijkstra's algorithm, known as Contraction Hierarchies [37].

It is however worth pointing out that this project does not use any of the state of the art speed-up techniques to deal with Dijkstra's algorithm's complexity. Instead, it relies on clustering methods to diminish the size of the matrix underlying Dijkstra's algorithm.

Dijkstra's Algorithm

Dijkstra's algorithm is a widely used algorithm for determining the shortest path from a source vertex to all other vertices in a weighted graph. In the context of route planning, the nodes that make up the weighted graph are simply the road intersections between two or more roads, while the connections between these intersections represent the individual roads. Each road is associated with a cost, which is why the term weighted is used to describe the network of nodes. The cost, still within the context of route planning, typically represents the distance or the travel time between various connections among the nodes. Additionally, it is possible to incorporate a combined cost, which could be the sum of two or more attributes, such as travel time and energy consumption. By employing specific multiplicative coefficients for each cost components, this combined cost can yield a journey that, depending on the choice of coefficients, may prioritize either the minimization of travel time or energy consumption. The algorithm's objective is to provide the correct sequence of nodes to traverse in order to minimize the mentioned cost from a generic starting node to an arrival node.

To achieve this objective, the algorithm starts by setting the distance of the source vertex to 0 and the distances of all other vertices to ∞. Then, it iterative selects the vertex with the smallest tentative distance and explores its neighboring vertices, updating their distances if a shorter path is found. This process continues until all vertices have been visited or the destination vertex is reached. Operating in this manner, the Dijkstra algorithm has a time complexity of $\mathcal{O}(n \cdot \log(n) + m)$, where n denotes the number of vertices and m represents the number of edges.

However, it is important to underline that the aforementioned algorithm cannot handle graphs with negative edge weights. One of the main reasons the Dijkstra algorithm fails with negative edge costs is the concept of a *greedy* approach it employs. It always selects the vertex with the smallest tentative distance for exploration, assuming it has found the optimal solution so far. However, in the presence of negative edge costs, this assumption is no longer valid. Negative edge costs can create scenarios where a vertex with a larger tentative distance can actually lead to a shorter overall path due to the negative cost. The algorithm, following its *greedy* nature, might incorrectly discard such possibilities as it always favors vertices with smaller tentative distances. This can result in suboptimal or incorrect shortest path calculations.

One of the pioneering works that addressed the challenge of incorporating negative edge costs into the application of the Dijkstra algorithm for optimizing energy-efficient routes for BEVs within street networks was the study conducted by [38]. The study highlighted two fundamental distinctions from the conventional shortest path problem: firstly, edge costs may be partially negative due to the energy recuperation capability of BEVs. Secondly, battery constraints must be satisfied, prohibiting a feasible path from including a node with a battery load less than zero (energy depletion) or a battery load exceeding the battery capacity (overcharging). However, [39] demonstrated that one-to-one queries can be resolved with the same complexity as Dijkstra's algorithm by representing battery constraints as edge cost functions and leveraging Johnson's shifting method to transform them into non-negative edge costs.

Johnson's Shifting Technique

Johnson's shifting method [36] aims to transform the graph and eliminate negative edge weights, thereby making it suitable for Dijkstra's algorithm. The main concept behind Johnson's shifting method involves introducing a shifting factor that is added to each edge weight. The shifting factor is carefully chosen to ensure that all edge weights become non-negative. By applying this shifting operation, the graph is transformed in a way that eliminates any negative cycles.

Bellman-Ford's Algorithm

An alternative algorithm to Dijkstra's for finding the shortest path is Bellman-Ford's algorithm [35]. It operates by iterative relaxation of the edges of the graph, gradually improving the estimated distances to the nodes until the solution converges to the optimal one. It follows a dynamic programming approach, where the shortest path from the source node to each destination node is built up one edge at a time. Initially, all nodes are assigned a distance value of ∞, except for the source node which is set to 0. Then, for each edge in the graph, the algorithm checks if the distance to the destination node can be improved by taking the current edge. If so, the distance is updated accordingly. This process is repeated for a number of iterations equal to the number of nodes minus one, ensuring that the algorithm has sufficient opportunities to optimize the distance values. If, after this number of iterations, any distance value can still be improved, it indicates the presence of a negative cycle. In the context of the Bellman-Ford algorithm, the existence of a negative cycle within the network is deemed unacceptable, as it would imply the presence of a path that, if perpetually traveled, would lead to an incessant reduction in the overall cost of the cycle.

The main advantage of the Bellman-Ford algorithm over Dijkstra's is its capability to determine the shortest path in various types of graphs, including those with positive, zero, or negative edge weights. However, it comes with a computational complexity of $\mathcal{O}(n \cdot m)$, where n denotes the number of vertices and m represents the number of edges, which is higher than that of Dijkstra's algorithm.

Contraction Hierarchies

The specific formulation of the edge cost functions adopted in route planning algorithms, enables the adaptation of an acceleration strategy for shortest path queries known as contraction hierarchies [37]. In a pre-processing phase, the graph is enriched with shortcuts that effectively enhance the speed of path finding queries. This is achieved through a process of gradually contracting the nodes of the graph, effectively removing them from consideration. If a contracted node lies on the shortest path between two neighboring nodes, a direct edge (shortcut) connecting

those neighbors is added to ensure the preservation of the shortest path. The determination of such shortcuts involves performing a shortest path search using Dijkstra's algorithm between each pair of neighbors. Each node is assigned a level based on the order of contraction, with higher levels indicating later contractions and the possibility of replacing shortcuts from lower-level nodes. To query the shortest path, a bidirectional search using Dijkstra's algorithm is conducted, with each side only traversing to nodes of higher levels until they meet. This approach significantly reduces the number of visited nodes, resulting in rapid queries without the issue of exponential growth in function descriptions.
However, the pre-processing stage can be computationally demanding, especially for large-scale maps spanning entire countries. As more nodes undergo contraction, the remaining uncontracted nodes become densely connected, leading to significant computational costs when contracting the last few nodes. To address this issue, [40] proposed a strategy that selectively contracts only a subset of nodes, achieving reasonable pre-processing times by contracting merely 99.5% of the nodes. The resulting graph, consisting of the remaining uncontracted nodes, is referred to as the core graph [41]. While this approach significantly reduces pre-processing time, it may lead to longer query times.

The development of this Master's Thesis exclusively relies on the utilization of Dijkstra's algorithm, as deeper explained in the subsequent Section 3.1.2. Accordingly, a node network will be constructed wherein nodes correspond to charging stations present on the map and edge costs represent travel time from one node to another, encompassing recharge time in instances where destination node coincides with a charging station. Since edge costs are always positive in the context of travel time between nodes, there is no need to apply Johnson's shifting techniques. Consequently, the adoption of Dijkstra's algorithm proves more advantageous in terms of computational efficiency when compared to Bellman-Ford's algorithm [35]. Conversely, this Master's Thesis does not incorporate the utilization of any contraction hierarchies. Rather, the set of edges constituting the complete map remains unaltered throughout the entirety of the study, as the primary interaction entails accessing it as a client of a third-party software. This software grants access to the map while providing the requisite information (speed profile) for the construction of a secondary cost matrix that will serve as the foundation for executing the Dijkstra algorithm. The specific approach employed by the third-party software to determine the fastest path between arbitrary origin and destination edges is beyond the scope of this study. Instead, the software is treated as a black box, receiving inputs and generating outputs.
It is crucial to emphasize that, unlike the aforementioned work by [39], battery constraints are not initially integrated within the map. Instead, they are taken into

consideration subsequently, after receiving the speed profile from the external software. This enables the execution of simulations utilizing a vehicle model, facilitating the energy consumption estimation. From a purely mathematical standpoint, the solution derived in this study is not rigorous, as the inclusion of battery constraints within the map itself is deemed essential. There may arise situations wherein the external software calculates fastest path overlooking battery constraints, resulting in excessive depletion and the need for a recharge. Modifying the map to include these constraints would enable the software to optimize the route, minimizing battery usage and avoiding deviation to charging stations. Nonetheless, this slight deviation from the mathematical rigor is justified by the outcomes of simulations conducted by [18], comparing the results attained using a system that integrates battery constraints within the node network and adjusts the edge costs, with a system that exclusively pursues the fastest route without factoring in battery constraints. The results reveal a discrepancy of approximately 0.2% between the former, incorporating battery constraints, and the latter, omitting them - an entirely acceptable variance within the context of this research endeavor.

Chapter 3

Methodology

In the subsequent sections, the operational framework of the program will be elucidated. Initially, a comprehensive overview of the solution will be provided, encompassing the concepts delineated in Figure 3.1. Subsequently, in-depth analyses will be provided regarding the route planning algorithm, the vehicle consumption model, the novel charging station clustering tool and the several speed-up techniques employed to lower processing time.

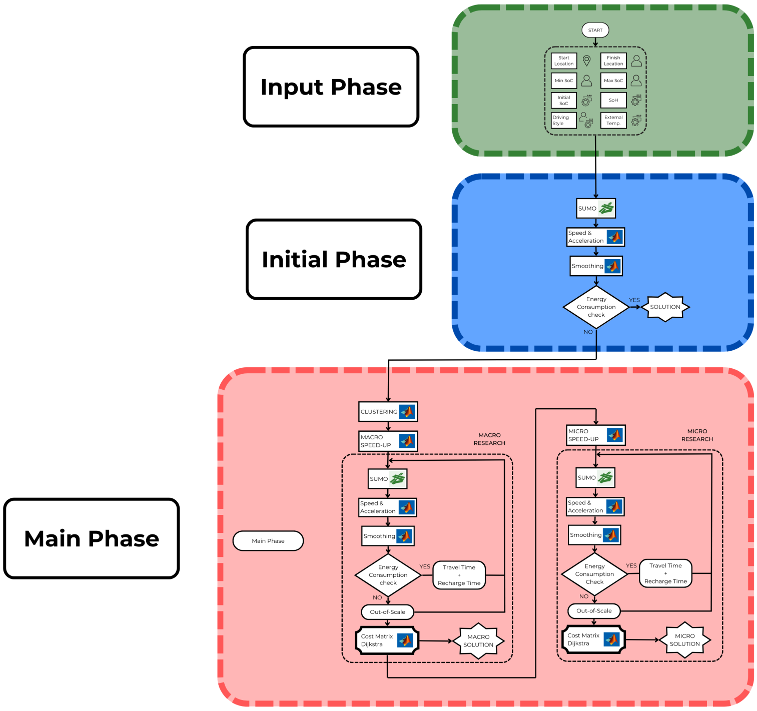


Figure 3.1: Working Principles - Summary

3.1 Working Principle

In order to briefly introduce what will be extensively described in the subsequent sections, referring to Figure 3.1, it can be noted that the project presented in this Master's Thesis is divided into three parts. In the first part, called the *Input Phase*, the user is required to input the fundamental parameters for the proper functioning of the program. In the second part, called *Initial Phase*, a preliminary check is carried out to determine if the journey with the coordinates set by the user, the environmental and vehicle conditions can be completed without the need to stop for battery recharge. If this isn't feasible, the third part, referred to as the *Main Phase*, is entered. This section, which not only constitutes the most extensive section and where the greatest efforts have been concentrated, is also the one that ensures the definition of the optimal route to minimize travel time taking in consideration the need of stopping for recharging the battery.

3.1.1 Input Phase

The program execution starts by asking the user to input the fundamental parameters for its operation. In this Master's Thesis project, all these parameters are manually provided by the user, but in a future application of the software in a real vehicle, some of them could be automatically acquired by the software through internal hardware, GPS or learned by the software itself.

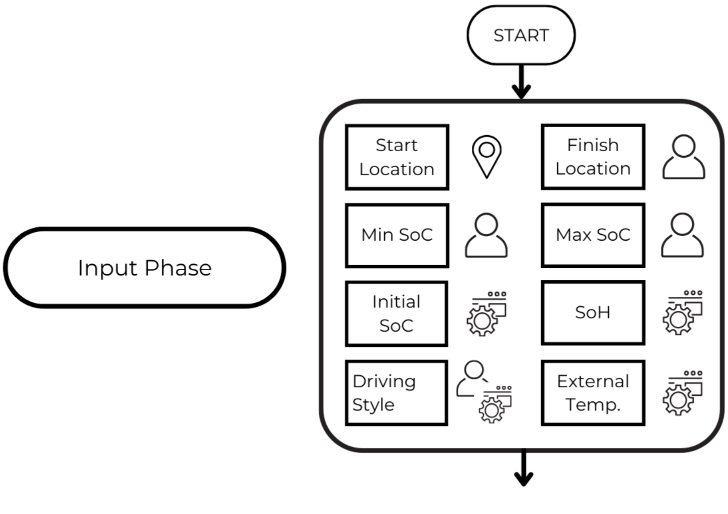


Figure 3.2: Working Principles - Input Phase

Referring to Figure 3.2, the first parameters requested are the starting and destination coordinates of the trip. In a practical application, the starting coordinates could be obtained automatically by the software through the integrated GPS in the vehicle, while the destination coordinates would need to be entered manually by the user.

Next, the user is prompted to input critical parameters for preserving battery life when properly configured. These parameters include minimum desired battery's SoC threshold, which represents the lower limit that should never be violated during the journey, along with desired maximum battery's SoC level for each charging cycle. In a future application on a real hardware, these two parameters could be automatically set by the vehicle manufacturer, with the possibility of being adjustable by the user as needed.

Finally, the user is required to provide initial battery's SoC, number of passengers, external temperature and adopted driving style. In the context of a future implementation in a real vehicle, the initial battery SoC, as well as the number of passengers and external temperature, are parameters that could be easily acquired by onboard hardware. As for the driving style, there could be several possible options: the user could be able input it manually, or the driving style could be chosen through an internal learning system that analyzes the user's driving behavior and selects the most suitable one [42]. In this project, the choice of the driving style has impacts on both the maximum speed and the maximum acceleration/deceleration that the vehicle undergoes during the travel of different routes, as highlighted in Table 3.1.

Table 3.1: Driving Style Parameters

Drving Style	**Max Acc.**	**Max Speed**
Eco	1 m/s^2	25 m/s
Normal	2.5 m/s^2	40 m/s
Sport	9 m/s^2	50 m/s

3.1.2 Initial Phase

At this stage, the actual execution of the program begins according to the steps highlighted in Figure 3.3. First, a check is performed to immediately verify the non-stop journey feasibility, considering the retrieved parameters and the constraints set by the user, as specified in Section 3.1.1. The same procedure is later applied to verify the reachability of each charging station, relying on the adoption of the external software SUMO [43] and the utilization of the TraCI4Matlab [44] functions, enabling the communication between SUMO and Matlab. The working environment and the specific functions employed will be thoroughly elucidated in Section 3.2.

In response, the program retrieves the minimum travel time baseline route, that means *without stops*, from the starting to the destination coordinates, presented as a list of various edges constituting the entire trajectory. Subsequently, employing TraCI4Matlab functions, length and maximum allowed speed of each edge are extracted from the aforementioned list. By performing simple mathematical operations, starting from the knowledge of the length and travel speed of each edge, speed and acceleration profiles as functions of time are then obtained.

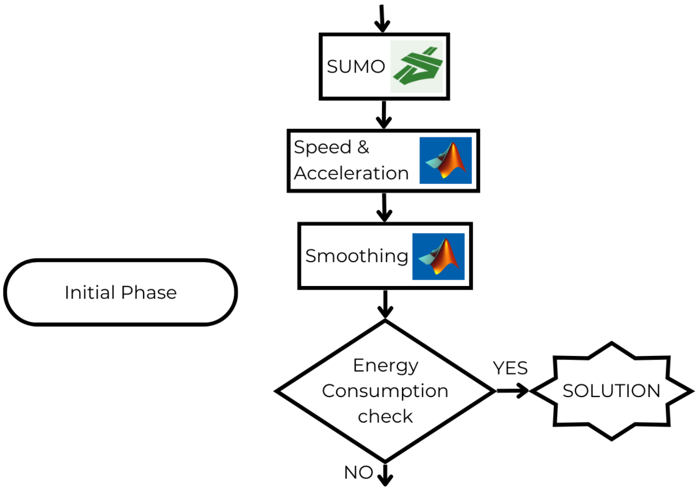


Figure 3.3: Working Principles - Initial Phase

The two obtained profiles undergo an additional process, defined as *Smoothing*, before being utilized for the validation of energy consumption with the vehicle model. This process was not included in the original model [17] and serves as a replacement for the effect of noise. The noise was initially introduced to simulate a non-uniform trend characterized by continuous small positive and negative accelerations. However, its purpose of use differs from the current process. The objective of this process is to refine the acceleration profile and, consequently, the speed profile, in order to mitigate peak acceleration and deceleration values to comply with the constraints set by the user's driving style set in Section 3.1.1. Three driving styles are available: Eco, Normal, and Sport. Each driving style is associated with specific speed and acceleration limits, as detailed in Table 3.1.
The calculation of energy consumption profile is performed based on the newly obtained speed and acceleration smoothed profiles. This profile is essential in determining the feasibility of the trip without the need of any stops. The detailed calculation of the energy consumption profile is provided in Section 3.3. The trip feasibility corresponds to the possibility of performing such trip maintaining the battery SoC level above the pre-determined limits established during the Section 3.1.1. If the trip is deemed feasible, the problem is solved and the result is presented

to the user. Conversely, if the trip is found to be unfeasible, the program moves to the subsequent phase, wherein the objective is to identify the minimum travel time route from the departure point to the destination incorporating charging stops to recharge the battery up to the designated level specified by the user, as described in Section 3.1.1.

3.1.3 Main Phase

As depicted in Figure 3.4, this phase is characterized by the utilization of Dijkstra's algorithm as the primary tool for determining the shortest path from origin to destination, while considering the time required for each recharge.

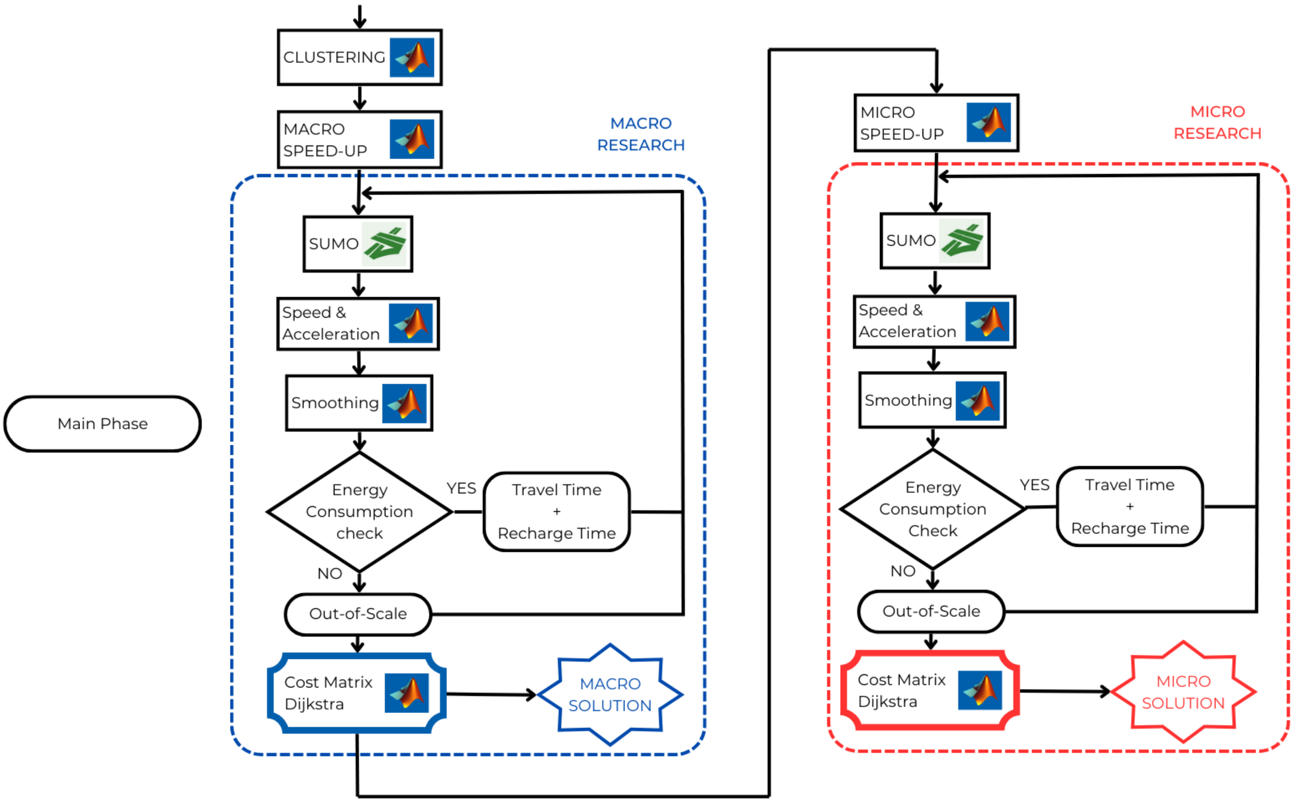


Figure 3.4: Working Principles - Main Phase

Dijkstra's algorithm, as introduced in Section 2.3, is employed when the objective is to find the minimum cost of going from one node to another within a network of nodes. These nodes correspond to the charging stations that are present on the map, in addition to the starting and ending locations specified by the user. On the other hand, the network branches represent the cost of moving between nodes within the network. In this project, the cost associated with each branch is the summation of two values: the travel time from the starting node to the destination node and the recharge time at the destination node. The travel time is acquired providing to SUMO the coordinates of all the nodes inside the net. These coordinates can correspond to user's specified start and end locations at Section 3.1.1 or the various charging stations. Recharge time, on the other hand, is calculated as the time required to reach the specified SoC level once the node has been reached. An approximate indication of how the cost for each connection is calculated is provided by the Eq.3.1, which represents the cost associated with the interconnection between a generic starting node i_{th} and a generic ending node j_{th}:

$$cost^{i \to j} = t^{i \to j}_{travel} + t^{i \to j}_{charge} \approx t^{i \to j}_{travel} + \frac{E^{i \to j}_{travel}}{P^{j}_{charge}} \tag{3.1}$$

Naturally, if the destination node coincides with the user's start or end location, the recharge time would be null due to the absence of any charging station at that position.

As mentioned in Section 3.1.2, not all routes are feasible due to the possibility of the battery SoC level to fall below the user-defined minimum or even reaching complete depletion. In such cases, the cost of the journey is no longer the sum of the two parameters defined earlier. Instead, an out-of-scale value (∞, 10^{10} , ...) is assigned to the edge. This value is sufficiently high to ensure that it is never considered during the computation of the minimum cost path using Dijkstra's algorithm.

Operating in this manner yields a square cost matrix of size $(n+2) \times (n+2)$, where n represents the number of charging stations present on the map, with the addition of the departure and arrival locations of the trip specified by the user. The value at position ij within the matrix represents the cost associated with traveling from the i^{th} starting node to the j^{th} destination node.

3.1.4 Speed-Up Techniques

As will be analysed in Section 3.4, the large number of stations (n) coupled with the average processing time required to calculate the cost for each cell within the cost matrix, results in a time-consuming and computationally intensive matrix construction process. These factors have prompted the development of methods to accelerate the construction of the cost matrix.

The first step to accelerate the construction of the cost matrix is represented by an innovative clustering system for charging stations based on their geographical location. It has been introduced in order to significantly reduce the number of stations and, consequently, the size of the cost matrix. Additionally, a speed-up mechanism has been implemented to streamline the matrix construction process, enabling for the computation of costs only for the relevant branches/trips that contribute to the final outcome. A more comprehensive explanation of the clustering system and of the speed-up mechanisms is presented in Section 3.4, for clustering, and in Section 3.5, for the speed-up mechanisms.

Macro - Micro Research

Operating accordingly with Figure 3.4, the research for the minimum travel time path, from the starting point to the destination, is divided into two distinct stages: the *macro-research* and the *micro-research*.
During the *macro-research*, the focus is on defining the minimum travel time path using clustered charging stations. These clusters represent sets of stations grouped together under a single geographic location. The output of Dijkstra's algorithm in this stage is a sequence of geographic areas, rather than individual charging stations, inside each of which *micro-research* will define the best places to stop minimizing travel time. The term *macro* has been adopted to emphasize the focus on macro-geographic areas that encompass multiple stations.
Subsequently, *micro-research* phase entails resolving the same problem of finding the path that minimizes cost using Dijkstra's algorithm, but exclusively considering the recharging stations located within the clusters identified as the output solution of *macro-research*. The output of Dijkstra's algorithm in *micro-research* is the precise sequence of charging stations where stopping is required to travel from starting point to destination, minimizing total travel time. This elucidates the rationale behind the adoption of the term *micro* as it focuses on micro-scale geographic areas, providing a fine-tuning of the ultimate outcome by comparing charging stations with closely located geographical positions, where the difference in travel distance from the entire journey is significantly reduced.

The internal components of *macro* and *micro-research* flowchart, represented in Figure 3.4, exhibits the same structure. Both research methods comprise a speed-up block followed by a work cycle that employs TraCI4matlab functions to establish communication between the SUMO environment and Matlab followed by Dijkstra's algorithm resolution block, employed to determine the shortest path within the cost matrix.
The concept of speed-up techniques aims to expedite the creation of the cost matrix by mitigating the performance bottleneck caused by TraCI4matlab functions, as outlined in Section 3.4. During the subsequent computational phase utilizing SUMO software, only trips deemed essential for determining the shortest path will be considered, while non-essential trips will be rapidly assigned an out-of-scale default value. This approach excludes them from Dijkstra's algorithm application,

as conceptually introduced at the beginning of Section 3.1.3. Consequently, the same final outcome can be achieved with reduced processing time.
Cycling block encompasses the utilization of SUMO environment for the speed profile creation and feasibility assessment of the trip in terms of energy consumption, according to user's inputs outlined in Section 3.1.1. This block cycle adheres to the operational principle, introduced in Section 3.1.3, of assigning an out-of-scale value to infeasible trips or cells within the cost matrix and a cost equal to the combined travel time and recharge time for the feasible trips. Energy consumption verification for the definition of the feasibility of a trip relies on the formulas introduced in Section 3.3.1.

3.2 Working Environment

3.2.1 Software Involved

For this project, two distinct software applications were employed: Matlab and SUMO [43], an open-source traffic simulation software designed for simulating traffic scenarios and analyzing transportation systems. Matlab was employed as the platform for executing the co-simulation, seamlessly integrating with SUMO software through the utilization of TraCI4Matlab [44]. TraCI4Matlab is a specialized implementation of the TraCI protocol [45] tailored to Matlab environment, facilitating smooth communication between the two software tools. This enables the seamless interaction between Matlab and SUMO, where SUMO acts as the server fulfilling Matlab's requests as the client [46]. Co-simulation serves the purpose of validating the defined route by providing a more accurate simulation of the vehicle dynamics. Speed and acceleration profiles are obtained from SUMO co-simulation, which operates with a time step of 1 second.The decision to employ SUMO for route definition and speed profile generation is driven by its optimization capabilities, allowing for route computation within milliseconds. Additionally, SUMO's status as a traffic simulator makes it a suitable choice for future developments where the impact of traffic congestion may be taken into account.

3.2.2 Road Network

The road network utilized in this study, as depicted in Figure 3.5, is the same adopted in the original project [17], encompassing the primary thoroughfares of the northwestern region of Italy.

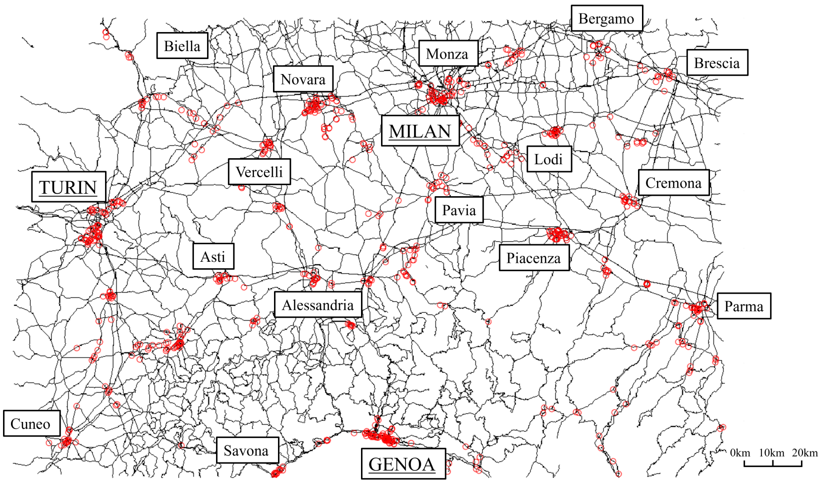


Figure 3.5: Road Network with Recharging Stations

The network is represented within SUMO framework and sourced from OSM website [47], an online cartographic service constantly updated and constructed by volunteer contributors, accessible under an open-content license.
As indicatively illustrated in Figure 3.6, the road network in SUMO is composed of three key components: junctions, edges, and lanes. Junctions serve as the network nodes where each edge initiates or terminates. Edges denote unidirectional connections linking two junctions and form the fundamental building blocks for route representation.

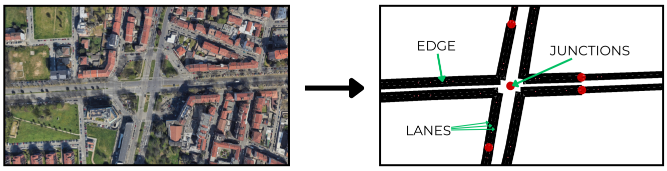
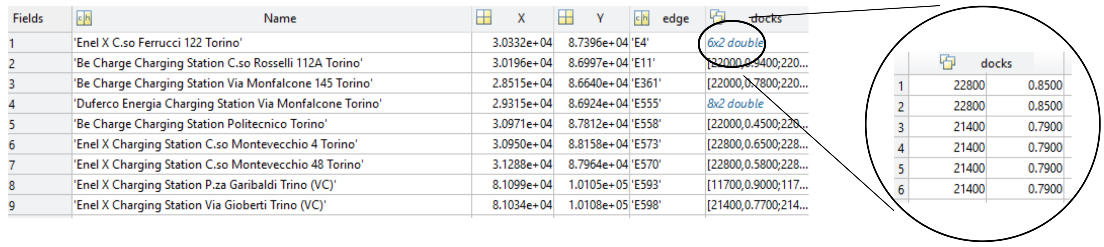


Figure 3.6: Edges, Junctions and Lanes creation

In Matlab, routes are expressed as ordered arrays of cells, with each cell indicating the corresponding edge name. Lanes, as sub-components of edges, encompass vital details such as speed limits, lengths and permissible vehicle types. Within the context of this paper, lanes assume particular importance as they store the speed and length information for each edge. The constitutive elements of the network are consolidated in a *.xml* file, where their respective properties are stored. It is worth noting that, while the vehicle model and simulation both account for road slope, the network itself lacks altitude information, rendering it unable to incorporate slope considerations during route creation and subsequent energy consumption simulations. Additionally, temperature data, though user's input, follows a simplified representation, with a single temperature value throughout the whole map, a simplification utilized in previous studies [48] [49]

3.2.3 Charging Stations

While SUMO has its own representation for charging stations, identified as portions of lanes, throughout this work it was preferred to store them in an external Matlab table. This choice was taken to be able to easily access such database bypassing SUMO when performing the planning operation. In order to seamlessly merge the two logics, a triangular arrangement of roads has been introduced. The vehicle reaches the designated charging station location and traverses the three consecutive edges added outside the core network, finally halting at the third edge, which represents the station. When simulating the following step of the travel, SUMO will generate a vehicle departing from the same edge
As shown in Figure 3.7, which displays screenshots taken in the Matlab environment,



Fields	Name	X	Y	edge	docks
1	'Enel X C.so Ferrucci 122 Torino'	3.0332e+04	8.7396e+04	'E4'	6x2 double
2	'Be Charge Charging Station C.so Rosselli 112A Torino'	3.0196e+04	8.6997e+04	'E11'	[22000,0.9400;220...
3	'Be Charge Charging Station Via Monfalcone 145 Torino'	2.8515e+04	8.6640e+04	'E361'	[22000,0.7800;220...
4	'Duferco Energia Charging Station Via Monfalcone Torino'	2.9315e+04	8.6924e+04	'E555'	8x2 double
5	'Be Charge Charging Station Politecnico Torino'	3.0971e+04	8.7812e+04	'E558'	[22000,0.4500;220...
6	'Enel X Charging Station C.so Montevecchio 4 Torino'	3.0950e+04	8.8158e+04	'E573'	[22800,0.6500;228...
7	'Enel X Charging Station C.so Montevecchio 48 Torino'	3.1288e+04	8.7964e+04	'E570'	[22800,0.5800;228...
8	'Enel X Charging Station P.za Garibaldi Trino (VC)'	8.1099e+04	1.0105e+05	'E593'	[11700,0.9000;117...
9	'Enel X Charging Station Via Gioberti Trino (VC)'	8.1034e+04	1.0108e+05	'E598'	[21400,0.7700;214...

	docks	
1	22800	0.8500
2	22800	0.8500
3	21400	0.7900
4	21400	0.7900
5	21400	0.7900
6	21400	0.7900

Figure 3.7: Road Network saved in a Matlab Struct

charging station details are stored in a Matlab struct, where each row corresponds to a specific charging station. The first column of the struct contains the charging station's name in order to provide to the user more detailed information on where to stop and charge the battery, thereby improving the human-machine interface. The second and third columns contain the (X,Y) coordinates of charging stations within the local coordinate system, which are essential for the subsequent charging station clustering process. The fourth column, crucial for route planning, stores the edge name, which is given as an input to SUMO to find the road from start to station or from station to arrival. Lastly, the fifth column contains a table for each row, representing all the docks of the charging stations. Due to the considerable number of charging stations considered in this research and the manual data set compilation, the table does not include charger types. However, this information can be easily incorporated. The database comprises around 700 charging stations, added by manually extracting data from Google Maps. The selection aimed to achieve equitable coverage across the entire map of northwestern Italy while emphasizing denser charging station representation within major cities.

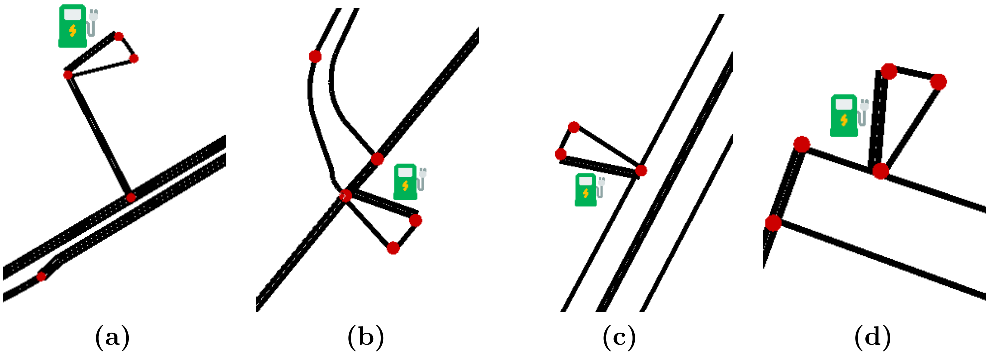


Figure 3.8: Examples of Recharging Stations inside the Road Map

3.3 Vehicle Consumption Model

In this section, the mathematical model employed to approximate the vehicle under study is presented. Specifically, vehicle type *b)* utilized in the original project [17], whose fundamental parameter values are listed in Table 3.2.

Table 3.2: Vehicle Main Parameters

Parameter	**Symbol**	**Value**
Vehicle Mass	m	1320 kg
Frontal Area	A	2.38 m^2
Aero-Drag Coeff.	C_d	0.30
Inertia Factor	$C_{inertia}$	0.05
Coast Down (Constant)	f_0	0.005 N
Coast Down (Quadratic)	f_2	$3.5 \cdot 10^{-6}$ $N/(m/s)^2$
EM Torque Rated	T_{rated}	400 Nm
EM Base Speed	w_{base}	4000 rpm
EM Max Speed	w_{max}	12000 rpm
EM transm. ratio	g_{ratio}	4
EM transm. eff.	η_{gear}	0.99
Battery Energy	E_{Batt}	24.15 kWh
Battery RTE	RTE	0.95
Tyre Data	-	175/55 R20

Moreover, the data to compare numerical simulations conducted using the newly developed mathematical model presented in this project were provided, as well as the previous mathematical model presented in [17]. Notably, the principal distinctions between the new and old model derive from the consideration of a variable motor efficiency based on the power demanded by the driver [25], together with a coefficient accounting for the motor size [25] and the electric motor's inability to regenerate braking at low speeds due to insufficient available torque [25].

3.3.1 Mathematical Formulation

In Figure **??**, an explanatory diagram of the direction of power flow used during the current section is provided. The direction of power is considered positive when the battery supplies energy to the motor and negative when the battery receives energy from the motor, which functions as a generator.

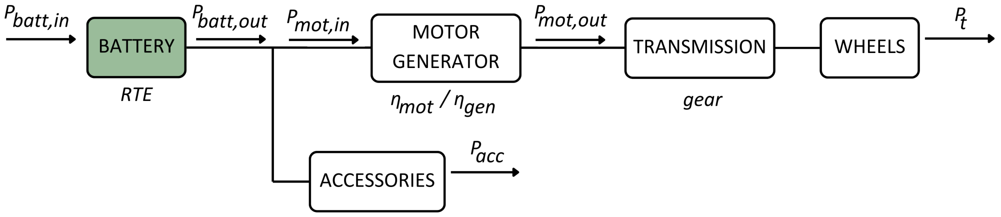


Figure 3.9: Power flows and Efficiencies in Main Components

Traction Power at wheels

As a starting point, a generic physics-based model serves as the theoretical framework for estimating the tractive effort F_t (in N) required to overcome resistance forces and accelerate the vehicle. Specifically, the tractive effort refers to the propulsive force transmitted to the ground through the wheels, driving the vehicle forward. This concept is defined by Eq. 3.2:

$$F_t = F_{aero} + F_{roll} + F_{hc} + F_{acc} + F_{inertia} \tag{3.2}$$

The aerodynamic drag, F_{aero}, is the force that acts in opposition to the vehicle's motion through the air, as defined by Eq. 3.3:

$$F_{aero} = \frac{1}{2}\rho A_f C_d v^2 \tag{3.3}$$

where ρ is the air density (in kg/m^3), A_f is the frontal vehicle area (in m^2), C_d is the aerodynamic drag coefficient, and v is the linear vehicle speed (in m/s).

The rolling resistance force, F_{roll}, arises primarily from the interaction between the vehicle's tires and the road surface, as represented by Eq. 3.4 , where f_0 (in N) and f_2 (in $N/(m/s)^2$) are respectively the constant and quadratic term of the coast down resistance, while v represents the vehicle speed (in m/s):

$$F_{roll} = f_0 + f_2 \cdot v^2 \tag{3.4}$$

The hill climbing force, F_{hc}, denotes the gravitational force component that acts on a vehicle when ascending a slope α, as defined by Eq. 3.5:

$$F_{hc} = mg \cdot sin(\alpha) \tag{3.5}$$

In accordance with Newton's second law of motion, the force F_{acc} required to achieve linear acceleration in a vehicle is expressed by Eq. 3.6

$$F_{acc} = m \cdot a \tag{3.6}$$

where a is the linear acceleration of the vehicle (in m/s^2).

The inertia force $F_{inertia}$, associated with the rotating components of the vehicle, is influenced by the powertrain's moment of inertia. Since this parameter is often not readily available, a common approach is to employ a mass correction factor $C_{inertia}$ to account for the rotational inertia acceleration [25], as shown in Eq. 3.7. A common approximation for $C_{inertia}$ is the one adopted in Table 3.2.

$$F_{inertia} = C_{inertia} \cdot m \cdot a \tag{3.7}$$

The overall tractive effort, F_t, is considered positive when the battery supplies power to the motor, and negative when the motor operates as a generator, supplying power back to the battery. The traction power, P_t (in W), required to propel the vehicle at a given speed v, (in m/s), can be represented by Eq. 3.8:

$$P_t = F_t \cdot v \tag{3.8}$$

Transmission System

Despite there being simpler vehicle models in which the power drawn from the battery is a simple function of driving speed [18], in the model proposed by this Master's Thesis, motor efficiency is represented by a map as a function of motor speed and torque. For this reason, it is necessary to appropriately model a transmission system.
As a first distinctive element, in this vehicle model, the wheels are not directly connected to the motor shaft. Instead, a gear system is used to convert the torque generated by the motor into the torque required at the wheels. Considering the gear ratio of the transmission system as g_{ratio}, the angular speed of the motor, $\omega_{mot,out}$ (in rad/s), can be determined using Eq. 3.9:

$$\omega_{mot,out} = \omega_{wheel} \cdot g_{ratio} = \frac{v}{R} g_{ratio} \tag{3.9}$$

where ω_{wheel} is the angular speed of the wheel (in rad/s), R is the tyre radius (in m) and v (in m/s) is the vehicle speed.

Taking into account transmission system gear efficiency η_{gear}, the mechanical power $P_{mot,out}$ can be expressed, as in Eq. 3.10, for both the cases where traction power is provided to the wheels or energy is recovered in generator mode. Additionally, Eq. 3.11 provides the motor output torque $T_{mot,out}$ (in Nm):

$$P_{mot,out} = \begin{cases} P_t \cdot \eta_{gear} & P_t < 0 \\ P_t / \eta_{gear} & P_t > 0 \end{cases} \tag{3.10}$$

$$T_{mot,out} = \frac{P_{mot,out}}{\omega_{mot,out}} \tag{3.11}$$

Motor

In contrast with the approach adopted by [17], which utilizes a constant motor efficiency, in the proposed Master's Thesis, the construction of the efficiency curves is based on rational and linear fitting, using efficiency values at specific load points, as done by [25]. These load points include 0%, 25%, 50%, 75%, and 100% of the rated load. The assumed part-load efficiency curve for an IM typically varies between 88% and 92%, while a BLDCM ranges from 92% to 95% [25].
By utilizing the characteristic points presented in Table 3.3 and considering the efficiency range for each motor type, the load-efficiency curve can be approximated using a piecewise function, as expressed in Eq.3.12:

$$\eta_{mot}, \eta_{gen} = \begin{cases} (c_1 \cdot x + c_2)/(x + c_3) & 0 \leq x < 0.25 \\ d_1 \cdot x + d_2 & 0.25 \leq x < 0.75 \\ e_1 \cdot x + e_2 & x \geq 0.75 \end{cases} \tag{3.12}$$

where x denotes the mechanical power of the motor $P_{mot,out}$ (in W) as a fraction of its rated power $P_{mot,rated}$ (in W). Figure 3.10 in used to visualize the efficiency curve defined by Eq.3.12 as a function of the aforementioned x.

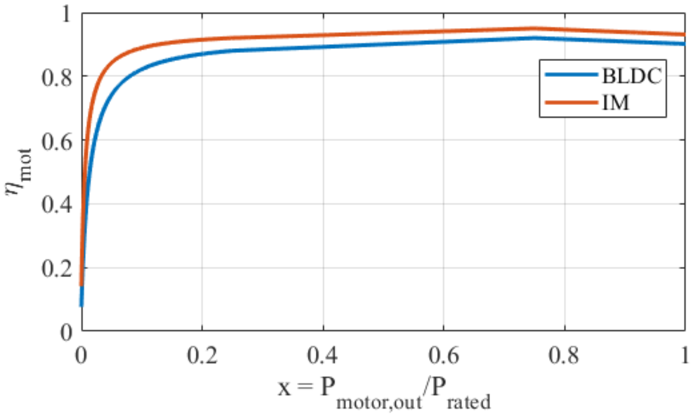


Figure 3.10: Motor efficiency curve for IM and BLDC motor

Table 3.3: Nominal motor efficiency at part and full rated load

Percent Load	**Motor Eff.**
0%	0
25%	$minEff$
50%	$0.98 \cdot maxEff$
75%	$maxEff$
100%	$0.98 \cdot maxEff$

Table 3.4: Coefficients of piece wise function for motor/generator efficiency

	IM		**BDLC**	
	Motor Mode	*Generator Mode*	*Motor Mode*	*Generator Mode*
c_1	0.924300	0.925473	0.942269	0.942545
c_2	0.000127	0.000148	0.000061	0.000067
c_3	0.012730	0.014849	0.006118	0.006732
d_1	0.080000	0.075310	0.060000	0.057945
d_2	0.860000	0.858605	0.905000	0.904254
e_1	-0.073600	-0.062602	-0.076000	-0.066751
e_2	0.975200	0.971034	1.007000	1.002698

One important attribute of electric motors is their tendency to exhibit higher efficiency as their size increases [25]. The correlation between motor size, indicated by rated power, and efficiency is established by considering the prescribed minimum efficiency requirements for electric motors [28]. The average efficiency of motors is determined for each rated power class, and subsequently divided by the maximum average efficiency. The results of these computations are represented in Table 3.5 with the indication of the value of the normalization factor for each rated power class. This computation yields to Eq. 3.13, where the normalization factor ($normfactor$), which serves as coefficient for the motor efficiency calculated by Eq. 3.12, enables the incorporation of motor size as a factor in determining its efficiency.

$$P_{mot,in} = \begin{cases} P_{mot,out} \cdot \eta_{gen} \cdot normfactor & P_t < 0 \\ P_{mot,out}/(\eta_{mot} \cdot normfactor) & P_t > 0 \end{cases} \tag{3.13}$$

Table 3.5: Efficiency normalization factor ($normfactor$)

P_{rated} $[kW]$	30	45	55	75	90	110	132	160	200-375
$normfactor$	0.968	0.978	0.981	0.987	0.990	0.993	0.996	0.998	1.000

In this project, it was assumed that the torque-speed relationship in generator mode mirrors that in motor mode, as long as the braking torque remains below or equal to the maximum generator torque, an assumption already established in the literature [25]. Conversely, if the braking torque surpasses the torque limit, the remaining power is wasted as heat due to mechanical braking. Additionally, the regenerative braking system does not exert any braking force at low vehicle speeds, as the available torque is insufficient [25].

In order to simulate this operation, the vehicle model incorporates a speed-dependent regeneration factor, denoted as $regenfactor$. This factor represents the proportion of available regenerative braking power that can practically be recovered based on the vehicle's speed v (in m/s). Notably, there exists a threshold speed v_1 (in m/s) that must be exceeded for the electrical machine to initiate energy regeneration [26]. Furthermore, the EV machine achieves its maximum regeneration capability for speeds surpassing another threshold speed v_2 (in m/s). For speeds between v_1 and v_2, we assume that the percentage of recoverable braking power increases linearly with the vehicle's speed until reaching the maximum regeneration capability. In this project, the two speed thresholds were set as $v_1 = 10km/h$ and $v_2 = 30km/h$.

Considering the aforementioned details, Eq.3.13 can be re-formulated as follows:

$$P_{mot,in} = \begin{cases} P_{mot,out} \cdot \eta_{gen} \cdot normfactor \cdot regenfactor & P_t < 0 \\ P_{mot,out}/(\eta_{mot} \cdot normfactor) & P_t > 0 \end{cases} \tag{3.14}$$

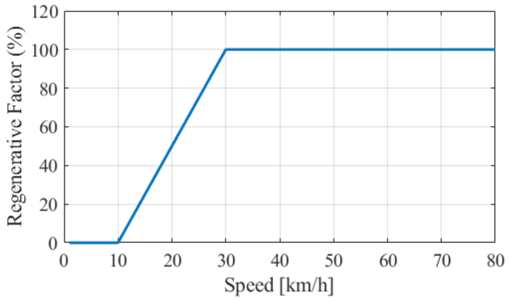


Figure 3.11: Speed-dependent regeneration factor

Battery

In this project, energy and power losses during battery charging and discharging, due to conversion of electric energy into chemical and vice-versa, are expressed through battery round-trip efficiency factor (RTE). Table 3.6 shows the relevant efficiencies of typical battery types [25][50]. Battery power flow in each time instance is expressed by Eq.3.15

$$P_{total} = \begin{cases} P_{battery,out} \cdot \sqrt{RTE} & P_{battery,out} < 0 \\ P_{battery,out}/\sqrt{RTE} & P_{battery,out} > 0 \end{cases} \tag{3.15}$$

where P_{total} (in W) and RTE denote the total power and round trip efficiency of the EV battery respectively, while $P_{batter,yout}$ (in W) is the electric output of the battery to provide power P_{ac} (in W) to the accessories of the EV, and supply the motor with electric power ($P_{motor,in} > 0$) or receive electric power from the generator ($P_{motor,in} < 0$), as given in Eq.3.16:

$$P_{battery,out} = P_{motor,in} + P_{ac} \tag{3.16}$$

Table 3.6: Energy efficiency of typical battery types for EVs

Battery Type	Energy Eff.
Lead Acid	>80%
NiCd	75%
NiMH	70%
Li-ion	>95%

The cumulative energy consumption $E(t)$ can be computed by Eq.3.17:

$$E(t) = E(t-1) + \int_{t-1}^{t} P_{total}(t)\, dt \tag{3.17}$$

Recharge Power Profile

Unlike the approach adopted in the initial project [17], where a constant power charging profile was used regardless of the SoC of the battery, this Master's Thesis project adopts a variable charging profile that depends on the battery's SoC. As highlighted by [18], the CP-CV protocol is considered the closest to real-world charging, with an error margin of less than 2%. In paper [18], the authors decided to divide the charging profile into two parts: an initial phase of constant power charging for SoC ranging from 0% to 80%, followed by a subsequent phase of constant voltage charging from 80% to full battery charge. This charging protocol is mathematically represented by Eq. 3.18

$$P_{charge}(SoC) = \begin{cases} P_{charge,max} & SoC \leq 80\% \\ P_{charge,max} \cdot \frac{1-SoC}{0.2} & SoC > 80\% \end{cases} \tag{3.18}$$

Additionally, Figure 3.12 presents the charging power profile as a function of time. It is evident that the charging power is not constant, but exhibits a decrease after a certain point, where the charging power starts to diminish. This decline, corresponding to reaching an SoC of 80%, results in extended expected charging times compared to the previous model [17]. This observation is particularly significant when considering battery charging beyond the 80% threshold, whereas below this threshold, the outcome remains unchanged.

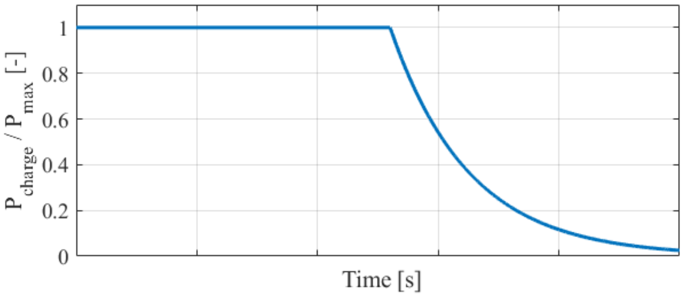


Figure 3.12: Charging Power profile as a function of Time

From the way the charging power profile was formulated in Eq. 3.18, it is possible to calculate the charging time (t_{charge}) required to reach a certain SoC at the end of the charging (SoC_{end}), starting from an initial SoC (SoC_{start}), using the formulas provided in Eq. 3.19, where E_{batt} is the maximum energy the battery is capable to store:

$$t_{charge} = \begin{cases} \frac{(SoC_{end}-SoC_{start})\cdot E_{batt}}{P_{charge,max}} & SoC \leq 80\% \\ max(\frac{(SoC_{end}-SoC_{start})\cdot E_{batt}}{P_{charge,max}}, 0) - \frac{E_{tot}\cdot 20\%}{P_{charge,max}} \cdot log(\frac{1-SoC_{end}}{20\%}) & SoC > 80\% \end{cases} \tag{3.19}$$

3.3.2 Model Comparison

This Section is focused on the comparison between the new model, whose mathematical formulation has been presented in Section 3.3.1, and the old model, adopted in project [17], from the point of view of the energy consumption calculation and required processing time.
At first, a comparison of the energy consumption of the two models has been performed. The speed profile utilized in this section is depicted in Figure 3.13. This speed profile was derived by inputting the coordinates of two edges present in the map into the SUMO software, following a methodology similar to that employed in the execution of Section 3.1.2. Following the approach described in Section 3.1.2, the raw speed profile obtained from SUMO was subsequently subjected to a *Smoothing* process, employing the driving style conditions categorized as *normal*, as indicated in Table 3.1.
The adopted speed profile is characterized by a duration of more than 8000 seconds. It exhibits a notable variety of speeds, encompassing urban segments, suburban stretches and an extensive highway section featuring speeds surpassing 130 km/h. Furthermore, it encompasses acceleration and deceleration of varying magnitude and duration, enabling the assessment of transient performance.

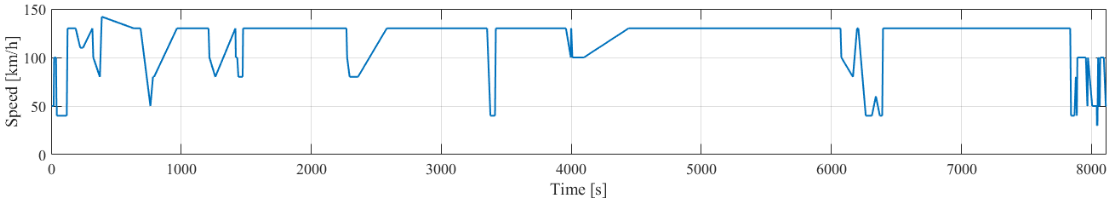


Figure 3.13: Speed Profile

Firstly, we focused on comparing the tractive power demanded by the wheels in the two models under investigation. To visualize this performance comparison, Figure 3.14 illustrates the discrepancy between the tractive power calculated using the new model and that derived from the old model, as a function of time.

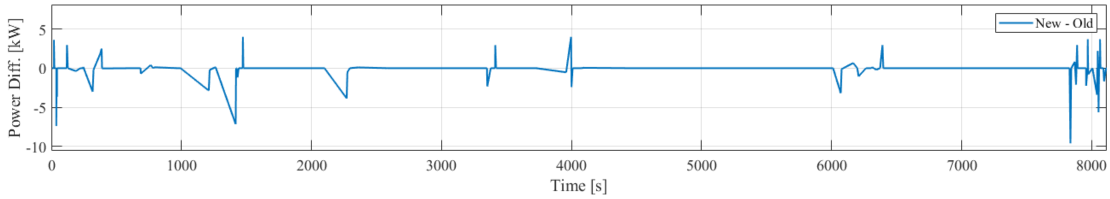


Figure 3.14: Difference in Power Requests at Wheel Level (New-Old)

It is crucial to emphasize that the fundamental dissimilarity between these two

models lies in the incorporation of an inertia factor, which considers the rotational inertia of the power-train components. As a result, the graph depicted in Figure 3.14 exhibits non-zero values solely at instances where discrepancies in speed occur, signifying the influence of inertia. When the vehicle operates at a constant speed, the discrepancy in the power demanded by the wheels between the two models becomes negligible. Conversely, during acceleration, the new model exhibits a tendency to require higher power from the motor, as a portion of the power is allocated to overcome the inertia of the power-train. Conversely, during deceleration, a reduced power demand is observed.
However, in terms of energy requested to the battery, the anticipated trend of the newly developed model, with higher power demands during acceleration and lower power requirements during deceleration, is not evident. As depicted in Figure 3.15, battery energy requirement is lower in the new model compared to its predecessor.

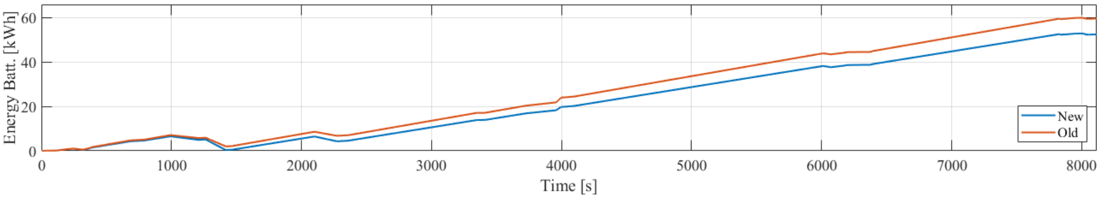


Figure 3.15: Energy Requests at Battery Level

The discrepancy can be attributed to the introduction of novel efficiencies and corrective parameters, which differ from those employed in the previous model. Notably, the overall vehicle efficiency is no longer treated as a constant; instead, it is contingent upon demanded power and speed of the electric motor. Furthermore, the inclusion of a regenerative capacity factor for the electric motor has been introduced, which now depends on the instantaneous vehicle speed.

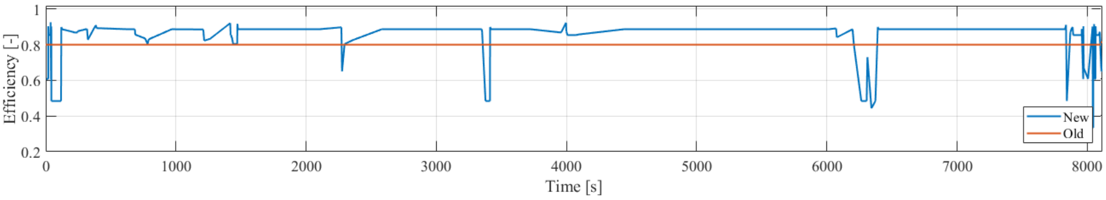


Figure 3.16: Combined Efficiency of Battery and EM

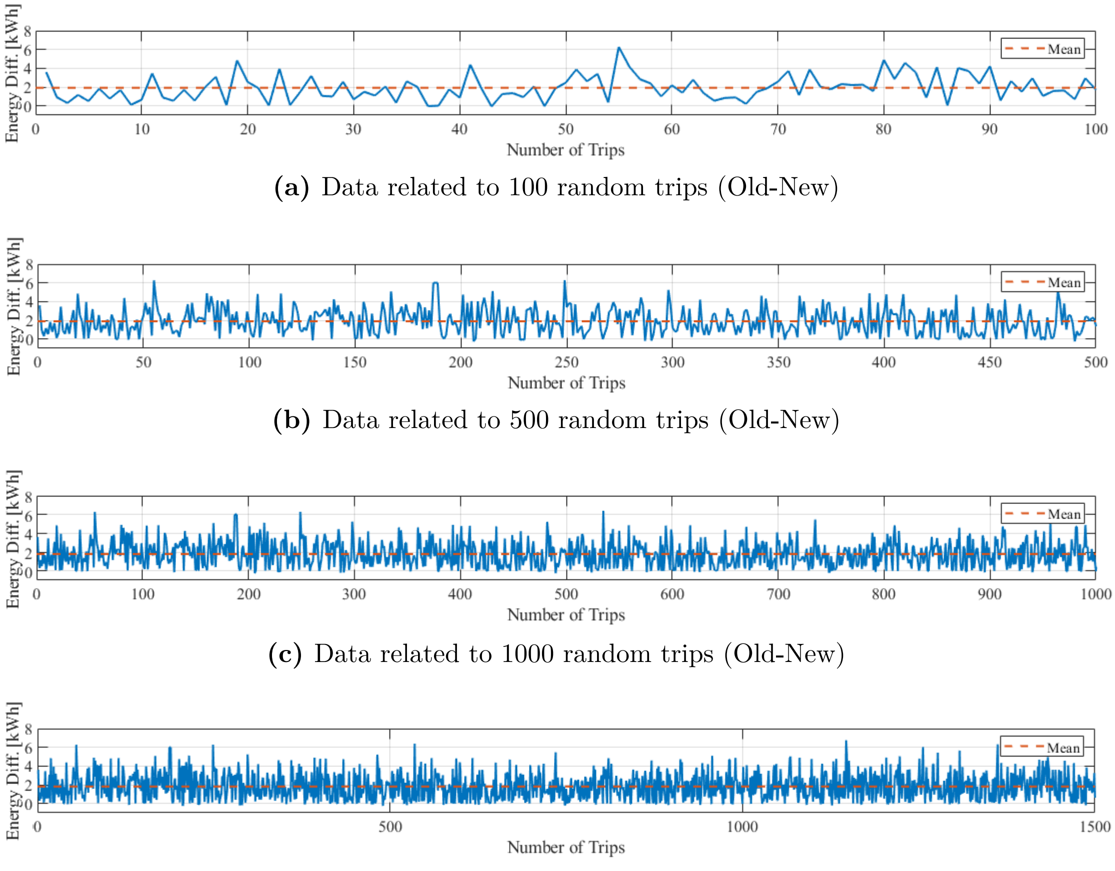
As demonstrated in Figure 3.16, the primary source of disparity in battery energy demand stems from the divergent combined efficiency of the EM and battery. Over long stretches of the journey, the combined efficiency in the new model surpasses that of the constant efficiency employed in the previous model. Notably, during periods of constant high travel speed, which constitute a significant portion of the

travel duration, the power required at the wheels remains the same for both old and new models, but the augmented combined efficiency in the new model yields reduced energy consumption. Nevertheless, there exist specific instances, typically occurring at lower travel speeds, where the efficiency computed with the new model falls short in comparison to the constant efficiency adopted in the previous model. This discrepancy arises from the combined influence of the EM's efficiency being less pronounced at lower power demands and the regenerative factor substantially nullifying recharging when the vehicle's speed is low.
After analyzing the differences between the two models, specifically focusing on subsets of a single journey, the objective shifted towards obtaining a broader perspective on the disparities between these models. To accomplish this, conducted simulations involving a varied number of trips (100, 500, 1000, and 1500 random trips), while considering the disparity in battery energy consumption between the two models. Furthermore, taking into account the duration of each trip, computed an indicative value for the energy consumption difference between old and new models, using a standardized mean reference trip length of 100 km. The resulting data have been presented in Table 3.7 for each simulation cycle. In addition, Figure 3.17 exhibits individual outcomes obtained from the simulation cycles involving 100, 500, 100 and 1500 trips.

Table 3.7: Battery's Energy Request - New Model vs. Old Model

N° of Trips	Mean Distance $[km]$	**Mean Battery Diff. (Old-New)** $[kWh]$	
		Over each Trip	Over 100 km
100	126.26	1.817	1.441
500	124.18	1.949	1.570
1000	124.24	1.861	1.497
1500	123.90	1.915	1.549

In this Master's Thesis project, focus extends beyond the final outcome of each program segment to encompass the consideration of processing time. The aim is to develop a program that not only yields reliable results, but also operates with optimal processing speed. Thus, comprehensive analysis has been conducted, as depicted in Table 3.8, comparing the processing times associated with each simulation cycle. The evaluation was performed using the same set of random trips employed in the creation of Table 3.7. Consistent with previous approach, detailed information for each simulation cycle have been provided, including the number of random trips considered, the average distance traveled per trip, the average processing time and the processing time normalized per 100 km trips for the old model. Additionally, the corresponding values for the new model have been presented, both in terms of average processing time and processing time normalized per 100 km.



(a) Data related to 100 random trips (Old-New)

(b) Data related to 500 random trips (Old-New)

(c) Data related to 1000 random trips (Old-New)

(d) Data related to 1500 random trips (Old-New)

Figure 3.17: Cumulative Battery Energy Consumption Difference (Old-New)

It is crucial to emphasize that the processing time presented in Table 3.8 exclusively pertains to the verification of energy consumption, with no involvement in the route computation from the origin to the destination. Furthermore, it is crucial to emphasize that the comparison was conducted by analyzing the outcomes derived from a set of random trips within the operational map depicted in Figure 3.5. This analysis omitted any stops for battery recharging and did not account for the battery's maximum energy storage capacity. This approach was deliberate, as the primary objective was to gather indicative data pertaining to the variance in energy consumption from the battery for a generic 100-kilometer route.

The subsequent analysis, reported in Table 3.8, specifically addresses the comparison between the processing time required for energy consumption calculation and that needed to generate the velocity profile from the starting point to the endpoint. Notably, processing time significantly increases with the new vehicle model in comparison to the original model. This disparity arises from the introduction of novel functions for efficiency determination, inevitably elongating the processing time when compared to the previous model, where all such efficiencies were restricted to a single constant value.

Table 3.8: Processing Time - New Model vs. Old Model

N° of Trips	Mean Distance $[km]$	**Mean Proc. Time (Old)** $[ms]$		**Mean Proc. Time (New)** $[ms]$	
		$[ms]$	$[ms/100km]$	$[ms]$	$[ms/100km]$
100	126,26	0,019	0,015	0,073	0,058
500	124,18	0,009	0,007	0,066	0,053
1000	124,24	0,006	0,006	0,057	0,046
1500	123,90	0,005	0,005	0,045	0,036

The considerable increase in processing time could potentially undermine the advantage offered by the improved estimation of energy consumption. However, the project finds support in Table 3.9, which presents a comparison between the processing time for energy consumption using the new model and the processing time required for route calculation, from which the velocity profile is derived. Table 3.9 provides a comprehensive overview of each simulation cycle, including the number of trips, the average distance traveled per trip, the average processing time normalized to 100 km for energy consumption calculation with the new model, as well as the average processing time and the normalized one used for route calculation and the subsequent velocity profile definition.

Table 3.9: Processing Time - Energy Consumption vs. Route Definition

N° of Trips	Mean Distance $[km]$	**Mean Proc. Time (New)** $[ms]$		**Mean Proc. Time Routes** $[ms]$	
		$[ms]$	$[ms/100km]$	$[ms]$	$[ms/100km]$
100	126,26	0,073	0,058	267	212
500	124,18	0,066	0,053	293	236
1000	124,24	0,057	0,046	264	213
1500	123,90	0,045	0,036	213	172

The results clearly demonstrate that the processing time of energy consumption calculation requires orders of magnitude less time compared to the route calculation. Therefore, the increase in processing time for energy consumption calculation has no practical impact on the overall processing time.

3.4 DBSCAN Clusterization Algorithm

As previously mentioned in Section 3.1.3, this Master's Thesis project proposes the utilization of a clustering system for charging stations in order to accelerate the construction of the cost matrix and subsequently apply the Dijkstra's algorithm to determine the optimal sequence of nodes that minimizes the overall cost, specifically the total travel time.

The concept of clustering charging stations draws inspiration from the way humans typically plan their EV journeys. It involves either opting to stop within a general geographical area that encompasses numerous charging stations meeting the desired requirements (power, availability, cost, etc.) or specifically selecting a particular charging station within a specific geographic location.

Among the plethora of existing clustering algorithms, the choice falls upon a novel approach known as the DBSCAN algorithm [51]. This algorithm allows for the identification and grouping of highly dense geographic regions containing charging stations, while also identifying isolated stations situated in distinct geographic areas as outliers.

In a more detailed analysis, the DBSCAN algorithm [51] is a clustering algorithm widely used in various domains for identifying clusters based on data density. Unlike traditional clustering algorithms, DBSCAN does not rely on predefined cluster shapes or centroid assumptions, making it particularly suitable for datasets with varying densities and complex cluster structures.

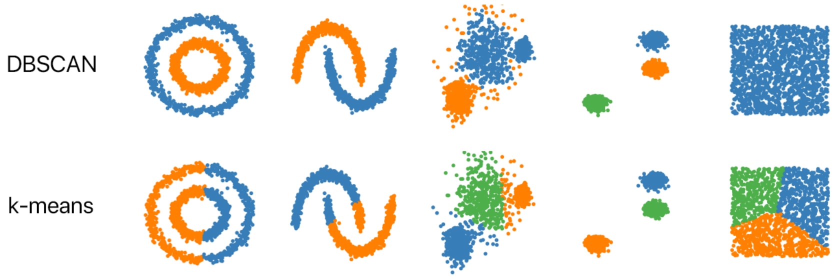


Figure 3.18: DBSCAN vs. k-means clustering [52]

The fundamental principle of DBSCAN lies in the notion of density. The algorithm starts by selecting an unvisited data point randomly and examines its neighborhood within a specified distance threshold (ϵ). If the number of points within this neighborhood surpasses a user-defined minimum threshold ($minPts$), a new cluster is formed. The algorithm continues to expand the cluster by iteratively exploring the neighborhoods of newly added points, effectively connecting them to the existing cluster.

DBSCAN categorizes data points into three types: core points, border points, and noise points. As shown in Figure 3.19, core points are characterized by having a sufficient number of neighboring points within the distance threshold, indicating a dense region. Border points have fewer neighbors, but are still within the distance threshold of another core point. Noise points, on the other hand, do not satisfy the density criteria and are considered outliers, not assigned to any cluster. In Figure 3.19, the core points depicted in red and the border points shown in yellow form a single cluster. On the other hand, the blue point represents a noise point that does not belong to any specific cluster.

By designating noise points as outliers, the algorithm effectively handles noisy or sparse regions in the dataset, ensuring accurate cluster identification. This aspect holds significant importance within our project, given the presence of numerous isolated charging stations that are far from major urban areas. This particular characteristic allows the algorithm to efficiently identify these charging stations as outliers.

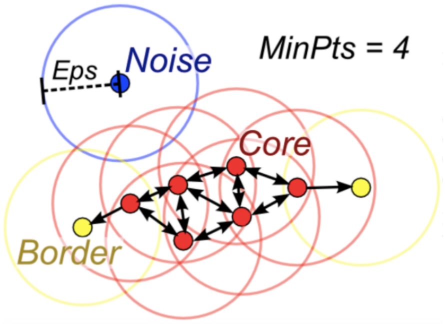
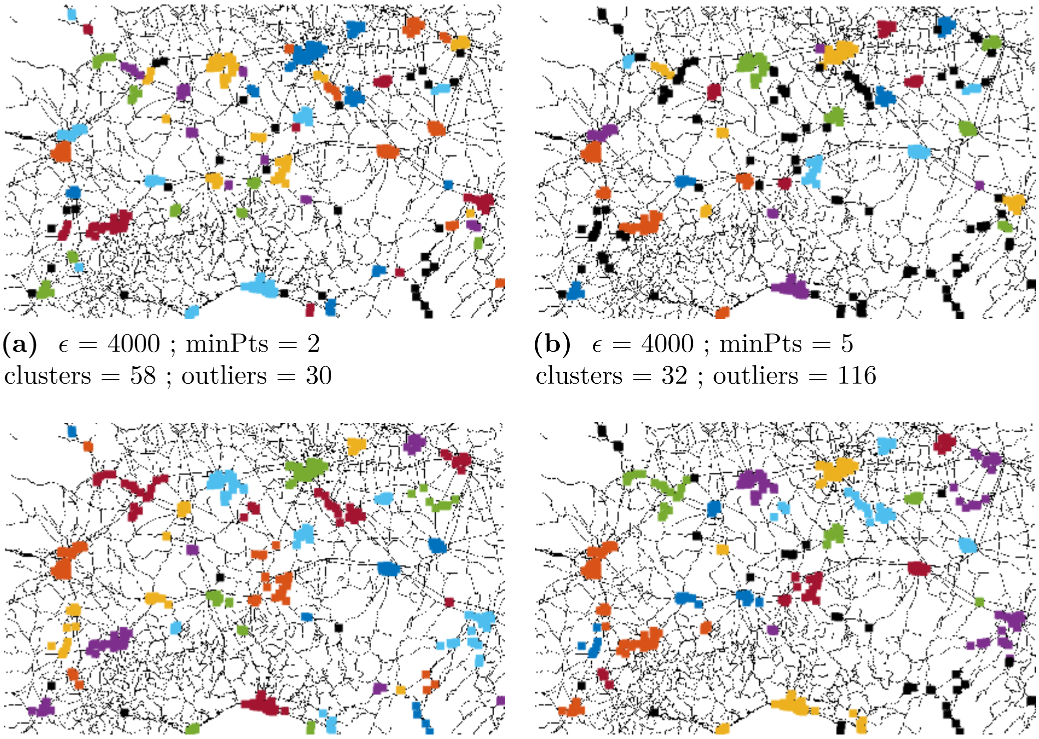


Figure 3.19: DBSCAN at work ($MinPts = 4$)

However, selecting appropriate parameters for DBSCAN is crucial for obtaining meaningful results. The ϵ parameter determines the distance threshold for defining a neighborhood, while the $minPts$ parameter sets the minimum number of points required to form a dense region. These parameters demand careful consideration and experimentation to achieve optimal clustering outcomes, often guided by domain knowledge and data characteristics.

In Figure 3.20, the results of four distinct clustering processes conducted with four different pairs of ϵ and $minPts$ parameters are presented. Geographic locations within the same cluster are assigned the same color, while black points denote locations identified as outliers through clustering. It is worth noting that locations with the same color may belong to different clusters due to their significant spatial separation. This discrepancy arises from the greater number of clusters compared to the available colors for representation. Hence, color classification should primarily be utilized when distinguishing adjacent clusters. The analysis of Figure 3.20

reveals how the selection of different ϵ and *minPts* values can significantly impact the clustering of charging stations. Consequently, this affects the number of nodes within the network, regardless of whether it pertains to *macro* or *micro-research*. This observation underscores the importance of carefully choosing these parameters to achieve accurate and meaningful clustering results in the context of charging station analysis.



(a) $\epsilon = 4000$; minPts = 2
clusters = 58 ; outliers = 30

(b) $\epsilon = 4000$; minPts = 5
clusters = 32 ; outliers = 116

(c) $\epsilon = 8000$; minPts = 2
clusters = 41 ; outliers = 4

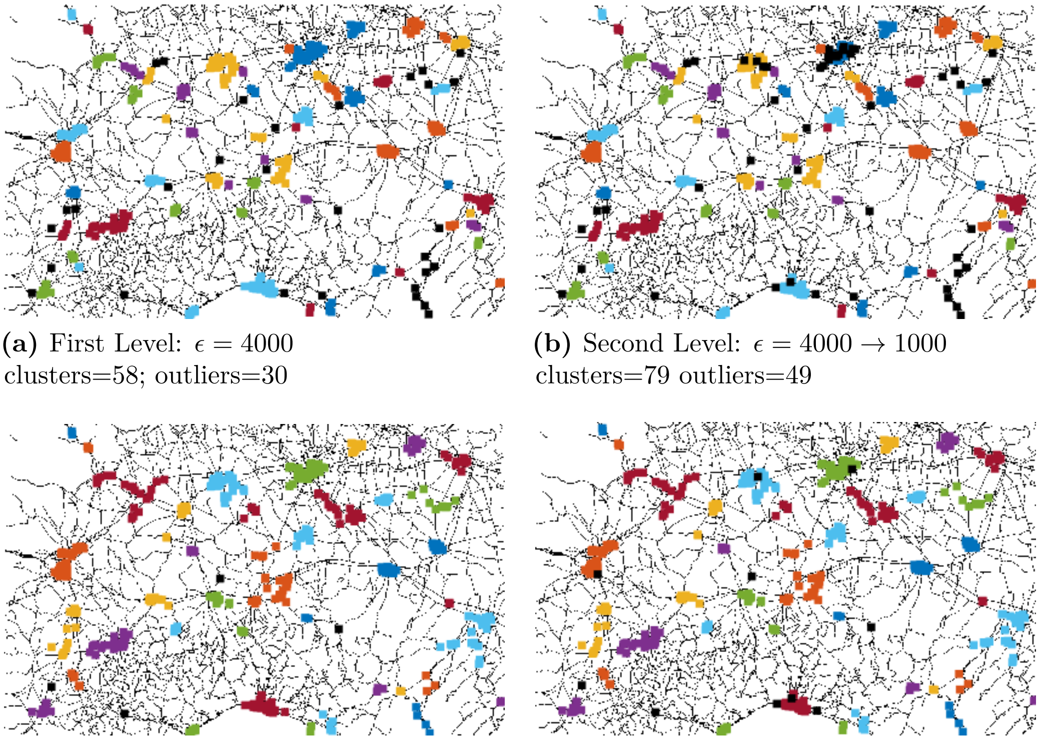
(d) $\epsilon = 8000$ minPts = 5
clusters = 31 ; outliers = 40

Figure 3.20: Influence of ϵ and minPts on the clustering

Defining a single set of ϵ and *minPts* values that can effectively capture the characteristics of all possible journeys on the map presents a challenge. The main issue arises from the difficulty of achieving at the same time a reduced number of nodes in the clustering output, while maintaining efficient processing times during *macro-research*, without compromising the identification of clusters containing a substantial number of charging stations, which would considerably increase the processing time during *micro-research*.

To address this challenge, a dual-level clustering approach is proposed. This approach involves two distinct levels, each characterized by a specific pair of ϵ and *minPts* values. The primary objective of the first level is to perform clustering

with higher ϵ and *minPts* values, effectively reducing the number of nodes and consequently improving the processing time during *macro-research.* In the second level, the focus shifts to clusters that encompass a significant number of charging stations, surpassing a predefined threshold. Only the charging stations within these densely populated clusters undergo a secondary clustering process with lower ϵ and *minPts* values. As a result, these clusters are subdivided into multiple smaller clusters, each exhibiting a reduced number of charging stations, that will lead to a reduce processing time during *micro-research.*
The effectiveness of this two-level clustering technique is demonstrated in Figure 3.21. Notably, this approach minimally impacts the node count within the *macro-research* network, while substantially decreasing the number of stations that need to be examined during *micro-research*, ultimately enhancing the efficiency of the overall analysis process.



(a) First Level: $\epsilon = 4000$
clusters=58; outliers=30

(b) Second Level: $\epsilon = 4000 \rightarrow 1000$
clusters=79 outliers=49

(c) First Level: $\epsilon = 8000$
clusters=41 outliers=4

(d) Second Level: $\epsilon = 8000 \rightarrow 2000$
clusters=55 outliers=10

Figure 3.21: Two Level Clusterization

3.5 Pruning Techniques

The overall processing time is primarily influenced by two factors: the computational time required by SUMO to calculate the speed profile from the origin node to the destination node, along with the associated smoothing process discussed in Section 3.4; as weel as the processing time associated with the feasibility verification of the journey, as extensively elaborated in Section 3.3.
As mentioned in Section 3.3, the introduction of a new vehicle model has resulted in an increase in processing time. However, as highlighted in Table 3.9, the processing time for performing the feasibility check is significantly smaller compared to the time spent by SUMO in calculating the speed profile. Hence, it can be inferred that the majority of the processing time is dedicated to determine the speed profile with SUMO, with only a marginal increment caused by the subsequent feasibility check. These findings strongly support the adoption of a more sophisticated vehicle model, as the benefits in terms of energy consumption prediction far outweigh the drawbacks arising from the increased processing time.
In a more general context, taking into account the conducted simulations reported in Table 3.9, it has been determined that the SUMO software, complemented by the Traci4Matlab function family, requires an average of 0.2 seconds to calculate the speed profile and apply smoothing techniques for mitigating acceleration peaks and valleys. This step involves creating a network branch, which will later undergo later the Dijkstra algorithm's.

Even when considering the reduction of charging stations through clustering in the *macro-research*, as discussed in Section 3.4, it becomes apparent that the complete construction of the cost matrix consumes a significant amount of time. For instance, referring to the example depicted in Figure 3.20(a), clustering with an $\epsilon = 4000$ and $MinPts = 2$ yields a network comprising 109 nodes, encompassing both clusters and outliers. Considering that the resulting cost matrix has dimensions of $(109+2) \times (109+2)$, it entails 12321 cells, each requiring the computation of velocity profiles and travel feasibility. Based on the average processing time depicted in Table 3.9, this operation would demand approximately 2370 seconds, equivalent to over 39 minutes. Clearly, such a computational duration is deemed unacceptable for real time software implementation. Moreover, the situation does not improve even if different values of ϵ and $MinPts$ are employed. As demonstrated in the same Figure 3.20(c), adopting an $\epsilon = 8000$ and $MinPts = 2$ results in a network of 59 nodes, comprising clusters and outliers, and leads to a matrix of size 61×61 with 3721 cells. On average, the processing time required for this computation amounts to approximately 744 seconds, or 12 minutes. It is important to highlight that the aforementioned rough processing time solely pertains to the initial phase, known as *macro-research*. The subsequent phase, known as *micro-research*, which involves examining individual stations within each cluster, still needs to be accounted for. Consequently, while the utilization of ϵ and $MinPts$ values that reduce the total number of nodes yields a processing time advantage during *macro-research*, it inadvertently leads to an increase in nodes within the new network during *micro-research*. This is due to the fact that, on average, a higher number of charging stations exist within each cluster.

Through multiple simulations conducted without any form of pruning techniques, several distinct phenomena were observed. Firstly, it was noted that the majority of the cells within the cost matrix exhibited values that were out of scale. As explained in Section 3.1.3, such values indicate the non feasibility of a trip in terms of energy consumption. Secondly, the algorithm's selection of charging stations, determined by Dijkstra's algorithm, predominantly favored locations along the direct path connecting the user-defined origin and destination nodes. Thirdly, in cases where feasible, the Dijkstra algorithm consistently prioritized charging stations with higher power capacity to minimize the time spent on battery charging. Lastly, when a trip involved multiple stops, the algorithm shows the tendency to elongate each intermediate leg by selecting charging stations that allowed for significant battery discharge, while still adhering to the user-defined limits outlined in Section 3.1.1.

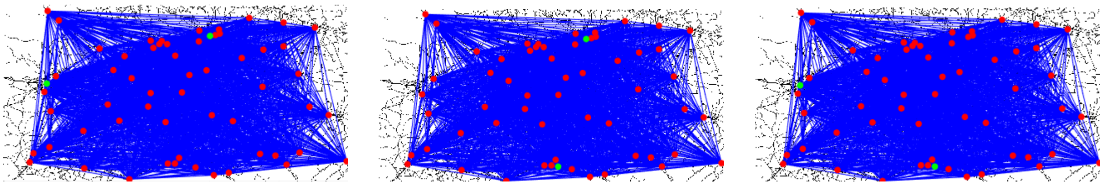
Through the analysis of these phenomena, the idea has emerged to incorporate some pruning techniques before utilizing SUMO. The aim is to partially construct the cost matrix without relying on SUMO. This construction will exclusively focus on cells/trips that are deemed irrelevant for determining our own trip. The approach involves pro-actively identifying these cells assigning them an out-of-scale value, in a similar manner to what was described in Section 3.1.3. This ensures that they are not considered by Dijkstra's algorithm when calculating the path that minimizes total travel time. The significant advantage of these pruning techniques lies in the fact that the early identification of these cells/trips, and the subsequent value assignment, can be accomplished in a fraction of the time required by SUMO. By minimizing the reliance on SUMO, which constitutes the primary processing time bottleneck, only trips considered potentially useful will undergo its evaluation, while all other trips/cells can be swiftly assigned out-of-scale values.

In the following, the elements that have contributed to the pruning techniques in both *macro-research* and *micro-research* will be listed. The two pruning systems differ between the two phases, as in the first phase it is necessary to identify the best geographical areas for the stops, while in the second phase, it is already known that these geographical areas are crucial for achieving an optimal result. The pruning techniques will be presented using three key routes as examples: from

Turin to Milan, from Milan to Genoa, and from Turin to Genoa. Through figures it is shown how, progressively, pruning techniques reduce the number of remaining useful connections, leaving only those where further verification, using the SUMO environment, is truly necessary.

3.5.1 Pruning for *Macro-Research*

As previously mentioned, this section focuses on defining the various steps involved in the implementation of the pruning techniques for macro-research. The pictures from Figure 3.22 to Figure 3.29 presents the map used in this project, where red and green points are interconnected by blue lines. The two green points represent the user-defined origin and destination points, as specified in Section 3.1.1, while the red points indicate the charging stations identified after applying the clustering technique. Specifically, the DBSCAN algorithm was employed with characteristic values of $\epsilon = 8000$ and $MinPts = 2$, resulting in a network comprising 57 nodes, including both clusters and outliers. Furthermore, the blue lines connecting pairs of nodes represent the branches of our node network, representing specific cells in the cost matrix and potential travel routes.
In accordance with Figure 3.22, it is observed that, in the absence of any pruning algorithm, the journeys from Turin to Milan, from Milan to Genoa, and from Genoa to Turin exhibit an equal number of connections, amounting to 11990. Considering a starting number of nodes $n = 57$, this count of connections was obtained starting from $(n + 2) \cdot (n + 2) = 3481$ connections present in the entire matrix. Then $(n + 2) = 59$ cells on the diagonal of the matrix, were excluded since they represent trips from a node to itself. An additional $2 \cdot (n + 1) = 116$ cells representing journeys where the starting node serves as the destination or vice versa, were also excluded.



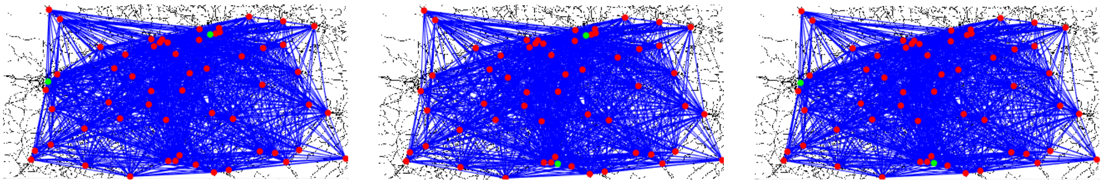
(a) Turin to Milan - 3306 **(b)** Milan to Genoa - 3306 **(c)** Genoa to Turin - 3306

Figure 3.22: Node net at *macro-research* with no pruning

Battery Range Pruning

The first pruning step takes into account the maximum range achievable by the electric vehicle under examination, without the need for recharging. Naturally, this

maximum range strongly depends on the driving speed and it results challenging to predict in advance. However, the aim of this initial step is to consider generally long journeys that would yield an out-of-scale value within the cost matrix, indicating complete battery depletion and surpassing the user-defined SoC limits. These lengthy trips are often traversed primarily via highways or expressways to minimize travel time. This provides us with additional insights into potential energy consumption. Considering the vehicle's consumption at highway and expressway speeds, we assumed an average consumption rate of approximately 20 kWh per 100 km. By dividing the battery capacity by this average consumption, we obtained an approximate value for the maximum distance that can be covered on a single charge. Subsequently, any branches connecting two nodes with an air distance greater than the calculated maximum distance are deemed unfeasible and assigned an out-of-scale value. This ensures they are not taken into account during the determination of the path that minimizes travel time. It is crucial to clarify that the distance considered for comparison with the maximum distance achievable on a single charge is the air distance between two nodes. This distance is calculated solely based on geographical coordinates, without utilizing SUMO and it is always shorter than the actual distance covered by the vehicle from the starting node to the destination node through the road network.
As demonstrated in Figure 3.23, employing this first pruning step results in precisely 2274, 2285, and 2275 connections for the trips from Turin to Milan, from Milan to Genoa, and from Genoa to Turin, respectively.



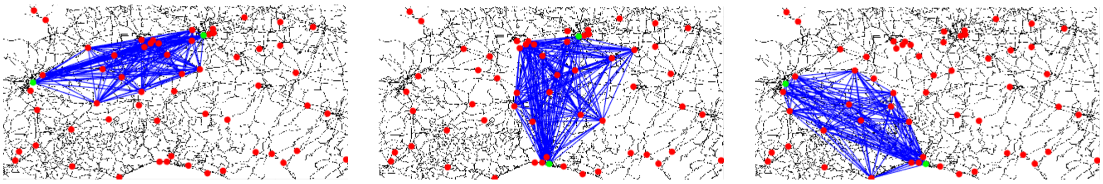
(a) Turin to Milan - 2274 **(b)** Milan to Genoa - 2285 **(c)** Genoa to Turin - 2275

Figure 3.23: Node net at *macro-research* with 1 step of pruning

Geometrical Pruning

In Figure 3.23, it is evident that the network of nodes remains extensive and intricate even after applying the first step of pruning step. Notably, there are numerous nodes, along with their respective connections, that are unlikely to play a significant role in route planning. For instance, during the journey from Turin to Milan, it is implausible to consider stopping near Genoa. Similarly, transiting through Brescia is inconceivable when traveling from Turin to Genoa. Thus, it

becomes imperative to restrict the nodes under consideration solely to those situated along the journey between the origin and destination nodes. To accomplish this, a bounding rectangle was created based on the coordinates of the user-defined starting and destination nodes. For any connection where either node fell outside this bounding rectangle, an out-of-scale value was assigned within the cost matrix. The construction of the bounding rectangle involved drawing a direct line connecting user-defined starting and destination nodes. From this line, two parallel lines were generated at a specified distance, effectively forming a corridor of specified width. The width of the corridor is set at either 40 km or 30 km, depending on whether the air distance between the coordinates of the starting and destination charging stations is less than 200 km (40 km) or greater (30 km). The corridor was then intersected by two orthogonal lines passing through the user-defined starting and destination points.
As illustrated in Figure 3.24, this second pruning step resulted in the following number of connections for the respective journeys: 306 for Turin to Milan, 1249 for Milan to Genoa, and 419 for Genoa to Turin.



(a) Turin to Milan - 306 **(b)** Milan to Genoa - 340 **(c)** Genoa to Turin - 209

Figure 3.24: Node net at *macro-research* with 2 steps of pruning

Angular Pruning

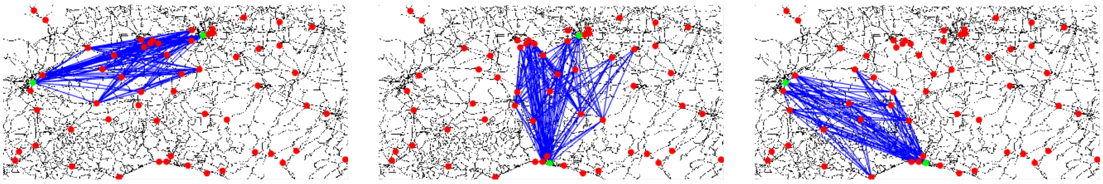
As depicted in Figure 3.24, the network of connections between nodes now exhibits a higher degree of alignment with the intended objectives, providing initial insights into user-defined journey's starting and ending nodes, as well as the designated geographical areas for battery recharge stops. Nevertheless, the total number of connections for all three routes remains considerably high. To address this issue, a criterion to identify journeys that do not substantially bring us closer to the desired final node has been employed. Specifically, for each viable journey showcased in Figure 3.24, the angle formed between the straight line connecting the user-defined starting and ending nodes and the straight line linking a given starting and ending node within the network has been examined. If the resulting angle exceeds a predetermined acceptance angular threshold, the corresponding journey is deemed impractical as it fails to significantly converge towards the target destination. By implementing this methodology, we have derived a revised node network comprising only journeys that effectively propel us towards the intended endpoint, as they follow trajectories that, within a specific angular tolerance, closely align with the original user-specified journey. Consistent with the preceding two pruning stages, any routes/cells that violate the constraints imposed by this pruning mechanism have been assigned an out-of-scale value, ensuring their exclusion from the calculation of the minimum-cost path.

This second phase of pruning encompasses an additional advantage that may not be immediately evident by simply visually comparing the reduction in connections depicted Figure 3.25. In the context of this preparatory stage prior to utilizing the SUMO environment, it is important to note that the cost matrix is constructed as a nearly symmetric square matrix. This near symmetry arises from the fact that cell *ij* represents the total travel time required to travel from the i^{th} starting node to the j^{th} destination node, while the symmetric cell at position *ji* represents the travel cost from j^{th} starting node to the i^{th} destination node. Considering the way in which the initial two pruning steps were designed, when a connection from the i^{th} node to the j^{th} node has not been pruned out by the first two steps, it follows that the symmetric connection from the j^{th} node to the i^{th} node is also still considered plausible.

By introducing this third step of pruning, if a connection/cell *ij*, between a starting i^{th} node and a destination j^{th} node, is still considered plausible thanks to its alignment with the original user-defined journey, the symmetric cell at position *ji* will be regarded as implausible. This is because the journey originating from the j^{th} node and ending at the i^{th} node displays a divergent direction compared to the connection/cell *ij*. Hence, if the connection or cell *ij* is oriented towards our desired destination, it is guaranteed that the connection or cell *ji* will lead in the opposite direction, away from the intended destination. It is important to emphasize that, even though it's certain that if the connection/cell *ij* is plausible, in terms of its alignment with the user-defined trip, the connection/cell *ji* is not aligned, the opposite cannot be automatically assumed. In simpler terms, if connection/cell *ij* is not plausible, it does not imply that connection/cell *ji* is automatically plausible. Consequently, during the third stage of pruning, it is essential to monitor of all connections/cells deemed plausible after the second pruning stage. Nonetheless, this approach offers a substantial advantage in terms of the efficacy of pruning techniques, leading to the reduction of over half of the connections/cells that would otherwise have redundantly been considered. This efficiency gain stems from the fact that, in addition to eliminating all connections/cells *ji* for which the counterpart *ij* is plausible, both connections/cells *ij* and *ji* can be pruned if neither aligns, within a defined tolerance, with the user-defined trip.

As illustrated in Figure 3.25, this second pruning step resulted in the following number of connections for the respective journeys: 126 for Turin to Milan, 440 for

Milan to Genoa, and 151 for Genoa to Turin.



(a) Turin to Milan - 126 **(b)** Milan to Genoa - 128 **(c)** Genoa to Turin - 86

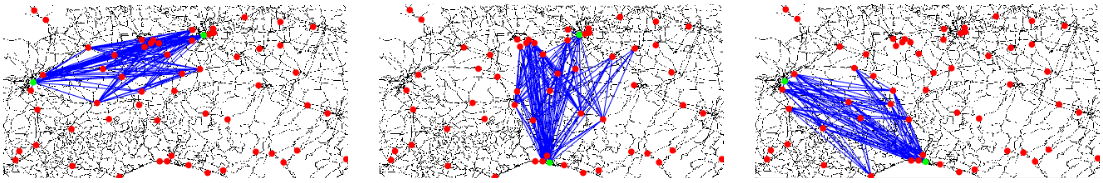
Figure 3.25: Node net at *macro-research* with 3 steps of pruning

Short Trip Pruning

The subsequent pruning step is derived from the analysis of results obtained before implementing any pruning techniques. It was observed that the Dijkstra algorithm exhibits a tendency to favor longer trips, aiming to maximize battery discharge before initiating a subsequent recharge. This trend is attributed to the non-constant profile of the charging power, which tends to decrease beyond a SoC of 80%. Despite the total energy consumption being equal across the entire journey, opting for multiple short trips and more frequent recharging when the SoC is above 80% results in charging at a lower power level than the maximum permitted. Consequently, the program prioritizes minimizing the total travel time by predominantly charging the energy with a SoC below 80% while utilizing the maximum permissible power. To address this trend, a fourth pruning step is introduced, which identifies trips covering shorter distances and excludes them from further analysis within the SUMO environment. To identify these trips, the analysis begins with the set of trips that were deemed valid after the previous pruning step. The straight-line distance between the starting and destination nodes of each trip is considered. If this distance falls below a certain fraction of the maximum range achievable with a single battery charge, the trip/cell is assigned an out-of-scale value, following the approach adopted in previous steps. The maximum distance that our vehicle can cover is determined based on the assumption made during the initial pruning step, which estimated an average energy consumption of approximately 20 kWh per 100 km for each trip. By leveraging this average consumption value and the known energy capacity of the battery, an average maximum travel range and the corresponding acceptance fraction for this pruning step are computed.

It is important to note that this pruning step is not applied to connections connections departing from the starting point of the travel. This decision is justified by the consideration that, in certain cases, like starting with a low SoC of the battery, it is advantageous to halt the journey at the earliest opportunity, potentially utilizing

charging stations with high power capacities. This strategic pause allows for a subsequent continuation with a comfortably high SoC, effectively mitigating range anxiety concerns from both the driver's and passenger's perspective.
As illustrated in Figure 3.26, this fourth pruning step resulted in the following number of connections for the respective journeys: 111 for Turin to Milan, 391 for Milan to Genoa, and 139 for Genoa to Turin.



(a) Turin to Milan - 111 **(b)** Milan to Genoa - 119 **(c)** Genoa to Turin - 82

Figure 3.26: Node net at *macro-research* with 4 steps of pruning

Charging Power Pruning

This new step of pruning arises, similar to the previous one, from the analysis of results obtained prior to the implementation of pruning techniques. It has emerged that the Dijkstra algorithm exhibits a tendency to favor trips where the charging station power is maximized within the vehicle's allowable limits. This preference aims to minimize the charging time and, consequently, the overall travel duration. In order to beforehand identify such trips/cells, an out of scale value has been assigned to the remaining trips from the previous step of pruning that feature a destination node linked to a clustered charging station with a maximum power below the permissible threshold. It is worth noting that, since the representative charging station of the entire cluster offers a charging power greater than or equal to any other station within the cluster, it is appropriate to focus this acceleration check solely on that particular station. It is important to emphasize that this acceleration stage provides significant advantages when adopting a charging power above the minimum requirement (for example 50 kW instead of the minimum 22 kW). As demonstrated in Figure 3.28, substantial reductions in the number of nodes within the network were achieved, showcasing the efficacy of this approach.
As illustrated in Figure 3.28, this fifth pruning step resulted in the following number of connections for the respective journeys: 78 for Turin to Milan, 276 for Milan to Genoa, and 92 for Genoa to Turin.

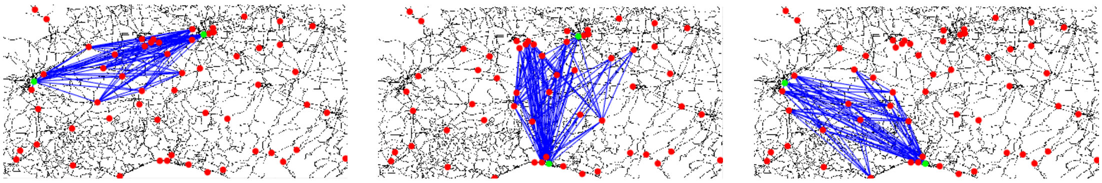

(a) Turin to Milan - 78 (b) Milan to Genoa - 106 (c) Genoa to Turin - 72

Figure 3.27: Node net at *macro-research* with 5 steps of speed-up

Isolated Nodes Pruning

With the introduction of the latest step, the pruning techniques reach their conclusion, as all the elements identified during the analysis of results and cost matrices, without the application of any pruning technique, have been appropriately transformed into corresponding constraints.

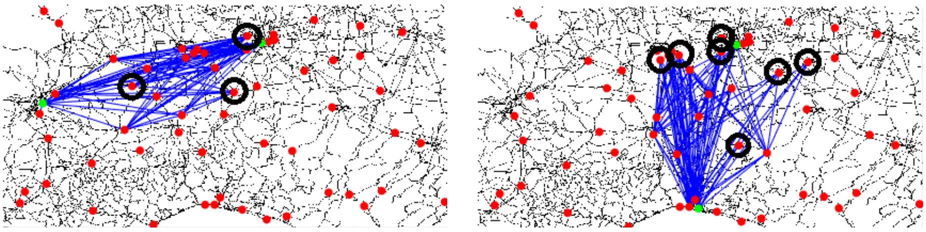

(a) Turin to Milan - 78 (b) Milan to Genoa - 276

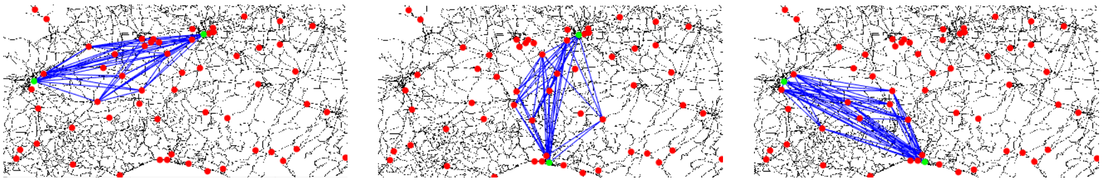
Figure 3.28: Node net at *macro-research* with 5 steps of speed-up

However, as highlighted in Figure 3.28 by the black circles around certain network nodes, there are still numerous trips/cells that do not contribute to defining the optimal path. These trips are characterized by originating from nodes that are not destinations of any other connection or arriving at nodes from which no new connection originates.
To eliminate these fictitious connections, which would never be utilized for generating a path, a comprehensive review of the cost matrix is conducted multiple times, ensuring that if the i^{th} row exclusively contains out-of-scale values, the i^{th} column must also be filled with the same out-of-scale values. Naturally, the same principle is applied by initially considering the j^{th} column and subsequently the j^{th} row. This approach guarantees that if the i^{th} node is not the departure point of any connection, it will also not serve as the destination for any connection. Any attempt to reach such a node would be futile, as the journey cannot progress further. Similarly, if the j^{th} node is not the destination of any trip, it is illogical to initiate any additional trips from it. Of course, it is necessary to exclude the

user-defined starting and ending nodes from this pruning step.

It is crucial to emphasize that this pruning step, which comprehensively reviews the cost matrix by examining and modifying the values within, must be executed multiple times. There is a potential risk that sequentially reviewing the matrix from the first to the last row and column and replacing the values within may generate new rows and columns consisting solely of out-of-scale values. Consequently, new nodes may emerge from which no connections originate or to which no connections arrive. Nevertheless, after just a few reviews of the cost matrix (less than 10 given size of the matrix) it can be certain that the pruning step has been correctly completed.

As illustrated in Figure 3.29, this final pruning step resulted in the following number of connections for the respective journeys: 42 for Turin to Milan, 55 for Milan to Genoa, and 40 for Genoa to Turin.



(a) Turin to Milan - 42 **(b)** Milan to Genoa - 41 **(c)** Genoa to Turin - 51

Figure 3.29: Node net at *macro-research* with 6 steps of pruning

3.5.2 Pruning for *Micro-Research*

In a similar manner to the pruning techniques employed for *macro-research*, the development of these secondary pruning techniques were derived from the analysis of numerous simulations conducted without employing any *micro-research* pruning methods. Specifically, only pruning techniques for the *macro-research* were implemented to extract essential trends and streamline the construction process of the second cost matrix.

As mentioned, the focal point was shifted away from the *macro-research* phase, which aimed to identify the macro-geographic areas for charging stops, redirecting it towards the *micro-research* phase, where the necessary charging locations within these pre-determined clusters were already known. During this phase, the simulations conducted without *micro-research* pruning techniques revealed a prominent tendency: the charging stations deemed suitable for stopping consistently adhered to the sequence of clusters identified as the outcome of the Dijkstra's algorithm during the *macro-research*.

This observed trend affirms the significant benefits of simplifying the construction of the second cost matrix by incorporating an *out-of-scale* value for all cells/trips that hold no relevance in determining the optimal route. Notably, it is essential to underscore that the second cost matrix employed for the *micro-research* outcome is a square matrix encompassing all feasible connections between the user-defined starting and destination nodes, outlined in Section 3.1.1, as well as all the nodes representing charging stations located within the clusters identified at the conclusion of the *macro-research*, using the Dijkstra's algorithm.

For instance, if a particular journey, as dictated by the *macro-research*, necessitates two charging stops, the cost matrix will encompass numerous cells/trips that can be deemed irrelevant in advance neglecting the utilization of the external SUMO software, leading noteworthy improvements in processing time. Considered irrelevant in advance are cells/trips representing connections between user defined starting node and charging stations situated within the second cluster, as they fail to adhere to the prescribed sequence indicated by the *macro-research* output. Similarly, every cells/trips between nodes within the first or second cluster and the user defined starting node, exhibiting a reversed sequence compared to that specified by the *macro-research*, as well as cells/trips between charging stations within the same cluster that provide no substantive contribution to determining the optimal path are assigned with an *out-of-scale* value.

To summarize, *micro-research* pruning techniques expect to consider as irrelevant any cells/trips connecting the user-defined starting or destination node with nodes representing charging stations that do not conform to the ordered sequence of clusters identified as the output of the *macro-research*.

Chapter 4

Results

In this chapter, the goal is the actual implementation and analysis of the various additions proposed throughout this project. These additions encompass the introduction of a new vehicle model, described in Section 3.3, the utilization of a clustering algorithm as outlined in Section 3.4, and the incorporation of speed-up techniques elaborated in Section 3.5. Consequently, these implementations will be compared against the baseline minimum travel time path research software presented by [17]. The primary metrics employed to assess the effectiveness of our work are the total travel time and processing time required to execute the operations.

The focus has been on comparing the outcomes obtained with the original model [17], with those achieved through an intermediary solution employing the same approach of minimum travel time path research detailed in [17] together with the new vehicle model. Lastly, the comparison is concluded by presenting the results obtained through the utilization of the new vehicle model, as well as the novel minimum travel time path research system enabled by clustering and various speed-up techniques.

In order to test the enhancements made to the base program, a series of simulations were conducted to validate its performance across medium and long-distance trips. To establish meaningful comparisons, short-duration trips were intentionally excluded from the analysis. The reason behind this decision stems from the minimal number of charging stops required, which would make it challenging to discern discrepancies between the various vehicle models and path search systems.

To ensure a robust evaluation of the newly implemented features across diverse work scenarios, identical routes were employed while varying user-defined initial parameters. This approach facilitated the examination of the new minimum travel time path research system's sensitivity to those user-defined factors. Furthermore, we explored the impact of external ambient temperature and driving style on the system's performance. For comprehensive coverage, all simulations were systematically repeated with variations in the maximum allowable charging power, selecting either 22 or 50 kW, to verify the sensitivity of the enhancements to this parameter. The initial stage of the investigation involves comparing the performance of different minimum travel time path research systems using a maximum charging power of 22 kW, which represents the minimum power output provided by every charging station featured in the map illustrated in Figure 3.5.

Within this first phase of simulations, an examination was undertaken evaluating the performance of the route planning tools across six distinct routes. Specifically, the routes encompassed the connections between Turin and Milan, Milan and Cuneo, Piacenza and Turin, Cuneo and Parma, Turin and Brescia, and Milan and Genoa. With reference to Figure 3.5 it can be seen that these six routes not only have variations occur in total travel distance, but also in travel direction. Consequently, different roads, highways, regions and charging stations were visited and utilized along each route.
To thoroughly assess the impact of various user-defined parameters, numerous simulations were executed for each of the aforementioned routes, systematically altering the user-defined initial parameters. Initially, all simulations adopted a standard configuration, wherein SoH was set at 100%, initial SoC at 80%, reachable SoC threshold during every journey at 20%, SoC threshold at the destination at 20%, no passengers were considered and a normal driving style was employed. Subsequently, the initial configuration was systematically modified to investigate the sensitivity of the system to various factors. Adjustments included reducing the SoH of the battery, decreasing the initial SoC value, increasing the reachable SoC threshold during the journey, as well as the threshold at the destination. Moreover, the external ambient temperature was altered within a range of $\pm 10°C$, and the driving style parameters were varied accordingly.

To effectively present the simulation results and provide comprehensive insights, Table 4.1, 4.2 and 4.3 captures essential data for each program iteration, encompassing base, intermediate and final solution. Data includes: total travel time, number of stops required for battery charging, processing time, and final SoC value upon arrival. Notably, in the case of the final program, which integrates clustering and speed-up techniques, a division was made in the results of the travel and processing time. This division enabled a separated examination of the *macro* and *micro-research* outcomes, facilitating the evaluation of the positive impact of *micro-research* on the total travel time, along with its potential negative effect on the processing time.
Furthermore, certain routes within the tables have been marked in red to draw attention to those that do not fully adhere to the user-defined constraints in Section

3.1.1. The results obtained from these specific routes, along with their influence on the comparison, will be further examined in the subsequent section dedicated to presenting the results.

Table 4.1: Results relative to Base Logic and Base Vehicle Model - 22 kW

Trip	SoH - startSoC - minSoC - endSoC - Temp - numPass - Style	Travel Time (*sec*)	N° Stops	Process Time (*sec*)	Final SOC (%)
TORINO - MILANO	100 - 80 - 20 - 20 - 21 - 0 - Normal	13897	5	324.51	77.11
	90 - 80 - 20 - 20 - 21 - 0 - Normal	13893	5	266.44	76.53
	100 - 60 - 20 - 20 - 21 - 0 - Normal	15032	5	312.10	76.82
	100 - 80 - 20 - 40 - 21 - 0 - Normal	13897	5	321.82	77.11
	100 - 80 - 20 - 20 - 21 - 0 - Sport	20407	5	142.45	72.69
MILANO - CUNEO	100 - 80 - 20 - 20 - 21 - 0 - Normal	22883	7	452.73	75.50
	100 - 40- 20 - 20 - 21 - 0 - Normal	25615	7	401.70	75.38
	100 - 80 - 20 - 40 - 21 - 0 - Normal	23438	7	479.84	75.31
	100 - 80 - 20 - 20 - 14- 0 - Normal	23343	7	336.65	85.23
	100 - 80 - 20 - 20 - 21 - 3 - Eco	15168	2	85.23	28.96
PIACENZA - TORINO	100 - 80 - 20 - 20 - 21 - 0 - Normal	19595	7	488.92	86.81
	90 - 80 - 20 - 20 - 21 - 0 - Normal	20578	6	407.02	86.65
	100 - 80 - 20 - 40 - 21 - 0 - Normal	19595	7	583.02	86.81
	100 - 80 - 20 - 40 - 21 - 3 - Normal	21666	6	355.39	86.51
	100 - 80 - 20 - 20 - 21 - 0 - Sport	23926	6	236.75	0.00
CUNEO - PARMA	100 - 80 - 20 - 20 - 21 - 0 - Normal	23250	5	309.15	26.39
	100 - 60 - 20 - 20 - 28 - 0 - Normal	26218	6	210.55	40.60
	100 - 80 - 20 - 60 - 21 - 0 - Normal	23411	5	280.60	26.62
	100 - 80 - 20 - 20 - 21 - 3 - Eco	19276	2	81.71	55.90
TORINO - BRESCIA	100 - 80 - 20 - 20 - 21 - 0 - Normal	21850	7	470.06	63.65
	85 - 80 - 20 - 20 - 21 - 0 - Normal	20418	7	296.07	58.25
	100 - 80 - 20 - 60 - 21 - 0 - Normal	21542	6	492.07	66.27
	100 - 80 - 20 - 20 - 30 - 0 - Normal	22696	6	165.07	64.35
	100 - 80 - 20 - 20 - 21 - 0 - Sport	30258	7	255.69	42.88
MILANO - GENOVA	100 - 80 - 20 - 20 - 21 - 0 - Normal	14567	4	415.21	85.91
	90 - 80 - 20 - 20 - 21 - 0 - Normal	14247	5	438.15	83.99
	100 - 60 - 20 - 20 - 21 - 0 - Normal	15472	5	456.63	84.50
	100 - 80 - 20 - 40 - 21 - 0 - Normal	14567	4	401.90	85.91
	100 - 80 - 20 - 20 - 21 - 0 - Sport	20618	5	206.26	79.18

Table 4.2: Results relative to Base Logic and New Vehicle Model - 22 kW

Trip	SoH - startSoC - minSoC - endSoC - Temp - numPass - Style	Travel Time (*sec*)	N° Stops	Process Time (*sec*)	Final SOC (%)
TORINO - MILANO	100 - 80 - 20 - 20 - 21 - 0 - Normal	13208	3	205.72	87.76
	90 - 80 - 20 - 20 - 21 - 0 - Normal	14738	6	373.05	87.70
	100 - 60 - 20 - 20 - 21 - 0 - Normal	14969	4	267.31	88.18
	100 - 80 - 20 - 40 - 21 - 0 - Normal	13208	3	190.21	87.76
	100 - 80 - 20 - 20 - 21 - 0 - Sport	10710	2	58.27	3.20
MILANO - CUNEO	100 - 80 - 20 - 20 - 21 - 0 - Normal	21837	5	323.67	85.56
	100 - 40- 20 - 20 - 21 - 0 - Normal	20862	5	278.49	26.98
	100 - 80 - 20 - 40 - 21 - 0 - Normal	19476	4	270.06	47.96
	100 - 80 - 20 - 20 - 14- 0 - Normal	21055	4	185.58	56.83
	100 - 80 - 20 - 20 - 21 - 3 - Eco	16566	3	155.38	38.65
PIACENZA - TORINO	100 - 80 - 20 - 20 - 21 - 0 - Normal	14782	4	281.45	59.67
	90 - 80 - 20 - 20 - 21 - 0 - Normal	18379	4	268.23	52.52
	100 - 80 - 20 - 40 - 21 - 0 - Normal	14782	4	286.85	59.67
	100 - 80 - 20 - 40 - 21 - 3 - Normal	19244	4	241.5	53.70
	100 - 80 - 20 - 20 - 21 - 0 - Sport	13163	2	64.06	28.00
CUNEO - PARMA	100 - 80 - 20 - 20 - 21 - 0 - Normal	22679	5	302.10	34.34
	100 - 60 - 20 - 20 - 28 - 0 - Normal	24360	6	209.38	41.31
	100 - 80 - 20 - 60 - 21 - 0 - Normal	21850	4	263.58	49.28
	100 - 80 - 20 - 20 - 21 - 3 - Eco	20266	3	151.93	57.62
TORINO - BRESCIA	100 - 80 - 20 - 20 - 21 - 0 - Normal	19250	5	253.56	61.93
	85 - 80 - 20 - 20 - 21 - 0 - Normal	21117	7	281.1	62.02
	100 - 80 - 20 - 60 - 21 - 0 - Normal	19202	4	211.24	71.13
	100 - 80 - 20 - 20 - 30 - 0 - Normal	20801	5	126.32	60.36
	100 - 80 - 20 - 20 - 21 - 0 - Sport	21417	4	124.12	41.56
MILANO - GENOVA	100 - 80 - 20 - 20 - 21 - 0 - Normal	14035	4	402.31	87.07
	90 - 80 - 20 - 20 - 21 - 0 - Normal	13868	4	355.73	86.44
	100 - 60 - 20 - 20 - 21 - 0 - Normal	15049	4	386.65	86.98
	100 - 80 - 20 - 40 - 21 - 0 - Normal	14035	4	416.39	87.07
	100 - 80 - 20 - 20 - 21 - 0 - Sport	11439	2	101.35	14.17

Table 4.3: Results relative to New Logic and New Vehicle Model - 22 kW

Trip	SoH - startSoC - minSoC - endSoC - Temp - numPass - Style	Travel Time Macro (*sec*)	Travel Time Micro (*sec*)	N° Stops	Process Time Macro (*sec*)	Process Time Micro (*sec*)	Final SOC (%)
TORINO - MILANO	100 - 80 - 20 - 20 - 21 - 0 - Normal	9206	9206	2	22.41	28.92	33.01
	90 - 80 - 20 - 20 - 21 - 0 - Normal	9286	9286	2	44.01	26.43	1.48
	100 - 60 - 20 - 20 - 21 - 0 - Normal	9997	9858	2	22.41	26.92	34.99
	100 - 80 - 20 - 40 - 21 - 0 - Normal	9356	9284	2	48.61	25.28	1.41
	100 - 80 - 20 - 20 - 21 - 0 - Sport	9214	9072	2	22.35	25.29	33.44
MILANO - CUNEO	100 - 80 - 20 - 20 - 21 - 0 - Normal	13016	12932	2	24.59	83.51	6.68
	100 - 40- 20 - 20 - 21 - 0 - Normal	14760	14676	3	24.58	83.99	9.67
	100 - 80 - 20 - 40 - 21 - 0 - Normal	14316	14054	3	47.51	83.53	9.71
	100 - 80 - 20 - 20 - 14- 0 - Normal	14642	14366	3	43.71	91.83	9.08
	100 - 80 - 20 - 20 - 21 - 3 - Eco	9660	9660	1	21.69	85.03	3.46
PIACENZA - TORINO	100 - 80 - 20 - 20 - 21 - 0 - Normal	11792	11620	3	30.82	43.54	13.73
	90 - 80 - 20 - 20 - 21 - 0 - Normal	11706	11535	3	24.25	40.97	14.96
	100 - 80 - 20 - 40 - 21 - 0 - Normal	12150	12150	2	42.72	38.10	3.75
	100 - 80 - 20 - 40 - 21 - 3 - Normal	13146	13141	2	49.60	38.52	9.51
	100 - 80 - 20 - 20 - 21 - 0 - Sport	11992	11815	3	26.99	39.25	12.64
CUNEO - PARMA	100 - 80 - 20 - 20 - 21 - 0 - Normal	18471	18436	4	23.77	92.43	12.03
	100 - 60 - 20 - 20 - 28 - 0 - Normal	19637	19637	4	21.60	123.67	8.93
	100 - 80 - 20 - 60 - 21 - 0 - Normal	20378	20240	5	64.82	123.10	23.43
	100 - 80 - 20 - 20 - 21 - 3 - Eco	16497	16487	3	79.47	27.49	11.63
TORINO - BRESCIA	100 - 80 - 20 - 20 - 21 - 0 - Normal	15683	15679	3	30.12	87.44	14.72
	85 - 80 - 20 - 20 - 21 - 0 - Normal	15552	15552	3	21.35	104.04	16.89
	100 - 80 - 20 - 60 - 21 - 0 - Normal	16302	16302	3	66.94	115.35	10.18
	100 - 80 - 20 - 20 - 30 - 0 - Normal	16142	16142	3	27.06	90.93	13.57
	100 - 80 - 20 - 20 - 21 - 0 - Sport	15693	15693	3	30.04	108.98	18.33
MILANO - GENOVA	100 - 80 - 20 - 20 - 21 - 0 - Normal	7506	7506	1	24.05	44.41	0.73
	90 - 80 - 20 - 20 - 21 - 0 - Normal	8993	8983	1	36.70	50.93	2.89
	100 - 60 - 20 - 20 - 21 - 0 - Normal	8297	8297	1	24.50	48.51	0.70
	100 - 80 - 20 - 40 - 21 - 0 - Normal	9048	9038	1	42.04	45.46	1.70
	100 - 80 - 20 - 20 - 21 - 0 - Sport	8964	8964	1	44.84	37.69	2.00

In order to enhance the interpretation of simulation results, graphical representations have been employed. Figure 4.1 illustrates the outcomes concerning total travel time, while Figure 4.2 provides an overview of the processing times across different simulations obtained with a maximum charging power of 22 kW.

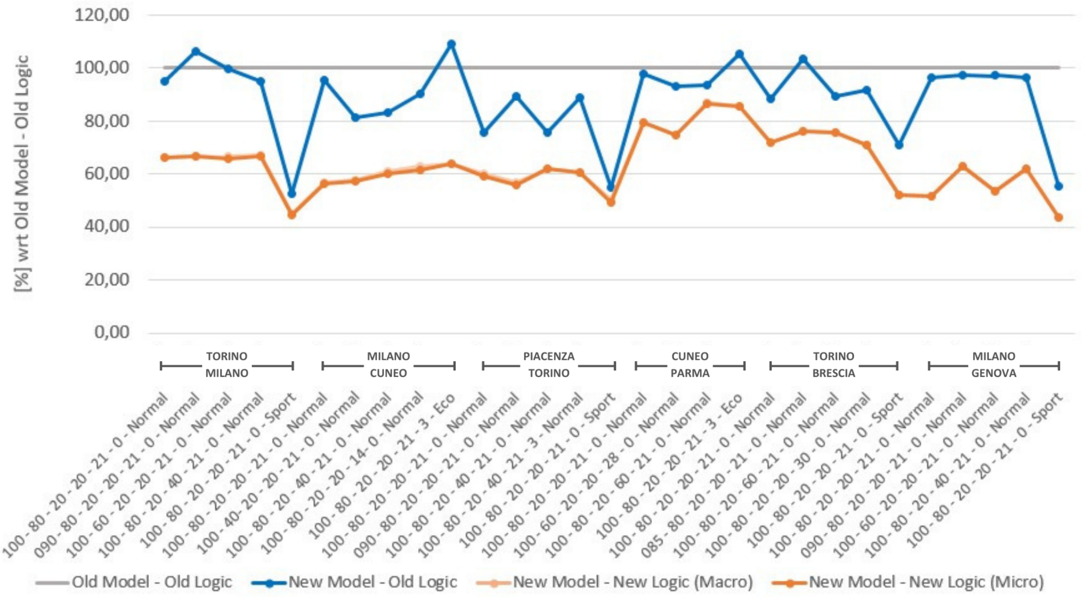


Figure 4.1: Travel Time Results - 22 kW

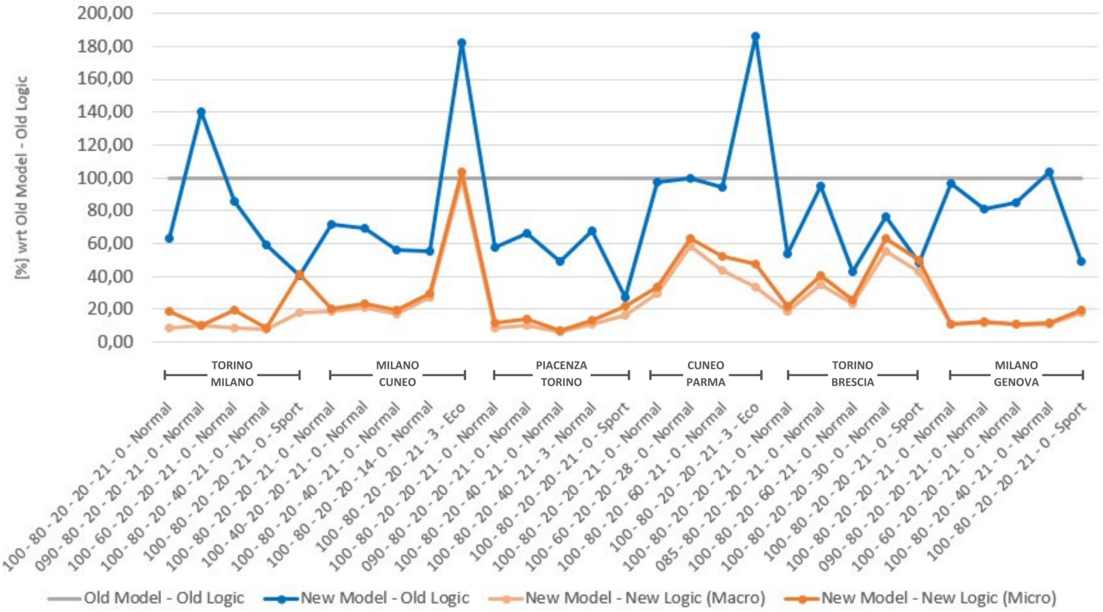


Figure 4.2: Processing Time Results - 22 kW

These two figures serve as crucial tools for evaluating the efficacy of the implemented modifications, presenting the simulation results as a percentage fraction in relation to the baseline program [17]. The values lower than unity, or 100%, indicate a reduced travel or processing times compared to the results obtained with the base program [17], thereby highlighting an improvement in program output. The utilization of percentage fractions as a representation has been favored over direct reporting as it allows for a more comprehensive analysis of the results and the identification of performance trends, despite variations in total travel time and processing time across all the different routes.

Within Figures 4.1 and 4.2, four distinct lines are featured. The grey line, consistently maintained at 100% horizontally, represents the results attained with the base program. The blue line corresponds to the results achieved through the intermediate program, which incorporates the minimum travel time path research program presented in the base model and the newly introduced vehicle model. The final two lines, characterized by light and dark orange respectively, denote the results derived from the final program. As already mentioned, this final program allows to distinguish the results between those obtained with *macro-research* only from those obtained with *micro-research*.

When examining the results presented in Figure 4.1, it becomes evident that the intermediate program, incorporating the minimum travel time path research system of the base program and the new vehicle model, consistently achieves a lower total travel time compared to the original model in the majority of tests. As demonstrated in Section 3.3.2, the new vehicle model successfully reduces energy consumption during each trip, resulting in fewer stops for battery charging and effectively compensating for the increased travel time associated with a non-constant battery power charging profile. In general, the total travel time calculated with the intermediate program is able to reach 80% of the one obtained with the original program. However, it is worth noting that specific combinations of user-defined parameters and routes can lead to a slightly higher total travel time compared to the original model.

Furthermore, the intermediate program demonstrates a notable reduction in processing time. By employing a vehicle model that consumes less energy, the intermediate program achieves processing times that can be up to 40-80% lower than those of the original model. This improvement in processing time is directly linked to the reduction in the number of stops and iterations required to check all potential charging stations between the starting and ending points. These findings highlight the positive impact of the intermediate program on both travel time and processing efficiency, confirming the effectiveness of the new vehicle model and its associated path search system. It is worth noticing that, in a similar manner to the travel time, there are combinations of user-defined parameters and routes, where even the processing time exceeds that of the base program.

Regarding the comparison between the intermediate and final programs, it is clear that the final program consistently achieves a lower total travel time compared to the intermediate program. Generally, the travel time calculated with the final program ranges from 50-80% of the travel time calculated with the original program. A considerable portion of the travel time reduction can be attributed to achieving a lower final SoC of the battery. The introduction of the new shortest path search program, incorporating the Dijkstra's algorithm alongside clustering and speed-up techniques, enables a more effective selection of charging stations, thereby optimizing the overall outcome. This approach efficiently utilizes the battery's energy, leading to fewer charging stops during the journey, in line with the constraints specified by the user.

Furthermore, an additional improvement is observed in the reduction of processing time from the intermediate to the final program. The integration of clustering and speed-up techniques has led to a significant decrease in the number of nodes/stops considered, consequently resulting in a reduction in processing time.

Through the analysis of the results obtained with the final program, it can be observed that the total travel time calculated using the *macro* or *micro-research* is practically equal, while the inclusion of the *micro-research* introduces a significant

increase in processing time. This observed trend is fully justified as the role of the *micro-research* is primarily to refine the travel time determined by the *macro-research*, which plays a more prominent role in identifying the optimal battery charging areas across the entire map. However, it is important to note that the *micro-research* can lead to notable negative impacts on the overall processing time. Based on the conducted simulations, it can be argued that the exclusive utilization of the *macro-research* approach is justified, focusing on the examination of a single charging station for each cluster. Although this approach may result in slightly less precise travel time calculations, it offers substantial benefits in terms of processing time reduction.
In order to conclude the analysis of the results obtained with a maximum charging power by the vehicle equal to 22 kW, it is crucial to highlight the existence of particular routes, as evident in Table 4.2 highlighted in red, where the intermediate program fails to respect the user-defined limits for the minimum SoC value upon arrival and during the journey. This aspect carries significant importance as it significantly influences the interpretation of the travel and processing time graphs, ultimately leading to inaccurate comparisons between the intermediate and final programs. Consequently, the results derived from the intermediate program can be considered slightly underestimated, as full compliance with the user-defined constraints would have inevitably increased both total travel and processing time.

Moving on to the analysis of the results presented from Table 4.4 to 4.6, which depict the total travel and processing time results obtained with a maximum charging power of 50 kW, we can draw similar conclusions as those derived from the study conducted with a lower charging power.

Table 4.4: Results relative to Base Logic and Base Vehicle Model - 50 kW

Trip	SoH - startSoC - minSoC - endSoC - Temp - numPass - Style	Travel Time (*sec*)	N° Stops	Process Time (*sec*)	Final SOC (%)
TORINO - MILANO	100 - 80 - 20 - 20 - 21 - 0 - Normal	9305	3	37.75	42.45
	090 - 80 - 20 - 20 - 21 - 0 - Normal	11843	9	144.26	86.45
	100 - 60 - 20 - 20 - 21 - 0 - Normal	11395	6	125.63	87.54
	100 - 80 - 20 - 40 - 21 - 0 - Normal	8837	3	55.25	41.06
	100 - 80 - 20 - 20 - 21 - 0 - Sport	13185	5	49.61	83.23
MILANO - CUNEO	100 - 80 - 20 - 20 - 21 - 0 - Normal	16216	6	106.26	74.54
	100 - 40- 20 - 20 - 21 - 0 - Normal	17569	6	111.29	74.85
	100 - 80 - 20 - 40 - 21 - 0 - Normal	16381	6	122.11	74.82
	100 - 80 - 20 - 20 - 14- 0 - Normal	17131	8	85.79	73.99
	100 - 80 - 20 - 20 - 21 - 3 - Eco	13479	2	26.26	44.28
CUNEO - PARMA	100 - 80 - 20 - 20 - 21 - 0 - Normal	16872	6	96.24	38.50
	100 - 60 - 20 - 20 - 28 - 0 - Normal	17894	6	62.76	40.60
	100 - 80 - 20 - 60 - 21 - 0 - Normal	18362	8	161.42	41.36
	100 - 80 - 20 - 20 - 21 - 3 - Eco	15247	2	24.52	19.79
TORINO - BRESCIA	100 - 80 - 20 - 20 - 21 - 0 - Normal	15116	7	107.07	63.77
	100 - 80 - 20 - 60 - 21 - 0 - Normal	14809	5	84.05	57.73
	100 - 80 - 20 - 20 - 30 - 0 - Normal	15421	6	60.18	64.55
ALESSANDRIA - BRESCIA	90 - 80 - 20 - 20 - 21 - 0 - Normal	15632	8	172.51	86.78
	100 - 80 - 20 - 40 - 21 - 0 - Normal	21651	9	306.80	86.73
	100 - 80 - 20 - 20 - 21 - 0 - Sport	20614	6	356.02	83.45
MILANO - GENOVA	100 - 80 - 20 - 20 - 21 - 0 - Normal	10535	4	112.37	83.56
	90 - 80 - 20 - 20 - 21 - 0 - Normal	10712	5	122.36	80.37
	100 - 60 - 20 - 20 - 21 - 0 - Normal	11411	5	138.04	81.23
	100 - 80 - 20 - 40 - 21 - 0 - Normal	10535	4	112.18	83.56
	100 - 80 - 20 - 20 - 21 - 0 - Sport	13988	6	78.20	74.83

Table 4.5: Results relative to Base Logic and New Vehicle Model - 50 *kW*

Trip	SoH - startSoC - minSoC - endSoC - Temp - numPass - Style	Travel Time (*sec*)	N° Stops	Process Time (*sec*)	Final SOC (%)
TORINO - MILANO	100 - 80 - 20 - 20 - 21 - 0 - Normal	8571	3	62.47	54.09
	090 - 80 - 20 - 20 - 21 - 0 - Normal	9839	4	77.17	61.21
	100 - 60 - 20 - 20 - 21 - 0 - Normal	9066	3	73.99	53.77
	100 - 80 - 20 - 40 - 21 - 0 - Normal	8571	3	70.85	54.09
	100 - 80 - 20 - 20 - 21 - 0 - Sport	9076	2	18.92	41.05
MILANO - CUNEO	100 - 80 - 20 - 20 - 21 - 0 - Normal	14320	4	85.81	48.03
	100 - 40- 20 - 20 - 21 - 0 - Normal	15947	5	122.94	26.80
	100 - 80 - 20 - 40 - 21 - 0 - Normal	14453	4	100.51	47.94
	100 - 80 - 20 - 20 - 14- 0 - Normal	15587	4	52.20	56.93
	100 - 80 - 20 - 20 - 21 - 3 - Eco	14357	3	72.86	42.52
CUNEO - PARMA	100 - 80 - 20 - 20 - 21 - 0 - Normal	16576	6	126.38	44.45
	100 - 60 - 20 - 20 - 28 - 0 - Normal	18206	8	98.26	43.70
	100 - 80 - 20 - 60 - 21 - 0 - Normal	18123	6	130.76	44.44
	100 - 80 - 20 - 20 - 21 - 3 - Eco	16351	3	37.94	58.00
TORINO - BRESCIA	100 - 80 - 20 - 20 - 21 - 0 - Normal	13465	4	68.81	71.12
	100 - 80 - 20 - 60 - 21 - 0 - Normal	13695	5	103.21	61.72
	100 - 80 - 20 - 20 - 30 - 0 - Normal	14705	5	40.42	60.22
ALESSANDRIA - BRESCIA	90 - 80 - 20 - 20 - 21 - 0 - Normal	15843	6	229.72	86.13
	100 - 80 - 20 - 40 - 21 - 0 - Normal	13545	4	91.44	60.09
	100 - 80 - 20 - 20 - 21 - 0 - Sport	12056	6	110.92	46.11
MILANO - GENOVA	100 - 80 - 20 - 20 - 21 - 0 - Normal	10432	4	129.95	84.74
	90 - 80 - 20 - 20 - 21 - 0 - Normal	10466	4	119.41	84.86
	100 - 60 - 20 - 20 - 21 - 0 - Normal	11202	5	144.86	82.66
	100 - 80 - 20 - 40 - 21 - 0 - Normal	10432	4	140.74	84.74
	100 - 80 - 20 - 20 - 21 - 0 - Sport	8676	2	23.53	14.58

Table 4.6: Results relative to New Logic and New Vehicle Model - 50 *kW*

Trip	SoH - startSoC - minSoC - endSoC - Temp - numPass - Style	Travel Time Macro (*sec*)	Travel Time Micro (*sec*)	N° Stops	Process Time Macro (*sec*)	Process Time Micro (*sec*)	Final SOC (%)
TORINO - MILANO	100 - 80 - 20 - 20 - 21 - 0 - Normal	7311	7311	2	39.96	17.78	1.58
	090 - 80 - 20 - 20 - 21 - 0 - Normal	7540	7430	3	34.40	10.08	3.49
	100 - 60 - 20 - 20 - 21 - 0 - Normal	7925	7816	3	39.96	11.68	4.07
	100 - 80 - 20 - 40 - 21 - 0 - Normal	7499	7499	2	41.22	10.25	1.85
	100 - 80 - 20 - 20 - 21 - 0 - Sport	7691	7600	3	38.58	10.04	3.08
MILANO - CUNEO	100 - 80 - 20 - 20 - 21 - 0 - Normal	11123	11019	3	26.79	41.09	9.28
	100 - 40- 20 - 20 - 21 - 0 - Normal	11863	11758	4	26.79	35.09	13.12
	100 - 80 - 20 - 40 - 21 - 0 - Normal	11667	11667	4	60.03	34.66	3.92
	100 - 80 - 20 - 20 - 14- 0 - Normal	11268	11161	3	23.28	34.08	8.6
	100 - 80 - 20 - 20 - 21 - 3 - Eco	10433	10433	3	23.53	36.48	5.27
CUNEO - PARMA	100 - 80 - 20 - 20 - 21 - 0 - Normal	15995	15539	4	27.09	64.91	13.57
	100 - 60 - 20 - 20 - 28 - 0 - Normal	17619	17245	6	65.66	47.27	25.42
	100 - 80 - 20 - 60 - 21 - 0 - Normal	15996	15996	4	62.62	47.22	6.47
	100 - 80 - 20 - 20 - 21 - 3 - Eco	15571	15571	3	62.65	47.26	4.52
TORINO - BRESCIA	100 - 80 - 20 - 20 - 21 - 0 - Normal	11847	11800	4	27.73	87.39	5.91
	100 - 80 - 20 - 60 - 21 - 0 - Normal	12841	12841	4	74.36	84.73	4.23
	100 - 80 - 20 - 20 - 30 - 0 - Normal	12299	12135	5	24.74	102.52	8.28
ALESSANDRIA - BRESCIA	90 - 80 - 20 - 20 - 21 - 0 - Normal	8369	8249	2	63.13	64.36	13.27
	100 - 80 - 20 - 40 - 21 - 0 - Normal	8400	8280	2	65.82	72.66	10.65
	100 - 80 - 20 - 20 - 21 - 0 - Sport	8360	8169	2	63.57	65.77	12.27
MILANO - GENOVA	100 - 80 - 20 - 20 - 21 - 0 - Normal	6325	6225	1	24.05	20.81	0.76
	90 - 80 - 20 - 20 - 21 - 0 - Normal	7436	7436	2	81.80	15.12	1.58
	100 - 60 - 20 - 20 - 21 - 0 - Normal	6673	6673	1	24.05	20.54	0.7
	100 - 80 - 20 - 40 - 21 - 0 - Normal	7160	7160	2	57.31	19.96	2.98
	100 - 80 - 20 - 20 - 21 - 0 - Sport	6380	6380	1	23.41	17.64	0.53

First, it is worth noting that, as previously observed for the lower charging power, the intermediate program predominantly achieves a lower total travel time compared to the base program. Generally, the total ravel time obtained with the intermediate program ranges between 80-100% of the result obtained with the base program, with only a few exceptions falling below 80% or slightly exceeding 100%. This once again attests to the effectiveness of employing a vehicle model that consumes less energy, resulting in a considerable reduction in the number of required charging stops. Such optimization compensates for the utilization of a non-constant charging power profile. Conversely, the processing time exhibits considerable variability, as demonstrated by the figures and the comparable number of routes that display both higher and lower processing times than the base model.

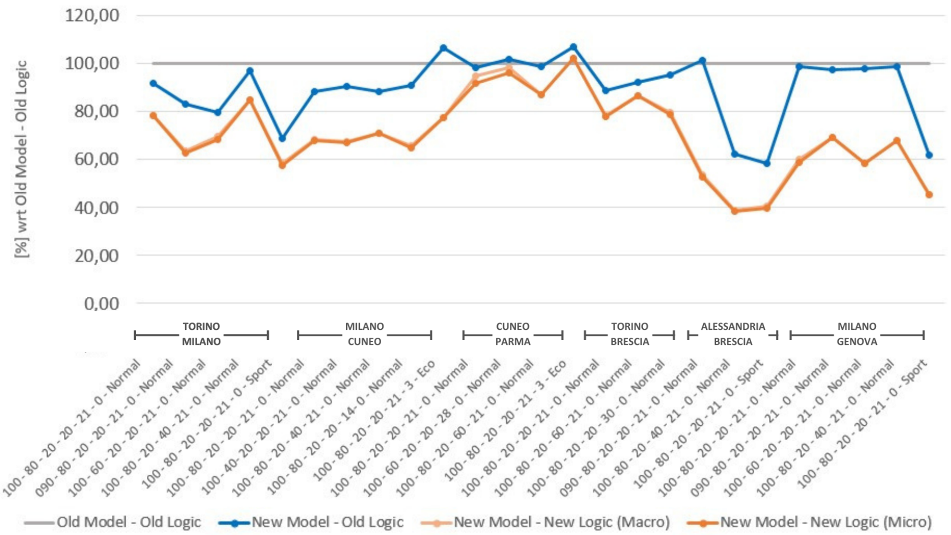


Figure 4.3: Travel Time Results - 50 *kW*

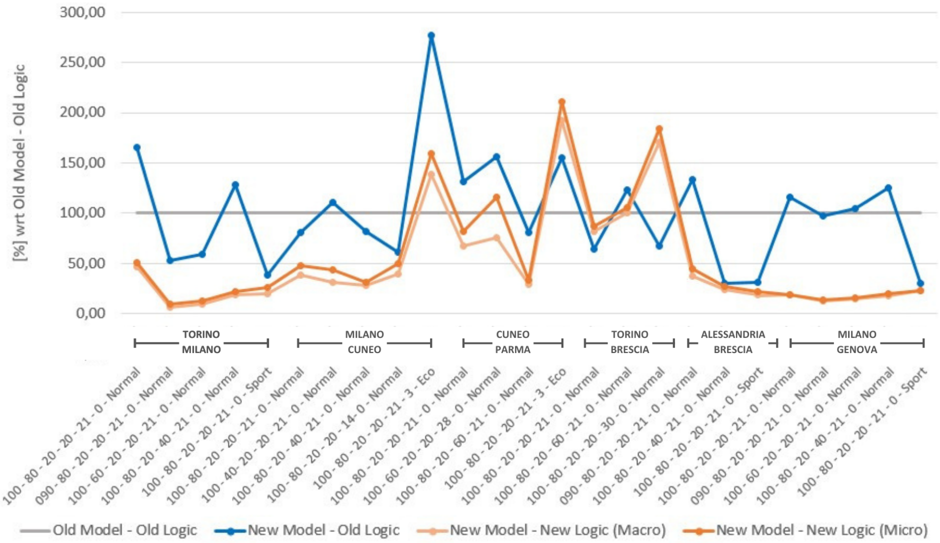


Figure 4.4: Processing Time Results - 50 *kW*

Directly addressing the results depicted in Figure 4.3, it is evident that the total travel time achieved with the final program consistently outperforms that of the intermediate program, which aligns with the earlier observations. However, the increased allowable charging power has led to a reduced number of charging stations to consider along the journey, thereby diminishing the effectiveness of the clusterization and speed-up techniques. Consequently, the percentage difference in travel time results between the intermediate and final programs is less pronounced compared to the scenario with the lower charging power. An in-depth analysis of these outcomes, coupled with the utilization of the same vehicle model and charging profile between the intermediate and final programs, once again underscores the efficacy of the various clustering strategies and speed-up techniques, complemented by the implementation of Dijkstra's algorithm. Collectively, these methodologies optimize the selection of charging stops along the shortest path, surpassing the logic of the base program.
Nevertheless, the superiority of the final program over the intermediate one in terms of processing time is not as evident. This discrepancy is particularly noticeable in those routes characterized by specific user-defined parameter combinations, wherein the final program may exhibit higher processing times than its intermediate counterpart. This trend becomes more evident during extensive journeys spanning the entire map. For instance, routes such as Cuneo to Parma or Turin to Brescia generally manifest higher processing times in the final program, while shorter routes such as the one from Turin to Milan or Turin to Genoa improved results in both travel and processing time when compared to the intermediate program.

Chapter 5

Conclusion

Throughout this Master's Thesis, the project has been presented in alignment with the actual evolution of the program, starting from the original program [17] and taking into consideration the enhancements it has introduced to the state of the art in EV route planning software.
The first step involved modifying the vehicle model and the battery recharging profile within the minimum travel time path research algorithm. The vehicle model, built upon the framework proposed by [25], facilitated more precise energy consumption predictions by more accurately modeling the electric motor's efficiency through a power-dependent engine map. Moreover, regenerative braking efficiency was incorporated via a regeneration factor linked to the vehicle's speed [25]. The comparative analysis between the original and the new vehicle models illuminated how the combined motor and battery efficiency consistently achieve a reduction in battery consumption of approximately 1.5 kWh per 100 km traveled.
In parallel to the advancement in the vehicle model, a strategic decision was made to introduce a novel battery charging model, formulated through in depth analysis as elaborated by [18]. This model proposed adopting a CP-CV charging protocol through a comparison with real world data, replacing the previous constant power recharging protocol.

The analysis of initial simulations, carried out with the aim of identifying the minimum travel time path between a user defined origin and destination, employing the new vehicle model together with the original research algorithm, shows significant challenges concerning final outcomes, processing times and recommended recharging stations. A thorough examination of the obtained results and the algorithm utilized in the original project underscored the necessity of improvements.
These enhancements were not only confined to the vehicle model and battery charging process, but extended to the algorithm itself employed for the minimum travel time path research. Consequently, the decision was made to comprehensively

rewrite the research algorithm, leveraging the architecture of Dijkstra's algorithm. According to this methodology, each charging station was conceptualized as a node within a network. The cost associated with each link between nodes represented the shortest travel time between the starting and ending stations, incorporating the battery charging time up to the level specified by the user.

It is crucial to emphasize that this approach deviates from the conventional minimum travel time path research programs based on the Dijkstra's algorithm. As elucidated in Section 2.3, the usual procedure entails constructing a network of nodes, designating each intersection on the map as a node, with links representing road segments between intersections. Each link is assigned a cost, representing travel time. The Dijkstra's algorithm is applied to this network of nodes to identify the less expensive path from a starting node to an arbitrary destination node. Starting from this base version, a specific variant tailored for electric vehicles was developed [38], which also accounts for energy consumption within the cost function of each link. This ensures that the optimal path does not consume more energy than initially available in the battery.
The decision to employ the Dijkstra's algorithm in an unconventional manner stemmed from two considerations. First, during the script of the original project [17], the choice was made to utilize external software [43] to effortlessly obtain the shortest path from an initial location to an arbitrary destination. This bypassed the need to define a cost matrix, with simply provide the software with a map containing the costs of each road segment. However, this software did not incorporate energy consumption as a factor in the cost function of each segment. Consequently, it might output a path that was not feasible in terms of energy consumption, potentially leading to a situation where, at the end of the journey, more energy was consumed than initially stored in the battery. Therefore, it became imperative to verify the feasibility of the journey from the perspective of energy consumption, accounting for the battery's initial conditions. During the construction of the node network between the various charging stations, if the path suggested by the external software was found to be unfeasible in terms of energy consumption, that segment was excluded from the network.
Conducting a posteriori check without incorporating energy consumption in the cost function of each road segment results in a loss of mathematical precision in the final solution. In simpler terms, it yields paths with longer travel times than the minimum possible if energy consumption were included, as detailed earlier. Nevertheless, the results presented in [18] demonstrate that the paths obtained by including energy consumption in the cost function compared to those obtained by applying a posteriori checks exhibit minimal disparities, justifying the unconventional construction of the node network and the subsequent use of the Dijkstra algorithm.

As delineated earlier in Section 3.5, the substantial number of connections within the Dijkstra's network, when applied to each charging station, in conjunction with the average time required to calculate the cost of each network link and the requisite updates of these costs for every travel request to accommodate specific user-provided data, necessitated targeted strategies to expedite the network construction phase. An additional effort was undertaken to address this challenge. The objective was further streamlining the network's complexity by formulating an algorithm capable of discarding links representing unfeasible travel scenarios, those inconsequential to the desired path, or those typically excluded due to inherent constraints that prolonged the total travel time.
Taking inspiration from the cognitive processes involved in human route planning, the approach was to deem journeys reliable only when their intermediate starting and ending positions fell within a defined-width imaginary corridor. This corridor was designed to originate from and terminate at the user-specified starting and ending point. Nonetheless, it is imperative to acknowledge a hidden limitation. The effectiveness of these techniques is most pronounced when dealing with routes between relatively closed positions in relation to the overall dimensions of the map. In scenarios where the distance between the starting and ending positions is closed to the maximum defined by the map, the spatial layout of the infrastructure, encompassing highways and their associated recharging stations, finishes to exhibit linear characteristics. In such cases, two important risks appears. Firstly, there exists the possibility that segments of the route may extend beyond the boundaries of the imaginary corridor created by the speed-up techniques, thereby elevating the risk of not accounting plausible charging stations. To mitigate this danger, an adaptation involves increasing the corridor's width in accordance with the spatial distance between the starting and ending positions, thereby guaranteeing the inclusion of all feasible charging stations. Nonetheless, this adaptation presents a secondary risk: the potential inclusion of an excessive number of charging stations within the corridor. This, in turn, would substantially increase processing time, thereby nullifying the benefits derived from the truncation done by the speed-up techniques.
This double problem is a well-known risk associated with the use of a straight corridor that inadequately fits the intricate configuration of the infrastructure depicted on the map. A plausible solution to this issue would have necessitated the construction of a corridor with a variable form, tailored to the principal infrastructure features situated between the initial starting and concluding positions. However, such solution would have involved a considerable investment of labour time, so it has been replaced in favor of a solution adopting a straight corridor with variable width based on the distance between starting and ending position.

In conclusion, the integration of speed-up techniques through the establishment of a straight corridor with variable width, built upon the distance spanned between the starting and ending positions, constitutes an indispensable element that facilitated the integration of the Dijkstra's algorithm into the project, all while maintaining acceptable processing times. It is imperative to underline that all the optimization strategies outlined have notably enhanced the overall system performance. Specifically, the duration of travel and data processing times have been significantly refined. As demonstrated in Section 4, the implementation of algorithms built on the structure of Dijkstra's node network for optimal route finding consistently resulted in diminished travel times. This achievement stems from the ability to discern the most suitable charging stations through the construction of a node network, coupled with the capability to traverse it with utmost freedom. Conversely, regarding processing time, a more faded dynamic emerges. Broadly speaking, it can be affirmed that as the maximum charging power permissible by the vehicle diminishes, the number of charging stations capable of providing that level of energy increases. In these scenarios, the new search logic proves more advantageous in terms of processing time than what was achievable through the original program. This scenario represents the optimal context where the clusterization and truncation executed through the speed-up techniques manifest particular effectiveness in optimizing the process. Nevertheless, with the augmentation of the maximum permissible charging power of the vehicle, a reduction in the number of stations capable of offering that power is observed. Consequently, the number of recharging stations normally taken into account during the research for the travel path diminishes. However, this situation is translated into a decreased positive impact of clusterization and truncation within the decision making process pertaining to processing time.

Within the purview of this Master's Thesis, the innovations introduced in this project encompassing the vehicle model, battery recharging process and the algorithm for selecting recharging stations to minimize the total travel time, have substantially contributed to the overall refinement of the original program. Specifically, the decision to incorporate clusterization of charging stations predicated on their geographical density, coupled with the integration of a sequence of speed-up strategies for truncating less relevant branches within the node network concerning the user's desired path, unlocks promising horizons within the realm of optimal route planning. This progression can be further extended into the use of artificial intelligence given that , as introduced in Section 3.4, these strategies were formulated with analogies to the current cognitive approach to route planning. However, the integration of computational approaches amplifies these capabilities, substantially enhancing computational performance and, consequently, markedly reducing processing times.

Bibliography

[1] Christian Brand and Alistair Hunt. *The Health costs of air pollution from cars and vans.* June 6, 2018. DOI: 10.13140/RG.2.2.25409.86886 (cit. on p. 1).

[2] *Cars, Trucks, Buses and Air Pollution | Union of Concerned Scientists.* URL: https://www.ucsusa.org/resources/cars-trucks-buses-and-air-pollution (visited on 08/31/2023) (cit. on p. 1).

[3] Sandra Wappelhorst. «The end of the road? An overview of combustion engine car phase-out announcements across Europe». In: (May 11, 2020). Publisher: ICCT: International Council on Clean Transportation. URL: https://policycommons.net/artifacts/3801824/the-end-of-the-road-an-overview-of-combustion-engine-car-phase-out-announcements-across-europe/4607651/ (visited on 05/23/2023) (cit. on p. 1).

[4] *Arab oil embargo | History, Cause, Impact, & Definition | Britannica.* URL: https://www.britannica.com/event/Arab-oil-embargo (visited on 08/31/2023) (cit. on p. 2).

[5] Mike [D-WA-4 Rep. McCormack. *H.R.8800 - 94th Congress (1975-1976): Electric Vehicle Research, Development, and Demonstration Act.* Archive Location: 1975-07-22. Sept. 17, 1976. URL: https://www.congress.gov/bill/94th-congress/house-bill/8800 (visited on 08/31/2023) (cit. on p. 2).

[6] *Oct. 1973: General Motors prototype urban electric car.* DOCUMERICA. URL: https://www.documerica.org/documerica-photographs/oct-1973-general-motors-prototype-urban-electric-car (visited on 08/31/2023) (cit. on p. 2).

[7] Ronnie Schreiber. *Decades before GM's EV1, these postal Electruck Jeeps delivered the goods.* Hagerty Media. Nov. 15, 2019. URL: https://www.hagerty.com/media/automotive-history/decades-before-gm-ev1-these-postal-electruck-jeeps-delivered-the-goods/ (visited on 08/31/2023) (cit. on p. 2).

[8] *Toyota Prius*. In: *Wikipedia*. Page Version ID: 1172842381. Aug. 29, 2023. URL: https://en.wikipedia.org/w/index.php?title=Toyota_Prius&oldid=1172842381 (visited on 08/31/2023) (cit. on p. 2).

[9] *Tesla Roadster (first generation)*. In: *Wikipedia*. Page Version ID: 1171162991. Aug. 19, 2023. URL: https://en.wikipedia.org/w/index.php?title=Tesla_Roadster_(first_generation)&oldid=1171162991 (visited on 08/31/2023) (cit. on p. 2).

[10] Jonn Axsen, Kenneth S. Kurani, and Andrew Burke. «Are batteries ready for plug-in hybrid buyers?» In: *Transport Policy* 17.3 (May 1, 2010), pp. 173–182. ISSN: 0967-070X. DOI: 10.1016/j.tranpol.2010.01.004. URL: https://www.sciencedirect.com/science/article/pii/S0967070X10000156 (visited on 06/02/2023) (cit. on pp. 2, 3).

[11] Ona Egbue and Suzanna Long. «Barriers to widespread adoption of electric vehicles: An analysis of consumer attitudes and perceptions». In: *Energy Policy*. Special Section: Frontiers of Sustainability 48 (Sept. 1, 2012), pp. 717–729. ISSN: 0301-4215. DOI: 10.1016/j.enpol.2012.06.009. URL: https://www.sciencedirect.com/science/article/pii/S0301421512005162 (visited on 06/02/2023) (cit. on pp. 2, 3).

[12] *What's The Real World Highway Range Of Today's Electric Cars? We Test To Find Out*. InsideEVs. URL: https://insideevs.com/reviews/443791/ev-range-test-results/ (visited on 06/02/2023) (cit. on p. 2).

[13] *2020–2023 global chip shortage*. In: *Wikipedia*. Page Version ID: 1171699999. Aug. 22, 2023. URL: https://en.wikipedia.org/w/index.php?title=2020%E2%80%932023_global_chip_shortage&oldid=1171699999 (visited on 09/01/2023) (cit. on p. 2).

[14] Arno Kwade, Wolfgang Haselrieder, Ruben Leithoff, Armin Modlinger, Franz Dietrich, and Klaus Droeder. «Current status and challenges for automotive battery production technologies». In: *Nature Energy* 3.4 (Apr. 2018). Number: 4 Publisher: Nature Publishing Group, pp. 290–300. ISSN: 2058-7546. DOI: 10.1038/s41560-018-0130-3. URL: https://www.nature.com/articles/s41560-018-0130-3 (visited on 06/02/2023) (cit. on p. 3).

[15] *Electric-car battery costs: Tesla $190 per kwh for pack, GM $145 for cells*. URL: https://www.greencarreports.com/news/1103667_electric-car-battery-costs-tesla-190-per-kwh-for-pack-gm-145-for-cells (visited on 06/02/2023) (cit. on p. 3).

[16] *Total Cost of Ownership (TCO) for Electric Vehicles (EV) vs Internal Combustion Engine Vehicles (ICE)*. URL: https://nickelinstitute.org/ (visited on 09/01/2023) (cit. on p. 3).

[17] Alberto Ponso, Angelo Bonfitto, and Giovanni Belingardi. «Route Planning for Electric Vehicles Including Driving Style, HVAC, Payload and Battery Health». In: *Energies* 16.12 (Jan. 2023). Number: 12 Publisher: Multidisciplinary Digital Publishing Institute, p. 4627. ISSN: 1996-1073. DOI: 10.3390/en16124627. URL: https://www.mdpi.com/1996-1073/16/12/4627 (visited on 06/13/2023) (cit. on pp. 3, 5, 10, 18, 23, 26, 29, 33, 34, 54, 58, 64, 65).

[18] Sven Schoenberg and Falko Dressler. «Planning Ahead for EV: Total Travel Time Optimization for Electric Vehicles». In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. 2019 IEEE Intelligent Transportation Systems Conference (ITSC). Oct. 2019, pp. 3068–3075. DOI: 10.1109/ITSC.2019.8917335 (cit. on pp. 5, 8, 9, 14, 28, 33, 64, 65).

[19] *Electric vehicle route planning using real-world charging infrastructure in Germany - ScienceDirect.* URL: https://www.sciencedirect.com/science/article/pii/S2590116821000412?casa_token=dIZGtJkjZTIAAAAA:70qM0xGCfUxbMbJqXqUeqQQ_pxM7oeMYGreBe4hl4pageM3XcECUZLAPRMW7GVCic9LNEBh6NQ (visited on 09/01/2023) (cit. on p. 5).

[20] Giovanni De Nunzio and Laurent Thibault. «Energy-optimal driving range prediction for electric vehicles». In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. 2017 IEEE Intelligent Vehicles Symposium (IV). June 2017, pp. 1608–1613. DOI: 10.1109/IVS.2017.7995939 (cit. on p. 5).

[21] *S2RC: A multi-objective route planning and charging slot reservation approach for electric vehicles considering state of traffic and charging station - ScienceDirect.* URL: https://www.sciencedirect.com/science/article/pii/S1319157822000738 (visited on 09/01/2023) (cit. on p. 5).

[22] William J. Smith. «Can EV (electric vehicles) address Ireland's CO2 emissions from transport?» In: *Energy*. The 3rd International Conference on Sustainable Energy and Environmental Protection, SEEP 2009 35.12 (Dec. 1, 2010), pp. 4514–4521. ISSN: 0360-5442. DOI: 10.1016/j.energy.2010.07.029. URL: https://www.sciencedirect.com/science/article/pii/S0360544210004044 (visited on 05/22/2023) (cit. on p. 6).

[23] Shichun Yang, Cheng Deng, Tieqiao Tang, and Yongsheng Qian. «Electric vehicle's energy consumption of car-following models». In: *Nonlinear Dynamics* 71.1 (Jan. 1, 2013), pp. 323–329. ISSN: 1573-269X. DOI: 10.1007/s11071-012-0663-0. URL: https://doi.org/10.1007/s11071-012-0663-0 (visited on 05/22/2023) (cit. on p. 6).

[24] Chiara Fiori, Kyoungho Ahn, and Hesham A. Rakha. «Power-based electric vehicle energy consumption model: Model development and validation». In: *Applied Energy* 168 (Apr. 15, 2016), pp. 257–268. ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2016.01.097. URL: https://www.sciencedirect.

com/science/article/pii/S030626191630085X (visited on 05/22/2023) (cit. on p. 6).

[25] Konstantinos N. Genikomsakis and Georgios Mitrentsis. «A computationally efficient simulation model for estimating energy consumption of electric vehicles in the context of route planning applications». In: *Transportation Research Part D: Transport and Environment* 50 (Jan. 1, 2017), pp. 98–118. ISSN: 1361-9209. DOI: 10.1016/j.trd.2016.10.014. URL: https://www.sciencedirect.com/science/article/pii/S1361920915302881 (visited on 05/22/2023) (cit. on pp. 6, 7, 26, 28–32, 64).

[26] James Larminie and John Lowry. *Electric Vehicle Technology Explained*. Google-Books-ID: FwXcCmT1OQUC. John Wiley & Sons, Sept. 17, 2012. 342 pp. ISBN: 978-1-119-94273-3 (cit. on pp. 6, 31).

[27] Valery Vodovozov, Zoja Raud, Tonu Lehtla, Anton Rassolkin, and Nikolai Lillo. «Comparative analysis of electric drives met for vehicle propulsion». In: *2014 Ninth International Conference on Ecological Vehicles and Renewable Energies (EVER)*. 2014 Ninth International Conference on Ecological Vehicles and Renewable Energies (EVER). Mar. 2014, pp. 1–8. DOI: 10.1109/EVER.2014.6844125 (cit. on p. 6).

[28] «Commission Regulation (EC) No 640/2009 of 22 July 2009 implementing Directive 2005/32/EC of the European Parliament and of the Council with regard to ecodesign requirements for electric motorsText with EEA relevance». In: (July 22, 2009) (cit. on pp. 7, 30).

[29] Mahnoosh Alizadeh, Hoi-To Wai, Anna Scaglione, Andrea Goldsmith, Yue Yue Fan, and Tara Javidi. «Optimized path planning for electric vehicle routing and charging». In: *2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. 2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton). Sept. 2014, pp. 25–32. DOI: 10.1109/ALLERTON.2014.7028431 (cit. on p. 7).

[30] Kong Soon Ng, Chin-Sien Moo, Yi-Ping Chen, and Yao-Ching Hsieh. «Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries». In: *Applied Energy* 86.9 (Sept. 1, 2009), pp. 1506–1511. ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2008.11.021. URL: https://www.sciencedirect.com/science/article/pii/S0306261908003061 (visited on 05/22/2023) (cit. on p. 7).

[31] Tobias Zündorf. «Electric Vehicle Routing with Realistic Recharging Model». In: (Nov. 24, 2014) (cit. on p. 8).

[32] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. «A Formal Basis for the Heuristic Determination of Minimum Cost Paths». In: *IEEE Transactions on Systems Science and Cybernetics* 4.2 (July 1968). Conference Name: IEEE Transactions on Systems Science and Cybernetics, pp. 100–107. ISSN: 2168-2887. DOI: 10.1109/TSSC.1968.300136 (cit. on p. 10).

[33] Ali Haghani and Soojung Jung. «A dynamic vehicle routing problem with time-dependent travel times». In: *Computers & Operations Research* 32.11 (Nov. 1, 2005), pp. 2959–2986. ISSN: 0305-0548. DOI: 10.1016/j.cor.2004.04.013. URL: https://www.sciencedirect.com/science/article/pii/S0305054804000887 (visited on 09/02/2023) (cit. on p. 10).

[34] E. W. Dijkstra. «A note on two problems in connexion with graphs». In: *Numerische Mathematik* 1.1 (Dec. 1, 1959), pp. 269–271. ISSN: 0945-3245. DOI: 10.1007/BF01386390. URL: https://doi.org/10.1007/BF01386390 (visited on 05/22/2023) (cit. on p. 10).

[35] Samah W. G. AbuSalim, Rosziati Ibrahim, Mohd Zainuri Saringat, Sapiee Jamel, and Jahari Abdul Wahab. «Comparative Analysis between Dijkstra and Bellman-Ford Algorithms in Shortest Path Optimization». In: *IOP Conference Series: Materials Science and Engineering* 917.1 (Sept. 2020). Publisher: IOP Publishing, p. 012077. ISSN: 1757-899X. DOI: 10.1088/1757-899X/917/1/012077. URL: https://dx.doi.org/10.1088/1757-899X/917/1/012077 (visited on 06/02/2023) (cit. on pp. 10, 12, 13).

[36] Donald B. Johnson. «Efficient Algorithms for Shortest Paths in Sparse Networks». In: *Journal of the ACM* 24.1 (Jan. 1977), pp. 1–13. ISSN: 0004-5411, 1557-735X. DOI: 10.1145/321992.321993. URL: https://dl.acm.org/doi/10.1145/321992.321993 (visited on 05/22/2023) (cit. on pp. 10, 12).

[37] Robert Geisberger, Peter Sanders, Dominik Schultes, and Daniel Delling. «Contraction Hierarchies: Faster and Simpler Hierarchical Routing in Road Networks». In: *Experimental Algorithms*. Ed. by Catherine C. McGeoch. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2008, pp. 319–333. ISBN: 978-3-540-68552-4. DOI: 10.1007/978-3-540-68552-4_24 (cit. on pp. 10, 12).

[38] Andreas Artmeier, Julian Haselmayr, Martin Leucker, and Martin Sachenbacher. «The optimal routing problem in the context of battery-powered electric vehicles». In: (Apr. 6, 2010) (cit. on pp. 11, 65).

[39] Jochen Eisner, Stefan Funke, and Sabine Storandt. «Optimal Route Planning for Electric Vehicles in Large Networks». In: *Proceedings of the AAAI Conference on Artificial Intelligence* 25.1 (Aug. 4, 2011). Number: 1, pp. 1108–1113. ISSN: 2374-3468. DOI: 10.1609/aaai.v25i1.7991. URL: https://ojs.aaai.

org/index.php/AAAI/article/view/7991 (visited on 05/22/2023) (cit. on pp. 11, 13).

[40] Sabine Storandt. «Route Planning for Bicycles — Exact Constrained Shortest Paths Made Practical via Contraction Hierarchy». In: *Proceedings of the International Conference on Automated Planning and Scheduling* 22 (May 14, 2012), pp. 234–242. ISSN: 2334-0843. DOI: 10.1609/icaps.v22i1.13495. URL: https://ojs.aaai.org/index.php/ICAPS/article/view/13495 (visited on 05/22/2023) (cit. on p. 13).

[41] Julian Dibbelt, Thomas Pajor, and Dorothea Wagner. «User-Constrained Multimodal Route Planning». In: *ACM Journal of Experimental Algorithmics* 19 (Feb. 3, 2015), pp. 1–19. ISSN: 1084-6654, 1084-6654. DOI: 10.1145/2699886. URL: https://dl.acm.org/doi/10.1145/2699886 (visited on 05/22/2023) (cit. on p. 13).

[42] Xin Xia, Zonglin Meng, Xu Han, Hanzhao Li, Takahiro Tsukiji, Runsheng Xu, Zhaoliang Zheng, and Jiaqi Ma. «An automated driving systems data acquisition and analytics platform». In: *Transportation Research Part C: Emerging Technologies* 151 (June 1, 2023), p. 104120. ISSN: 0968-090X. DOI: 10.1016/j.trc.2023.104120. URL: https://www.sciencedirect.com/science/article/pii/S0968090X23001092 (visited on 09/21/2023) (cit. on p. 17).

[43] *Eclipse SUMO - Simulation of Urban MObility.* Eclipse SUMO - Simulation of Urban MObility. URL: https://www.eclipse.org/sumo/ (visited on 06/11/2023) (cit. on pp. 17, 23, 65).

[44] *TraCI4Matlab.* June 11, 2023. URL: https://www.mathworks.com/matlabcentral/fileexchange/44805-traci4matlab (visited on 06/11/2023) (cit. on pp. 17, 23).

[45] Andrés F. Acosta, Jorge E. Espinosa, and Jairo Espinosa. «TraCI4Matlab: Enabling the Integration of the SUMO Road Traffic Simulator and Matlab® Through a Software Re-engineering Process». In: *Modeling Mobility with Open Data.* Ed. by Michael Behrisch and Melanie Weber. Lecture Notes in Mobility. Cham: Springer International Publishing, 2015, pp. 155–170. ISBN: 978-3-319-15024-6. DOI: 10.1007/978-3-319-15024-6_9 (cit. on p. 23).

[46] Andres Acosta. *TraCI4Matlab.* original-date: 2015-06-30T15:06:58Z. July 4, 2022. URL: https://github.com/pipeacosta/traci4matlab (visited on 05/25/2023) (cit. on p. 23).

[47] *OpenStreetMap.* OpenStreetMap. URL: https://www.openstreetmap.org/ (visited on 05/25/2023) (cit. on p. 24).

[48] Theresia Perger and Hans Auer. «Energy efficient route planning for electric vehicles with special consideration of the topography and battery lifetime». In: *Energy Efficiency* 13.8 (Dec. 1, 2020), pp. 1705–1726. ISSN: 1570-6478. DOI: 10.1007/s12053-020-09900-5. URL: https://doi.org/10.1007/s12053-020-09900-5 (visited on 05/25/2023) (cit. on p. 24).

[49] Adam Thor Thorgeirsson, Stefan Scheubner, Sebastian Fünfgeld, and Frank Gauterin. «An Investigation Into Key Influence Factors for the Everyday Usability of Electric Vehicles». In: *IEEE Open Journal of Vehicular Technology* 1 (2020). Conference Name: IEEE Open Journal of Vehicular Technology, pp. 348–361. ISSN: 2644-1330. DOI: 10.1109/OJVT.2020.3031699 (cit. on p. 24).

[50] Mehrdad Ehsani, Yimin Gao, Stefano Longo, and Kambiz Ebrahimi. *Modern Electric, Hybrid Electric, and Fuel Cell Vehicles*. Google-Books-ID: ImhQD-wAAQBAJ. CRC Press, Feb. 2, 2018. 573 pp. ISBN: 978-0-429-99824-9 (cit. on p. 32).

[51] *DBSCAN*. In: *Wikipedia*. Page Version ID: 1156762207. May 24, 2023. URL: https://en.wikipedia.org/w/index.php?title=DBSCAN&oldid=1156762207 (visited on 05/26/2023) (cit. on p. 39).

[52] *DBSCAN*. original-date: 2020-02-07T13:43:19Z. Aug. 1, 2023. URL: https://github.com/NSHipster/DBSCAN (visited on 09/21/2023) (cit. on p. 39).