Fleet Management and Energy Management of Shared Autonomous Electric Vehicles

Elena Golchin

A Thesis

in

The Department

of

Mechanical, Industrial, and Aerospace Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science (Industrial Engineering) at

Concordia University

Montreal, Quebec, Canada

March 2021

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: Elena Golchin

Entitled: Fleet Management and Energy Management of Shared Autonomous Electric Vehicles

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Industrial Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

____Chair
Dr. Onur Kuzgunkaya

____Examiner
Dr. Youmin Zhang

____Examiner
Dr. Amir Aghdam

____ Thesis Supervisor(s)
Dr. Anjali Awasthi

Approved by ____
Dr. Martin Pugh Chair of Department or Graduate Program Director

____

Dr. Mourad Debbai Dean, Gina Cody School of Engineering and Computer Science

Date 30/3/20

ABSTRACT

Fleet Management and Energy Management of Shared Autonomous Electric Vehicles

Elena Golchin

Nowadays, many cities are supplied with one or several fleets of shared vehicles as an alternative method for daily transportation. These fleets could be improved in many aspects if they employ autonomous or electric autonomous vehicles. Zero greenhouse gas (GHG) emission, fewer personnel requirements, automatic self-charge evaluation and process, and smooth driving are some of these improvements. This essay models a fleet comprised of shared electric autonomous vehicles (SEAV), giving one-way transportation service for short-distance travels. The case study of the area around Montreal Olympic Park is suggested to test the model's validity. In this regard, five stations are introduced on the streets surrounding the park. Different combinations of the fleet parameters are considered to generate 105 various scenarios. A multi-agent model is developed in the NetLog toolkit to simulate such a fleet under each scenario. The results show that a fleet of 20 vehicles of a 50km range with 30 parking spots can meet the demand under all considered customer arrivals frequencies with 90% or above performance rates. Another finding from the results suggests that the impact of different ranges on the fleet's performance compared to the fleet size and demand probability is almost negligible. The last part investigates the number of parking spots on fleet performance. The results suggest that the fewer the number of parking spots, the fewer rejected customers but at the same time, the higher empty travels.

Keywords: Autonomous electric vehicle, shared transportation fleet, multi-agent simulation, NetLogo

Acknowledgment

Through this thesis's writing, I had the privilege to receive help and support from many individuals. Herby, I would like to show my gratitude towards those who made this thesis's completion come true.

First, I would like to sincerely thank my supervisor, Dr. Anjali Awasthi, who gave me the opportunity to join her research team. Ever since, she has guided me through this journey with her wisdom, right-to-the-point comments, intelligence, and creativity.

I would also like to thank Gina Cody school's admission office and faculty for giving me the chance to continue my studies at Concordia University.

I cannot thank my family enough for the amount of support I have received from them my whole life and especially the last two years.

I want to thank my friend Mr. Peyman Jafari for his help and guidance concerning GIS data.

Last but not least, I would like to thank my friends Mehdi Shahparnia, Mona Saghae, Mina Saghae, Amirhessein Shajeri, and Babak Zamiri, who cheered me up through the ups and downs of this journey.

Contribution

The contribution of this thesis is threefold:

First, finding the combinations of the number of vehicles, the number of parking spots, and vehicle ranges that satisfy the demand with 90% or above performance while under different customer arrival rates.

Secondly, introducing a model that can be used as a framework for studying a small shared fleet's performance giving last-mile transportation service in any other road network.

Thirdly, providing the simulation with regard to the case study's GIS map and having the vehicles make the trips in the exact pattern of the real world.

Fourth, using a creative method for the movement of agents that lets them not only to move on the right path but also to move meaningfully with regard to the time step.

Table of Contents

List of figures……………………………………………………………………………………..ix

List of tables………………………………………………………………………………………x

1 Introduction 1
1.1 Background 1
1.1.1 Electric Vehicles 1
1.1.2 Autonomous Vehicle 2
1.1.3 A Fleet of Shared Vehicles 3
1.2 Problem Statement 4
1.3 Research Objectives 5
1.4 Dissertation Outline.. 6
1.5 Summary 6
2 Literature Review 7
2.1 Introduction 7
2.2 Autonomous Vehicles (AV).. 8
2.3 Shared Autonomous Vehicles (SAV) 10
2.3.1 One-way SAV fleet... 11
2.3.2 SAV Taxi 13
2.4 Personal Electric Vehicles... 17
2.5 Shared Electric Vehicles 19
2.5.1 One-way SEV fleet 20
2.5.2 Free-floating SEV 22
2.6 Shared Autonomous Electric Vehicles (SAEV)... 23
2.6.1 One-way SAEV fleet 24
2.6.2 SAEV Taxi.. 28
2.7 Problem Solving Methodology 31
2.8 Summary 34
3 Modeling Approach... 35
3.1 Introduction 35
3.2 One-way SAEV Fleet Selection.. 35
3.3 Unified Modeling Language (UML)... 36
3.3.1 Class Diagrams 37
3.3.2 Activity Diagram 42

3.3.3 Use Case Diagrams 44
3.3.4 Sequence Diagram 49
3.4 Summary 51
4 Agent-Based Simulation of a Fleet of SAEVs: Case Study of Olympic Park 52
4.1 Introduction 52
4.2 Agent-Based Modeling (ABM) 52
4.3 Simulation Software 54
4.4 Agent-Based Modeling of Montreal Olympic Park Using NetLogo 57
4.5 Model Assumptions 60
4.6 Model Execution 61
4.7 Summary 66
5 Numerical Results and Discussions 67
5.1 Introduction 67
5.2 Scenario Generation 67
5.3 Results 69
5.3.1 Rejected Customers 70
5.3.2 Performance 72
5.3.3 Discharged Vehicles 74
5.3.4 Rebalancing Policy 76
5.4 Summary 79
6 Conclusion and Future Work 80
6.1 Conclusion 80
6.2 Future work 81
7 References 83
8 Appendix 88

List of figures

Figure 3-1. Physical Class Diagram... 38
Figure 3-2. Resources Class Diagram... 40
Figure 3-3. Control Class Diagram... 41
Figure 3-4. Making a Trip Activity Process 43
Figure 3-5. Making a Trip Use Case Diagram... 46
Figure 3-6. Vehicle Entering Station Use Case Diagram 48
Figure 3-7. Making a Trip Sequence Diagram 50
Figure 4-1. Map of roads around Montreal Olympic Park area brought in NetLogo interface.... 56
Figure 4-2. Interface tab of the developed model of this study 57
Figure 4-3. Candidate stations 59
Figure 5-1. Number of rejected customers vs. number of vehicles for demand probability of 0.5 70
Figure 5-2. Number of rejected customers vs. number of vehicles for demand probability of 0.2 71
Figure 5-3. Number of rejected customers vs. number of vehicles for demand probability of 0.1 71
Figure 5-4. Performance vs. number of vehicles when demand probability is 0.5...................... 73
Figure 5-5. Performance vs. number of vehicles when demand probability is 0.2...................... 73
Figure 5-6. Performance vs. number of vehicles when demand probability is 0.1...................... 74
Figure 5-7. Number of discharged vehicles vs. number of vehicles when demand probability is 0.5... 75
Figure 5-8. Number of discharged vehicles vs. number of vehicles when demand probability is 0.2... 75
Figure 5-9. Number of discharged vehicles vs. number of vehicles when demand probability is 0.1... 76
Figure 5-10. Number of rejected customers vs. number of parking spots when the fleet has 20 vehicles of 50km range 77
Figure 5-11. Number of empty travels vs. number of parking spots when the fleet has 20 vehicles of 50km range 78
Figure 5-12. Number of discharged vehicles vs. number of parking spots when the fleet has 20 vehicles of 50km range 78

List of tables

Table 2-1. List of articles for fleet management problem 32
Table 2-2. List of articles for energy management problem.. 33
Table 2-3. List of articles for both fleet and energy management problem.................................. 33
Table 5-1. Possible values for different parameters.. 69

List of acronyms

ABM	Agent-Based Modeling
AEV	Autonomous Electric Vehicle
AMoD	Autonomous Mobility on Demand
AV	Autonomous Vehicle
BEV	Battery Electric Vehicle
CAEV	Connected Autonomous Electric Vehicles
CS	Charging Station
EV	Electric Vehicle
GHG	Green House Gas
HDV	Human-Driven Vehicle
IBR	Index-Based Redistribution
ICEV	Internal Combustion Engine Vehicle
LP	Linear Programing
MATSim	Multi-Agent Transport Simulation
MINLP	Mixed Integer Non-linear Programing
MIP	Mixed Integer Programming
MoD	Mobility on Demand
MPA	Model Predictive Approach
SAEV	Shared Autonomous Electric Vehicle
SAV	Shared Autonomous Vehicle
SEV	Shared Electric Vehicle
SNN	Simple Nearest Neighbor
SoC	Sate of Charge
UML	Unified Modeling Language
V2G	Vehicle-to-Grid
VKT	Vehicle Kilometer Traveled

Chapter 1:

1 Introduction

1.1 Background

The invention of internal combustion engine vehicles (ICEV) and their accessibility to individuals have made urban transportation faster and easier than it used to be in the past. A machine that used to help humans in their everyday lives is now making it impossible to live in big cities. Congestion, greenhouse gas (GHG) emissions, human driver errors, and the high price of owning a car despite being idle most of the time are significant shortfalls of current means of transportation. The emergence of novel technologies such as electric vehicles (EV), autonomous vehicles (AV), and autonomous electric vehicles (AEV), in addition to new methods of transportation such as fleets of shared cars, can lead to a more promising approach for urban transport. In the following subsections, each of the technologies and methods mentioned above, as well as their benefits and shortfalls, are briefly described.

1.1.1 Electric Vehicles

One of the significant disadvantages of ICEVs is their tailpipe emissions, which have negatively affected major cities' air quality. Due to the zero tailpipe emission of EVs, they seem to be a better alternative. Although some argue that this statement is not entirely correct since providing the needed electricity for charging EV batteries requires fossil fuel burning, which is a source of carbon dioxide emission. In response, Taiebat and Xu (2019) claim that with locating electricity generating plants outside cities, it is easier to manage the produced air pollutants, hence not directly affecting the cities' air quality. The other solution that they suggest is employing renewable energy sources for generating the required electricity, such as solar plates or turbines.

Despite the advantages of EV implementation in the environment, some barriers still exist to be accepted by most people. These barriers include battery limitation, limitations in the number of charging infrastructure, charging time management, and range anxiety. Range anxiety refers to the anxious feeling of not fulfilling a trip with an EV and losing all battery energy before reaching the destination. These concerns are usually unnecessary because, with the improvement of battery technologies, many vehicles can efficiently function to complete trips inside a city. However, for traveling long distances between cities, EVs might not be a good option. Other transportation methods, such as fleets comprised of shared EVs (later introduced in section 1.1.3), will relieve the user from any charging-related concern.

1.1.2 Autonomous Vehicle

Another emerging technology that has promising aspects for improving urban transportation is AVs. Since human errors play a significant role in road congestions, AVs' implementation could lead to error erasing. Increasing fuel efficiency, avoiding fatal crashes, and providing service for everyone are some of its other benefits (Fagnant & Kockelman, 2015).

To understand the concept of a fully autonomous vehicle, according to the SAE Standard J3016, Nieuwenhuijsen et al. (2018) developed a table for different levels of vehicle automation. Six automation level for the dynamic driving task is mentioned which are as follows:

- *Level 0 (No Automation).* The human driver is in charge of all the dynamic driving tasks, even when the vehicle is equipped with a warning or intervention system.
- *Level 1 (Driver Assistance).* The vehicle is equipped with a driving assistant system that can perform either steering or acceleration/de-acceleration. All the other elements of the dynamic driving task need to be done by the human driver.
- *Level 2 (Partial Automation).* This level is very similar to level 1, with the variation that the driving assistant system performs both tasks of steering and acceleration/de-acceleration.
- *Level 3 (Conditional Automation).* The vehicle can perform all the elements of dynamic driving tasks at this level but expects the human driver to act accurately when the system indicates human intervention.
- *Level 4 (High Automation).* This level is the same as level 3 except that the system can perform when the human driver is not acting in respond to an intervention request.
- *Level 5 (Full Automation).* The vehicle can perform all the elements of dynamic driving tasks under any road or climate condition, which is also feasible for a human driver.

Similar to EVs, there are some barriers to AV implementation as well. Liability in the case of an accident occurring, high price, legislation, interaction with other non-automated road actors, sufficiently developed AI are some of these obstacles (Mounce & Nelson, 2019).

1.1.3 A Fleet of Shared Vehicles

An alternative method introduced for urban transportation instead of personal vehicles is profiting from a shared vehicle fleet. Since most personal cars are only in use 5% of the time and are parked for the rest of 95%, owning a vehicle while considering its cost, oil prices, and insurance fees seems to be an inefficient way of transportation. Unlike the developing countries in which cars are considered an asset, they are usually regarded as transportation costs in developed countries. Many people prefer to lease a car for a few years instead of buying one. When a good quality car-sharing service exists, people can experience a convenience level close to owning a vehicle with fewer costs. As a result, car sharing has gained popularity in many modern cities nowadays.

There are some different approaches to providing a vehicle-sharing service. The first one is the two-way method in which people will borrow a vehicle from a station, and after they have finished their trip, they need to return that vehicle to the same station. The other method which is more convenient for the customers but might lead to more complexity in fleet management is the one-way method. In this method, customers can borrow a vehicle from a station and return it to any other station, including the first one from which they have borrowed. The third method is called free-floating, in which there are no stations, and only with the help of an application, customers find idle vehicles near them. They can drive the vehicle and park it inside the borders of a predefined region in the city. One last method that is only applicable in using a fleet comprised of AVs is the taxi mode. The AVs will pick up the customer from their depot and leave them at their desired destination.

1.2 Problem Statement

The information provided in the previous section indicates that using a fleet consisting of shared vehicles that are electric and autonomous can overcome some existing implementation barriers. Using a fleet of shared autonomous electric vehicles (SAEV) is a convenient, safe, and environmentally friendly method for urban transportation, which will be feasible in near future.

Managing such a fleet in order not only to gain profit but also keep a high level of service quality is the main challenge. Many researchers have tried to find the best solution in this area, stated in-depth in the second chapter. Some of the problems in this regard are such as following:

- *Redistribution*. This problem only happens in one-way fleets/free-floating and is when some stations/regions run out of vehicles while some other stations/regions face an overload of vehicles and run out of parking spots. In this case, to keep customers satisfied, the service provider needs to find a solution for reacquiring the system's equilibrium. Some proposed solutions are using trucks for transporting the cars or using a reward system for promoting the customers to park in a station facing car deficiency.
- *Energy management*. Whether a fleet comprises EVs or ICEVs, some time for recharging or refueling must be considered in the vehicle's schedule. Nevertheless, this is more important for EVs since recharging is considerably more time-consuming than refueling.
- *Customer wait time*. In order to gain more performance satisfaction, the customer wait time should be as minimized as possible.
- *Cost minimization*. Like any other business to gain profits, the costs should be minimized while keeping the service quality at a predefined level or above. This matter is essential considering that ride fees cannot be very pricey so that the fleet could compete with other transportation methods.

1.3 Research Objectives

Not all the problems mentioned in the previous section can be tacked in this study. Here the objective is to develop an agent-based simulation model for a small fleet comprised of SAEVs.

Since these problems are case-dependent, in the case study providing services to reach different tourism attractions near Olympic Park in Montreal is considered. Modeling a one-way transportation service comprised of a few stations and developing different scenarios to find the optimal solution for the case study is our work's primary objective.

1.4 Dissertation Outline

The rest of this study is as ordered as follows:

In the second chapter, a literature review on EVs, AVs, and shared fleets comprised of them is reviewed, and their method of problem-solving is discussed.

In chapter three, the general method for optimizing a fleet comprised of SAEVs is considered.

Chapter four presents the case study of Olympic Park.

Chapter 5 contains numerical results and discussions.

Finally, in chapter six, the conclusion of this study and the future research prospective are stated.

1.5 Summary

In this chapter, a brief review of new technologies and methods that alter urban transportation towards being more sustainable and less costly is introduced. Then the limitations and problems towards employing them by the majority of people and governments are enumerated. Further, the main goal of this research and outline of the following chapters is stated.

Chapter 2:

2 Literature Review

2.1 Introduction

Using a fleet of shared vehicles instead of private cars or public transportation for daily travels is a new alternative and is becoming popular in many cities. In this regard, many researchers devote their time and resources to investigate such shared fleets' improvement methods. To tackle some challenges of everyday transportation such as lack of parking space, availability of vehicles, and lack of driving abilities, the implementation of autonomous vehicles in a shared fleet is proposed. An autonomous car can pick up travelers at the start point and drop them off at their destination. Afterward, it can either go back to its designated parking station, refuel, or pick up another customer. Nowadays, cars are equipped with smart technologies that allow them to act autonomously in special situations such as adaptive cruise control, lane-keeping guidance, collision warning, blind-spot warning, and parking assistance. Even with these developments, the application of fully autonomous vehicles is not popular or even legalized broadly (Mounce & Nelson, 2019). The adoption of such new technology depends on different factors. People with a high educational level, people living in dense areas, younger generations, and households without a car are more willing to adopt autonomous vehicles (Liljamo, Liimatainen, & Pöllänen, 2018). Many contributions in the literature are devoted to examining the capability of adopting such technology in our daily life to improve the quality of daily transportation while moving towards sustainability.

One possible way to move towards a sustainable city is electric cars' implementation instead of combustion-based ones. When it comes to using EVs, there are always some obstacles that prevent people from adopting them. These obstacles include range anxiety (fear of running out of charge without reaching the destination), availability of charging structure, and charging time. A natural synergy happens when using EVs in a shared autonomous fleet since the barriers mentioned above can be tackled in such a system. (Chen & Kockelman, 2016).

Regardless of whether a fleet is composed of electric cars or combustion-based ones, the primary objectives of most of the optimization problems are minimizing customer wait time and empty vehicle travel while maximizing the profit. With these objectives in mind, fleet management problems such as the number of required vehicles, routing policy, relocation strategy, and energy management problems such as critical charging level and charging policy can be answered.

In the following sections, works conducted in the literature regarding the optimization of a fleet of shared AVs and AEVs are reviewed. Section 2 studies the works conducted on autonomous vehicles, and in section 3, shared fleets of them are discussed. In section 4, electric vehicles and later in section 5, shared fleets of them are mentioned. Finally, shared autonomous electric vehicles are discussed in section 6. A conclusion is given in section 7.

2.2 Autonomous Vehicles (AV)

Fagnant & Kockelman (2015) investigate the significant benefits of employing autonomous vehicles instead of manual ones. These benefits are such as:

- Safety
- Congestion reduction

- Accessibility by a wide range (people with or without driving ability)
- Parking fee savings
- The possibility of having more productive travel times for former drivers

In addition to these benefits, they also study the barriers to AV implementation. These barriers are such as:

- Vehicle cost
- AV certification
- Legislation and liability
- Electronic security
- Data privacy

Nieuwenhuijsen et al. (2018) implement a system dynamics approach to find the market penetration of autonomous vehicles quantitatively in the long run. They consider the six levels of automation from level 0 (fully manual) to level 5 (fully automated), and each level has its fleet size, technology maturity, price, and customer utility. They assume that customers are constantly comparing the utility of the automation level that their vehicle brings to the utility of higher levels of automation. Gradually such behavior leads to the adoption of higher levels of autonomy. They tested various scenarios and policy adoptions in the case study of the Netherlands. The results show that market adoption is highly sensitive to alteration of policies and scenarios.

Stern et al. (2019) studied the effect of a sparse number of autonomous vehicles on overall GHG emission of traffic flow. They suggest that the stop-and-go behavior of human drivers, which leads to constant acceleration and deceleration of vehicles, is a GHG emission and energy loss source. By designing an experiment using one AV and 22 human-driven cars, they show that 5% of AVs designed to stabilize the traffic flow could lead to 15% (carbon dioxide) and 73% (nitrogen oxide) reduction in vehicular emission. Connected autonomous vehicles (CAV) are the kind of AVs connected to each other and the infrastructure. Such technology existing in the traffic flow will lead to safer and smoother driving patterns (Ye & Yamamoto, 2019).

While many researchers consider a fleet of shared autonomous vehicles when they model the future of daily urban transportation, Correia & van Arem (2016) consider a scenario in which personal AVs and public transportation are the only options for motorized urban transportation. They consider a model in which each AV satisfies all the household members' trips in the morning, parks itself, and later in the afternoon returns members home. In this model, any other travel during the day, such as going to the gym, is also included. They study the case of the city of Delft in the Netherlands. Their results show that such a transportation method could lead to lower travel costs and higher travel quality but would lead to higher congestion.

2.3 Shared Autonomous Vehicles (SAV)

Autonomous vehicles have the potential to modify the shape of future transportation and consequently urban living. González-González, Nogués, & Stead (2019) use a backcasting approach to identify the positive and negative impacts of AV utilization on urban living from social, economic, and environmental aspects. Martínez-Díaz & Soriguera (2018) delve into AV technology, its effects on mobility, people’s acceptability, and required new legislation through a review of former works available in the literature.

This section considers works conducted for modeling and studying different types of problems that one might face offering an SAV fleet service. For providing such services usually, there are three methods. The first one is when there are several stations in the study region, and customers should pick up the cars from these stations and bring the car back to its first place (two-way trip model). The second method is when there are stations, but there is no necessity to bring back the car to its first location, and customers can pick up or drop off the vehicles from their nearest station (one-way trip model). The third method is picking up customers at their start location and dropping them off at their destination (taxi model). In the following, we categorize each work based on its service providing method.

2.3.1 One-way SAV fleet

This service-providing method is more convenient for the customers than the two-way method since many trips are one-way, and the traveler might not want to get back to the start node right away. The biggest problem with this method is that eventually, the system's equilibrium is lost. Some stations run out of vehicles while some run out of parking space, leading to customer dissatisfaction. To prevent such discomfort, service providers need to relocate the excess of cars from the overpopulated station to the ones with a shortage of cars.

Correia & Antunes (2012) consider the doing relocation of cars at the end of each day. They develop a mixed-integer problem (MIP) to find the best location for depots of a one-way carsharing system while considering three different scenarios for meeting the demand to maximize the profit. The first scenario is when the system meets all the demand. The second one has the right to reject any demand if it was not profitable, even if there would be enough resources to meet the demand. The last one is a hybrid of the two such that it meets the demand when cars exist at the depot and rejects it otherwise. Their results indicate that the most profitable scenario is the last one when the system meets the demand if resources are available.

Pavone et al. (2012) address the rebalancing problem by assuming that vehicles can autonomously drive to the station where there is a shortage in demanded vehicles; thus, rebalancing is done automatically. With a fluid model, a linear problem is introduced to minimize the number of vehicles needed to meet customers' demands. The solution to this problem is a policy that is tested in a simulated situation via MATLAB and hardware experiments. The results indicate that the provided policy can rebalance the model so that there is always an excess of vehicles in caparison with the customers in each station.

Fagnant & Kockelman (2014) investigate the environmental benefits and the vehicle miles traveled (VMT), and cold vehicles start utilizing SAV instead of personal cars. An agent-based model is introduced by the authors, who initially finds the required fleet size for having reasonable customer wait times. Other than enough vehicles, every 5 minutes, the unoccupied cars are relocated to be around more demanding areas to decrease waiting times. Twenty-five Different traveling scenarios are generated based on trip demands, trip patterns, traffic level, service area limits, vehicle relocation strategies, and the number of vehicles. Their simulation for 100 days demonstrates that the number of personal cars that a person needs is decreased by ten times by using SAV. On the other hand, because of relocation policies, the total distance traveled is 11% increased. These results indicate that for most cases, the emission of undesired gases to the environment is decreased.

Spieser et al. (2014) are one of the first researchers to model a big size problem based on a city's real transportation data. They try to find the right number of autonomous vehicles in a fleet of Mobility-on-Demand to meet the request of all customers who were initially using personal cars for transportation and now have switched to using a shared fleet. The number of vehicles should be small enough such that providing the service would be financially feasible. They have also considered different transportation systems using personal cars, autonomous shared cars, dual-mode shared cars, autonomous personal cars, and taxi drivers and compared these systems from an economic point of view. The real taxi data from Singapore is used as the data for the case study. The results show that each vehicle of the fleet can substitute three personal cars. Finally, the problem was solved by using mathematical modeling.

2.3.2 SAV Taxi

The most convenient method for the customers is the taxi model in which autonomous vehicles drive to the passenger, and after finishing the service, they park themselves. Levin et al. (2017) compare several scenarios such as SAVs used as a taxi, ridesharing, and using personal AVs. They conclude that using an SAV fleet outperforms AVs in the ride-sharing scenario. They provide a general framework for the investigation of SAVs efficacy. This framework makes it possible to model a variety of problems while considering realistic traffic assumptions. Meaning that while SAV is an option for transportation, several other options exist, such as personal cars. They believe that the works done so far on the subject are not economically reliable since many of these works fail to model the traffic flow's real situation. An event-based framework is proposed to solve the model. The problem of optimizing ridesharing is NP-hard and not solvable in real-time frames. As a result, a heuristic approach is considered for its optimization.

To study the effects of replacing all the daily travels carried out via personal vehicles with SAVs, Levin (2017) proposes a linear routing optimization problem to minimize travel time. Every demand must be met in his model, and demands are considered to be known in advance. Several scenarios are suggested to test the effect of such a transportation system on traffic congestion. For rush hour cases, when the demand is asymmetric, using SAVs instead of personal cars would result in higher congestion. However, for times when the demand is symmetric, no adverse effect on congestion is reported. Besides, the model finds the optimal number of used AVs, which is less than the available ones to stay in optimality.

Bsaybes, Quilliot, & Wagler (2017) consider transportation inside an industrial area via individual public autonomous vehicles (VIPA). They develop a mathematical model of the problem and come up with two algorithms. One for solving the offline problem and the other for implementing a replan strategy for the online problem. These two algorithms show that the replan strategy can effectively perform compared to the optimal solution acquired from the offline problem.

Liu et al. (2017) study the possibility of using a fleet of SAVs to transport people in Texas's Austin region using a multi-agent transportation simulation (MATSim) toolkit. This region lacks efficient public transportation. As a result, it is considered the right candidate for the utilization of an SAV fleet. This work primarily generates travel demand per traveler, simulates travel, reports the results, and studies other impacts such as greenhouse emission and energy consumption. In this regard, four scenarios based on \$0.50, \$0.75, \$1, and \$1.25 fare rates are generated, and the simulation results show that 50.9, 12.9, 10.5, and 9.2% respectively switch to SAVs. The results show that when the fare rate is low, people with access to human-driven vehicles (HVs) have the tendency to switch to SAVs, but those without HVs have this intention only when the trip is short (less than 10 miles).

Hörl (2017) Investigates the ability of simulation in showing people's reaction towards new ways of transportation (AVs as one-passenger taxies or car-pooling ones). The one-passenger taxi provides faster service but has a more expensive fare rate than the car-pooling taxies. Other choices that have been considered in this study are using buses, walking, and using personal cars. The author uses MATSim to gain the results from travelers' preference between different transportation methods while changing the fleet size. This study provides a useful framework for future research.

Hyland & Mahmassani (2018) define six different strategies for AV to traveler assignment in a shared-use AV mobility service (SAMS) without ridesharing. The first two strategies are the simple FCFS, and the last four are more complex, requiring solving an optimization problem. These assignment strategies are conveyed to the fleet from a central operator. They used agent-based simulation based on a Manhattan gird network with two artificial demand sets and one acquired from Chicago taxi data. Their results show that the optimization-based assignment strategies, in terms of relative empty vehicle miles traveled and customer average wait time, are far better than the simple FCFS ones when the fleet size is small relative to the demand. The last strategy, which considers assigned and unassigned travelers as well as the idle and en-route vehicles in the optimization problem, outperforms other strategies in most cases. Their work is very novel and practical. However, it does not consider road congestion, refueling AVs, and the historical data for demand.

Alam & Hbib (2018) investigate the impacts of using a fleet of SAVs during rush hours in the morning on traffic congestion while considering human-driven vehicles (HV) on the road. They model the city of Halifax as their case study. Vehicles are kept in 32 stations in the city. When there is a demand in the system, a vehicle is assigned to pick up the customer only if the travel time from the vehicle destination to the customer is less than a waiting time threshold; otherwise, the person will have no eligible vehicle will use an HV. They tested four various fleet sizes with different percentages of SAV and HV utilizations via microsimulation. Results show that when the fleet size is equal to 900 and 20% of the trips are made via SAVs, the average speed is increased in the first hour (7 am to 8 am), the average speed is increased, and vehicle kilometers traveled (VKT) is decreased.

Winter et al. (2018) use simulation to show the effects of changes in the level of demand, the capacity of vehicles, dwell time of vehicles, and initial distribution of vehicles on the required fleet size and operation cost in an automated demand responsive transport service (ADRTS) system. The system allows travelers who have the same origin and destination to be combined with being served with one vehicle. The dwell time is the time that each vehicle is supposed to wait to gain more passengers before moving towards its destination. Comparing the system's operational cost with the bus service shows that they are both in the same range. A case study of the city Arnhem in the Netherlands is investigated. Results show that with the minimum fleet size to meet the demand, each passenger trip's operational costs are between 0.84 and 1.22 Euros. Also, for the same fleet size, the passenger idle time is between 2 to 6 minutes.

Babicheva et al. (2018) compare different algorithms for reactive and proactive redistribution of AVs. Reactive distribution is the relocation of vehicles to pick up a customer, and proactive redistribution is carried out towards a demand that is not yet in the system. However, due to historical data, there is a good chance for its occurrence. Four reactive and two proactive redistributions and their combination are tested. The results show that the combination of the introduced proactive algorithm, Index-based Redistribution (IBR), with reactive Simple Nearest Neighbor (SNN) would lead to the least maximum and average waiting time and the queue length. Their work emphasizes the importance of predicting future demand and moving towards responding to it.

Starting with a thorough literature review, Hörl et al. (2019) simulate the autonomous mobility on demand (AMoD) for the city of Zurich in Switzerland. They consider four different existing policies in the literature with various fleet sizes while considering peak and off-peak travel patterns to shape the desired scenarios. Two of these policies find the optimal pattern for discharging vehicles to the customers, and the other two find the optimal vehicle redistribution. They used the MATSim surface to model the problem to compare the attractiveness of policies from the aspect of providing the least traveling fees and wait times. Their results show that cost and wait times are highly dependent on each policy's choice and the fleet size. Fleets of less than 7000 vehicles lead to long wait times, and fleets of more than 14000 vehicles dictate infeasible traveling fees to the system. They found that traveling fees are higher than subsidized public transportation and personal cars in the short term for this case study. Nevertheless, in the long term, this AMoD can compete with personal cars. Also, the traveling fee is far lower than the case of using taxis.

2.4 Personal Electric Vehicles

With the emergence of electric vehicles, some people have switched to using them in the past years. However, still many people have resistance against them even though these cars would benefit the environment. Kang, Feinberg, & Papalambros (2015) claim that the primary reason for the cold welcome towards EVs is the consumer range anxiety. Range anxiety as described earlier is the existing concern among users that an EV would run out of charge during a trip before reaching a charging point. Since the operating range depends on the vehicle and its battery attributes, the availability of charging station (CS), and charging time, to improve it, both EV manufacturers and CS operators should provide profitable products and services. The authors propose a cooperative business model in which EV manufacturers and CS operators act as one and share benefits and losses. Their optimization model is made of three subsystems: the demand model, the EV design model, and the CS location network model. They show that this model, compared with the sequential one (in which each business has its own gains and losses), is performing far better.

Pourazarm, Cassandras, & Wang (2016) seek to find the fastest route considering EVs' charging range. First, they model a mixed-integer non-linear problem (MINLP) for a single electric vehicle, which is hard to solve but can be decomposed into two subproblems: linear problems (LPs). One tries to find the fastest route for the vehicle, and the other tries to find the recharge amount. Further, they consider multi-vehicles in the system, and by clustering them into sub-flows, they once again model the same problem as they did for a single-vehicle. The decomposed subproblems are not a simple linear problem this time. As a result, another formulation is proposed. In this formulation, the sub-flows can be divided into several optimal routes to lessen the former model's complication with MINLP formulation. Then a transportation system with both EVs and non-electric vehicle are considered. Finally, the price of acting selfishly (optimality of single-vehicle) is compared with and the price of acting optimally as a group (sub-flow optimality). Robust mathematical modeling and reasoning can be considered as the main strength of their work. Although, their work has some limitations that can be the target of future studies. For example, the vehicles' flow is considered deterministic, and the effect of uncertainty is not considered.

Electric vehicles benefit the environment by their zero GHG emission characteristic. They can help the renewable energy resource generators by acting as a storage when electricity production is higher than the demand. This extra electricity can be used to run or give back to the grid for peak demand times. These are the characteristics of smart charging strategies that are studied in several works. Hu et al. (2016) review these works to find the positive and negative points of each strategy dictated by the fleet operator to the EV owners. These strategies are called centralized control, transactive control, and price control. Their results show that the best of them so far is centralized control.

Due to the environmental benefits of employing electric vehicles instead of combustion-based ones, Usman et al. (2017) propose a model to concur with the range anxiety of using EVs. They consider the charging possibility of EVs at home or in the workplace. When the state of charge (SoC) is not high enough to complete a trip, charging at a fast-charging station is proposed. Detouring to get to a charging station is considered a factor that decreases the overall utility. They used the travel data of Flanders, Belgium, to simulate such a transportation system.

Yi & Shirk (2018) consider the problem of optimizing the charging act (minimizing cost and time of charging) of connected autonomous electric vehicles (CAEV) for personal usage. The vehicles are connected to an infrastructure which means they are aware of the price, location, and waiting time of charging stations. They are autonomous, which means they can drive and perform the process of charging without a driver. They developed a framework and then a real-time updating algorithm for the stochastic energy cost prediction. Later, they developed a multi-stage model and a dynamic algorithm to make optimal charging decisions such as where to charge or how much to charge.

2.5 Shared Electric Vehicles

To profit from the benefits of EVs and at the same time respond to customer concerns in adopting EVs such as range anxiety, using electric vehicles instead of combustion-based ones in a ride-hailing service system is introduced. Using electric cars will add to the problem's complexity since charging considerations must be added to the model. However, it profits the customers since they do not need to worry about running out of charge during a trip. Like the SAV problems, several service providing methods, such as the one-way trip model and two-way trip model, can be considered in these systems. The taxi model would not be applicable since the car cannot drive itself to the customer. Nevertheless, there is another format in which there are no stations, and customers can pick up or park a car in a predefined region. Such a service-providing system is called the free-floating model.

2.5.1 One-way SEV fleet

Hafez, Parent, & Proth (2001) were pioneers in considering managing an electric shared vehicle fleet. They conducted two problems for redistribution and recharge of the shared cars available in a suburb of Paris. A redistribution is carried out when the current arrangement is considered unfavorable, which means there are shortages or overflows in the number of cars in the stations. The number of cars to be redistributed for each station is modeled as an integer linear programming. Later this ILP is relaxed to be solved. By knowing the number of cars to be moved as a constraint, they modeled the problem to minimize redistribution duration. This problem is NP-hard. As a result, heuristic procedures are proposed. The recharge problem is solved by considering that the distribution is optimized. A cost function is introduced consisting of customer dissatisfaction because of a lack of energy of the cars available in stations and cost incurred when a vehicle cannot complete its journey due to lack of energy for a predefined energy level threshold. This problem aims to find the energy level threshold which minimizes such cost function.

Rigas, Ramchurn, & Bassiliades (2015) propose one of the first works which consider managing electric shared cars to maximize the total number of serviced customers in an MoD system comprised of EVs. Using Mixed-Integer Programming (MIP), they can find the optimal solution for problems comprising of few hundreds of tasks. They also propose a greedy heuristic algorithm that can converge to optimality 90% of the time when using real data of Washington DC city for problems with the size of thousands of tasks and EVs. Then with the assumption that each station has charging facilities, they propose a battery swap optimization algorithm.

Biondi, Boldrini, & Bruno (2016) try to find the right location for stations and the number of parking spots for each station in an MoD system made of EVs. To do so, they propose a stochastic optimization problem since car demand cannot be precisely predicted. Their result shows that only four or fewer parking spots would suffice in many cases, but there is also the need to have a few big stations with up to 15 parking spots. In the next step, they try to determine the fleet's energy demand while considering different assumptions. One of these assumptions is to suppose a system made of power-sharing EVs. In this system, EVs can transfer energy to each other, which is demonstrated to lead to a decrease in the cost of building charging stations without significantly affecting the state of charge (SoC) of the vehicles. Although, they tackle these two problems separately.

Brendel et al. (2018) propose a model to optimize the utilization of battery electric vehicles (BEV) in a fleet made of BEVs and internal combustion engine vehicles (ICEV) from the energy management perspective. In their model, a framework is introduced to predict the required SoC for completing a trip. The other aspect of the model is the charging schedule of BEVs with a battery-life expanding approach. EVs will be sent to be charged as little frequently as possible, and they will be plugged into the charger until fully charged. Later in the model, a flowchart is introduced for vehicle selection with prioritizing BEVs. If there is an available BEV with enough charge to complete the trip, it is chosen by the customer; otherwise, the ICEV is chosen. Since this is a one-way carsharing problem, the relocation is also considered to balance each station's supply.

The rebalancing is composed of having employees undertake it or giving the customers incentives to park the cars in the model's desired parking spot. Simulation results show better utilization of BEVs with no incident running out of charge during the trips.

Mounce & Nelson (2019) elaborate on the potentials of using one-way electric car sharing. Their work is composed of reviewing the literature on the subject. They have gathered information on the advantages, barriers, and potentials of autonomous vehicles, car-sharing, and electric cars. They conclude that with the adoption of a SEV fleet in addition to good public transportation, many transportation problems such as road congestion, lack of parking space, and environmental problems such as greenhouse emissions would be encountered.

2.5.2 Free-floating SEV

Wang & Cheu (2013) continued the work of Lee et al. with the implication of electric vehicles instead of combustion-based ones. In this work, the objective is to minimize the number of taxies needed to meet the Singapore population's demand while considering vehicle charging requirements. In this model, taxies of the fleet are reserved by customers. A mathematical formulation is used to model the problem, and since it is an NP-hard problem, three heuristics with a two-phase approach are used to gain a solution on a real-time scale. These heuristics are as follows: nearest neighbor, sweep, and earliest time insertion. Numerical results show that the earliest time insertion solution is the best choice among all. Later, three different scenarios for battery capacity and recharging times were developed. Results show that a more significant capacity with a longer charging time scenario serves best in terms of taxi drivers' revenue and the number of visits needed to the charging station. Another experiment also highlights that adding the number of charging stations would not significantly impact performance metrics. This study is based on a small transportation network but, with modification, can be used for other networks as well.

Bischoff & Maciejewski (2014) modeled a fleet of electric taxis for transportation in a small city while simulating the traffic. Four different types of scenarios for traffic flow are designed. With the use of a Multi-Agent Transport Simulation (MATSim), they drew the following conclusions. During non-peak hours, using a fleet of electric vehicles as taxies have a performance as promising as a fleet of conventional taxies. However, during peak hours with the increased demand, a conventional taxi has better performance.

Bauer et al. (2019) continued their former work to find economic justification for using a ridesharing fleet composed of electric vehicles. They used an agent-based model already developed in Bauer, Greenblatt, & Gerke (2018). They demonstrated that with the implementation of sufficient charging stations (three to four 50kW chargers per square mile), a SEV fleet with an average battery range could meet the demand of customers of a fleet composed of internal combustion engine vehicles (ICEV). They show that implementing the cost of charging infrastructure given that chargers are utilized at least 15% of the times (3.5 hours in each day) plus the electricity price will be less than gasoline cost for the equivalent fleet combined of ICEVs. With these findings, they conclude that regulations that mandate electrifying vehicles due to reducing GHG emissions are feasible with today's technology.

2.6 Shared Autonomous Electric Vehicles (SAEV)

As already mentioned in this chapter, range anxiety, ease of access to charging infrastructure, and the fact that the charging process is time-consuming, are the significant barriers to EV adoption.

Four emerging technologies entitled vehicle automation, wireless charging, shared mobility, and vehicle-to-grid (V2G) integration can make up for these barriers (Taiebat & Xu, 2019).

2.6.1 One-way SAEV fleet

Boyacı et al. (2015) try to find the optimal number of vehicles, relocation personnel, and stations as well as the optimal location of stations for a one-way car-sharing fleet. By seeing that relocation is a must in a one-way car-sharing, they propose a MILP model to solve the problem. The MILP is followed by a heuristic while considering a virtual hub for the cars and using branch and bound procedure to find a solution on a real-time scale. Solving the problem as a multi-objective model helps find the desired level in the existing trade-off between the user's benefit and the operating system's benefit since both are seen in the objective function with assigned weights.

Iacobucci et al. (2018) are some of the first authors to consider SAEVs as both a transportation method and an electricity storage. They investigate the profitability of providing a SAEV fleet for the service provider. They also test the capability of the AEVs to act as a spinning and non-spinning reserve for the grid, which means that the vehicles can provide transportation services. However, they can also be considered an operating reserve for the grid to give electricity to the grid when the demand is high and absorb the over-generated electricity when the demand is low. To solve the transportation problem, they utilized a simulation model based on Tokyo's transportation data and proposed an algorithm to optimize the charging schedule based on electricity prices. Their work shows that each AEV in the fleet can replace 7 to 10 private cars.

By considering EVs and AVs characteristics and the fact that they complete each other's inefficiencies, Chen, Kockelman, & Hanna (2016) suggest that employing a fleet of SAEV would lead to a more practical method of car sharing. As a result, they investigate the capability of a fleet of shared electric vehicles, which are also autonomous and self-charged to answer the demand for everyday travel in an urban area. They use a discrete-time agent-based simulation that can be considered an extension of Fagnant and Kockelman's (2014) work. They consider different types of vehicles with various charge ranges and different types of charging stations and generate several scenarios and consider a gridded city based on Austin, Texas as the case study. The simulation shows that the most cost-efficient scenario is an 80-mile range vehicle and a level II charging station. Also, each 80-mile range vehicle with a level II charging station can replace 3.7, and each 200-miles range vehicle with the same charging station type can replace 5.5 privately owned cars. Furthermore, each 80-mile range vehicle with a level III charging station can replace 5.4, and each 200-miles range vehicle with the same charging station type can replace 6.8 privately owned cars. In the last case study, Austin's actual travel demand is considered, which leads to 5 to 9 privately owned car replacements.

Chen & Kockelman (2016) try to find the market share of SAEVs while competing with private vehicles and transit. They establish values such as the value of travel time for each method of transportation. They then assigned different fare rates for the SAEV services. Their results show that transit is more sensitive to the share rate of the SAEVs. Also, for longer trips, SAEVs are more preferred than private vehicles.

Zhang, Rossi, & Pavone (2016) introduced a wholistic model predictive approach (MPC) for fleet optimization of autonomous mobility on demand (AMoD) with the capability of adding real-world constraints such as charging level. Model predictive is a controlling approach that works to solve an optimization problem sequentially, and optimization policies are established for each time step up until a fixed horizon. In this regard, they developed a Lyapunov stable MPC algorithm and compared it with other existing algorithms in the literature. They then solved the model as a mixed-integer linear program with binary variables to serve all the customers, conduct an efficient rebalancing, and finally maximize the vehicles' charge state at the end of the time horizon. Two algorithms were conducted with and without the charging constraint. Using real taxi data showed that the proposed algorithms are far better from the customer wait time aspect than the existing ones.

Iacobucci, McLellan, & Tezuka (2019) propose an optimization model for relocation and vehicle charging based on Zhang et al. model by considering the charging level as not only a constraint but also a decision variable. With such innovation, they were able to add a V2G policy to their model. They conducted the optimization model as a MILP by proposing a two-time-scaled algorithm. The relocation problem is carried out to gain the optimized waiting time. Then the results are set as constraints of the second layer, which is the optimization of charging problems to minimize electricity costs. Rebalancing is optimized during a 15-30 minutes time scale, while the charging problem needs several hours to be optimized. Finally, the authors conducted a simulation with the data from Tokyo Trip Survey 2008. They concluded that since electricity prices are not wildly variant in Japan, the V2G policy does not have considerable benefits. However, when the electricity prices are more variant as they generated data based on gamma distribution, the model outperforms significantly.

Kang, Feinberg, & Papalambros (2017) try to find the optimized fleet size and assignment schedule, number and location of charging stations, vehicle attributes, and service fee in one general model for the shared AEV fleet. They also compare AEV and AV shared vehicle fleet from an economic aspect. They proposed an optimization system model that takes four subsystems into account. AEV design, service demand, CS location, and fleet assignment are these subsystems. The optimization is carried out so that the system-level fleet assignment connects all the other subsystems. The result of their simulation shows that both AEV and AV are marketable though the profit from AEV in their business scenario is slightly higher than AV. Also, the simulations show that even though membership fees and driving rates are higher in autonomous rather than regular car-sharing services, due to the benefits of autonomous pickup and return, it is still worthy for the customers and has its market share.

Scheltes & Correia (2017) used a fleet of one-passenger AEVs for compensating the shortfalls of public transportation as a last/first-mile travel system. One of the inconveniences of using public transportation is the last/first mile of the travel when the only way of reaching stations is time-consuming and difficult (walking or biking). They consider the Delft Zuid station near the University of Delft's campus to study the daily transportation from different faculties to the station. The vehicle used in their study can travel in the same line as bikes. As a result, the speed cannot be greater than the average cycling speed. Their results show that these vehicles can bring higher travel utility for pedestrians but not for cyclists unless there is possibility of manual driving to reach higher speeds.

Battery capacity and the time-consuming process of charging are two main reasons that will result in less utilization for a shared fleet of vehicles when using electric cars than conventional ones. Zhang, Liu, & He (2019) try to find a solution that leads to a good utilization percentage for a one-way fleet of shared electric vehicles. They propose a space-time-battery flow network model with two considerations. One is to optimize the vehicle allocation to the customers, and the second one is to add vehicle relay to the former optimization problem. Vehicle relay is defined as a situation when there is a demand for a long trip, but there is no vehicle with enough SoC to undertake it. In that case, the customer is suggested to use a vehicle to get to a transitional station and then switch to another one to complete his/her journey. Since the mathematical model is hard to solve and unable to give instant answers, a diving heuristic algorithm is introduced. When data of Shanghai's case study is inserted in the model, the results show that the optimization allocation will bring an agreeable vehicle utilization percentage when demanded trips are short. Also, when the demand for long trips is high, the model with relay consideration performs better. In this model, the relocation of vehicles that focus on many former works in the literature may or may not be considered.

2.6.2 SAEV Taxi

Awasthi et al. (2011) model a fleet of a particular type of AEVs (Cybercar) with the use of a centralized fleet management system (CFMS) with the objectives of employing a minimum number of cars to meet the demand of customers while being satisfied. Customer satisfaction is acquired when the customer waiting time is minimized. Each demand is requested in advance, and then a cybercar is assigned to meet the demand. Ridesharing is suggested as a solution to minimize the number of required cars. The fleet operator controls the assignment of each car to several customers. He/She can also update the routing of the car in real-time manners with consideration of the road congestion. Recharging is met by defining a threshold for giving service. If a vehicle's charge status is less than that threshold, It is sent to the stations to recharge.

Farhan & Chen (2018) consider the possibility of employing ridesharing in a fleet of SAEVs from the fleet operation perspective. Based on the work of Chen et al. (2016), they utilize a discrete-time agent-based simulation to solve the problem. Also, they optimize the routing with the help of a Tabu Search algorithm. Their work shows that as much as the number of vehicle capacity to be shared is increased, the number of required fleets and the charging infrastructure is decreased.

However, on the other hand, the passenger wait times are also increased. The best results are acquired when the cars' capacity is equal to two since, by having the benefits of ridesharing, the wait times are still in a tolerable zone.

Yi, Smart, & Shirk (2018) believe that an autonomous car is more agreeable when it is electric rather than a combustion engine since controlling an electric device is easier for a computer. Also, it is safer for an electric car to be recharged autonomously rather than being refueled. Upon such reasoning, they study the energy management of a fleet of SAEVs based on New York City's taxi date. First, they develop a grid for stochastic energy consumption based on the velocity of deriving and ambient temperature. Then they suggest eco-routing of vehicles to minimize energy consumption. Their results show that the ambient temperature has a significant on the energy consumption of AEVs.

Bauer, Greenblatt, & Gerke (2018) use agent-based simulation to find the required fleet size, vehicle battery capacity, and number and location of charging infrastructure to meet the Manhattan city's taxi demand while minimizing the operational costs. Their model runs an algorithm to find the optimal place and number of charging stations using an eliminating procedure. Every passenger will wait a maximum of 10 minutes to be picked up. A car is assigned to a passenger if it can make it there by 10 minutes and has enough charge to complete the trip plus the trip to the charging station. If that was not possible, a new taxi is created to meet the demand. This way, the required fleet size to meet the demand will be determined while complying with the simulation's policy to create the minimum number of vehicles possible. Unlike former works in the literature, as soon as a taxi becomes idle, it goes to a charging station and stays there until fully charged or assigned to another trip, which means there is no need to fall below a predefined critical threshold making a charging trip. This attitude makes it possible to meet the demand with lower battery capacity. The result of their simulation finds the optimal battery range (50-90 mi) and charging infrastructure (66 chargers per square mile, rated at 11 kW or 44 chargers per square mile, rated at 22 kW), which have considerably lower fees than a conventional taxi. Comparing such a fleet with an automated fleet of internal combustion engine vehicles (ICEV) or hybrid vehicles also depicts lower fees. Besides, the optimal fleet was compared with an automated fleet composed of ICEVs. The results indicate 73% GHG emission and 58% energy consumption reduction for the proposed model.

Loeb et al. (2018) try to find a more realistic way for modeling a SAEV fleet for the same region studied by Chen et al. (2016). They aim to first provide a more realistic network for modeling the problem to investigate the reliability of the results provided by former authors. Besides, they develop the model by increasing the number of possible scenarios and introducing new charging strategies. To solve the problem, they utilized the model provided by Bösch et al. (2016) and added a charging process and infrastructure consideration to the problem. The conducted simulation enables them to find robust charging locations. With the resulting locations, they investigate the effect of fleet size, charging time, and battery range on the customers' average waiting time. The result of their simulation shows that the number of required charging stations depends on the AEV range. The most prominent factor in reducing the customer waiting time is increasing the number of vehicles. Decreasing the charging time (by using fast-charging equipment) can also reduce wait times, but it has less effect than the fleet size increase. Also, they observe that increasing the range of vehicles to more than 175Km has no significant impact on waiting times.

Zhang et al. (2020) are one of the few researchers that have tackled fleet management and energy management of autonomous electric vehicles (AEV) as a holistic problem at the same time. They study a ride-hailing service with a unique agent-based simulation called BEAM. In this simulation, the driving, charging, and parking behavior of vehicles are considered. Via this simulation, they can find the location and time of charging demand of each vehicle. With this information and with the use of a hybrid algorithm based on K-mean clustering, they can find the best locations for the charging stations. The economic and environmental benefits of employing such a fleet for an urban area (San Francisco Bay Area) are reported. Various scenarios while considering several different fleet sizes, battery capacities, and charger powers are studied.

2.7 Problem Solving Methodology

Throughout the literature, two principal problem-solving methodologies for solving fleet-related, energy-related, or combination of both have been used by different researchers. These two effective methods are mathematical modeling and simulation. For solving a problem with the aid of mathematical modeling and exact procedures, when the size of the problem gets bigger for saving time and computation effort, researchers move towards employing heuristic algorithms. Simulation is the other method that has gained popularity in this field, especially agent-based simulation. Development of simulation has gone to the point that a specific software for solving large-scale transportation problems, called Multi-agent Transport simulation (MATSim), is introduced to the literature.

In the following tables, some of the works reviewed in this chapter with their problem-solving methodology are depicted. Since transportation problems can very case study dependent, the case study region is also included in the summary tables. Tables 2-1 show studies considering only the fleet management aspect, and table 2-2 shows problems only focusing on the fleet's energy management. Table 2-3 summarizes works considering fleet management and energy management of a shared transportation providing service. Even though a considerable number of researchers take into account both fleet and energy-related problems, many of them do not solve these two types simultaneously, and they optimize each part separately. As a result, research focusing on both fleet and energy aspects of a shared mobility service is relatively sparse. This essay aims to study different scenarios for a small service provider while considering both energy and fleet at the same time.

Another shortfall in literature is the travel pattern simplification. Most of the works conducted so far do not consider the actual path that a vehicle needs to take to perform a trip. In this study, however, with the help of the GIS map implemented in the software surface, the actual path that a vehicle would take in the real world to fulfill a trip is considered.

Table 2-1. List of articles for fleet management problem

Authors	Year	Type of Vehicle	Region	Solution
Pavone et al.	2012	AV	_	LP
Correia & Antunes	2012	AV	Lisbon - Portugal	MIP
Fagnant & Kockelman	2014	AV	Based on Austin-Texas-USA	ABS
Spieser et al.	2014	AV	Singapore	Mathematical Modeling
Boyaci et al.	2014	AEV	Nice - France	MILP
Rigas et al.	2015	EV	Washington DC, USA	MIP - Heuristic
Bsaybes et al.	2017	AV	An industrial area	Mathematical Modeling
Levin	2017	AV	_	Linear optimization model
Levin et al.	2017	AV	Austin (down town)	Linear Routing Optimization
Liu et al.	2017	AV	Austin-Texas-USA	MATSim
Horl	2017	AV	Based on Sioux Falls City	MATSim
Alam & Hbib	2018	AV	Halifax, Canada	Microsimulation
Winter et al.	2018	AV	Arnhem, Netherlands	Simulation
Babicheva et al.	2018	AV	Saclay, France	Mathematical Modeling
Hyland & Mahmassani	2018	AV	Manhattan grid network	ABS
Hörl et al.	2019	AV	Zurich, Switzerland	MATSim

Table 2-2. List of articles for energy management problem

Authors	Year	Type of Vehicle	Region	Solution
Bischoff & Maciejewski	2014	EV	Mielec, Poland	MATSim
Brendel et al.	2018	EV	_	Simulation
Yi et al.	2018	AEV	New York City, USA	Simulation

Table 2-3. List of articles for both fleet and energy management problem

Authors	Year	Type of Vehicle	Region	Solution
Hafez et al.	2001	EV	Paris - France	MILP - Heuristic
Awasthi et al.	2011	AEV	_	Simulation
Wang & Cheu	2013	EV	Singapore	Mathematical Modeling - Heuristic
Zhang et al.	2016	AEV	_	MILP with Binary Variables
Chen et al.	2016	AEV	Based on Austin-Texas-USA	Discrete-time ABS
Biondi et al.	2016	EV	Netherlands	Stochastic Optimization
Scheltes & Correia	2017	AEV	Delft University, Netherlands	ABS
Kang et al.	2017	AEV	Ann Arbor -Michigan - USA	Simulation
Bauer et al.	2018	AEV	Manhattan, New York, USA	ABS
Farhan & Chen	2018	AEV	Based on Austin-Texas-USA	Discrete-time ABS
Iacobucci et al.	2018	AEV	Tokyo - Japan	Simulation/Optimization algorithm
Leob et al.	2018	AEV	Austin, Texas 6-county region	MATSim
Zhang et al.	2019	AEV	Shanghai, China	Mathematical Modeling - heuristic
Bauer et al.	2019	EV	New York & San Francisco, USA	ABS
Iacobucci et al.	2019	AEV	Tokyo - Japan	MILP
Zhang et al.	2020	AEV	San Francisco Bay Area, USA	ABS (BEAM)

2.8 Summary

In this chapter, a review of the literature on emerging shared mobility services using novel technologies of autonomous vehicles and electric vehicles is proposed. Due to these technologies' benefits, our urban transportation will be drastically changed in the near future. Several varieties of employing different types of vehicles in the fleet are addressed. Since the scenario of employing autonomous electric vehicles in a shared fleet seems to be the most beneficial and can make up for former shortfalls, in our future work, we tackle the fleet and energy management of SAEVs.

Chapter 3:

3 Modeling Approach

3.1 Introduction

"Essentially, all models are wrong, but some are useful" (George Box, 1987)

Following chapter 2, it is decided to model a fleet of SAEVs because of the resulting benefits that both autonomous and electric vehicles bring to the table. In this chapter, the general idea behind the proposed model is introduced, and with the help of Unified Modeling Language (UML), different aspects of the model are discussed.

3.2 One-way SAEV Fleet Selection

The service-providing fleet that is studied in this writing comprises electric and autonomous vehicles so that they can cover each other's shortfalls and benefit from the advantages of both of these technologies. Providing a shared transportation service requires high utilization of fleet vehicles in order to be profitable. In this regard, according to Loeb et al. (2018), employing EVs for the fleet is more advantageous since EVs have lower energy and maintenance costs in comparison with conventional ICEVs therefore, their high utilization is less costly. Another reason for using EVs is their environmental advantage, such as zero GHG emission, less noise pollution, the possibility of employing renewable energy sources for providing the needed electricity. Employing AVs is also beneficial for fleet providers due to the omission of human driver salary.

3.3 Unified Modeling Language (UML)

UML, as Rumbaugh, Jacobson, and Booch, its developers, describe it, is a general-purpose modeling language for capturing different aspects of a software system. The aim behind modeling a system from various aspects is:

- So that people besides the developers can understand the model
- To show the design of the system
- To be able to browse into different sections of the system
- To maintain the system towards arriving updates
- To control the information

UML addresses the system's static structure by considering different role-playing discrete objects in the system and their interactions. It also models the dynamic behavior of those objects through time. Since its developers intended to model the broad applications of software systems, it cannot be small. However, they also aimed to make it as simple as possible to be applicable by more people (Rumbaugh et al., 1999).

Several ULM diagrams are introduced to catch the service-providing system's dynamic and dynamic aspects, each having its characteristics and benefits. Throughout the following subsections, some of the most useful UML diagrams are introduced, and the relevant diagrams for the case of a SAEV service-providing fleet are created.

3.3.1 Class Diagrams

The general format of the diagram is such that a three-sectioned rectangle represents each class. The top section contains the class's name, the middle section contains the class's attributes, and the bottom section contains its methods. Also, the type of relationship between all the classes apparent in the diagram is modeled. Four different types of relationships are depicted with the following considerations.

Inheritance: An open arrow represents this type of relationship. When a child class is connected to a class by the inheritance relationships, it gets all the attributes and methods of its parent class. This type of relationship helps us not repeat the same attributes and methods for each subclass and instead just write it once in the parent class.

Association: A simple line is drawn from one class to the other with the type of association written on top to show such a relationship.

Aggregation: When a subclass is connected to a parent class with a line ending with an open diamond, this represents the aggregation relationship. An aggregation relationship is a type when the subclass can exist outside of a parent class.

Composition: This is the opposite of the aggregation relationship, and the subclass cannot exist outside of the superclass. Meaning if the parent class is destroyed, the subclass is also going to be destroyed automatically. This type of relationship is depicted using a line ending with a closed diamond.

One more demonstration by UML class diagrams is called multiplicity: the number of each class per the other class it is connected to, which includes an actual number or a possible range. Visibility is also another feature of UML diagrams that is shown by inserting different signs before each attribute or method. The most used signs are plus and minus signs. The plus sign shows that an attribute or method is public; therefore, other classes can reach it. On the other hand, the minus sign shows that an attribute or method is private and can only be reached within the same class.

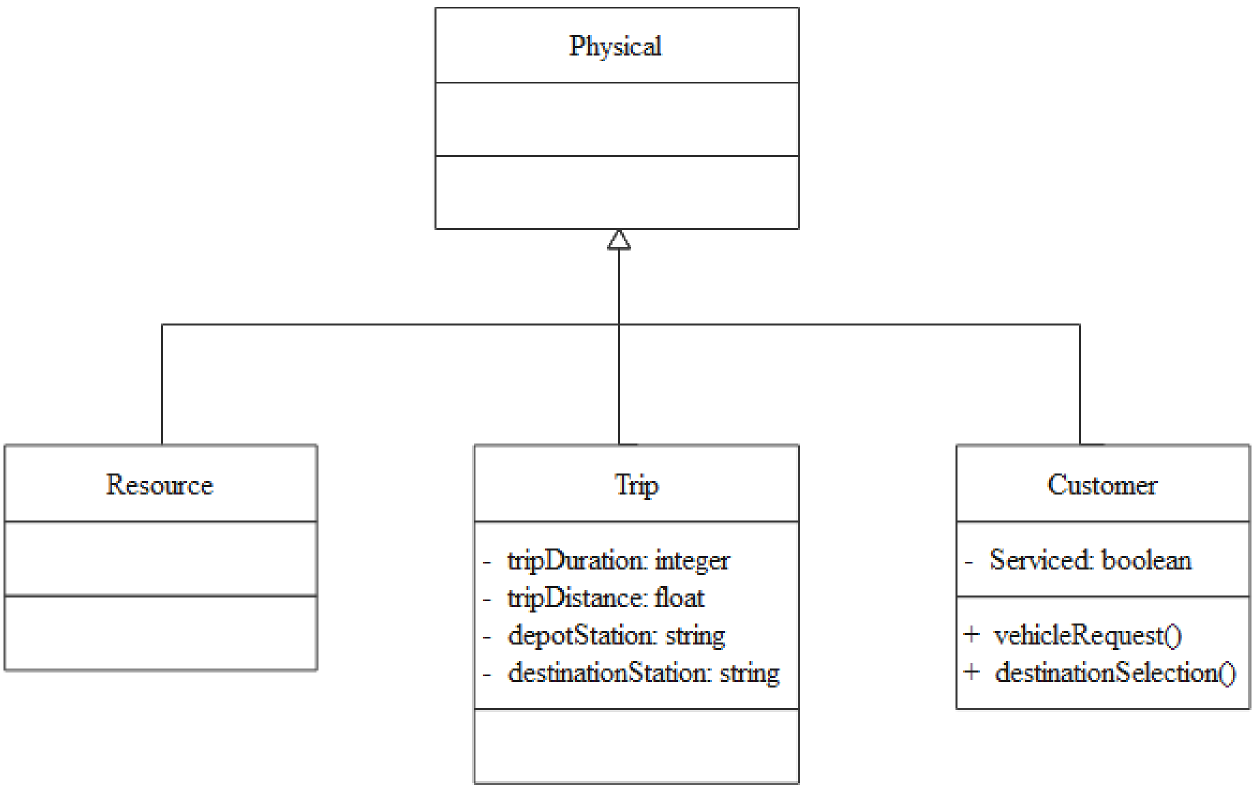

Figure 3-1. Physical Class Diagram

With class diagrams, the relationship between different parts of the proposed model in this thesis is depicted, and each class's attributes and methods are mentioned. In figure 3-1, all the physical classes involved in the model have been introduced: resources, trips, and the customers. Since resources are a broader subject, it is depicted separately as a class diagram with more details, but trip and customer class are explained in the following paragraphs.

In the simulation software, time is considered a discrete value containing integer numbers; thus, the trip duration can only be reported with integer numbers. The actual distance that a vehicle needs to move to fulfill a trip is considered as a whole number of kilometers. For each trip, there needs to be a depot station where customers will take the vehicle and a destination station to leave the vehicle.

One of the model assumptions is that customers will take a vehicle if and only if one is already available in their depot station due to the relatively short-distance nature of the trips. If there were no available vehicles in that station, the customer would decide to walk or use other public transportation methods. Whether a customer is serviced or not is showed by a boolean variable. A customer acts by requesting a trip and selecting a destination.

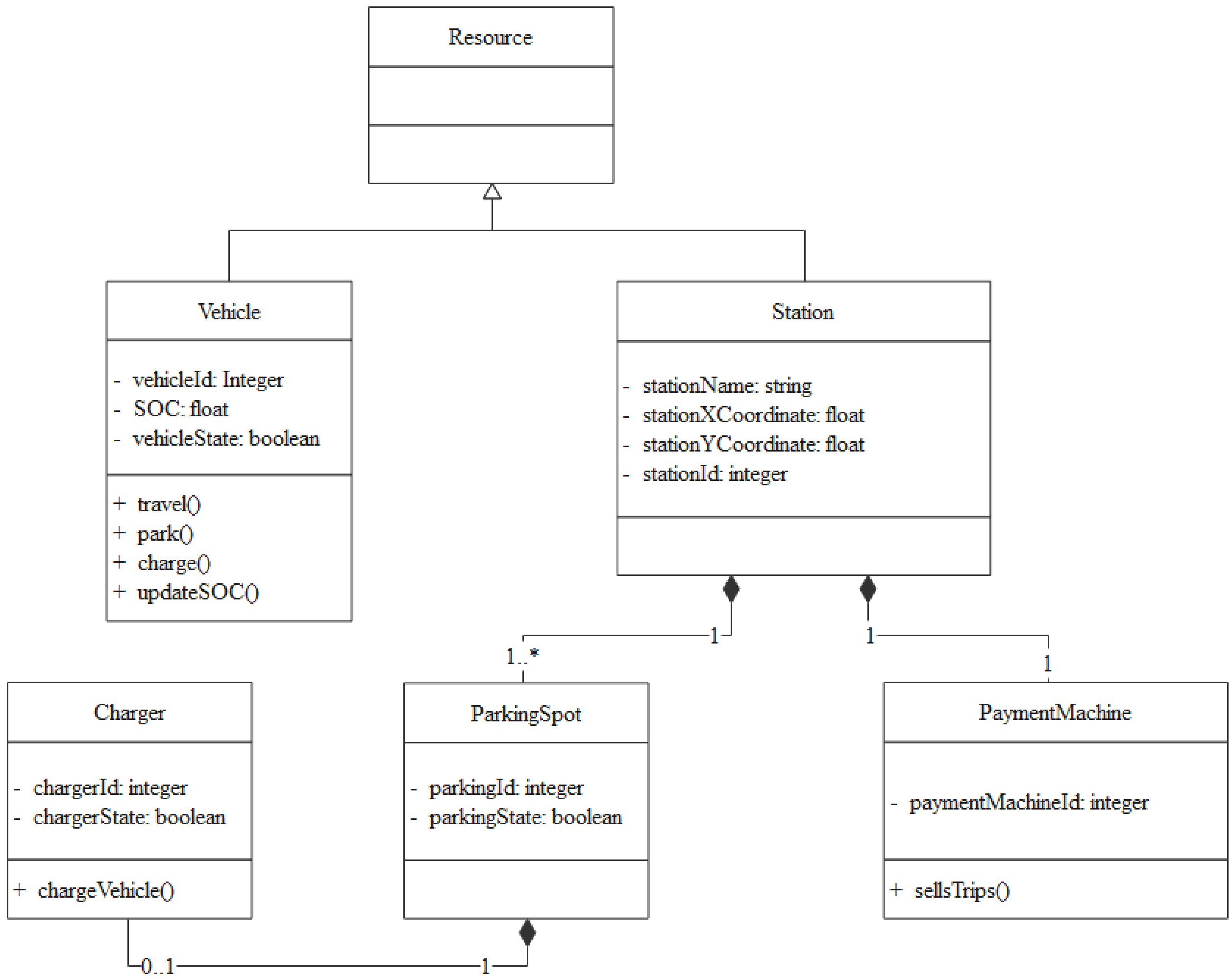

Figure 3-2. Resources Class Diagram

Figure 3-2 shows the class diagram of the resources. Two major types of resources are vehicles and stations. Each vehicle has a separate ID to be distinguished from other vehicles. SoC is the state of charge for the vehicle. When a vehicle makes a trip, its SoC will be reduced depending on the trip's duration and distance. The vehicle needs to update its SoC at the end of each trip so that if it goes below a threshold, it could plug itself into a vacant charger. Vehicles could be available or unavailable to give service to customers. A vehicle is unavailable when it is being charged, or it is completing a trip.

There is a composition relationship between the parking spot and payment machine and the station, meaning that there will not be a parking spot and payment machine if there were no stations. Each station has one payment machine that customers can use to pay for their trip. They can also use an app to make the payment similar to the system used by BIXI, the bicycle sharing service, in Montreal. Each parking spot can be equipped with a charger or not.

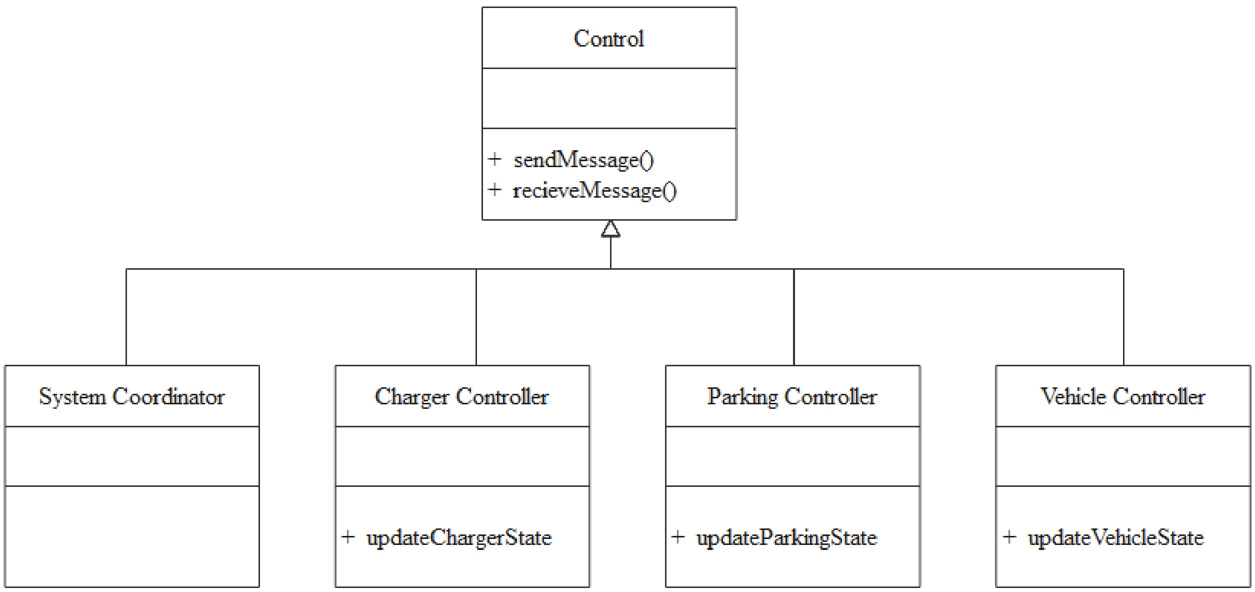

Figure 3-3. Control Class Diagram

The control class is composed of operations controlling different parts of the shared transportation system. The first one is the system coordinator, who is in charge of controlling all the operations. The charger controller which keeps track of each charger's state is the second one. When a vehicle reaches a station with an SoC below the critical threshold, the charger controller verifies if there is any vacant charger in that station or not. In the case when there is no vacant charger, the vehicle is sent off to another station. The third one is the parking controller, who determines the number of vacant parking spots. After completing a trip, when a vehicle arrives at the station, it needs to park itself at one of the available parking spots. If there were no parking spot available, the parking controller would send the vehicle to another station. The last part is the vehicle controller, which updates each vehicle's state in the system. When there is a demand for a vehicle in a particular station, this controller will verify vehicle availability to provide service to the customer.

3.3.2 Activity Diagram

The activity diagram simply shows the actions done to achieve an ultimate goal. Basic shapes used in an activity diagram are as follows:

- A closed circle shows the beginning of the flow of activities.
- Actions are inside the rounded-off rectangles.
- Decide stages are shown by a diamond.
- Arrows show the flow of the process.
- A solid bar showing the possibility of all the outcomes.

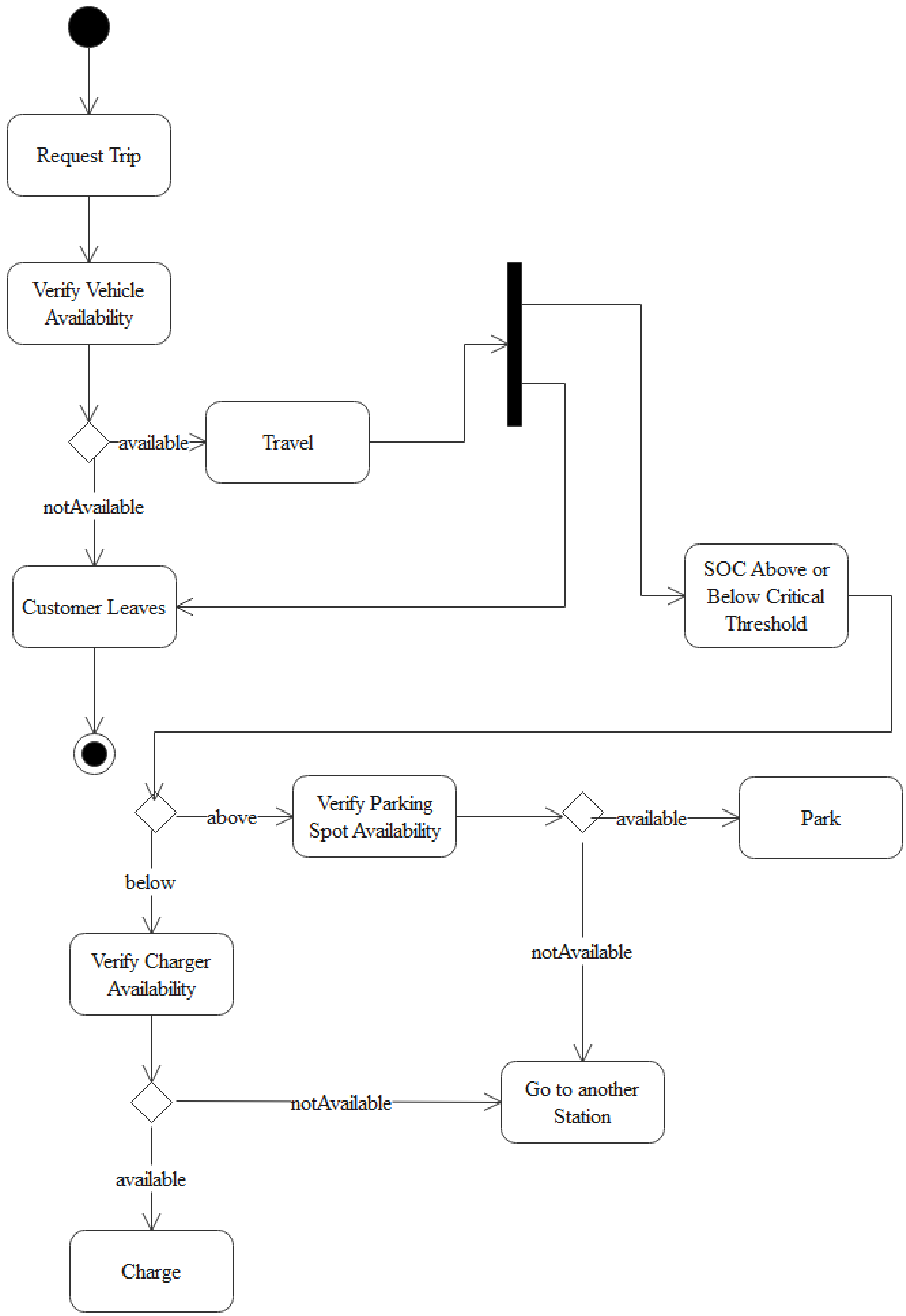

Figure 3-4. Making a Trip Activity Process

Figure 3-4 shows the activity diagram for making a trip. As can be seen from the diagram, the system will be triggered when there is a customer's request for a trip. After that, the system will verify any available vehicle in the customer's depot station. If there was an available vehicle, the travel could start. Otherwise, the customer will leave the system. This consideration is intending to simplify the model. Since trips are short, it is supposed that the customer has no intention of waiting for an available vehicle and will choose other transportation methods. After completing the trip, the customer will leave the system. At this moment, the vehicle checks its SoC. If its SoC is below a critical threshold and there exists a vacant charger at the station, it would plug into the charger. Otherwise, it will be sent to other stations. When the SoC was above the critical threshold, and there was a parking spot available at the station, it will park itself. Otherwise, it will be sent to another station be parked. A vehicle will be charged until reaching full battery or reaching a predefined percentage of charge and will not interrupt its charging process to provide service to a customer. According to battery longevity policies, such a decision is made to avoid extra costs due to battery replacement and be more eco-friendly, not to produce dead battery waste. A vehicle stays in a parking spot until it is summoned by the system to fulfill a trip request.

3.3.3 Use Case Diagrams

In a use case diagram, the aim is to depict a system and its outside actors' possible interactions. It also shows what type of service the system is providing. A use case diagram comprises four different parts: system, actors, use case, and relationships. The first part, which is the system, is shown by a big rectangle in the middle of the diagram. Actors, as the second part of the diagram, are demonstrated by stick figures. Depending on the actors' type, they could be placed on the right or the left side of the system's rectangle. There are two types of actors: primary actors and secondary actors. Primary actors are drawn on the system's left side, and they initiate interacting with the system. Secondary actors are depicted on the right side of the system, and they make a corresponding reaction to the initial action of the primary actors. The third part is the use cases shown with an oval shape, and they represent different actions carried out inside the system. The last part is the relationships which are differentiated by drawing several kinds of lines. Different types of relationships are as follows:

Association: A simple solid line shows this type of relationship.

Include: It is depicted by a dashed arrow starting from the base use case and ending in an included use case. On the dashed arrow, the word included is written in double chevrons. This type of relationship states that in order for a base use case to be executed, the included use case should be executed as well.

Extend: This type is also shown by a dashed arrow from the extended use case towards the base use case with the word extend written in double chevrons on top of it. The extended relationship indicates that when a base use case is executed, and some specific conditions are met, the extended use case also becomes executed.

Generalization: This type is similar to the inheritance relationship in a class diagram introduced earlier in this chapter. Like the inheritance, an open arrow shows this specific type of relationship between actors and between use cases.

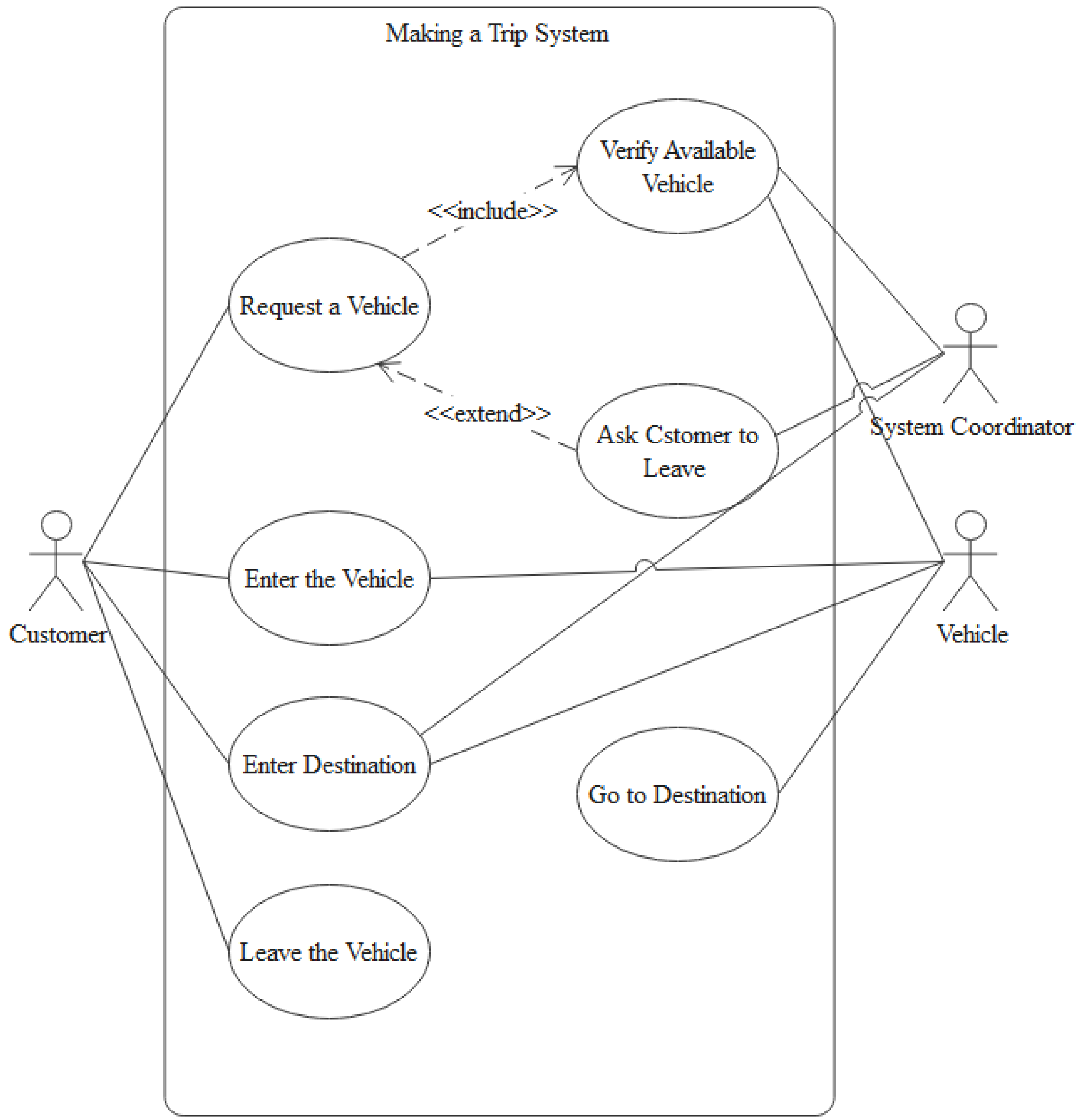

Figure 3-5. Making a Trip Use Case Diagram

Figure 3-5 shows the use case diagram for making a trip system. There are three actors involved in this process, the customer who will initiate interacting with the system by requesting a vehicle and the vehicle and system coordinator who react to this request. After a request is submitted, the system coordinator will verify if there are any available vehicles in the depot station or not. If there was availability, a vehicle is assigned to pick up the customer. After entering the vehicle, the customer will choose any other station as their destination, and by reaching that station, he/she will leave the vehicle. In the case where there was no available vehicle, the customer is asked to leave the system.

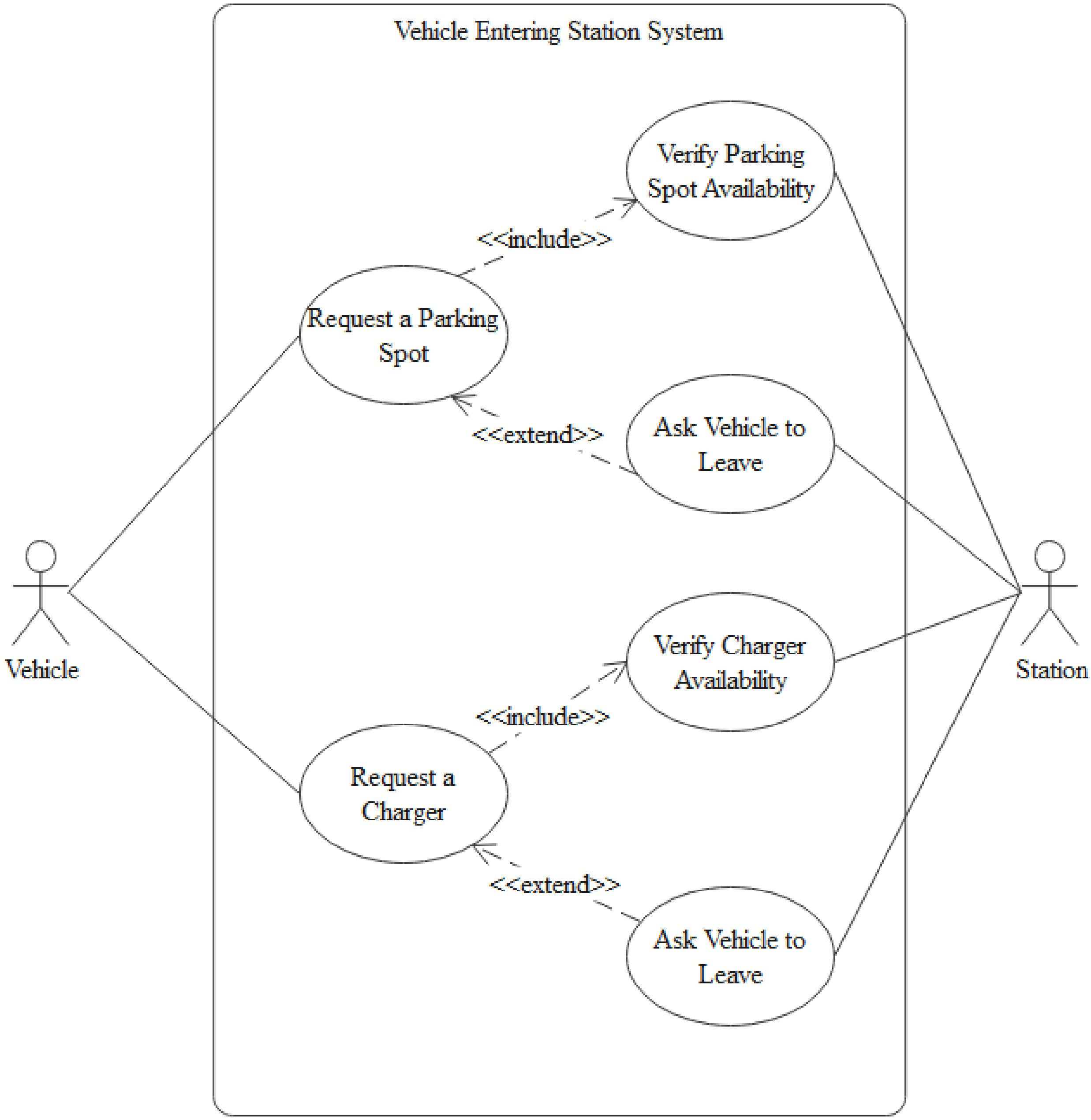

Figure 3-6. Vehicle Entering Station Use Case Diagram

Another use case diagram shown in this section is entering a station after fulfilling a trip for a vehicle which is depicted in figure 3-6. A vehicle needs to be parked when it has finished its travel. To do so, it will verify with that specific station if there are any available parking spots. If there were availability, it would be parked in that station. If not, it should leave this station and go to another one. Also, at the end of each trip, the vehicle needs to check its SoC. If the SoC were below a critical threshold, it would need a charger. Like the parking process, it will check the charger's availability. If no chargers were available in the destination station, it would leave for other stations. If not, it will stay in the same station and start the charging process.

3.3.4 Sequence Diagram

Sequence diagrams are made to show the interactions between different objects in the order of the timeline within a system. In other words, they simply show the sequence of events. Rectangles represent objects within the system, and a stick figure represents the actor who is outside the system but interacts with it. Vertical dashed lines represent the lifeline of the actor or objects. Each interaction between different parts is shown by a solid arrow labeled corresponding to the action or message. A returned message is depicted by a dashed arrow and responds to one object or actor to the other (a message from the receiving object back to the requesting). An alternative frame is used whenever there is a possibility of two sets for turns of events. An alternative frame is a rectangle divided in half to show two or more mutually exclusive options. Activation boxes are shown by a narrow rectangle how much and when an object is active. Figure 3-7 shows the sequence diagram related to the proposed SAEV transportation system.

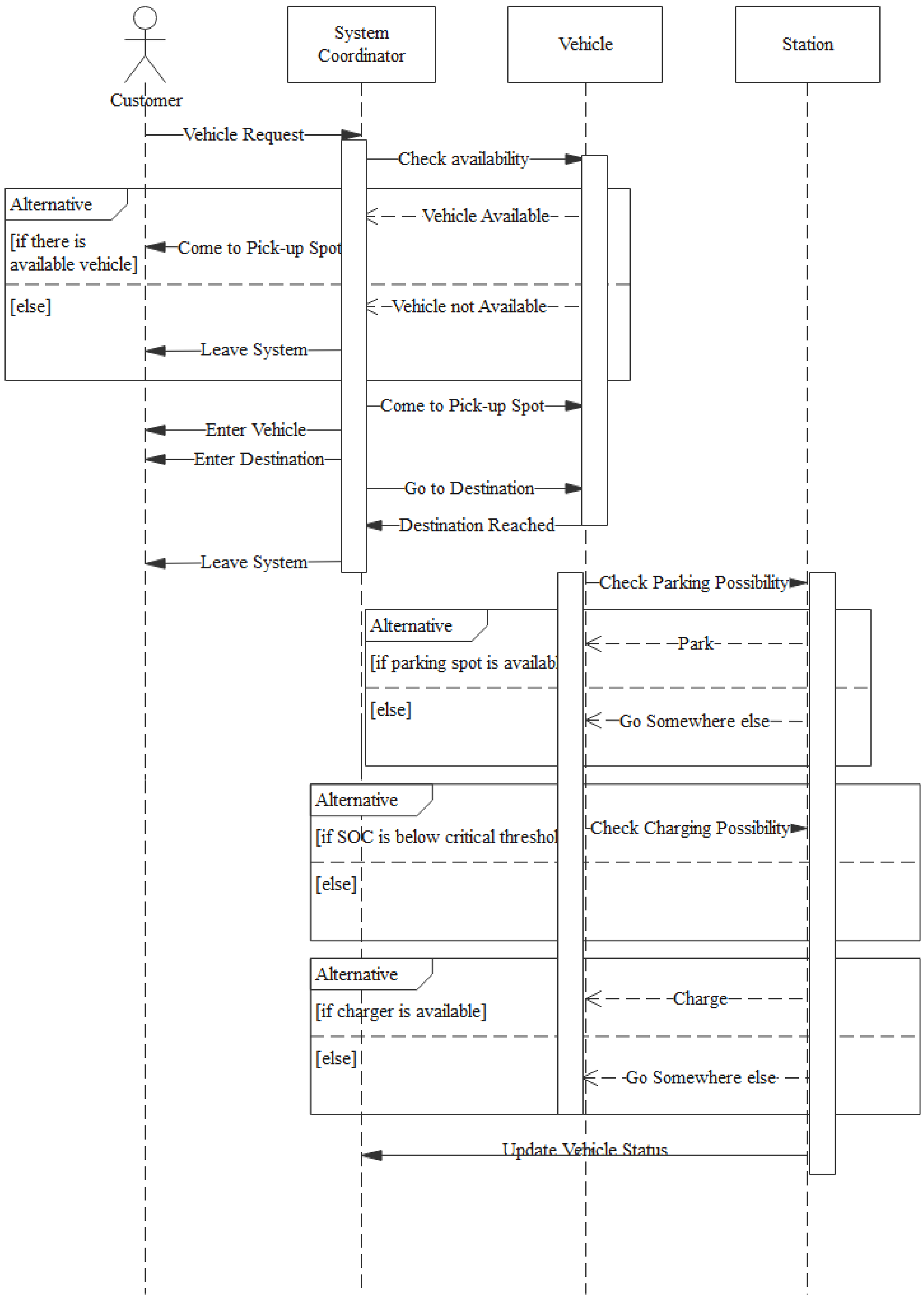

Figure 3-7. Making a Trip Sequence Diagram

3.4 Summary

In this chapter, with UML diagrams' help, different aspects of the proposed SAEV service providing system are introduced. Via the class diagram, different categories in the system and their attributes and methods in addition to their relationship towards each other are shown. The activity diagram depicted all the activities and their possible outcomes regarding such a transportation system. The use case diagram is utilized generally for introducing all the agents composing the system and their interactions with each other. A sequence diagram is used to show the possible actions concerning their relative time of occurrence in the system.

Chapter 4:

4 Agent-Based Simulation of a Fleet of SAEVs: Case Study of Olympic Park

4.1 Introduction

To study and capture the specifications of the SAEV fleet described in chapter three, with the help of the NetLogo simulation software, the simulation model of the case study of Montreal's Olympic Park is developed. In this chapter, the simulation model and the software used for creating it will be explained in more detail.

4.2 Agent-Based Modeling (ABM)

As some describe it, Agent-based modeling is "the third way of doing science," which enables us to study complex systems that were formerly impossible to study without simplifying assumptions. Take economics as an example that could only be studied with the assumption of a perfect market, long-run equilibrium, and homogenous agents. Now with the help of ABM can be analyzed with more realistic assumptions. Economic is not the only field that employs ABM. Many natural trends such as the spread of diseases, market penetration, extinction of civilizations, and many more can also be analyzed through ABM tools Macal and North (2009).

Although only recently being put into everyday use, ABM was not a new topic in science. As an example, the segregation model by Thomas Schelling, which was introduced in 1971, was one of the earliest ABMs presented. The idea behind the segregation model is that individuals can only tolerate a specific number of people other than their nation as their neighbors. If this number exceeds a threshold, they tend to move to other neighborhoods. Schelling used a checkerboard to represent such a model and solved the problem without the aid of a computer. Using a checkerboard for such a big problem was very time-consuming, but nowadays, this problem can be solved within seconds with computers' computation capacity.

The basic idea behind agent-based modeling is to study a system based on its agents' behavior and interactions, which means that each entity's behavior will determine the system's future state. In contrast with other methods which try to model the system as a whole, in agent-based modeling, the concept is to model the system from the ground-up.

Steps to develop an ABM to capture characteristics of a system can be summarized in three folds. First, it is essential to distinguish all the agents which form that system. Second, the specific attributes and variables relative to each of the agents must be defined. The final step is to imbed the interactions between the agents or between agents and their environment. To carry out the first step, it is crucial to know the major characteristics of an agent to distinguish them. According to Macal and North (2009), there is still unsettled debate on the exact definition of agents. However, based on their studies, they suggest the following definition capture significant characteristics of an agent:

- An agent is a definable unit with its attributes, decisions, and rules and is distinguishable from the environment or other agents.
- An agent can interact with other agents.
- An agent is autonomous in its behavior, meaning that it interacts with its environment or other agents in a manner that is governed by the agent itself and is not decided by a leader.

Some other characteristics mentioned by Macal and North (2009) which may or may not be seen in an agent are as follows:

- An agent may interact with its environment.
- An agent may be goal-orientated and continuously change its behavior and policies to reach an optimal situation.
- An agent may have memory and change its behavior based on past experience.

4.3 Simulation Software

Although an ABM can be developed simply using available programming languages such as Python or Java, using specifically designed ABM toolkits could considerably help the modeler. Since some relevant codes and commands are already embedded in these toolkits, they can save time and energy for model development. Several of such toolkits available are Repast, Swarm, MANSON, AnyLogic, and the one used in this study, NetLogo. While most of the toolkits mentioned above are usually used in massive projects, NetLogo is primarily used for educational purposes due to its simplicity; nevertheless, it can build problem-solving projects as well. It is a free toolkit designed to implement multi-agent-based modeling problems authored by Uri Wilensky in 1999. The Center for Connected Learning and Computer-Based Modeling at Northwestern University oversees NetLogo’s development. Being a multi-agent simulation toolkit makes it possible to define different agents' groups at the beginning of the code. The language used in this toolkit is called Logo, which was first introduced by Seymour Papert et al. in 1969. The logo is a single agent language designed to make the school students interested in computer programing. A robot called a turtle acted as the agent to make the toolkit interesting for students. The term turtle is still used for calling the agents in NetLogo.

Syntaxes used in NetLogo are similar to standard English, making it easy to write the commands even for people without much coding experience. Although, this simplicity does not make it unsuitable for complicated research projects. NetLogo has provided service to many research projects so far. The variety of sample problems available in NetLogo's library in different fields such as biology and economics, and mathematics makes it suitable for researchers with different backgrounds. The library option of NetLogo, a pool of example models, can help the researcher explore the proper use of code commands and develop their desired code based on a code example.

Another exciting feature about NetLogo is its extensions, enabling the modeler to bring different commands written in Java or other languages to their NetLogo model. As an example of these extensions with already available examples in the NetLogo, the library is the GIS extension. This extension makes it possible to bring GIS data to the model platform. As an example, figure 4-1 shows the map of all roads for the area near Montreal Olympic Park, which is brought to the interface of NetLogo using GIS extension.

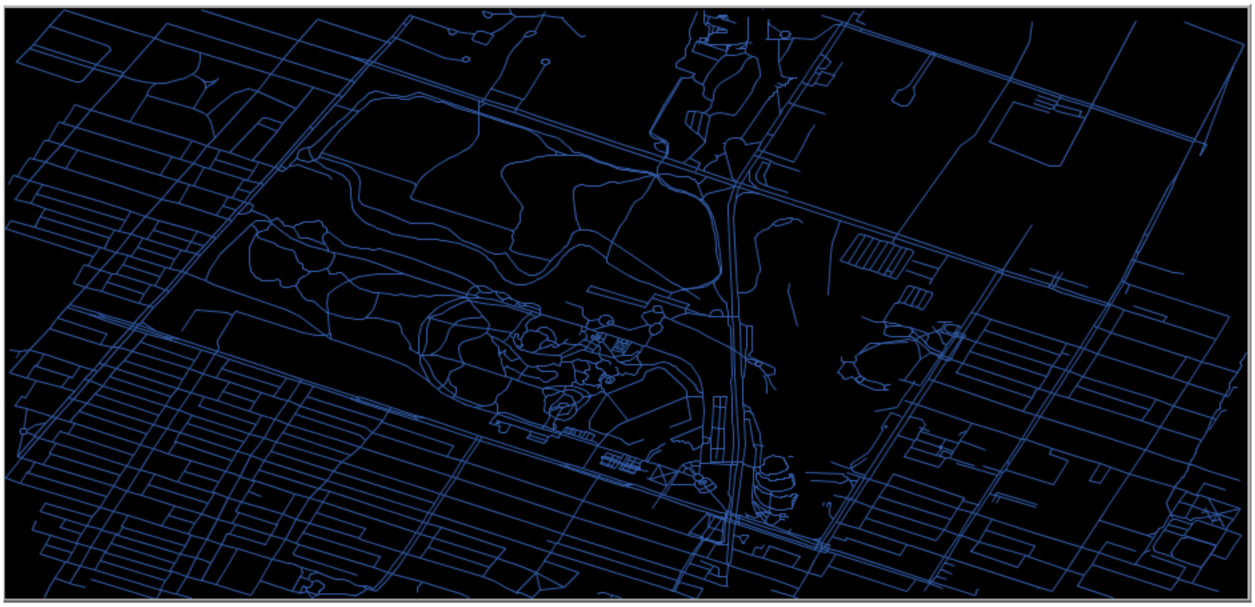

Figure 4-1. Map of roads around Montreal Olympic Park area brought in NetLogo interface

In NetLogo, each problem space is comprised of three different tabs. The first tab is called interface, which is where the model can be visualized, and with the use of different bottoms such as sliders, the initial values can be altered. Also, plots can be drawn to show the changes of a value during a run. The second tab is called Info, which is mostly space for the model developer to clarify their model's fundamental aspects in plain English. The last tab is the Code which is the principal place for modeling the problem using computer commands.

In the interface tab, a graphical environment exists which demonstrates the modeled system. This graphical environment is comprised of a black squares grid, named patches, and triangles representing agents. Aside from the turtles and patches, another character in the NetLogo models is called the observer, who runs and oversees the whole model. Shapes and colors can be changed according to the modeler's desire. This visualization capacity makes the model more understandable and helps the developer in the task of model verification, which means that one way of finding possible mistakes in modeling is to run the model and follow its visual development to see if it complies with expectations or not. Figure 4-2 shows the interface tab of the model created in this study. As it can be seen from this picture, parameters such as the number of vehicles, vehicle range, number of parking spots, critical threshold, maximum charging time, and demand probability can take different values using sliders. Also, reporters are added to this model to report the desired outputs for each time instant of the run. Turtles are asked to "put their pen down," which means to leave a trace wherever they go. This will significantly help examine if turtles are following the correct route or is there any deviation.

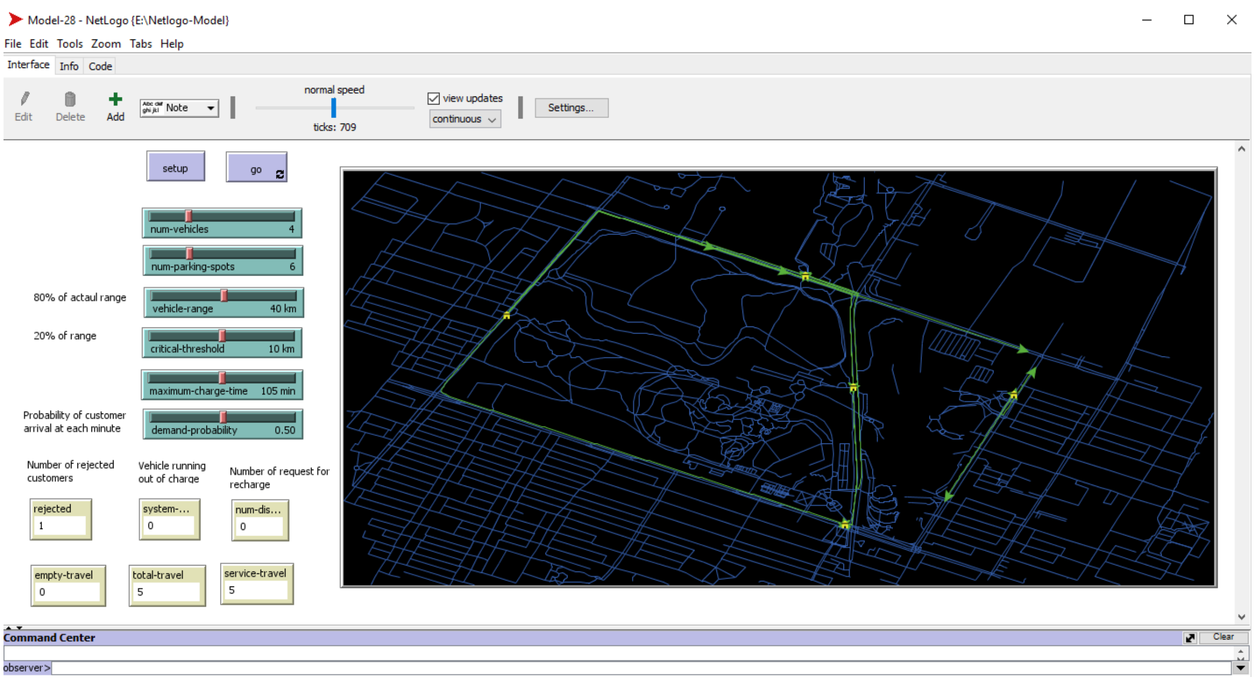

Figure 4-2. Interface tab of the developed model of this study

4.4 Agent-Based Modeling of Montreal Olympic Park Using NetLogo

As already mentioned in chapter three, this study tries to model a fleet of fully autonomous electric vehicles giving one-way transportation service for short distance travels. This full automation means that not only can the vehicles drive without any interference of a human driver, but also, they can charge themselves by using a robotic arm. A customer will randomly arrive at one of the stations at a time step. If by that time there was a vehicle available at the station for giving service, he/she will complete his/her travel by moving to another station. However, suppose there was no available vehicle at that moment. In that case, it is considered that the customer will leave the system immediately, and this unsatisfied departure would be counted through the simulation as a metric for calculating the performance.

Five stations on the streets surrounding the Montreal Olympic Park are considered to model such a fleet. Each of these stations is close to a tourist attraction or a metro station, and they are placed at a reasonable distance from each other to cover all of the park angles. The first station is on Pierre-de Coubertin Avenue near the Viau metro station. This consideration will help the travelers who take public transportation to access other parts of the park by using the shared fleet. The second station is on Viau street near the Golf Complex, and the third one is on Rosemont Boulevard near the entrance for Frédéric back Tree Pavilion. The fourth one is on route 125, close to the park's main entrance. This station is also easily accessible through the Pie-IX metro station. With these four stations, we can cover all the four streets surrounding Olympic Park. One last station is also added to Sherbrooke street's cross-section and the road leading to Montreal Insectarium. By adding this last station, more central parts of the park are reachable. Figure 4-3 shows the placement of these five stations on a picture taken from google maps.

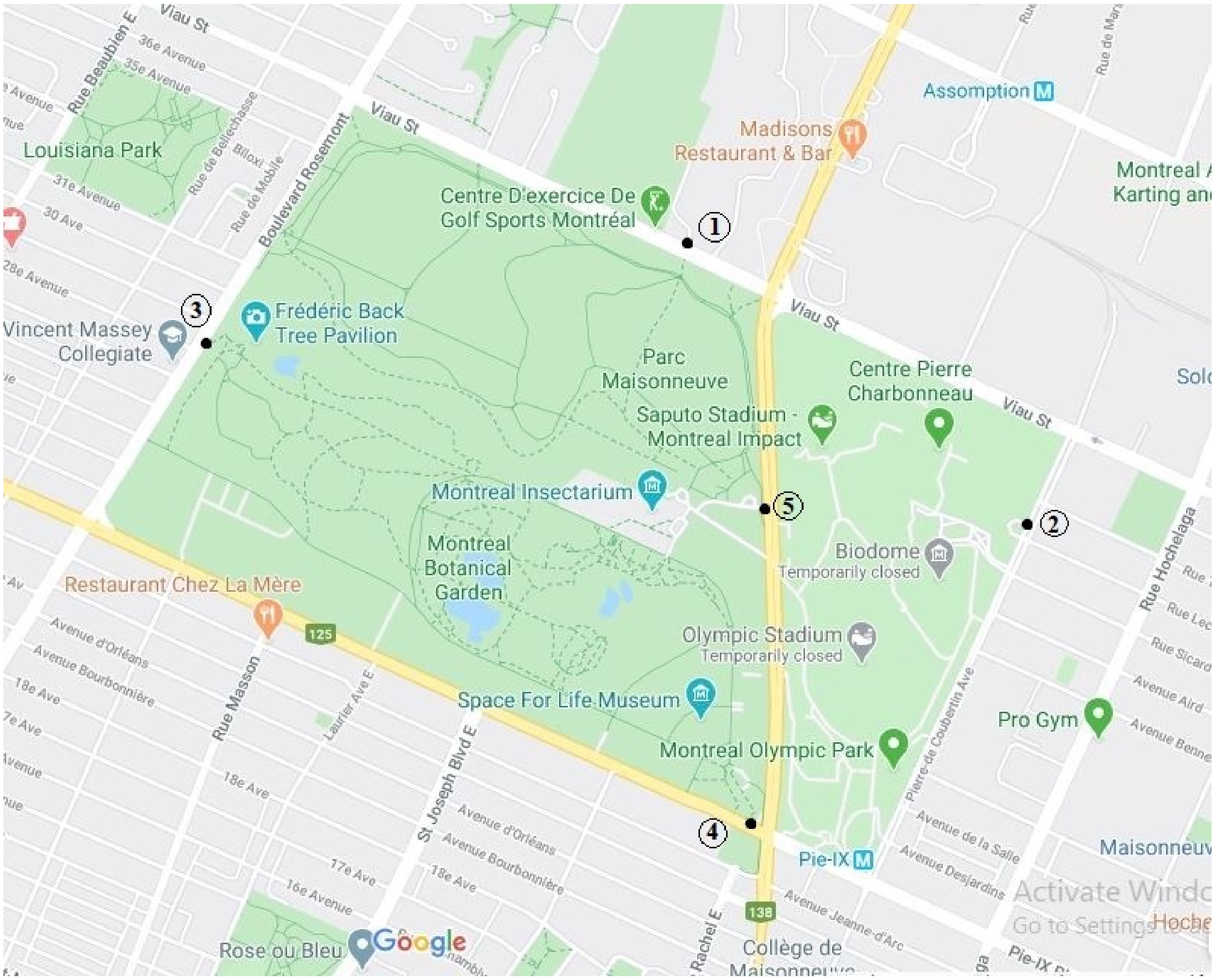

Figure 4-3. Candidate stations

As earlier explained, in one-way ridesharing, one consideration that improves the system is rebalancing. Since demand in some stations could be more significant than others, eventually, some stations would have an overpopulation of vehicles while others would suffer from vehicle shortage. This case usually occurs when the travel of the customers follows a pattern. For example, the pattern is shaped from the travel from areas near cities towards downtown in the morning's rush hours and vice versa in the evening. In order to maintain the quality of service providing, vehicles from overpopulated stations should be brought to stations with vehicle deficiency; in this study, rebalancing is not considered in the model due to the following reasons. First, the demand of the park visitors is entirely random and does not necessarily follow a pattern. Because, while someone is visiting one side of the park, another one could be vising the opposite side. As a result, the situation of becoming unbalanced is not as common as pattern-shaped travels. Secondly, considering the number of parking spots relatively small to the number of vehicles, automatically a station cannot be overpopulated since there is no parking place. Hence, they have to move to the next nearest station with available parking spots. The effect of the number of parking spots is later on studied in chapter 5.

4.5 Model Assumptions

It is evident that not a single model is able to capture all the aspects of a real-world system, and eventually, some simplifications must be made. In this regard, the simplification assumptions considered in the model are mentioned as follows:

- The traffic flow is neglected; hence, vehicles choose the fastest path possible for going from one station to another.
- It is considered that each parking spot is equipped with a charger. Meaning that if a vehicle was able to park in a station, it could defiantly charge as well.
- Since there was no data for the customer arrivals available, three different customer arrival rates are considered to shape high, medium, and low demand scenarios.
- Vehicles move at the constant speed
- Only 80% of the battery capacity is modeled so that charging with a constant rate would be valid. To protect batteries from early degradation, manufacturing companies design the charging process so that batteries get charged with a constant speed for charge values below

80% of the range. Nevertheless, towards the end of the charging process, meaning from 80% until 100% fully charged, the charging process will take more time with a shifting charging rate.

- Data acquired for range and charging time is based on hybrid vehicles and not EVs. For such a short distance model, we are only interested in using low-range vehicles for the fleet, but available EVs in the market have higher ranges required for this study.
- When a vehicle is making a trip to go to the nearest station with an available parking spot, there is no guarantee that the parking spot will still be available by the time it reaches that station. Because there is a chance that another vehicle has finished its travel and parked itself at the vacant parking spot, in such a situation, the system manager would again find another nearby station with a vacant parking spot.
- The SoC of vehicles remains intact when they are making a trip and will not continuously decrease during the trip. After a vehicle reaches its destination, the distance it has traveled will be deducted from its SoC.
- Customer arrival to each of the stations is sequential, and the case when two people arrive simultaneously at the same or different stations is neglected.

4.6 Model Execution

In this section, an explanation of the NetLogo model's steps is introduced to better clarify the model (the actual code written for the model can be found in the appendix section of this essay).

Before all, a simple definition for the objective function and constraints is stated. Considering that the service provider's objective is to minimize the number of its fleet size while maintaining good service quality, the optimization problem can be written as follows:

$$\text{Min } 5V \qquad (4.1)$$

$$s.t. \qquad \frac{TC - RC}{TC} \times 100 \geq 90 \qquad (4.2)$$

While V is the number of vehicles at the beginning of the simulation at each station and TC and RC are the number of total customers and number of rejected customers, respectively. The tolerance for the percentage of rejected customers is set to 10%, which means 90% of the customers will be serviced. Now, the general process of the simulation model can be gathered in few steps as follows:

Step1: At the beginning of the code, the global and local variables, as well as different sets of agents, are defined. Since NetLogo allows having multi-agents in the simulation, different groups of agents can be defined at the beginning of the code.

Step 2: In this step, all the commands that create the simulation environment are introduced. The observer would click on the "set up" bottom on the Interface tab, and all the following commands will be carried out in the subsequent order.

First, all the former runs so far, if any, will be deleted. Then, the GIS data of the desired case study, which is the surrounding streets of Montreal Olympic Park, is added to the code. Later, wherever there exists an angle on one of the studied roads, an agent is created at that specific location. These agents are called road vertices, and they are used to give direction to the vehicles and specify the location of stations.

Even though the GIS data is already implemented in the model, on the display section, nothing appears. Another piece of code is written to display the map of the roads. The last three lines of the "set up" command give direction for the station, vehicle and, parking spot implementation. Stations, vehicles, and parking spots are created as three different agent sets, and then they are moved to the predefined locations. The locations are found by assigning five specific road vertices as the desired spots for station implementation. In NetLogo, each agent has a unique identity number named the "who" number. The identity numbers of road vertices are used to find the correct locations. The number of vehicles and number of parking spots at each station can be altered in each run using a slider on the "Interface" tab of the model. The other parameter that can be changed regarding the vehicles by using another slider is their battery range.

Step 3: In this step, the simulation will be started by clicking on the "go" bottom. In the last line of this section commands, the word "ticks" is written to add a timer to the model. In NetLogo models, time is counted in discrete steps named "ticks." By considering the constant speed of 30km/h for each vehicle, every minute is calculated to be approximately equal to 42 ticks.

A trip is generated whenever there is a demand for it. Customers are randomly created to model the random arrival of customers in the real world by using the following condition.

$$if\ random\ \left(\frac{42}{p}\right) = 0 \qquad (4.3)$$

This condition will hold when the random whole number generated in the range $[0, \frac{42}{p} - 1]$ would be equal to zero. Each tick, a whole number in this range, is generated, and whenever that number is 0, one customer will be created. 42 ticks will be divided by p, the probability of customer arrival for each minute, to generate the expectancy of having one arrival after each $\frac{42}{p}$ ticks. When a customer is created, it will randomly move to one of the stations. Then, if there was any vehicle available, another station is chosen as the destination, and the vehicle would start its trip. On the other hand, if no vehicle were available to give service to the customer, he/she would leave the system, and this departure would be counted via formula 3.

$$RC = RC + 1 \qquad (4.4)$$

A vehicle is available when it is not fulling a trip. Hence it is parked in a station and is not getting charged as well. To demonstrate the vehicle availability, they are presented by the color green. Two other colors of white and red are also possible choices for each vehicle, each referring to different situations. Suppose there were no parking spots available in that destination. In that case, the system controller checks to find the closest station with a parking spot available and sends the vehicle to that station. A vehicle that is making a trip to park itself and is not giving service to a customer will change its color to white, and this trip will be counted as an empty trip. After the vehicle is parked, it will change color to green again if it has enough charge. Each time a vehicle is parked, it checks its SoC. If the charging level was below a predefined threshold, it will change color to red and stay idle until reaching 80% of its charging capacity. Then it will change color to green and stays in the station until being assigned to give service to a customer.

For the vehicle to move on the right path, it is crucial to know the depot and destination stations since each path from one station to another is separately defined. The vehicle will start its trip by facing the next road vertex in its path and moving towards that one step in each tick. When it is close enough to the road vertex meaning their distance is less than one patch, it will move to that road vertex location. When the vehicle reaches the road vertex, it will face the next road vertex and moves towards that. This process will continue until the vehicle reaches the last road vertex, which is the destination. Using this creative method of movement is one of the contributions of this essay which lets not only to go on the right path but also to move meaningfully with regard to the time step.

When a vehicle reaches the destination at the end of each trip, the kilometer it passed will be deducted from its SoC. Then the code would verify if the number of parking spots is bigger than the number of vehicles at the station. If this was true, the vehicle has a spot to park; hence it will end its trip. Otherwise, the code will determine which station has a parking spot available at the moment. The search for a parking spot will start by the nearest station to the current station to comply with the fuel efficiency policy. Then the vehicle would move towards the candidate station. By the time it reaches the station, it will go through the whole process of SoC adjustment and parking spot availability check. Because it takes some time to reach the station, during which the parking spot availability might have changed, this check needs to be done every time.

When a vehicle is finally parked, it will go through the charging process if its SoC is lower than the critical threshold. The vehicle will change its color to red and stop giving service to customers until reaching 80% of its range which is the maximum SoC for the vehicles during the simulation. A variable named "charge-time" is assigned to the vehicle, and for every tick of the model, it will be increased by one unit. To calculate the right time for the vehicle to be detached from the charger and become available again to give service formula 4.5 is used.

$$\frac{CT_{0-80} \times 42 \times (SoC_{80} - SoC)}{SoC_{80}} \qquad (4.5)$$

In this formula CT_{0-80} is charging time for a vehicle to reach 80% of its range while starting at 0 charge level. This parameter also should be changed based on the range capacity of the vehicle by using a slider. SoC_{80} is the state of charge of the vehicle when it is charged at 80% of its range, and SoC is the state of charge at the moment when the vehicle was parked. With this model specification, there is the possibility that if a vehicle wanders around a lot to find a parking spot or if the critical threshold for SoC is not high enough, the vehicle might run out of charge. The model counts the number of out-charged vehicles and reports that as the system error. If a system error is encountered in the simulation run, the critical threshold should be changed.

4.7 Summary

In this chapter, an introduction to the concept of ABM and its application is introduced. Later the methods and toolkits available for simulating an agent-based problem are enumerated, and an explanation of the toolkit utilized in this study is provided. Then the case study of Montreal Olympic Park is introduced. Finally, model assumptions and the method for writing the simulation code are provided.

Chapter 5:

5 Numerical Results and Discussions

5.1 Introduction

In this chapter, numerical experiments are introduced to test the model's validity. Three different demand probabilities representing high, medium, and low frequency are implemented in the model to compare the results under different customer arrival circumstances. The best scenario possible for the fleet owner can be dragged from the information provided in this section

5.2 Scenario Generation

Various scenarios are generated by considering different values for the number of vehicles, vehicle range, and level of demand. Seven value is considered for the number of vehicles at each station at the beginning of the simulation. These values are such as 2, 3, …, 8. The number of parking spots at each station also varies with the number of vehicles in a manner that for each scenario, the number of parking spots at each station is equal to the number of vehicles at each station plus 2. In order to test the effect of the fleet performance's vehicle battery capacity, five different possible vehicle ranges are assumed from 30km until 70km augmenting in 10 increments. For each of these ranges, different charging times and critical thresholds are assigned.

Charging times vary by many factors, one of which is the size of the battery. The charging times acquired for this study are estimates from different hybrid vehicle models available. In this manner, a vehicle with a 30km charge range needs 150 minutes to charge from 0 to 100. When the range is increased by ten increments, the charging time will as well increase with 30 minutes increments.

As already mentioned in chapter 4, one of the model assumptions is to consider vehicles at 80% of their charging rate at the beginning of the simulation. Also, vehicles will stop the charging process whenever they reach their 80% of range. According to Loeb et al. (2018), half of the charging time is spent to bring the battery from 0 to 80% of its range, and the other half will bring it from 80% until fully charged. Since ranges below 80% are modeled, only half of the charging time is seen in the simulation. Another factor that changes with the range of the vehicle is its critical threshold. To extend the battery's life, the critical threshold is considered to be 20% of the battery range.

By considering three demand levels with the probability of customer arrival in each minute equal to 0.5, 0.2, and 0.1, and all the aforementioned inputs for numbers of vehicles and battery range, a total of 105 different scenarios are generated. Table 5-1 shows all the possible values for these parameters. The model is run for 100 replications for each scenario, and values are averaged through the replications. The number of rejected customers, performance (percentage of the number of serviced customers divided by all the customers), number of empty travels, and the number needed charging service are reported as the outputs. System error is also added to the model as a controlling criterion that simply checks to see if a vehicle will run out of charge during a simulation or not. Since distances between stations are not considered high, low levels for critical threshold would result in almost no complete discharge of the battery, and the 20% of SoC threshold in almost all the cases would be big enough. The only time that this strategy seems to be unreliable is for 30km range vehicles; hence in that case threshold is considered to be 8 and not 6.

Table 5-1. Possible values for different parameters

Demand Level	0.1	0.2	0.5				
Number of Vehicles at Each Station	2	3	4	5	6	7	8
Number of Parking Spots at Each Station	4	5	6	7	8	9	10
Total Number of Vehicles	10	15	20	25	30	35	40
Total Number of Parking Spots	20	25	30	35	40	45	50
Vehicle Range (km)	30	40	50	60	70		
80% of Range (km)	24	32	40	48	56		
Critical Threshold	8*	8	10	12	14		
Charging Time (min)	150	180	210	240	270		
Half Time (min)	75	90	105	120	135		

* The actual threshold is 6, but with that value, there is a chance of running out of charge

Without seeing the actual results, it is logical that when the range is increased or the number of vehicles increases, the service quality will be improved. On the other hand, the provided service cost will also increase due to excess in equipment's number and cost. As a result, there is always a trade-off between service quality and system costs. Although the price of vehicles is not considered in this study, it is evident that the more the number, the higher the costs. Also, each vehicle's price depends on its battery capacity and qualifications, which means that a higher range battery could result in a higher vehicle price.

5.3 Results

The simulation is run for a weekend day, considering the demand would be highest compared to the rest of the week. The simulation's length is 10 hours for the hours that the park is operating; translating into ticks, it is equal to 25200. The results of the number of rejected customers, performance, and the number of discharged vehicles are discussed in the following subsections.

5.3.1 Rejected Customers

In this section number of rejected customers under different scenarios is studied. Figures 5-1, 5-2, and 5-3 show the number of rejected customers relative to the number of vehicles in the fleet for demand probability of 0.5, 0.2, and 0.1, respectively. These graphs also contain the information of each range capacity such that each line represents one level of vehicle range.

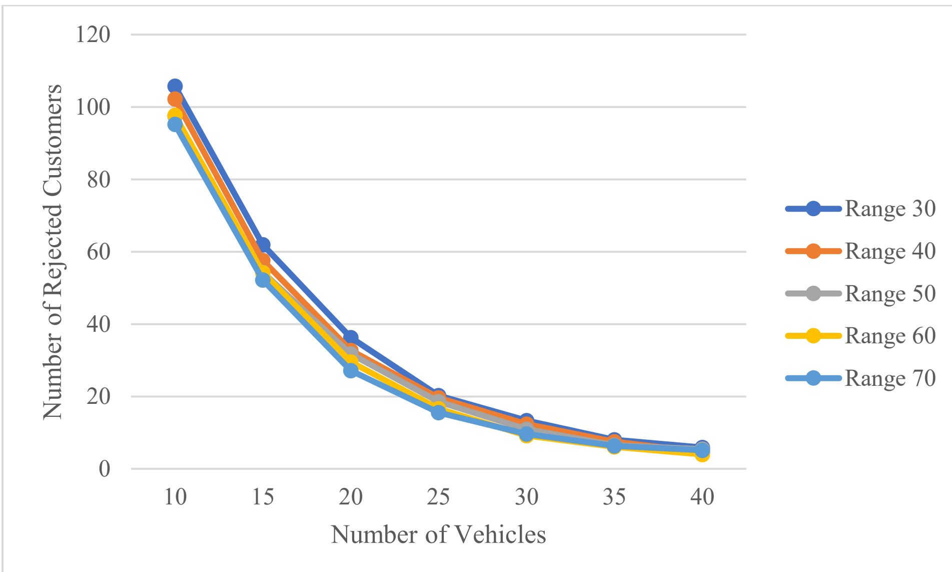

Figure 5-1. Number of rejected customers vs. number of vehicles for demand probability of 0.5

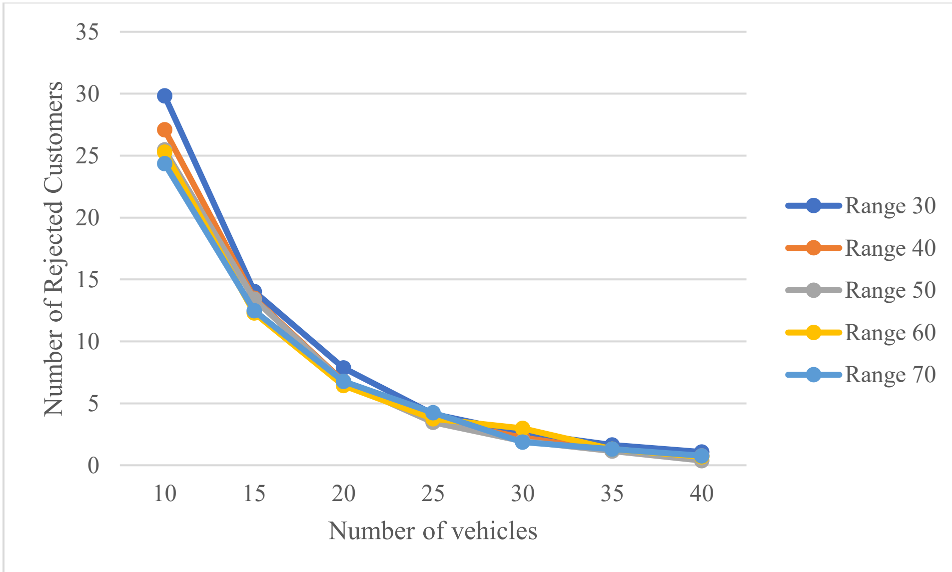

Figure 5-2. Number of rejected customers vs. number of vehicles for demand probability of 0.2

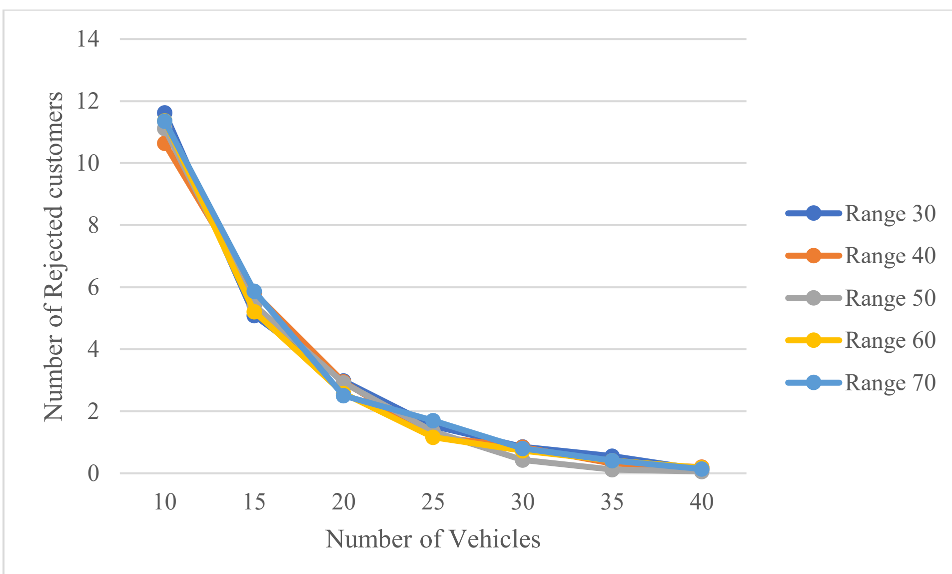

Figure 5-3. Number of rejected customers vs. number of vehicles for demand probability of 0.1

By comparing graphs showing the number of rejected customers with different fleet sizes and range capacities, it is concluded that factors that have an impact on the number of rejected customers are mostly fleet size and demand probability. While different ranges do not show a significant impact since in all three graphs, the line representing different vehicle ranges lay very close to each other. The number of rejected customers seems to decrease in an exponential manner with the increase in the number of vehicles. As a result, having a fleet comprised of 20 to 25 vehicles seems the right choice. Since range has no significant impact on the number of rejected customers, the case when the range is equal to 50km is chosen as the range for a base scenario. With this consideration, when the fleet has 25 vehicles, the number of rejected customers is equal to 18.52 when demand probability is 0.5. With 20 vehicles in a case when the demand probability is 0.2 and 0.1, the number of rejected customers is equal to 6.70 and 2.92, respectively. An acceptable limit for rejected customers is not a predefined number and depends on the service-providing company's policies.

5.3.2 Performance

To make the results less sensitive to the demand probability, another indicator is introduced named the performance. Performance is defined as the percentage of serviced customers divided by all the customers. Performance is also not considerably affected by the range of vehicles, and mostly fleet size and demand probability have an impact on it

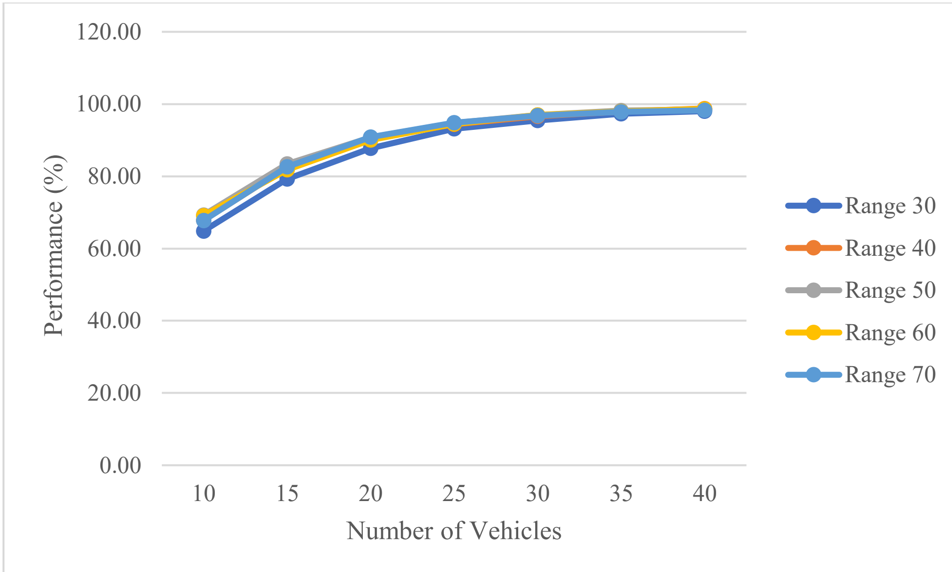

Figure 5-4. Performance vs. number of vehicles when demand probability is 0.5

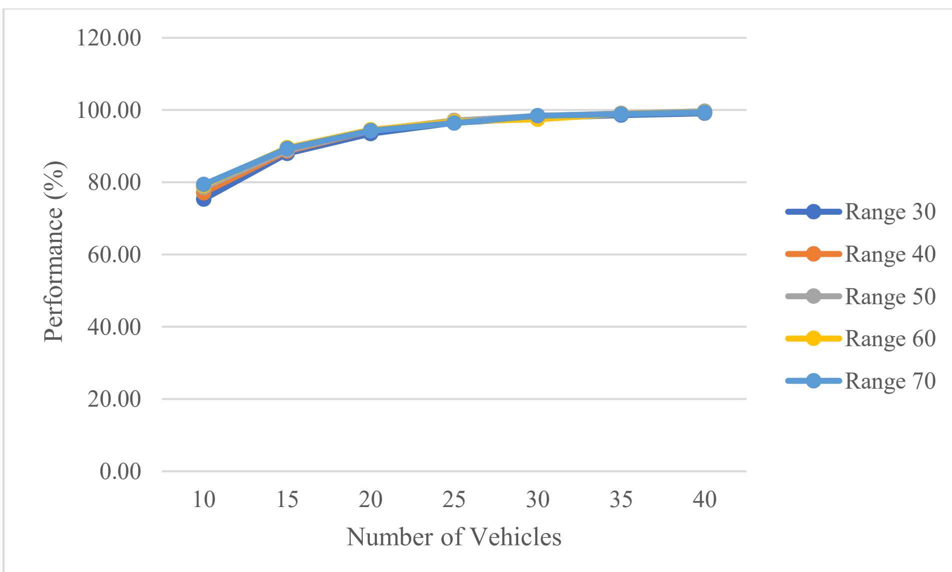

Figure 5-5. Performance vs. number of vehicles when demand probability is 0.2

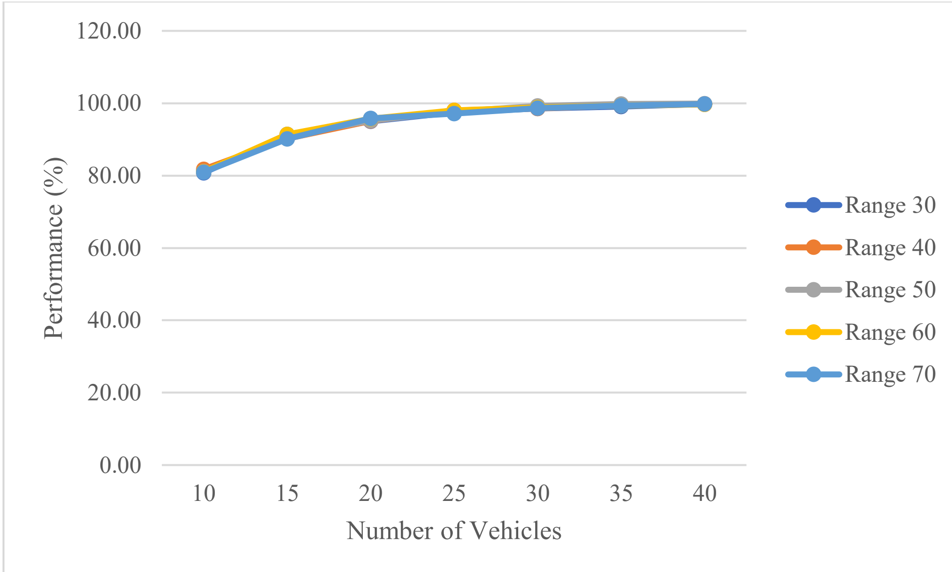

Figure 5-6. Performance vs. number of vehicles when demand probability is 0.1

As it can be observed from figure 5-4 to 5-6, - Another observation to conclude in this section is that when demand is high and fleet size is relatively small, the vehicle range's effect on the fleet performance is more particular. However, is not the case for higher lower demands and bigger fleet sizes as it can be seen lines defining different ranges almost overlap in such conditions.

5.3.3 Discharged Vehicles

So far, we have concluded the negligible effect of range on the number of rejected customers and overall fleet performance. In this section, the effect of the vehicle's range capacity on the number of times that a vehicle was discharged during a simulation run and needed to recharge itself is studied. Figures 5-7, 5-8, and 5-9 depict the number of recharging processes relative to vehicle size for different ranges under demand probability of 0.5, 0.2, and 0.1, respectively.

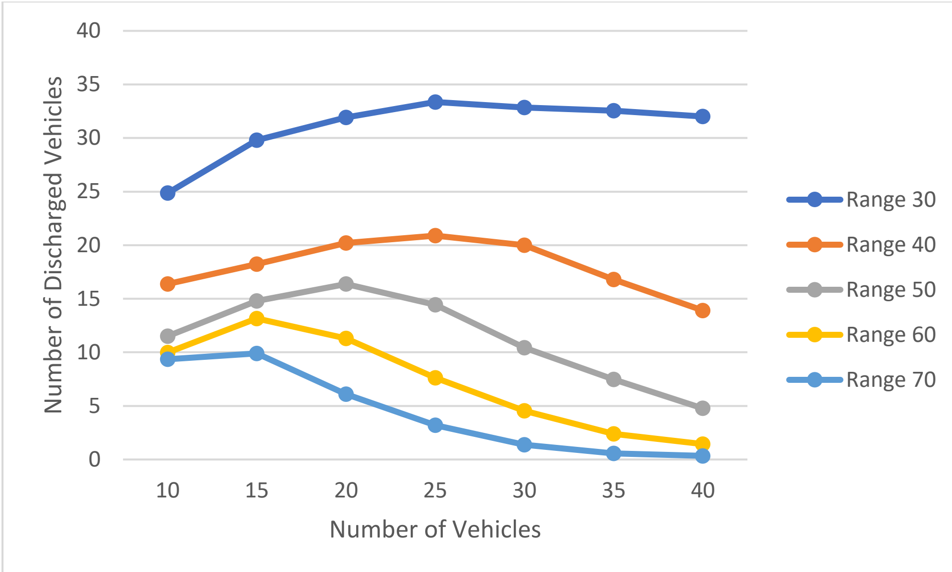

Figure 5-7. Number of discharged vehicles vs. number of vehicles when demand probability is 0.5

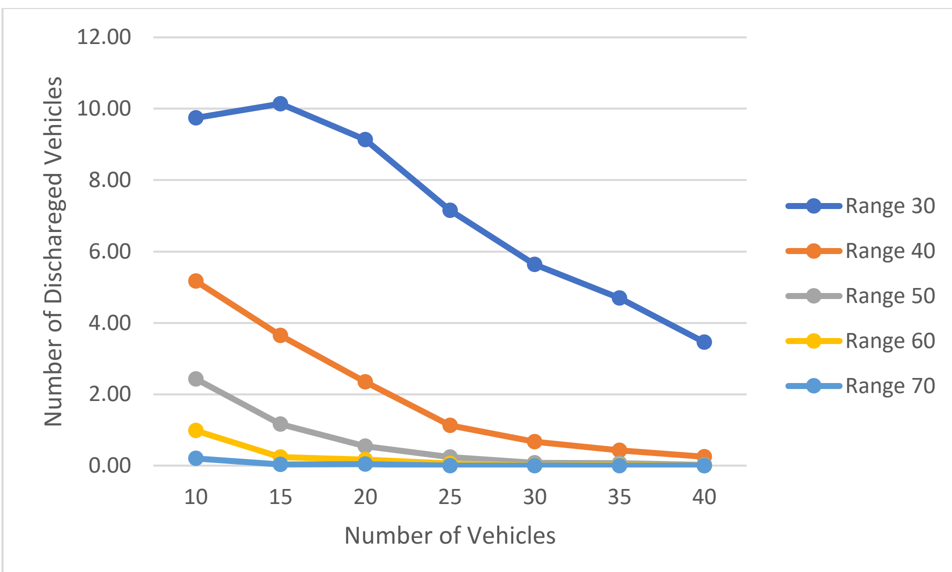

Figure 5-8. Number of discharged vehicles vs. number of vehicles when demand probability is 0.2

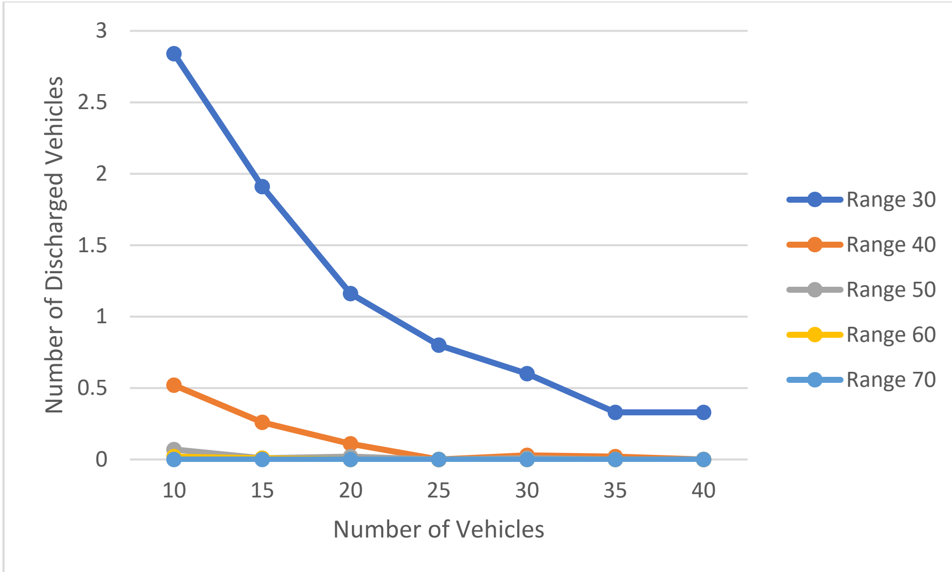

Figure 5-9. Number of discharged vehicles vs. number of vehicles when demand probability is 0.1

By comparing figure 5-7 to 5-9, it is concluded that demand level and the fleet size and battery range have a significant impact on the number of rechargers needed during a simulation run. When demand is low for ranges above 50km, there is almost no need for a charging process. For the case when demand is medium, vehicles with a battery capacity of 50km or above can provide service without having to recharge as little as two times or below.

5.3.4 Rebalancing Policy

As already mentioned earlier in this essay, the model built for this study does not provide a rebalancing but a semi-rebalancing one. When a station lacks vehicles or another station is too populated, vehicles do not move from the later station to the former. Nevertheless, the semi-rebalancing policy suggests limiting the number of parking spots. When the number of parking spots is relatively small to the number of vehicles, inevitably, vehicles need to move to other stations with an available parking spot and rebalance the system. Just like rebalancing, this strategy's downfall is that it could lead to more congestion and expenses due to empty travels.

This section studies the effect of different numbers for parking spots relative to the number of vehicles. Four different scenarios are introduced: the number of parking spots equals the number of vehicles, or each station has 1, 2, or 3 extra parking spots. Other factors concerning the problem are the number of vehicles and vehicle range set to 20 vehicles and 50km respectively. Hence, each experiment's number of parking spots is equal to 20, 25, 30, and 35. Figures 5-10, 5-11, and 5-12 show the number of rejected customers, the number of empty travels, and the number of discharged vehicles, respectively.

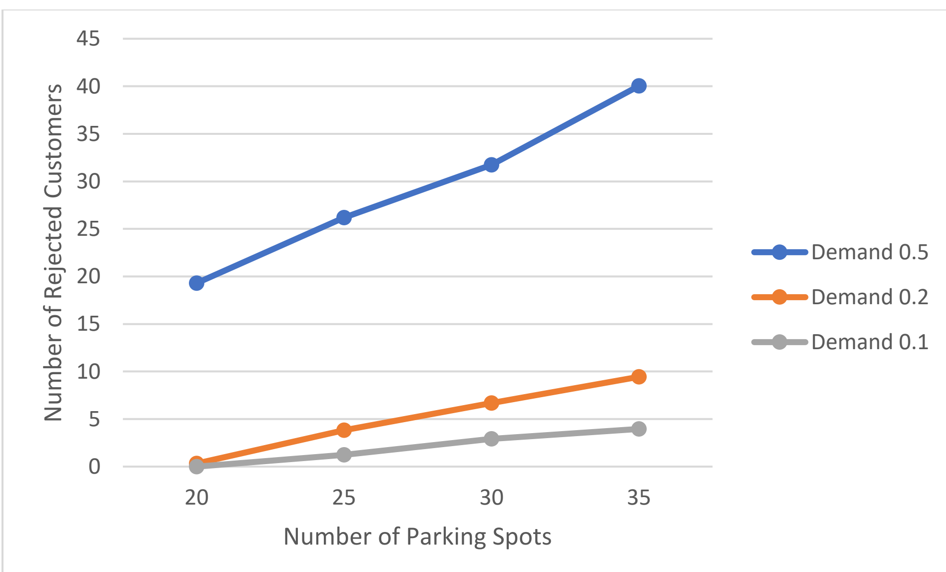

Figure 5-10. Number of rejected customers vs. number of parking spots when the fleet has 20 vehicles of 50km range

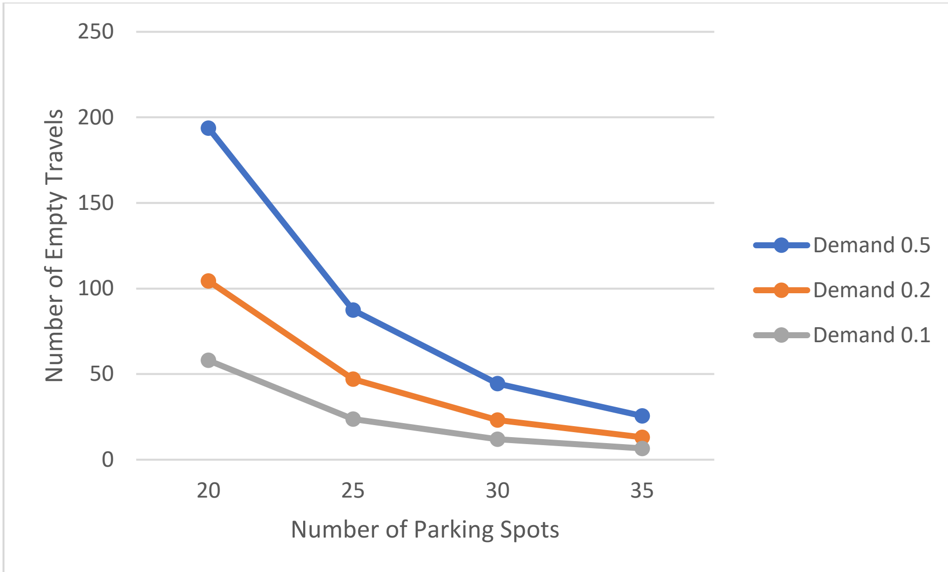

Figure 5-11. Number of empty travels vs. number of parking spots when the fleet has 20 vehicles of 50km range

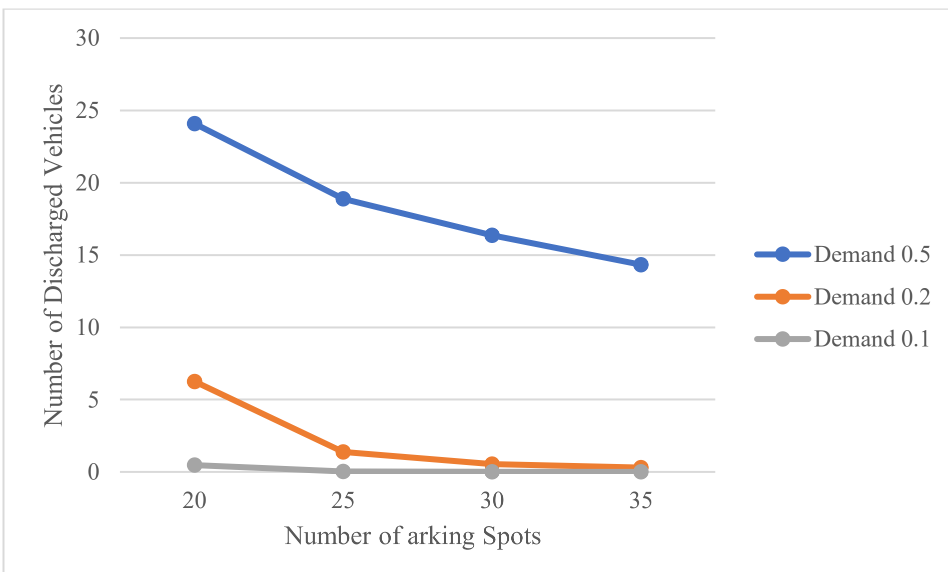

Figure 5-12. Number of discharged vehicles vs. number of parking spots when the fleet has 20 vehicles of 50km range

Observations show that with the increase in the number of parking spots, the number of rejected customers would increase. On the other hand, the number of empty travels and the number of requests for recharging would decrease. As much as the demand is high, the slope of these changes would be steeper.

5.4 Summary

This chapter introduces the numerical results to test the validity of the proposed model. First, 105 different scenarios are generated to represent various fleet sizes and battery ranges under high, medium, and low demand probability. The model runs each scenario for 100 replications, and the results are averaged through the replications. One semi-rebalancing policy is also tested to show the effect of different parking spots' capacity on the system's performance.

Chapter 6:

6 Conclusion and Future Work

6.1 Conclusion

This essay studies a fleet of shared autonomous electric vehicles from fleet and energy management aspects. NetLogo toolkit is used to develop the desired multi-agent simulation model. The area around Montreal Olympic Park is considered as the case study. The suggested model can either provide a transportation service to the customer or ask him/her to leave the system. It is also capable of capturing the parking and charging constraints of the SAEV fleet.

Under high, medium, and low customer arrival frequency, 10, 15, 20, 25, 30, 35, 40 fleet size and 30, 40, 50, 60, 70 vehicle range, 105 different scenarios are generated. Each scenario is run for 100 replications. The Numerical results suggest that a fleet comprised of 20 vehicles with a 50km range can meet the demand under all customer arrival scenarios with a 90% or above performance rate. While The number of vehicles in the fleet can exponentially affect the number of rejected customers and system performance, the range does not significantly impact the transportation service's performance. Later, the effects of a semi-rebalancing policy are tested. This policy aims to obtain the number of parking spots meaningfully as small as possible to increase the system's performance rate. In this regard, results of 12 scenarios with a 20 vehicles fleet size with 50km range capacity under all three different demand probabilities, and 20, 25, 30, 35 parking spots are gathered. The findings show that as much as the number of parking spots increases, the performance and number of empty travels decrease.

This model could act as a framework that can be adjusted to model a fleet for any other part of the world with a different road network. It only needs to bring the GIS data of that specific road network to the model's surface and define the station places. With path adjustments added to the code, another fleet in another part of the world can be studied. Also, by being one of few studies that consider the actual road network of the case study, this model could lead to more realistic results.

6.2 Future work

This section contains suggestions for future studies, which are as follows:

- Each station could have its demand probability. This model can be expanded such the possibility of two customers arriving simultaneously at two different stations would be probable.
- As mentioned earlier, the traffic flow is neglected in this study. For future work, the model can be developed in a way that considers traffic flow as well.
- One other factor that can help the decision-maker choose better fleet size levels, vehicle range and parking spots is cost. By adding this element to the model, it can demonstrate more meaningful results.
- This study's semi-rebalancing policy is to send the vehicles in need of a parking spot to the nearest station with an available parking spot at that moment. In future works, the fleet controller could suggest a station with the least number of vehicles to the vehicle searching for a parking spot.

- The results of the number of recharges needed during a 10-hour shift under all scenarios studied show that it does not exceed 35. As a result, implementing a charger for each parking spot is an unnecessary cost burden for the system. The future model could advance in this regard by only devoting a few chargers per station or considering a different location for charging purposes.

7 References

1. Alam, M. J., & Hbib, M. A. (2018). Investigation of the impacts of shared autonomous vehicle operation in Halifax, Canada using adynamic traffic microsimulation model. *Procedia Computer Science*, *130*, 496–503. Retrieved from https://reader.elsevier.com/reader/sd/pii/S0921800911001662?token=89A9FD1BAB070C7CC72797D15B37A6F8B82A4B32CAEBEA4D4DE4990193F26E3570CDC8B5E43AB29CC2704AF0041AD777
2. Awasthi, A., Chauhan, S. S., Parent, M., & Proth, J.-M. (2011). Centralized fleet management system for cybernetic transportation. *Expert Systems with Applications*, *38*(4), 3710–3717. https://doi.org/10.1016/J.ESWA.2010.09.029
3. Babicheva, T., Burghout, W., Andreasson, I., & Faul, N. (2018). The matching problem of empty vehicle redistribution in autonomous taxi systems. *Procedia Computer Science*, *130*, 119–125. https://doi.org/10.1016/J.PROCS.2018.04.020
4. Bauer, G. S., Greenblatt, J. B., & Gerke, B. F. (2018). Cost, Energy, and Environmental Impact of Automated Electric Taxi Fleets in Manhattan. *Environmental Science and Technology*, *52*(8), 4920–4928. https://doi.org/10.1021/acs.est.7b04732
5. Bauer, G. S., Phadke, A., Greenblatt, J. B., & Rajagopal, D. (2019). Electrifying urban ridesourcing fleets at no added cost through efficient use of charging infrastructure. *Transportation Research Part C: Emerging Technologies*, *105*, 385–404. https://doi.org/10.1016/J.TRC.2019.05.041
6. Biondi, E., Boldrini, C., & Bruno, R. (2016). The impact of regulated electric fleets on the power grid: the car sharing case. *2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a Better Tomorrow, RTSI 2016*.
7. Bischoff, J., & Maciejewski, M. (2014). Agent-based Simulation of Electric Taxicab Fleets. *Transportation Research Procedia*, *4*, 191–198. https://doi.org/10.1016/J.TRPRO.2014.11.015
8. Boyacı, B., Zografos, K. G., & Geroliminis, N. (2015). An optimization framework for the development of efficient one-way car-sharing systems. *European Journal of Operational Research*, *240*(3), 718–733. https://doi.org/10.1016/J.EJOR.2014.07.020
9. Brendel, A. B., Lichtenberg, S., Brauer, B., Nastjuk, I., & Kolbe, L. M. (2018). Improving electric vehicle utilization in carsharing: A framework and simulation of an e-carsharing vehicle utilization management system. *Transportation Research Part D: Transport and Environment*, *64*, 230–245. https://doi.org/10.1016/J.TRD.2018.01.024
10. Bsaybes, S., Quilliot, A., & Wagler, A. K. (2017). Fleet management for autonomous vehicles using flows in time-expanded networks. *Electronic Notes in Discrete Mathematics*, *62*, 255–260. https://doi.org/10.1016/J.ENDM.2017.10.044
11. Chen, T. D., & Kockelman, K. M. (2016). Management of a shared autonomous electric vehicle fleet: Implications of pricing schemes. *Transportation Research Record*, *2572*, 37–46. https://doi.org/10.3141/2572-05
12. Chen, T. D., Kockelman, K. M., & Hanna, J. P. (2016). Operations of a shared, autonomous, electric vehicle fleet: Implications of vehicle & charging infrastructure

decisions. *Transportation Research Part A: Policy and Practice*, *94*, 243–254. https://doi.org/10.1016/J.TRA.2016.08.020
13. Correia, G. H. de A., & Antunes, A. P. (2012). Optimization approach to depot location and trip selection in one-way carsharing systems. *Transportation Research Part E: Logistics and Transportation Review*, *48*(1), 233–247. https://doi.org/10.1016/J.TRE.2011.06.003
14. Correia, G. H. de A., & van Arem, B. (2016). Solving the User Optimum Privately Owned Automated Vehicles Assignment Problem (UO-POAVAP): A model to explore the impacts of self-driving vehicles on urban mobility. *Transportation Research Part B: Methodological*, *87*, 64–88. https://doi.org/10.1016/J.TRB.2016.03.002
15. Fagnant, D. J., & Kockelman, K. (2015). Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations. *Transportation Research Part A: Policy and Practice*, *77*, 167–181. https://doi.org/10.1016/J.TRA.2015.04.003
16. Fagnant, D. J., & Kockelman, K. M. (2014). The travel and environmental implications of shared autonomous vehicles, using agent-based model scenarios. *Transportation Research Part C: Emerging Technologies*, *40*, 1–13. https://doi.org/10.1016/J.TRC.2013.12.001
17. Farhan, J., & Chen, T. D. (2018). Impact of ridesharing on operational efficiency of shared autonomous electric vehicle fleet. *Transportation Research Part C: Emerging Technologies*, *93*, 310–321. https://doi.org/10.1016/J.TRC.2018.04.022
18. González-González, E., Nogués, S., & Stead, D. (2019). Automated vehicles and the city of tomorrow: A backcasting approach. *Cities*, *94*, 153–160. https://doi.org/10.1016/J.CITIES.2019.05.034
19. Hafez, N., Parent, M., & Proth, J. M. (2001). Managing a pool of self service cars. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 943–948.
20. Hörl, S., Ruch, C., Becker, F., Frazzoli, E., & Axhausen, K. W. (2019). Fleet operational policies for automated mobility: A simulation assessment for Zurich. *Transportation Research Part C: Emerging Technologies*, *102*, 20–31. https://doi.org/10.1016/J.TRC.2019.02.020
21. Hörl, Sebastian. (2017). Agent-based simulation of autonomous taxi services with dynamic demand responses. *Procedia Computer Science*, *109*, 899–904. https://doi.org/10.1016/J.PROCS.2017.05.418
22. Hu, J., Morais, H., Sousa, T., & Lind, M. (2016). Electric vehicle fleet management in smart grids: A review of services, optimization and control aspects. *Renewable and Sustainable Energy Reviews*, *56*, 1207–1226. https://doi.org/10.1016/J.RSER.2015.12.014
23. Hyland, M., & Mahmassani, H. S. (2018). Dynamic autonomous vehicle fleet operations: Optimization-based strategies to assign AVs to immediate traveler demand requests. *Transportation Research Part C: Emerging Technologies*, *92*, 278–297. https://doi.org/10.1016/J.TRC.2018.05.003
24. Iacobucci, R., McLellan, B., & Tezuka, T. (2018). Modeling shared autonomous electric vehicles: Potential for transport and power grid integration. *Energy*, *158*, 148–163. https://doi.org/10.1016/J.ENERGY.2018.06.024

25. Iacobucci, R., McLellan, B., & Tezuka, T. (2019). Optimization of shared autonomous electric vehicles operations with charge scheduling and vehicle-to-grid. *Transportation Research Part C: Emerging Technologies*, *100*, 34–52. https://doi.org/10.1016/J.TRC.2019.01.011
26. Kang, N., Feinberg, F. M., & Papalambros, P. Y. (2015). Integrated decision making in electric vehicle and charging station location network design. *Journal of Mechanical Design, Transactions of the ASME*, *137*(6). https://doi.org/10.1115/1.4029894
27. Kang, N., Feinberg, F. M., & Papalambros, P. Y. (2017). Autonomous electric vehicle sharing system design. *Journal of Mechanical Design, Transactions of the ASME*, *139*(1). https://doi.org/10.1115/1.4034471
28. Levin, M. W. (2017). Congestion-aware system optimal route choice for shared autonomous vehicles. *Transportation Research Part C: Emerging Technologies*, *82*, 229–247. https://doi.org/10.1016/J.TRC.2017.06.020
29. Levin, M. W., Kockelman, K. M., Boyles, S. D., & Li, T. (2017). A general framework for modeling shared autonomous vehicles with dynamic network-loading and dynamic ride-sharing application. *Computers, Environment and Urban Systems*, *64*, 373–383. https://doi.org/10.1016/J.COMPENVURBSYS.2017.04.006
30. Liljamo, T., Liimatainen, H., & Pöllänen, M. (2018). Attitudes and concerns on automated vehicles. *Transportation Research Part F: Traffic Psychology and Behaviour*, *59*, 24–44. https://doi.org/10.1016/j.trf.2018.08.010
31. Liu, J., Kockelman, K. M., Boesch, P. M., & Ciari, F. (2017). Tracking a system of shared autonomous vehicles across the Austin, Texas network using agent-based simulation. *Transportation*, *44*(6), 1261–1278. https://doi.org/10.1007/s11116-017-9811-1
32. Loeb, B., Kockelman, K., & Liu, J. (2018). Shared autonomous electric vehicle (SAEV) operations across the Austin, Texas network with charging infrastructure decisions. *Transportation Research Part C: Emerging Technologies*, *89*. https://doi.org/10.1016/j.trc.2018.01.019
33. Macal, C. M., & North, M. J. (2009). Agent-based modeling and simulation. *IEEE*, 86–98.
34. Martínez-Díaz, M., & Soriguera, F. (2018). Autonomous vehicles: theoretical and practical challenges. *Transportation Research Procedia*, *33*, 275–282. https://doi.org/10.1016/J.TRPRO.2018.10.103
35. Mounce, R., & Nelson, J. D. (2019). On the potential for one-way electric vehicle car-sharing in future mobility systems. *Transportation Research Part A: Policy and Practice*, *120*, 17–30. https://doi.org/10.1016/J.TRA.2018.12.003
36. Nieuwenhuijsen, J., Correia, G. H. de A., Milakis, D., van Arem, B., & van Daalen, E. (2018). Towards a quantitative method to analyze the long-term innovation diffusion of automated vehicles technology using system dynamics. *Transportation Research Part C: Emerging Technologies*, *86*, 300–327. https://doi.org/10.1016/J.TRC.2017.11.016
37. Pavone, M., Smith, S. L., Frazzoli, E., & Rus, D. (2012). Robotic load balancing for mobility-on-demand systems. *International Journal of Robotics Research*, *31*(7), 839–854. https://doi.org/10.1177/0278364912444766

38. Pourazarm, S., Cassandras, C. G., & Wang, T. (2016). Optimal routing and charging of energy-limited vehicles in traffic networks. *International Journal of Robust and Nonlinear Control*, *26*(6), 1325–1350. https://doi.org/10.1002/rnc.3409
39. Rigas, E. S., Ramchurn, S. D., & Bassiliades, N. (2015). Algorithms for Electric Vehicle Scheduling in Mobility-on-Demand Schemes. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, *2015-Octob*, 1339–1344. https://doi.org/10.1109/ITSC.2015.220
40. Rumbaugh, J., Jacobson, I., & Booch, G. (1999). *The Unified Modeling Language Reference Manual*. Addison-Wesley.
41. Scheltes, A., & de Almeida Correia, G. H. (2017). Exploring the use of automated vehicles as last mile connection of train trips through an agent-based simulation model: An application to Delft, Netherlands. *International Journal of Transportation Science and Technology*, *6*(1), 28–41. https://doi.org/10.1016/J.IJTST.2017.05.004
42. Spieser, K., Treleaven, K., Zhang, R., Frazzoli, E., Morton, D., & Pavone, M. (2014). *Toward a Systematic Approach to the Design and Evaluation of Automated Mobility-on-Demand Systems: A Case Study in Singapore*. https://doi.org/10.1007/978-3-319-05990-7_20
43. Stern, R. E., Chen, Y., Churchill, M., Wu, F., Delle Monache, M. L., Piccoli, B., … Work, D. B. (2019). Quantifying air quality benefits resulting from few autonomous vehicles stabilizing traffic. *Transportation Research Part D: Transport and Environment*, *67*, 351–365. https://doi.org/10.1016/J.TRD.2018.12.008
44. Taiebat, M., & Xu, M. (2019). Synergies of four emerging technologies for accelerated adoption of electric vehicles: Shared mobility, wireless charging, vehicle-to-grid, and vehicle automation. *Journal of Cleaner Production*, *230*, 794–797. https://doi.org/10.1016/J.JCLEPRO.2019.05.142
45. Usman, M., Knapen, L., Yasar, A.-U.-H., Bellemans, T., Janssens, D., & Wets, G. (2017). Optimal recharging framework and simulation for electric vehicle fleet. *Future Generation Computer Systems*. https://doi.org/10.1016/J.FUTURE.2017.04.037
46. Wang, H., & Cheu, R. (2013). Operations of a taxi fleet for advance reservations using electric vehicles and charging stations. *Transportation Research Record*, (2352), 1–10. https://doi.org/10.3141/2352-01
47. Winter, K., Cats, O., Correia, G., & van Arem, B. (2018). Performance analysis and fleet requirements of automated demand-responsive transport systems as an urban public transport service. *International Journal of Transportation Science and Technology*, *7*(2), 151–167. https://doi.org/10.1016/J.IJTST.2018.04.004
48. Ye, L., & Yamamoto, T. (2019). Evaluating the impact of connected and autonomous vehicles on traffic safety. *Physica A: Statistical Mechanics and Its Applications*, *526*, 121009. https://doi.org/10.1016/J.PHYSA.2019.04.245
49. Yi, Z., & Shirk, M. (2018). Data-driven optimal charging decision making for connected and automated electric vehicles: A personal usage scenario. *Transportation Research Part C: Emerging Technologies*, *86*, 37–58. https://doi.org/10.1016/J.TRC.2017.10.014
50. Yi, Z., Smart, J., & Shirk, M. (2018). Energy impact evaluation for eco-routing and charging of autonomous electric vehicle fleet: Ambient temperature consideration.

Transportation Research Part C: Emerging Technologies, *89*, 344–363. https://doi.org/10.1016/J.TRC.2018.02.018

51. Zhang, D., Liu, Y., & He, S. (2019). Vehicle assignment and relays for one-way electric car-sharing systems. *Transportation Research Part B: Methodological*, *120*, 125–146. https://doi.org/10.1016/J.TRB.2018.12.004
52. Zhang, H., Sheppard, C. J. R., Lipman, T. E., Zeng, T., & Moura, S. J. (2020). Charging infrastructure demands of shared-use autonomous electric vehicles in urban areas. *Transportation Research Part D: Transport and Environment*, Vol. 78. https://doi.org/10.1016/j.trd.2019.102210
53. Zhang, R., Rossi, F., & Pavone, M. (2016). Model predictive control of autonomous mobility-on-demand systems. *Proceedings - IEEE International Conference on Robotics and Automation*, 6019–6025. https://doi.org/10.1109/ICRA.2018.8460966

8 Appendix

```
; every 42 ticks is equal to 1 min when the speed of the car is equal to 30km/h

extensions [ gis ]

globals [
  roads-dataset
  landuse-dataset
  waterways-dataset
  natural-dataset
  rejected
  system-error num-discharged
  total-travel service-travel empty-travel
]

breed [ road-vertices road-vertex ]
breed [ vehicles vehicle ]
breed [ stations station ]
breed [ customers customer ]
breed [ parking-spots parking-spot ]

vehicles-own [
  depot destination
  charge discharged charge-time
  parking-search parked
  vg vi vt vj
  gv gi gt gj
  ig iv it ij
  tg tv ti tj
  jg jv ji jt
]

to setup
  clear-all
  setup-map
  highligth-road-vertices
  display-roads
  add-stations
  add-parking-spots
  add-vehicles
  reset-ticks
end

to setup-map
  set roads-dataset gis:load-dataset "data/roads.shp"
```

```
 set landuse-dataset gis:load-dataset "data/landuse.shp"
 set waterways-dataset gis:load-dataset "data/waterways.shp"
 set natural-dataset gis:load-dataset "data/natural.shp"
 gis:set-world-envelope (gis:envelope-union-of (gis:envelope-of roads-dataset)
                                               (gis:envelope-of landuse-dataset)
                                               (gis:envelope-of waterways-dataset)
                                               (gis:envelope-of natural-dataset))
end

to highligth-road-vertices
 set-default-shape road-vertices "circle"
 ask road-vertices [ die ]
 foreach gis:find-features roads-dataset "TYPE" "primary" [ vector-feature ->
   foreach gis:vertex-lists-of vector-feature [ vertex ->
     foreach vertex [ point ->
       let node-location gis:location-of point
       if not empty? node-location
       [ create-road-vertices 1
         [ set color white
           set xcor item 0 node-location
           set ycor item 1 node-location
           set hidden? true] ] ] ] ]
 foreach gis:find-features roads-dataset "TYPE" "secondary" [ vector-feature ->
   foreach gis:vertex-lists-of vector-feature [ vertex ->
     foreach vertex [ point ->
       let node-location gis:location-of point
       if not empty? node-location
        [ create-road-vertices 1
         [ set color white
           set xcor item 0 node-location
           set ycor item 1 node-location
           set hidden? true] ] ] ] ]
 foreach gis:find-features roads-dataset "NAME" "Avenue Pierre-De Coubertin" [ vector-feature
       ->
   foreach gis:vertex-lists-of vector-feature [ vertex ->
     foreach vertex [ point ->
       let node-location gis:location-of point
       if not empty? node-location
        [ create-road-vertices 1
         [ set color white
           set xcor item 0 node-location
           set ycor item 1 node-location
           set hidden? true] ] ] ] ]
end

to display-roads
```

```
  gis:set-drawing-color blue
  gis:draw roads-dataset 1
end

to add-stations
  set-default-shape stations "house"
  create-stations 5 [
    set color yellow
    set size 3
  ]
  (foreach (list (station 687) (station 688) (station 689) (station 690) (station 691))
    (list (road-vertex 492) (road-vertex 678) (road-vertex 183) (road-vertex 571) (road-vertex 22))
    [ [the-station the-location] -> ask the-station [move-to the-location] ])
end

to add-parking-spots
  ask stations [
    hatch-parking-spots num-parking-spots
    [ set hidden? true ]
  ]
end

to vehicle-defaults
  set color green
  set size 5
  set parked 1
  set charge (vehicle-range)
end

to add-vehicles
  ask station 687 [ hatch-vehicles num-vehicles [
    set depot 1
    vehicle-defaults ]
  ]
    ask station 688 [ hatch-vehicles num-vehicles [
    set depot 2
    vehicle-defaults ]
  ]
    ask station 689 [ hatch-vehicles num-vehicles [
    set depot 3
    vehicle-defaults ]
  ]
    ask station 690 [ hatch-vehicles num-vehicles [
    set depot 4
    vehicle-defaults ]
  ]
```

```
    ask station 691 [ hatch-vehicles num-vehicles [
    set depot 5
    vehicle-defaults ]
  ]
end

to go
  vehicle-charging
  trigger-trip
  trigger-parking-trip
  fulfill-trip
  tick
end

to trigger-trip
  set-default-shape customers "person"
  if random (42 / demand-probability) = 0 [
    create-customers 1 [ set size 4 ]
    ask customers [
      move-to one-of stations
      ifelse count vehicles-here with [ (discharged = 0) and (parking-search = 0) and (parked = 1) ]
        != 0
        [ ask one-of vehicles-here with [ (discharged = 0) and (parking-search = 0) and (parked = 1)
        ] [
          if depot = 1 [ set destination one-of [ 2 3 4 5 ] ]
          if depot = 2 [ set destination one-of [ 1 3 4 5 ] ]
          if depot = 3 [ set destination one-of [ 1 2 4 5 ] ]
          if depot = 4 [ set destination one-of [ 1 2 3 5 ] ]
          if depot = 5 [ set destination one-of [ 1 2 3 4 ] ]
          set parked 0
          start ]
        die ]
        [ set rejected rejected + 1
        die ]
  ] ]
end

to start
  (ifelse
    depot = 1 and destination = 2 [ golf-viau ]
    depot = 1 and destination = 3 [ golf-insectarium ]
    depot = 1 and destination = 4 [ golf-tree ]
    depot = 1 and destination = 5 [ golf-jardin ]
    depot = 2 and destination = 1 [ viau-golf ]
    depot = 2 and destination = 3 [ viau-insectarium ]
    depot = 2 and destination = 4 [ viau-tree ]
```

```
    depot = 2 and destination = 5 [ viau-jardin ]
    depot = 3 and destination = 1 [ insectarium-golf ]
    depot = 3 and destination = 2 [ insectarium-viau ]
    depot = 3 and destination = 4 [ insectarium-tree ]
    depot = 3 and destination = 5 [ insectarium-jardin ]
    depot = 4 and destination = 1 [ tree-golf ]
    depot = 4 and destination = 2 [ tree-viau ]
    depot = 4 and destination = 3 [ tree-insectarium ]
    depot = 4 and destination = 5 [ tree-jardin ]
    depot = 5 and destination = 1 [ jardin-golf ]
    depot = 5 and destination = 2 [ jardin-viau ]
    depot = 5 and destination = 3 [ jardin-insectarium ]
    [ jardin-tree ] )
end


to fulfill-trip
  ask vehicles [
    if gv > 0 [ golf-viau ]
    if gi > 0 [ golf-insectarium ]
    if gt > 0 [ golf-tree ]
    if gj > 0 [ golf-jardin ]
    if vg > 0 [ viau-golf ]
    if vi > 0 [ viau-insectarium ]
    if vt > 0 [ viau-tree ]
    if vj > 0 [ viau-jardin ]
    if ig > 0 [ insectarium-golf ]
    if iv > 0 [ insectarium-viau ]
    if it > 0 [ insectarium-tree ]
    if ij > 0 [ insectarium-jardin ]
    if tg > 0 [ tree-golf ]
    if tv > 0 [ tree-viau ]
    if ti > 0 [ tree-insectarium ]
    if tj > 0 [ tree-jardin ]
    if jg > 0 [ jardin-golf ]
    if jv > 0 [ jardin-viau ]
    if ji > 0 [ jardin-insectarium ]
    if jt > 0 [ jardin-tree ] ]
end

to vehicle-charging
  ask vehicles with [ discharged = 1 ] [
    set color red
    set charge-time charge-time + 1
    if charge-time >= ((maximum-charge-time * 42) * (vehicle-range - charge) / vehicle-range) [
      set discharged 0
```

```
      set charge-time 0
      set color green
      set charge (vehicle-range) ]
  ]
end

to trigger-parking-trip
  ask vehicles with [ parking-search = 1 ] [
  set color white
  set parking-search 0
  if depot = 1 [
    (ifelse
       count vehicles-on station 689 < count parking-spots-on station 689 [
         set destination 3
         golf-insectarium ]
       (count vehicles-on station 690 < count parking-spots-on station 690) and
       (count vehicles-on station 689 >= count parking-spots-on station 689) [
         set destination 4
         golf-tree ]
       (count vehicles-on station 688 < count parking-spots-on station 688) and
       (count vehicles-on station 690 >= count parking-spots-on station 690) and
       (count vehicles-on station 689 >= count parking-spots-on station 689) [
         set destination 2
         golf-viau ]
        [ set destination 5
          golf-jardin ] )
    ]
  if depot = 2 [
    (ifelse
       count vehicles-on station 691 < count parking-spots-on station 691 [
         set destination 5
         viau-jardin ]
       (count vehicles-on station 687 < count parking-spots-on station 687) and
       (count vehicles-on station 691 >= count parking-spots-on station 691) [
         set destination 1
         viau-golf ]
       (count vehicles-on station 689 < count parking-spots-on station 689) and
       (count vehicles-on station 687 >= count parking-spots-on station 687) and
       (count vehicles-on station 691 >= count parking-spots-on station 691) [
         set destination 3
         viau-insectarium ]
        [ set destination 4
          viau-tree ] )
    ]
  if depot = 3 [
    (ifelse
```

```
    count vehicles-on station 691 < count parking-spots-on station 691 [
      set destination 5
      insectarium-jardin ]
    (count vehicles-on station 687 < count parking-spots-on station 687) and
    (count vehicles-on station 691 >= count parking-spots-on station 691) [
      set destination 1
      insectarium-golf ]
    (count vehicles-on station 690 < count parking-spots-on station 690) and
    (count vehicles-on station 687 >= count parking-spots-on station 687) and
    (count vehicles-on station 691 >= count parking-spots-on station 691) [
      set destination 4
      insectarium-tree ]
     [ set destination 2
       insectarium-viau ] )
  ]
if depot = 4 [
  (ifelse
    count vehicles-on station 687 < count parking-spots-on station 687 [
      set destination 1
      tree-golf ]
    (count vehicles-on station 689 < count parking-spots-on station 689) and
    (count vehicles-on station 687 >= count parking-spots-on station 687) [
      set destination 3
      tree-insectarium ]
    (count vehicles-on station 688 < count parking-spots-on station 688) and
    (count vehicles-on station 689 >= count parking-spots-on station 689) and
    (count vehicles-on station 687 >= count parking-spots-on station 687) [
      set destination 2
      tree-viau ]
     [ set destination 5
       tree-jardin ] )
  ]
if depot = 5 [
  (ifelse
    count vehicles-on station 690 < count parking-spots-on station 690 [
      set destination 4
      jardin-tree ]
    (count vehicles-on station 687 < count parking-spots-on station 687) and
    (count vehicles-on station 690 >= count parking-spots-on station 690) [
      set destination 1
      jardin-golf ]
    (count vehicles-on station 689 < count parking-spots-on station 689) and
    (count vehicles-on station 687 >= count parking-spots-on station 687) and
    (count vehicles-on station 690 >= count parking-spots-on station 690) [
      set destination 3
      jardin-insectarium ]
```

```
        [ set destination 2
          jardin-viau ] )
    ]
  ]
end

to end-process
  set total-travel total-travel + 1
  if color = white [ set empty-travel empty-travel + 1 ]
  if color = green [ set service-travel service-travel + 1 ]
  if charge <= 0 [ set system-error system-error + 1 ]
  set color green
  ifelse count vehicles-here > count parking-spots-here
    [ set parking-search 1 ]
    [ set parked 1 ]
  if (charge < critical-threshold) and (parked = 1) [
    set discharged 1
    set num-discharged num-discharged + 1 ]

end

to-report node-gv
  (ifelse
    gv < 37 [ report road-vertex 406 ]
    gv < 38 [ report road-vertex 308 ]
    gv < 161 [ report road-vertex 400 ]
    [ report road-vertex 678 ] )
end

to golf-viau
    if distance road-vertex 678 = 0
    [ set gv 0
    set depot 2
    set charge charge - 1.97
    end-process
    stop ]
    pen-down
    face node-gv
    ifelse distance node-gv < 1
    [ move-to node-gv
    stop ]
    [ fd 1 ]
    set gv gv + 1
end

to-report node-gi
```

```
  (ifelse
    gi < 37 [ report road-vertex 406 ]
    gi < 38 [ report road-vertex 308 ]
    gi < 93 [ report road-vertex 192 ]
    gi < 94 [ report road-vertex 193 ]
    gi < 96 [ report road-vertex 194 ]
    gi < 98 [ report road-vertex 195 ]
    gi < 100 [ report road-vertex 196 ]
    gi < 121 [ report road-vertex 181 ]
    [ report road-vertex 183 ] )
end

to golf-insectarium
    if distance road-vertex 183 = 0
    [ set gi 0
    set depot 3
    set charge charge - 1.49
    end-process
    stop ]
    pen-down
    face node-gi
    ifelse distance node-gi < 1
    [ move-to node-gi
    stop ]
    [ fd 1 ]
    set gi gi + 1
end

to-report node-gt
  (ifelse
    gt < 75 [ report road-vertex 284 ]
    gt < 132 [ report road-vertex 494 ]
    gt < 133 [ report road-vertex 461 ]
    [ report road-vertex 571 ] )
end

to golf-tree
    if distance road-vertex 571 = 0
    [ set gt 0
    set depot 4
    set charge charge - 1.86
    end-process
    stop ]
    pen-down
    face node-gt
    ifelse distance node-gt < 1
```

```
    [ move-to node-gt
    stop ]
    [ fd 1 ]
    set gt gt + 1
end

to-report node-gj
  (ifelse
    gj < 37 [ report road-vertex 406 ]
    gj < 38 [ report road-vertex 308 ]
    gj < 93 [ report road-vertex 192 ]
    gj < 94 [ report road-vertex 193 ]
    gj < 96 [ report road-vertex 194 ]
    gj < 98 [ report road-vertex 195 ]
    gj < 100 [ report road-vertex 196 ]
    gj < 121 [ report road-vertex 181 ]
    gj < 139 [ report road-vertex 3 ]
    gj < 156 [ report road-vertex 5 ]
    gj < 160 [ report road-vertex 6 ]
    gj < 163 [ report road-vertex 9 ]
    gj < 170 [ report road-vertex 10 ]
    gj < 173 [ report road-vertex 12 ]
    [ report road-vertex 22 ] )
end

to golf-jardin
    if distance road-vertex 22 = 0
    [ set gj 0
    set depot 5
    set charge charge - 2.21
    end-process
    stop]
    pen-down
    face node-gj
    ifelse distance node-gj < 1
    [ move-to node-gj
    stop ]
    [ fd 1 ]
    set gj gj + 1
end

to-report node-vg
  (ifelse
    vg < 8 [ report road-vertex 660 ]
    vg < 9 [ report road-vertex 659 ]
    vg < 33 [ report road-vertex 401 ]
```

```
    vg < 34 [ report road-vertex 315 ]
    [ report road-vertex 492 ] )
end

to viau-golf
    if distance road-vertex 492 = 0
    [ set vg 0
    set depot 1
    set charge charge - 1.44
    end-process
    stop ]
    pen-down
    face node-vg
    ifelse distance node-vg < 1
    [ move-to node-vg
   stop ]
    [ fd 1 ]
    set vg vg + 1
end

to-report node-vi
  (ifelse
    vi < 8 [ report road-vertex 660 ]
    vi < 9 [ report road-vertex 659 ]
    vi < 33 [ report road-vertex 401 ]
    vi < 34 [ report road-vertex 315 ]
    vi < 102 [ report road-vertex 165 ]
    vi < 103 [ report road-vertex 192 ]
    vi < 104 [ report road-vertex 193 ]
    vi < 106 [ report road-vertex 194 ]
    vi < 108 [ report road-vertex 195 ]
    vi < 110 [ report road-vertex 196 ]
    vi < 131 [ report road-vertex 181 ]
    [ report road-vertex 183 ] )
end
to viau-insectarium
    if distance road-vertex 183 = 0
    [ set vi 0
    set depot 3
    set charge charge - 1.72
    end-process
    stop ]
    pen-down
    face node-vi
    ifelse distance node-vi < 1
    [ move-to node-vi
```

```
   stop ]
    [ fd 1 ]
    set vi vi + 1
end

to-report node-vt
  (ifelse
    vt < 62 [ report road-vertex 634 ]
    vt < 63 [ report road-vertex 201 ]
    vt < 64 [ report road-vertex 203 ]
    vt < 231 [ report road-vertex 143 ]
    vt < 234 [ report road-vertex 319 ]
    vt < 235 [ report road-vertex 320 ]
    [ report road-vertex 571 ] )
end

to viau-tree
    if distance road-vertex 571 = 0
    [ set vt 0
    set depot 4
    set charge charge - 3.18
    end-process
    stop ]
    pen-down
    face node-vt
    ifelse distance node-vt < 1
    [ move-to node-vt
    stop ]
    [ fd 1 ]
    set vt vt + 1
end

to-report node-vj
  (ifelse
    vj < 62 [ report road-vertex 634 ]
    vj < 63 [ report road-vertex 201 ]
    vj < 64 [ report road-vertex 203 ]
    [ report road-vertex 22 ] )
end

to viau-jardin
    if distance road-vertex 22 = 0
    [ set vj 0
    set depot 5
    set charge charge - 1.14
    end-process
```

```
    stop ]
    pen-down
    face node-vj
    ifelse distance node-vj < 1
    [ move-to node-vj
    stop ]
    [ fd 1 ]
    set vj vj + 1
end

to-report node-ig
  (ifelse
    ig < 14 [ report road-vertex 3 ]
    ig < 31 [ report road-vertex 5 ]
    ig < 35 [ report road-vertex 6 ]
    ig < 36 [ report road-vertex 48 ]
    ig < 40 [ report road-vertex 49 ]
    ig < 46 [ report road-vertex 50 ]
    ig < 56 [ report road-vertex 0 ]
    ig < 68 [ report road-vertex 1 ]
    ig < 95 [ report road-vertex 99 ]
    ig < 98 [ report road-vertex 100 ]
    ig < 99 [ report road-vertex 101 ]
    ig < 100 [ report road-vertex 122 ]
    ig < 102 [ report road-vertex 95 ]
    ig < 103 [ report road-vertex 117 ]
    ig < 104 [ report road-vertex 165 ]
    [ report road-vertex 492 ] )
end

to insectarium-golf
    if distance road-vertex 492 = 0
    [ set ig 0
    set depot 1
    set charge charge - 1.75
    end-process
    stop ]
    pen-down
    face node-ig
    ifelse distance node-ig < 1
    [ move-to node-ig
    stop ]
    [ fd 1 ]
    set ig ig + 1
end
```

```
to-report node-iv
  (ifelse
    iv < 14 [ report road-vertex 3 ]
    iv < 31 [ report road-vertex 5 ]
    iv < 35 [ report road-vertex 6 ]
    iv < 36 [ report road-vertex 48 ]
    iv < 40 [ report road-vertex 49 ]
    iv < 46 [ report road-vertex 50 ]
    iv < 56 [ report road-vertex 0 ]
    iv < 68 [ report road-vertex 1 ]
    iv < 95 [ report road-vertex 99 ]
    iv < 98 [ report road-vertex 100 ]
    iv < 99 [ report road-vertex 101 ]
    iv < 100 [ report road-vertex 122 ]
    iv < 101 [ report road-vertex 131 ]
    iv < 104 [ report road-vertex 134 ]
    iv < 106 [ report road-vertex 135 ]
    iv < 126 [ report road-vertex 397 ]
    iv < 166 [ report road-vertex 400 ]
    [ report road-vertex 678 ] )
end

to insectarium-viau
    if distance road-vertex 678 = 0
    [ set iv 0
    set depot 2
    set charge charge - 2.84
    end-process
    stop ]
    pen-down
    face node-iv
    ifelse distance node-iv < 1
    [ move-to node-iv
    stop ]
    [ fd 1 ]
    set iv iv + 1
end

to-report node-it
  (ifelse
    it < 14 [ report road-vertex 3 ]
    it < 31 [ report road-vertex 5 ]
    it < 35 [ report road-vertex 6 ]
    it < 38 [ report road-vertex 9 ]
    it < 45 [ report road-vertex 10 ]
    it < 48 [ report road-vertex 12 ]
```

```
    it < 50 [ report road-vertex 22 ]
    it < 193 [ report road-vertex 143 ]
    it < 197 [ report road-vertex 319 ]
    it < 198 [ report road-vertex 320 ]
    [ report road-vertex 571 ] )
end

to insectarium-tree
    if distance road-vertex 571 = 0
    [ set it 0
    set depot 4
    set charge charge - 2.8
    end-process
    stop ]
    pen-down
    face node-it
    ifelse distance node-it < 1
    [ move-to node-it
    stop ]
    [ fd 1 ]
    set it it + 1
end

to-report node-ij
  (ifelse
    ij < 14 [ report road-vertex 3 ]
    ij < 31 [ report road-vertex 5 ]
    ij < 35 [ report road-vertex 6 ]
    ij < 38 [ report road-vertex 9 ]
    ij < 45 [ report road-vertex 10 ]
    ij < 48 [ report road-vertex 12 ]
    [ report road-vertex 22 ] )
end

to insectarium-jardin
    if distance road-vertex 22 = 0
    [ set ij 0
    set depot 5
    set charge charge - 0.78
    end-process
    stop ]
    pen-down
    face node-ij
    ifelse distance node-ij < 1
    [ move-to node-ij
    stop ]
```

```
    [ fd 1 ]
    set ij ij + 1
end

to-report node-tg
  (ifelse
    tg < 48 [ report road-vertex 296 ]
    tg < 75 [ report road-vertex 305 ]
    tg < 140 [ report road-vertex 192 ]
    tg < 142 [ report road-vertex 95 ]
    tg < 143 [ report road-vertex 96 ]
    tg < 144 [ report road-vertex 165 ]
    [ report road-vertex 492 ] )
end

to tree-golf
    if distance road-vertex 492 = 0
    [ set tg 0
    set depot 1
    set charge charge - 1.85
    end-process
    stop ]
    pen-down
    face node-tg
    ifelse distance node-tg < 1
    [ move-to node-tg
    stop ]
    [ fd 1 ]
    set tg tg + 1
end

to-report node-tv
  (ifelse
    tv < 48 [ report road-vertex 296 ]
    tv < 75 [ report road-vertex 305 ]
    tv < 140 [ report road-vertex 192 ]
    tv < 208 [ report road-vertex 400 ]
    [ report road-vertex 678 ] )
end

to tree-viau
    if distance road-vertex 678 = 0
    [ set tv 0
    set depot 2
    set charge charge - 2.59
    end-process
```

```
    stop ]
    pen-down
    face node-tv
    ifelse distance node-tv < 1
    [ move-to node-tv
    stop ]
    [ fd 1 ]
    set tv tv + 1
end

to-report node-ti
  (ifelse
    ti < 48 [ report road-vertex 296 ]
    ti < 75 [ report road-vertex 305 ]
    ti < 140 [ report road-vertex 192 ]
    ti < 141 [ report road-vertex 193 ]
    ti < 143 [ report road-vertex 194 ]
    ti < 145 [ report road-vertex 195 ]
    ti < 147 [ report road-vertex 196 ]
    ti < 168 [ report road-vertex 181 ]
    [ report road-vertex 183 ] )
end

to tree-insectarium
    if distance road-vertex 183 = 0
    [ set ti 0
    set depot 3
    set charge charge - 2.1
    end-process
    stop ]
    pen-down
    face node-ti
    ifelse distance node-ti < 1
    [ move-to node-ti
    stop ]
    [ fd 1 ]
    set ti ti + 1
end

to-report node-tj
  (ifelse
    tj < 48 [ report road-vertex 296 ]
    tj < 75 [ report road-vertex 305 ]
    tj < 140 [ report road-vertex 192 ]
    tj < 141 [ report road-vertex 193 ]
    tj < 143 [ report road-vertex 194 ]
```

```
    tj < 145 [ report road-vertex 195 ]
    tj < 147 [ report road-vertex 196 ]
    tj < 172 [ report road-vertex 183 ]
    tj < 186 [ report road-vertex 3 ]
    tj < 203 [ report road-vertex 5 ]
    tj < 207 [ report road-vertex 6 ]
    tj < 220 [ report road-vertex 12 ]
    [ report road-vertex 22 ] )
end

to tree-jardin
    if distance road-vertex 22 = 0
    [ set tj 0
    set depot 5
    set charge charge - 2.88
    end-process
    stop ]
    pen-down
    face node-tj
    ifelse distance node-tj < 1
    [ move-to node-tj
    stop ]
    [ fd 1 ]
    set tj tj + 1
end

to-report node-jg
  (ifelse
    jg < 143 [ report road-vertex 143 ]
    jg < 146 [ report road-vertex 319 ]
    jg < 147 [ report road-vertex 320 ]
    jg < 151 [ report road-vertex 457 ]
    jg < 185 [ report road-vertex 572 ]
    jg < 230 [ report road-vertex 296 ]
    jg < 322 [ report road-vertex 192 ]
    jg < 324 [ report road-vertex 95 ]
    jg < 325 [ report road-vertex 96 ]
    jg < 326 [ report road-vertex 165 ]
    [ report road-vertex 492 ] )
end

to jardin-golf
    if distance road-vertex 492 = 0
    [ set jg 0
    set depot 1
    set charge charge - 3.87
```

```
    end-process
    stop ]
    pen-down
    face node-jg
    ifelse distance node-jg < 1
    [ move-to node-jg
    stop ]
    [ fd 1 ]
    set jg jg + 1
end

to-report node-jv
  (ifelse
    jv < 143 [ report road-vertex 143 ]
    jv < 146 [ report road-vertex 319 ]
    jv < 147 [ report road-vertex 320 ]
    jv < 151 [ report road-vertex 457 ]
    jv < 185 [ report road-vertex 572 ]
    jv < 230 [ report road-vertex 296 ]
    jv < 322 [ report road-vertex 192 ]
    jv < 324 [ report road-vertex 95 ]
    jv < 390 [ report road-vertex 400 ]
    [ report road-vertex 678 ] )
end

to jardin-viau
    if distance road-vertex 678 = 0
    [ set jv 0
    set depot 2
    set charge charge - 4.61
    end-process
    stop ]
    pen-down
    face node-jv
    ifelse distance node-jv < 1
    [ move-to node-jv
    stop ]
    [ fd 1 ]
    set jv jv + 1
end

to-report node-ji
  (ifelse
    ji < 143 [ report road-vertex 143 ]
    ji < 146 [ report road-vertex 319 ]
    ji < 147 [ report road-vertex 320 ]
```

```
    ji < 151 [ report road-vertex 457 ]
    ji < 185 [ report road-vertex 572 ]
    ji < 230 [ report road-vertex 296 ]
    ji < 322 [ report road-vertex 192 ]
    ji < 323 [ report road-vertex 193 ]
    ji < 325 [ report road-vertex 194 ]
    ji < 327 [ report road-vertex 195 ]
    ji < 329 [ report road-vertex 196 ]
    ji < 350 [ report road-vertex 181 ]
    [ report road-vertex 183 ] )
end

to jardin-insectarium
    if distance road-vertex 183 = 0
    [ set ji 0
    set depot 3
    set charge charge - 4.13
    set parking-search 0
    end-process
    stop ]
    pen-down
    face node-ji
    ifelse distance node-ji < 1
    [ move-to node-ji
    stop ]
    [ fd 1 ]
    set ji ji + 1
end

to-report node-jt
  (ifelse
    jt < 143 [ report road-vertex 143 ]
    jt < 146 [ report road-vertex 319 ]
    jt < 147 [ report road-vertex 320 ]
    jt < 151 [ report road-vertex 457 ]
    [ report road-vertex 571 ] )
end

to jardin-tree
    if distance road-vertex 571 = 0
    [ set jt 0
    set depot 4
    set charge charge - 2.01
    end-process
    stop ]
    pen-down
```

```
    face node-jt
    ifelse distance node-jt < 1
    [ move-to node-jt
    stop ]
    [ fd 1 ]
    set jt jt + 1
end
```