INTEGRATION OF PUBLIC TRANSPORTATION USING AUTONOMOUS VEHICLES

by
Akif Fidanoğlu
B.S., Civil Engineering, Boğaziçi University, 2011
M.Sc., Civil Engineering, Boğaziçi University, 2015

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in Civil Engineering
Boğaziçi University
2021

ACKNOWLEDGEMENTS

First, I need to express my sincerest gratitude to my supervisor Assoc. Prof. Ilgın Gökaşar for her continuous support and guidance. This study benefited greatly from her invaluable instruction, supervision, and mentorship. It has been a privilege to work with her for my PhD study.

Besides my supervisor, I would like to thank my thesis committee, Assoc. Prof. Özer Çinicioğlu, Assist. Prof. Mehtap Işık, Assoc. Prof. Erkan Çelik, and Assoc. Prof. Muhammet Deveci, who kindly accepted to take part in the thesis committee. I am highly grateful for their encouragement and insightful comments.

I would like to express my gratitude to those at BURULAŞ, for giving up their valuable time to discuss the studied problem with me and for providing me the data used in this thesis.

I also need to thank all my friends, especially Gürkan Aydemir and Mustafa Yeşilyurt, for all the fun we have had together during this period of my life and for their unconditional support.

I owe my deepest gratitude to my father Emrullah Fidanoğlu and especially my mother Sevim Fidanoğlu, who has sacrificed all she has for her children. Without her unconditional love and support, none of my achievements in life would have been possible. I am indebted to my brother Muhammet Fidanoğlu for his invaluable guidance and support. I also need to thank my wife Eda for her love and friendship. I am grateful for her continuous support and patience. Lastly, I need to thank my children, Yusuf Salih and Meryem, for being excellent kids.

ABSTRACT

INTEGRATION OF PUBLIC TRANSPORTATION USING AUTONOMOUS VEHICLES

While public transportation (PT) plays a critical role in urban mobility, the discomfort and the disutility of the last-mile trips make PT unattractive. We investigate the feasibility of shared autonomous vehicles (AV) in terms of providing an alternative on-demand transportation service for last-mile mobility to the conventional bus routes that have fixed routes and schedules. To this end, the bus routes that operate along the edges of the transportation network are selected. The origin and destination (OD) pairs of the trips made in these bus routes are inferred at an individual level. Then, the related vehicle routing problem is formulated by taking the characteristics of the proposed transportation service into consideration. Several solution methods are developed for both solution construction and solution improvement phases of the problem. An agent-based simulation framework is constructed to evaluate the performance of the solution methods with real-world data. The findings of the study indicate the success of the solution methods in solving a highly dynamic problem. The results show that the integration of PT using AVs is well-suited to improve the service quality in the last-mile mobility. The investment and the operational costs of the proposed transportation service are further analyzed and shown to be more advantageous than conventional buses with fixed routes.

ÖZET

OTONOM ARAÇLAR KULLANILARAK TOPLU TAŞIMA SİSTEMLERİNİN ENTEGRESYONU

Toplu taşıma sistemleri şehir içi ulaşımda kritik bir rol almakla birlikte, son-mil yolculuklarındaki konforsuzluk ve verimsizlik sebebiyle tercih edilebilirliğini kaybetmektedir. Bu çalışmada biz, paylaşılabilir otonom araçların son mil yolculukları için sabit rota ve sefer saatleri olan konvensiyonel otobüslere bir alternatif olma hususundaki fizibilitesini araştırmaktayız. Bu amaçla, ulaşım sisteminin uçlarında çalışan otobüs hatları seçilmiş ve bu hatlarda yapılan yolculukların başlangıç ve varış noktları kişi seviyesinde çıkarılmıştır. İlgili araç rotalama problemi önerilen taşıma hizmetinin özellikleri göz önünde bulundurularak formüle edilmiştir. Problemin çözüm oluşturma ve çözüm iyileştirme safhaları için birçok çözüm yöntemi geliştirilmiştir. Çözüm yöntemlerinin performanslarını değerlendirmek için ajan temelli simülasyon sistemi dizayn edilmiştir. Çalışmanın bulguları, kullanılan çözüm yöntemlerinin yüksek derecede dinamik bir problemin çözümünde başarılı olduğuna işaret etmektedir. Sonuçlar, otonom araçlarla sağlanan toplu taşıma entegrasyonun son mil ulaşımı için çok uygun olduğunu göstermiştir. Önerilen taşıma hizmetnin yatırım ve operasyonel maliyetleri de incelenmiş olup, sabit rotalı otobüs hatlarına nazaran daha avantajlı olduğu görülmüştür.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS/ABBREVIATIONS

ADC	Automated Data Collection
AFC	Automated Fare Collection
AV	Autonomous Vehicles
AVL	Automated Vehicle Location
DARP	Dial-a-ride Problem
DVRP	Dynamic Vehicle Routing Problem
OD	Origin and Destination
PDP	Pick-up and Delivery Problem
PT	Public Transportation
SA	Simulated Annealing
SVRP	Stochastic Vehicle Routing Problems
TS	Tabu Search
VRP	Vehicle Routing Problem
VRPSC	Vehicle Routing Problem with Stochastic Customers
VRPSD	Vehicle Routing Problem with Stochastic Demand
VRPTW	Vehicle Routing Problem with Time Windows

1. INTRODUCTION

1.1. The Motivation of the Dissertation

Transportation is a basic need for humans, enabling them to mobilize to different locations at local, national, and international levels. While it is an essential part of our lives, it is also one of the major users of energy. Energy used in transportation activities causes carbon emissions and contributes to global warming significantly.

Even with electric vehicles, environmental pollution will still be a concern since the electricity consumed by electric vehicles is generated mostly by fossil fuels. This situation will not dramatically change in the short term. In fact, fossil fuel is expected to be a primary resource for energy by over %60 in 2040 (World Energy Council, 2019). Therefore, increasing the overall efficiency of transportation systems is a must for decreasing fossil fuel consumption.

Private cars have occupied a great portion of passenger transportation and shaped personal mobility in the past century by providing fast door-to-door travel. However, the usage of private vehicles has increased traffic congestion, the need for parking spaces in urban areas, and oil consumption, thus greenhouse gases. In addition, the vast majority of the vehicles in urban areas are underutilized. Private cars spend 90 % of their lifespan as parked (Federal Highway Administration, 2011). While public transit is also one of the most used transportation modes, the discomfort, and the inconvenience experienced during ridership of PT, especially in the first and last-mile of the trips, make PT unattractive. There is an imbalance between these two transportation modes in terms of costs, comfort, and efficiency. In order to increase the efficiency of overall transportation systems in urban areas, it is necessary to have such a transportation service that combines advantageous characteristics of these two transportation modes.

Shared use services have allowed passengers to experience such transportation mode, which has the comfort of private cars and is a relatively cheaper option than owning a private vehicle. These services aim to decrease the empty seats during a trip, thus minimize the total vehicles needed and operation costs. By achieving these, ride-sharing services can help to reduce traffic congestion and fuel costs. For these reasons, ride-sharing services have become a common transportation mode in recent years. However, ride-sharing services still have some disadvantages, and fixed-route transits or private cars are more convenient for some trip purposes. Autonomous Vehicles (AV) are expected to eliminate these problems and become a more efficient transportation mode with ride-sharing operations. With the help of driverless vehicles, human-related performance limitations and the inefficiencies related to these limitations can be eliminated. With increased efficiency in operation, shared use AVs can significantly benefit passenger transportation in urban areas (Hyland, 2018).

Our study proposes a transportation service that consists of shared AVs and offers point-to-point transportation. This system is designed to replace the conventional bus routes that operate in the last-mile section of the passengers' multi-trips. The main idea behind selecting these routes is to establish a system that increases access to high-capacity transportation modes. Since mass transit is essential for an efficient transportation system in urban areas, continued investment in these mass transit modes should be guaranteed (Union of Concerned Scientists, 2017). Meanwhile, the conventional transportation services operating at the edges of the transportation networks must be replaced by the services that utilize the emerging technologies.

In this sense, we collected the trip data of bus routes that operate in a region near a subway station. The trips made in this region contain the characteristics of a typical last-mile problem. Origin and destination (OD) points of these trips were determined using the inference methods proposed in previous studies and the methods introduced in this study. After acquiring the OD pairs of the trips, the trips were simulated with shared autonomous vehicles. To solve the vehicle routing problem, several constructive and improvement heuristics were developed.

1.1. Last-mile Problem

Public transportation is an essential component of any sustainable urban transport system. Since it helps to reduce private car usage, thus traffic congestion, fuel consumption, and carbon emissions, municipalities are trying to increase the portion of PT in the transportation systems. However, it is crucial to serve sustainable and accessible public transportation to the passengers to make it preferable by the general public.

In major cities, the main public transportation modes are bus and rail services. However, most passengers need to walk, drive their private vehicles, and use taxis or other transportation modes to reach the nearest station or stop. These trips, which are the first or last leg of the daily trips, are defined as the first or last mile. Since in most of the cases, the first and last mile of the trips are inflexible and slow, while using public transportation, last mile or first mile experiences causes inconveniences for passengers (Scheltes and de Almeida Correia, 2017). This disadvantage in public transportation systems is one of the main deterrents that make PTs an undesirable transportation mode for passengers (H. Wang and Odoni, 2016). Even though various transportation modes are started to be used to eliminate the first and last mile problem in recent years, passengers still encounter limitations and difficulties such as long travel times, high costs, bad weather conditions when they prefer these new concepts. In this sense, convenient services for first and last-mile transportation need to be offered. By reducing the travel time of the commuters between the transit and their homes or workplaces and eliminating the difficulties in these trips, a more integrated public transit network potentially provides a cost-effective and sustainable door-to-door transportation (Chong et al., 2011).

1.2. Vehicle Routing Problem

There has been a rapid and significant evolution in passenger transportation. With the emergence of intelligent transportation systems and the smart phone technologies daily commute of passengers has become more efficient. Several companies like Lyft and Uber have used these technologies and offered users ride-sharing transportation services in recent years. In the near future, electrical and autonomous vehicle technology will revolutionize passenger transportation (Martin, 2019). The positive impact of these emerging technologies on the environment and the economy can be realized when efficient vehicle routing is achieved.

In this sense, planning and optimizing transportation services become vital, and problems faced during the operations of transportation services have attracted the attention of both the industry and academia. Researchers developed various methodologies that offer efficient solutions to operational problems. The routing problem is one of the most challenging problems in this domain. Therefore, VRP has become an important topic for various organizations and companies in the transportation of goods or passengers.

The VRP was first introduced by Dantzig et al. (1954) as Vehicle Routing Problem (VRP), which aims to find the set of optimal routes for vehicles that are supposed to visit all the customers. This basic problem has been extended to several variants that simulate real-life problems and include the complexity of real applications. With the advances in solution methods and technological developments, these problems have become one of the major research topics in operation research. With the introduction of real-life VRP applications that are more complex and large scale, the extent of VRP has grown considerably. According to Eksioglu, et al. (2009), the VRP literature has been growing exponentially at a rate of 6% each year.

The VRP can be defined as finding a set of routes for the vehicles to serve the trip requests with the consideration of problem constraints (Irnich, Toth, and Vigo, 2014). The task defined for VRP contains two steps, grouping the passengers into vehicles and finding the optimal routes. To utilize the emerging technologies efficiently, the routing problems we encounter in almost all transportation modes should be well defined and solved as optimal as possible.

There are several variants of VRP which formulate problems with different characteristics. The services that provide point-to-point transportation for passengers are classified as dial-a-ride problems (DARP) (Doerner and Salazar-González, 2014). In our study, we propose a transportation service similar to shared taxis. However, in our case, the vehicles in the fleet are autonomous. In addition, dynamic vehicle routing can be defined as the problem of dispatching vehicles to serve the trip requests revealed dynamically in real-time. The trip requests are highly dynamic in our studied problem since these requests were taken from a mass transit mode. In this sense, our studied problem can be defined as a dynamic dial-a-ride problem.

1.3. Why Autonomous Vehicles

Autonomous Vehicle (AV) technology is a breakthrough in the transportation domain. The inclusion of AVs in transportation networks is expected to have various impacts on the transportation industry (Wen et al., 2018). In recent years, online mobility-on-demand services have gained a significant share from conventional transportation modes (Rayle et al., 2016). It is anticipated that with the advances of AV technology, these services will attract more share in the transportation systems of major cities. Since they have a great potential to be utilized instead of certain types of public transportation modes, it will affect the ridership and general viability of PT. Instead of considering the AVs as a threat to PT or other transportation modes, it should be perceived as a great opportunity to increase the efficiency of the present transportation networks.

Integrated public transportation with autonomous vehicles is first discussed by Lenz and Fraedrich (2016). They illustrated the AVs as a service to increase the options of public transportation in less congested areas. Shen et al. (2018) utilized a simulation to examine the potential of AV services as a supporting system to bus operation and planning. Their study showed that an integrated transportation network would increase the efficiency of the process for both AV operators and public transportation agencies. Yan et al. (2019) focused on ride-sharing services like Uber and Lyft, with PT for first and last-mile transportation. They also studied the potential of these services as a substitute for low-demand routes. Wen et al. (2018) proposed a transit-oriented AV operation for integrated autonomous vehicles and public transportation systems. They focused on the opportunities between AV and PT when the AVs are utilized to support the existing PT network.

Several semi-autonomous features, from parking assistance systems to lane-keeping assistance systems, have already been utilized in vehicles. However, fully autonomous vehicles are not widely used in the transportation domain. With the utilization of fully autonomous vehicle technology, AVs will have significant potential benefits.

1.3.1. Capacity Gain

AVs are expected to have a shorter reaction time compared to human drivers. When the roads are occupied by only fully autonomous vehicles, the advantage of shorter reaction time will result in capacity gains on highway network up to %370 and on urban road network up to %80 (Brownell, 2013; Fernandes and Nunes, 2010).

1.3.2. New Users

With the elimination of human drivers, those who are not allowed or able to drive vehicles will have the chance to travel independently with AVs. In this sense, elderly people, adults without a license, unaccompanied children, people with mobility-restricting disabilities, and some other passenger groups will benefit from the advantage of AVs.

1.3.3. The Financial Burden of Private Car Ownership

While private car ownership comes with great comfort in the transportation experience compared to public transportation, it also has a significant financial burden apart from the purchasing cost such as insurance, repair costs, and taxes. For families owning multiple vehicles, these costs occupy a great portion of their yearly expenditures. AVs can provide the same comfort level by door-to-door transportation service without the costs related to private car ownership.

1.3.4. Non-driving Activities

Another advantage of AVs is the opportunity to perform non-driving activities. Because of the need for human control in traditional vehicles, drivers must focus on the road and the vehicle. The necessity of human control prevents drivers from enjoying any non-driving activities. With fully autonomous vehicles, drivers will become passengers in the vehicle and can utilize the trip duration with various activities such as reading a book, watching a movie, etc. With this advantage, AVs combine the comfort in private cars with the opportunity to enjoy non-driving activities in public transportation and serve better transportation service comfort-wise.

1.3.5. Traffic Congestion

It is predicted that a 33 percent increase in the world population by 2050 will have an impact on an increase in the number of cars. It is expected that around 1 billion additional cars will hit the roads (Voelcker, 2014). An increase in the number of vehicles with this magnitude will directly result in a need for extra infrastructure. Considering that the population will be more concentrated in cities in the future, the need for better transportation infrastructure will be a challenging problem for urban areas.

1.3.6. Parking Cost

In AV technologies, it is assumed that the vehicle will maneuver itself from its original parking lot to the passenger and vice versa (Lenz and Fraedrich, 2016). This means that the effort of passengers to reach and park the vehicle is eliminated in AVs. These vehicles can be moved to the place where it is demanded and leave the passenger at their destination, thereby providing a door-door service. With the utilization of AVs, land use for parking space in city centers will be considerably decreased (Heinrichs, 2016).

1.3.7. Safety

According to World Health Organization (WHO, 2018), road injuries are the 8th leading cause of death overall with the death of 1.3 million people each year. It is also the leading cause of death for people aged between 5-29 years. Furthermore, human errors, by far, the biggest reason behind these accidents. The study of the National Highway Traffic Safety Administration (NHTSA) concludes that over 90 percent of accidents are related to human errors (NHTSA, 2008). Since AV technologies focus on eliminating human drivers, AVs' utilization in transportation networks is expected to dramatically decrease crash rates. However, it is also possible that trust in AVs may encourage pedestrians or bicyclists to take more risks on roads (Kockelman et al., 2016).

1.3.8. Compliance with Fleet's Manager Operational Plans

Operational policies for a fleet are made for efficient fleet management. However, it is hard to force drivers to obey operational policies. Since in taxi or ride-sharing services, drivers have the autonomy to some extent, they tend to choose what is best for them among the alternatives and may ignore the overall efficiency of the fleet, full compliance with the operational policies of the fleet manager is hard to achieve. With a fleet of AVs, complete control over the vehicles can be supported, and the fleet manager can successfully implement policies that potentially increase the overall efficiency of the entire fleet. In order to achieve this, some AVs may need to serve in areas with low demand, where in a taxi example, no driver prefers to spend time in such places (Hyland, 2018).

1.3.9. Adapting to Pandemic Conditions

Because of the Covid19 pandemic, social distance in public transit has become very important for passengers. People fear being infected due to their interaction with other passengers. However, it is hard to find a public transportation system that offers the required personal space to its passengers. In this sense, shuttle or taxi-like vehicles are expected to occupy a bigger portion of the public transportation system. Moreover, since the drivers of buses or other PT vehicles interact with so many people on daily basis, the risk for them to be infected is exceptionally high. With the help of autonomous vehicle technology, the need for a human driver will no longer be present, and all the risk factors for the drivers will be eliminated.

1.3.10. Challenges with Autonomous Vehicles

Although there are several technical advantages of AVs, there is also a wide range of challenges that need to be eliminated in order to make AVs available to the general public. For instance, Lidar systems used by most AV companies are still very expensive while alternatives like using regular cameras in tandem with image recognition software provide a lower level of precision. Another difficulty arises from the bad weather conditions. Since the AVs observe the surrounding environment via cameras or sensors when the sight of AV is blurred by rain or snow, they become inoperable. Even in conditions where the sensor itself is not obscured but the lane dividers are AVs that cannot track the lanes. The computer systems that are integral to AVs are also vulnerable to hacking attacks that can lead to collisions and other costly disturbances in crowded cities.

Aside from the technical challenges, legislative problems have not been solved yet. When an accident occurs, there is no consensus about who is going to be the liable party: passenger, manufacturer, or software developer, etc. After all these challenges are resolved, people might still hesitate to use this new technology due to the resistance to adapt to new technologies. Several researchers have studied the key factors from safety to perceived usefulness that affect the adoption of AVs (Xu et al., 2018; Howard and Dai, 2014).

1.4. Presentation of Problem Studied

Since the conventional transportation modes of PT fail to provide a fast and flexible experience to passengers for the last mile, there is a need for new transportation concepts (Scheltes and de Almeida Correia, 2017). In order to have full integration of PT, these new transportation concepts should serve efficient transportation service that eliminates the disutility experienced in the conventional last-mile transportation modes.

This study aims to propose a transportation service that can replace the conventional bus routes operating in the last mile of public transportation. To achieve this, it is necessary to develop the models and solution methods for the routing problem that arose in the proposed service's execution. The characteristics of the proposed service are summarized as follows:

- The vehicles in the fleet are autonomous.
- The fleet is homogenous. All the vehicles have the same functional characteristics related to the service.
- Passengers should be delivered as soon as possible.
- Shared rides are allowed.

The proposed service offers point-to-point transportation. For this kind of problem, passengers make trip requests with their origins and destination of the trips. The optimization problem has two steps: assigning vehicles to passengers and generating the routes for each vehicle. This problem is defined as a dial-a-ride problem in the VRP literature. The problem consists of several constraints, and the relevant ones to our problem can be summarized as follows:

- Each passenger must be served. This constraint is relevant for the cases where the rejection of the trip request is not allowed.
- Each passenger must be picked up and delivered by the same vehicle. Passenger transfer is not allowed.
- Passengers must be picked up before being delivered.
- The number of passengers assigned to a vehicle cannot exceed the predefined vehicle capacity.

To generate the problem mentioned above, time, origin, and destination of trips are needed. For this purpose, the origin and destination of trips made in bus routes operating in the last mile are inferred. These origins and destinations and the time of the requests are used in the studied routing problem. An agent-based simulation framework is developed to evaluate the performance of the solution methods developed for the problem. The simulation is coded in Python 3.9. The solution algorithms used for the passenger-vehicle assignments and routing are embedded in the simulation.

1.5. Contributions of the Dissertation

The main contributions of the study can be summarized as follows:

- By extending the assumptions made in previous OD studies, a robust trip chaining algorithm is developed to find the OD pairs of the bus passengers.
- To the best of our knowledge, none of the previous studies used inferred OD pairs of bus passengers at an individual level for their routing problems. This study used real OD pairs of bus passengers for the routing problem.

- The study proposed a point-to-point transportation service containing a fleet of shared AVs that is designed to replace conventional bus routes operating in the last mile of PT.
- Exact and heuristic methods are widely utilized for solving routing problems. However, most of the previous studies used synthetic data for their research. The complexity of real-life problems cannot be fully perceived with synthetic data. In this study, the performance of the solution methods is tested with real data in a highly dynamic setup.
- We have introduced a new constructive heuristic method for the solution construction of the dynamic DARP. This method can be implemented in other solution methods or other variants of VRP.
- The neighborhood structure in improvement heuristics is designed to limit the computational effort required. With this method, the computational time of the related operators is kept in a reasonable range. By achieving this, the proposed algorithm becomes suitable for a dynamic environment.
- An agent-based simulation framework is constructed to evaluate the performance of the solution methods.
- The proposed model and the solution algorithms can be applied in present transportation systems. Although the vehicles are designed to be autonomous in our research, the conventional vehicles with drivers can be operated by the proposed model, and the solution methods can be used to solve the related routing problem.
- We have shown the economic feasibility of the proposed model in terms of both capital and operational cost.

1.6. Overview of the Dissertation

The rest of the dissertation is organized as follows:

- Chapter 2 overviews the literature for the VRP and its variants. The elements of VRP related to our studied problem are described in detail. Operational decisions which can be made during the execution of the proposed transportation service are discussed. Solution methods used to solve the VRP and its variants in previous studies are shown.

- Chapter 3 investigates the previous methods and the assumptions made in these methods for the OD inference of PT users. Further assumptions made in this study and some methods developed for the destination inferences are described in detail.
- Chapter 4 formulates the studied routing problem. The assumptions made for the related problem are shown. Constructive and improvement heuristics developed for the related problem are described in detail. The pseudo-codes for the solution methods are generated.
- Chapter 5 is dedicated to the elements of the simulation. Important parameters that affect the performance of the solution methods are analyzed.
- Chapter 6 examines the results of the simulation. The performances of the proposed solution methods are compared with each other in different settings.
- Chapter 7 finalizes the thesis by commenting on the main contribution of the study. The economic feasibility of the problem is analyzed in terms of investment and operational costs. In addition, some limitations of the studied problem are discussed in this chapter.

2. VEHICLE ROUTING PROBLEM

The vehicle routing problem is a delivery problem that many organizations and companies encounter daily while mobilizing goods or people. The VRP is a combinatorial optimization problem, and since the problem was first studied, there have been several exact and approximate solution methods developed by researchers. Like TSP, it is an NP-hard problem because VRP generalizes TSP.

Even though a huge literature and countless studies are focusing on VRP, researchers have still not reached a consensus on the definition of the VRP (Eksioglu, Vural, and Reisman, 2009). Especially for specific variants of VRP, different terminologies are used in the literature by various researchers.

2.1. Variants of VRP

Dantzig et al. (1954) published a paper introducing VRP to the literature as a generation of the Traveling Salesman Problem formalized by Flood (1956). TSP can be classified as a specific version of VRP. After this, various TSP papers were published. Clarke and Wright (1964) used multiple vehicles in their routing problem and generalized the problem such that it reflects the problems widely encountered in the logistics and transportation industry. After this classical formulation, various types of VRP were studied. One of the most common versions of VRP is the Capacitated Vehicle Routing Problem which vehicles have a limited capacity. Another widely studied variant of VRP is the Vehicle Routing Problem with Time Windows, where the customers need to be visited in a defined time frame. While some versions of VRP involve the transportation of goods, like Pick-up and Delivery Problem (PDP), others formalized the movement of passengers from their origins to destinations, such as Dial-A-Ride-Problem (DARP). Although the VRP has been studied extensively, and much progress has been achieved since it is first introduced, it is still a challenging research area (Ritzinger, Puchinger, and Hartl, 2016).

2.1.1. Classical Vehicle Routing Problem

The classical VRP, known as Capacitated Vehicle Routing Problem (CVRP), is designed to search for the set of routes for a fleet of vehicles with the minimum cost. The routes begin and end at the depot while the vehicles visit each customer once. The capacity of vehicles is not exceeded, and the route length restrictions should be met. Most of the exact solution algorithms consider the capacity restrictions, while the others have been developed with distance constraints. On the other hand, approximate solution methods deal with both capacity and distance constraints (Cordeau et al., 2007). Capacitated Vehicle Routing Problem is constructed to explore the routes for a specified number of vehicles and find the set of routes that gives the overall optimum result (Elshaer et. al. 2020). In CVRP, all vehicles should start their trips from a depot and end up in the same depot.

2.1.2. Vehicle Routing Problem with Time Windows (VRPTW)

Another important generalization of the classical vehicle routing problem is the Vehicle Routing Problem with Time Windows (VRPTW). In VRPTW, each vehicle visits the customers within a defined time window. A vehicle is allowed to visit the customer before a certain time and wait, but the visits after that time are not allowed (Cordeau et al., 2007). There are two types of time windows in VRPTW, VRP with soft time windows and VRP with hard time windows. In VRP with soft time windows, late arrival to the customers is not prohibited but comes with a penalty cost (Iqbal, Kaykobad, and Rahman, 2015). However, in VRP with hard time windows, visits after that time are not allowed.

The VRPTW is used in various fields, from food distribution to industrial waste collection (Golden, Assad, and Wasil, 2002). VRPTW generalizes the CVRP; therefore, it is an NP-hard problem. Thus, the researchers have focused on heuristic methods to find approximate solutions to VRPTW. Nevertheless, it is possible to find the optimal solutions for VRPTW through mathematical programming when the time intervals are narrow and the problem size is small (Cordeau et al., 2007).

2.1.3. Dynamic Vehicle Routing Problem (DVRP)

In most of the practical applications of VRP, parts of the information about the problem become available during the execution. This type of VRP where some of the input data are revealed dynamically is called dynamic VRP (DVRP), also called online or real-time VRP. Commonly used dynamic components of DVRP are the time of customer requests, service and travel times, and demands of the customers (Ritzinger, Puchinger, and Hartl, 2016).

Since DVRP reflects numerous real-life applications, it is extensively studied. With the inclusion of dynamic inputs, the problem becomes more complex. Due to the complexity of the problem, sometimes it is not easy to find feasible solutions for certain trip requests. In some of those cases, rejecting some service requests is allowed.

In DVRPs, reaction time becomes highly important since the decisions should be made dynamically in a short time. However, it is hard to give a good solution quickly since it requires high computation time. Therefore, the balance between decision quality and reactiveness should be achieved. For this purpose, Sleator and Tarjan (1985) introduced the competitive analysis, and it has been used to measure the performance of the online algorithms by several researchers.

2.1.4. Pick-Up and Delivery Problem (PDP)

In pickup and delivery problems, transportation requests include both origins and destinations. These types of transportation services refer to door-to-door or point-to-point transportation. Transportation may require the movement of passengers or goods from a pick-up location to a corresponding delivery location. For the cases in which the transportation of goods is considered, the term pick-up and delivery problem (PDP) is used (Parragh, Doerner, and Hartl, 2008; Battarra, Cordeau, and Iori, 2014). When the problem deals with the transportation of people, the term dial-a-ride problem (DARP) is used (Doerner and Salazar-González, 2014). Most of the DARPs are originally designed to solve the routing problem of transporting patients, disabled or elderly people (Hyland, 2018). Time window constraints for pick-ups and deliveries are generally applied to the problem either in the objective function or with problem constraints (Irnich, Toth, and Vigo, 2014).

2.1.5. Stochastic Vehicle Routing Problems (SVRP)

In most real-world applications, while there is uncertainty about the problem, historical data is available in advance. With the help of this data, one can forecast future events based on one or more components of the problem that follow a probability distribution. This type of VRP is called stochastic VRP (SVRP). There are three common cases of SVRP, VRP with stochastic customers (VRPSC), VRP with stochastic demand (VRPSD), VRP with stochastic travel and service times (VRPSTS) (Elshaer and Awad, 2020). Compared to the deterministic VRPs, SVRPs are more complex and require additional components for the solution method. Therefore researchers introduce a priori optimization step for solving SVRPs (Cordeau et al., 2007).

2.2. Elements of Vehicle Routing Problem

The proposed transportation service in our study is similar to shared autonomous taxis, which fall under the dynamic dial-a-ride problem. To be more specific, the studied routing problem can be classified as a multi-vehicle, dynamic, dial-a-ride problem with soft time-window constraints and fixed fleet size. Related but not limited to our problem, VRPs include various elements.

2.2.1. Rejections of Trip Requests

Rejection of trip requests sometimes can be allowed when the fleet operator considers the revenue generated by the vehicles. In the cases of unprofitable trips, the operator may decline the trip requests of the passengers. Moreover, the rejection can be relevant for especially highly dynamic problems where the possibility of encountering infeasible solutions is relatively high.

Some studies allow the fleet controller to reject the trip requests during the execution (Seow, Dang, and Lee, 2010), while others aim to minimize the number of rejected trip requests (Saint-Guillain, Solnon, and Deville, 2017; Pureza and Laporte, 2008). Since the proposed transportation service in our study aims to replace a public transportation service, it is expected to serve all the trip requests. Therefore, in the formulation of the studied problem, rejection of trip requests is not allowed.

2.2.2. Objective Functions

While in static VRPs common objective is to minimize the total cost of routes, in dynamic versions of VRPs, some other aspects are included in the objective function, which generally reflects the service level or the total revenue (Pillac et al., 2013). Also, some of the objectives, such as travel time or completion time, don't reflect meaningful parameters for the dynamic cases (M. W. P. Savelsbergh and Sol, 1995). In this sense, objective functions in dynamic VRPs are frequently distinguished from the objective functions in static VRPs (Psaraftis, 1995).

Even within the dynamic VRP literature, researchers used various objective functions. Bent and Van Hentenryck (2004) generated an objective function that maximizes the number of served passengers. Branke et al.(2005) aim to maximize the probability of new customers being inserted into fixed tours. Pureza and Laporte (2008) proposed an objective function that aims to minimize the number of rejected passengers.

The objective function used in the studied problem contains two parameters: total traveled distance and waiting time of the passengers. While the first one reflects the cost of the fleet operation, the latter stands for the service quality. It should also be noted that the waiting time of a passenger is defined as the gap between the service duration of the passenger and the travel time of the shortest path between the origin and destination of the passenger. The service duration starts with the passenger's trip request and ends with the delivery of the passenger. This definition challenges the solution method to find solutions close to the optimal solutions for passengers.

2.2.3. Time Window Structure

In the taxonomy proposed by Eksioglu et al. (2009), there are three types of time window structures: soft, strict, and mixed. When time windows for trip requests are applied, the vehicles should serve the trip requests within predefined time windows. On the other hand, there is a difference between applying strict and soft time windows to the studied problem. While strict time-window structure requires additional constraints in the problem, soft time-window structure is generally reflected in the objective function by including related parameters.

Our problem, which contains shared AVs, can be formulated with a strict or soft time-window structure. However, the problem has several constraints due to its characteristics. In addition to this, the dynamism and the size of the problem are considerably high. Introducing an additional constraint causes increased complexity hence requires extra computational effort. In order to avoid this, a soft time-window structure is used in this study, and the waiting time parameter is included in the objective function. This parameter is related to the service quality of the proposed transportation system.

2.2.4. En-Route Diversion

Some variants of routing problems allow vehicles to divert from their planned route and serve new requests. Especially in dynamic variants, allowing the diversion of en-route vehicles may contribute additional savings due to the possible assignments to the closer passengers that become available to the fleet controller during the execution of the routes. Lorini et al. (2011) argue the benefits of the en-route diversion based on the computational results of benchmark instances.

To find better solutions with the new requests that arrived dynamically, a planned route, which is the solution for the previous iteration's routing problem, should be decomposed, and reassignment should be carried out. However, reassignment of trip requests to the vehicles; in other words, diversion of en-route vehicles increases the studied problem's complexity and solution space.

Our problem is highly dynamic, and the diversion of the vehicles upon the new requests can change the solution quality significantly. Therefore, the diversion of vehicles or reassignment of the trip request is allowed for the studied problem. A solution generated in a decision epoch is decomposed in the next iteration, and a new solution is created. After reassigning the revised trip requests to the vehicles, the proposed algorithm aims to find optimal routes for each vehicle. For the passengers onboard, reassignment can not be made since the passenger transfer is not allowed in our study. However, the sequence of the onboard passengers' destinations can be rearranged with the revised assignments.

2.2.5. Evolution and Quality of Information

The evolution of information reflects variants of VRPs where some information becomes available to the planner during the execution of the service. Quality of information relates to the uncertainty on the available data (Pillac et al., 2013). According to this classification, there are four categories of vehicle routing problems shown in Table 2.1. Pillac et al. (2013) and Ho et al. (2018) used similar taxonomies for the reviews of dynamic vehicle routing problems and dial-a-ride problems, respectively.

Table 2.1. Taxonomy of routing problems by evolution and quality of information.

		Information Quality	
		Deterministic	Stochastic
Information Evolution	Known beforehand	Static and deterministic	Static and stochastic
	Changes over time	Dynamic and deterministic	Dynamic and stochastic

In static and deterministic problems, all the input used in the problem is known beforehand in such a way that routes of the vehicles can be determined before the execution. Once the execution starts, none of the input is changed. In static and stochastic routing problems, inputs are partially available to the planner. Also, only minor changes in the planned route can be accepted during the execution.

For the category of dynamic and deterministic, the planner has no distributional information to exploit. Pillac et al. (2013) state that exact information becomes available to the planner during execution in dynamic and deterministic problems. On the other hand, it is argued that deterministic routing problems cannot be dynamic. Only stochastic problems satisfy the characteristics of a dynamic problem (Lahyani, Khemakhem, and Semet, 2015).

In dynamic and stochastic VRPs, related problem inputs become available to the planner in real-time. On the other hand, the planner can exploit the available data on the studied problem when repositioning and routing vehicles. For example, although the real locations of the trip requests' origins and destinations are known during the execution, the fleet controller can take necessary action based on the spatio-temporal distribution of the previous trip requests in the studied region.

The routing problem studied in this research is considered to be a dynamic routing problem since all related inputs like request times, origin and destination locations, etc., reveal in real-time. Furthermore, the trip history of the studied bus routes is known beforehand and utilized in certain steps of the service execution. In this sense, our studied problem is under the umbrella of dynamic and stochastic routing problems.

2.2.6. Homogeneity of Vehicles and Users

Bodin and Golden (1981) categorized the type of vehicle fleet in two categories: homogeneous and heterogeneous. The vehicle fleet becomes heterogenous when the vehicles in the fleet are differentiated by the capacity or other characteristics (Ho et al., 2018). Most of the studies considering heterogenous vehicle fleets are motivated by the fact that transportation service providers need to acquire vehicles with different capacities and equipment to serve people with various limitations in real-life applications. The formulation of the routing problem is highly dependent on the characteristics of the vehicle fleet. The increase in the complexity of the problem challenges solution methods developed for the problem.

Due to the characteristics of the problem, some studies considered user heterogeneity. The introduction of different types of users is sometimes motivated by the operation strategies advantageous to the fleet operator (Molenbruch et al., 2017). User heterogeneity may also be needed due to the requirements in the studied problem (Ilani et al., 2014).

The problem in this dissertation aims to solve the routing problem for the trips made in public transportation. Since PT users don't differentiate in terms of their needs in transportation services, this study considers homogeneous vehicle fleets and users. One can argue that for disabled passengers, vehicles in PT should be equipped properly. Even in this case, instead of equipping some vehicles in the fleet, all of the vehicles should be designed to serve all PT users.

2.2.7. Road Network

The road network is one of the most crucial components of VRP that generally consists of arcs and nodes. The origins and destinations of the passengers and depots can be represented by the nodes, while the arcs can represent the roads. Some of the VRP studies use virtual networks which don't represent real road structure. These networks are simplistic and introduced by the researchers to test their solution algorithms for the related VRP. On the other hand, some studies used real road networks with the nodes representing the intersections and the arcs representing the roads that connect the nodes. On the other hand, some of the real networks have additional nodes in the arcs besides the nodes in the intersections. These additional nodes increase the precision of the studied networks; hence the studied problem reflects the real routing problem more. A real road network is used in our study, and additional nodes are included in the network to increase precision.

2.2.8. Ridesharing

Shared rides refer to the trips where the vehicle transports more than one passenger. After the launch of several shared mobility services like Car2Go, Zipcar, and others, the concept of shared mobility has improved its popularity in recent years with the increasing attention from academia and the transportation industry (Mourad, Puchinger, and Chu, 2019). Therefore, several VRP studies that deal with DARP consider ridesharing in their routing problems. In order to have the benefits of the shared rides, the vehicles are assumed to be shared by the passengers. It should also be noted that it seems almost impossible to introduce such a fleet that can serve the demand of the conventional busses with the limited number of vehicles.

2.2.9. Repositioning of Idle Vehicles

In some stochastic vehicle routing problems, spatio-temporal trip demand can be estimated based on the trip history of the service region. In these examples, it is possible to reposition the vehicles in such subregions where the future trip demand is expected to be higher than the presently available vehicles in that subregion. This strategy can significantly improve the overall efficiency of the vehicle fleet. On the other hand, it requires stochastic information of the trip demand and computational efforts in the fleet's operation. In this study, repositioning of the vehicles in the fleet is not considered to eliminate a further increase in computational effort.

2.2.10. Passenger Transfers

In most of the studied DARPs, passengers stay in the same vehicle for the entire trip. Some of the recent studies consider the possibility of transferring passengers between vehicles. Although transferring a passenger at a transfer point requires additional constraints hence increases the complexity of the studied problem, researchers were motivated by the advantages that can be gained by passenger transfers to improve the overall efficiency. Even if some passengers need to wait at the transfer points, and their journeys become less direct, significant savings can be achieved in the total trip duration and total traveled distance (Cortés, Matamala, and Contardo, 2010; Masson, Lehuédé, and Péton, 2014).

As stated above, considering passenger transfers increases the number of constraints and computational efforts needed to solve the studied problem. In our study, due to the problem size and the dynamism of the trip requests, solution algorithms require a high computational time. Therefore, the additional complexity in the problem is avoided. Moreover, the studied network is relatively small, and the introduction of transfer points in the network is not expected to give significant savings in the parameters of the objective function.

2.3. Solution Methods

A crucial factor for successfully solving sophisticated vehicle routing problems (VRP) is to offer reliable and flexible solutions. Due to the characteristics of VRPs, exact solution algorithms can only solve cases where the problem sizes are considerably small. Because the real routing problems we face in our daily life often exceed the sizes that the exact algorithms can solve, researchers have tried to generate heuristics or metaheuristics for the related routing problems in their studies. Especially in dynamic cases, the computational time needed to solve the problem becomes extremely important. Therefore, finding a good solution in a reasonable time is much more applicable to the real-world problem than searching extensively for the optimal solution.

On the other hand, in the application of dynamic dial-a-ride problems, which is considered to be the class of our studied problem, some metaheuristic methods can be computationally demanding (Häll and Peterson, 2013). Although several metaheuristics were used for VRPs in previous studies, additional constraints and the degree of dynamism of our studied problem require high computational effort. Therefore, we mainly focused on constructive and improvement heuristics to get feasible and good solutions in reasonable computational time. We also developed a simulated annealing algorithm for the solution improvements to diversify the generated solutions. Additional parameters and steps are included in the proposed SA algorithm.

2.3.1. Constructive Heuristics

Heuristics used in constructing the initial solution often provide a starting solution to the improvement heuristics (Laporte, Ropke, and Vidal, 2014). These heuristics aim to build feasible and good solutions by inserting a new request into a new route or an existing route. There are two main types of solution construction in the VRP literature: sequential and parallel construction methods. In sequential construction, additional vehicles can be used when no vehicle can be assigned to further requests. In the parallel construction method, the number of vehicles in the fleet is predefined, and all available vehicles are open to user-vehicle assignments.

Although the metaheuristic methods are effective in finding a good solution in such a way that they can even start from randomly generated initial solutions (Laporte, Ropke, and Vidal, 2014), construction heuristics are useful to acquire feasible solutions quickly. The dynamic variants of routing problems require fast initialization of the solution. Hence it is helpful to use construction heuristics in dynamic cases of VRP.

One of the most used insertion heuristics is the greedy insertion heuristic proposed by Jaw et al. (1986). In this method, each request is inserted into a route where the cost of the insertion has the minimum value. Although this heuristic is fast and simple, finding the optimal position for a request may require relatively high computational time in large-sized problems.

Another method widely used in previous studies is the method that assigns the passenger to the closest vehicles (Solomon, 1987). While this method is considerably advantageous in terms of computational time, the quality of the generated solutions is relatively worse.

The researchers propose various other construction heuristics. However, the performance of the construction heuristics is highly dependent on the characteristics of the studied problem. Hence one should find the best constructive heuristics for the studied problem. As discussed in Chapter 4, we proposed a constructive heuristic applied to the problem in the passenger-vehicle assignment phase. The heuristic searches available vehicles for a request within predefined radius intervals then makes the user-vehicle assignment.

2.3.2. Improvement Heuristics

Improvement heuristics are mostly used after the construction of solutions. Improvement heuristics used in VRPs contain two types of moves: intra-route and inter-route moves. Intra-route moves aim to improve the route of a particular vehicle by changing the order of the passengers' origins and destinations. For intra-route moves, heuristic methods used for TSP can be applied for VRPs also (Laporte, Ropke, and Vidal, 2014). Clarke and Wright (1964) proposed a saving heuristic that uses intra-route moves. This method is widely used to construct an initial solution for more advanced algorithms. Inter-route improvements, on the other hand, move the users from one route to another. Most of the studies combine these inter-route and intra-route heuristics to get better solutions.

One of the most used intra-route heuristics is the 2-opt search proposed by Lin (1965). This method can also be described as a λ-opt mechanism where λ refers to the number of edges that will be removed from the route. 2-opt search gives satisfactory results in a reasonable computational time. Potvin and Rousseau (1995) modified the 2-opt heuristic in a way that the exchange is made between different routes.

Various inter-route improvement methods were introduced by Savelsbergh (1992). These methods were further used by Prosser and Shaw (1996) for the VRPTW. One of these methods is the re-locate operator, which can be defined as removing a certain number of customers from their assigned vehicle and inserting them into other suitable routes. Another method is to exchange or swap in which passengers are swapped between the routes of different vehicles. However, it is important to note that intra-route improvement heuristics should be implemented within or after the inter-route heuristics to reach good solutions.

In improvement heuristics, two alternative methods are used for accepting the solutions, *best-improve* and *first-improve* strategies. In the first strategy, the heuristic searches the entire neighborhood to find the best improvement. The second strategy accepts the first better solution that is generated in the neighborhood of the current solution. While the first method gives better solutions, the second method has computational advantages. It is also possible to be stuck in the local optima in the best-improve strategy.

In our study, the routes of vehicles are solved mainly by exact methods. Therefore, there was no need to utilize intra-route improvement heuristics. On the other hand, for inter-route improvement, three different move strategies discussed in Chapter 4 were determined and tested with real data.

2.3.3. Metaheuristics

Metaheuristic algorithms are designed to find good solutions for sophisticated optimization problems in a reasonable time. The main goal of metaheuristic algorithms is to find near-optimal solutions instead of finding the optimal solutions. VRP is an NP-hard problem, and therefore, metaheuristic algorithms are widely used to solve it. They contain operations for both intensification and diversification. For the sake of diversification, most of the metaheuristic algorithms allow the moves that make the solution worse. This method allows the algorithms to explore a larger portion of solution space. Metaheuristics can be broadly classified into single-solution-based and population-based heuristics. While the population-based metaheuristics focus more on diversification, single-solution-based methods focus on intensification (Ho et al., 2018). In this research, we focused on the single-solution-based metaheuristics as these methods have given promising results in the previous VRP studies. One major advantage of these methods, especially for dynamic variants of VRP, is the ability to give good results in a short time.

The Tabu search (TS) method widely used in VRP studies was first introduced by Glover (1986). A tabu list that reflects the short-term memory is utilized to track the modifications made in the solution. The list prevents the algorithm from visiting the solutions that were found in previous iterations. Cordeau and Laporte (2003) used this method for a DARP example and introduced various diversification strategies, accepting infeasible solutions and penalizing the repeated moves. Due to its effectiveness and efficiency in DARPs, many researchers adapted TS in their studies (Beaudry et al., 2010; Ho and Haugland, 2011; Detti, Papalini, and Lara, 2017; Paquette et al., 2013; Guerriero, Bruni, and Greco, 2013; Pandi et al., 2018; Torgal, Dias, and Fontes, 2021). These studies adapted TS to handle more complex problems with real-life constraints.

Simulated annealing is a local search metaheuristic algorithm that has been a popular technique for optimization problems due to its convergence properties and ease of implementation (Nikolaev and Jacobson, 2010). The method is based on an analogy to the annealing process of solids in which a solid is heated to a certain temperature and cooled gradually until its regular crystal configuration is achieved. If the cooling process is arranged in a sufficiently slow manner, the final structural integrity of the solid is much superior. SA algorithm proposes an analogy between this thermodynamic behavior and the search for optimum solutions. At each iteration of the SA algorithm, the current solution and newly generated solution are compared. If the new solution is an improving solution, then it is accepted as the new current solution. Otherwise, the solution is accepted with a probability of $e^{\Delta/T}$ where the Δ is the variation between the objective functions of the current and new solution. This allows some portion of non-improving solutions to be accepted for the sake of escaping the local optima of the current solution. As the temperature decreases in each iteration, the probability of accepting non-improving solutions converges to zero.

The SA algorithm is used for several variants of VRP (Osman, 1993; Chiang and Russell, 1996; Kuo, 2010). Although it has not been used for DARPs as widely as other metaheuristic methods (Ho et al., 2018), some researchers utilized a standard SA using basic neighborhood structures in their related DARPs (Mauri, Antonio, and Lorena, 2009; Reinhardt, Clausen, and Pisinger, 2013) and obtained satisfactory results. Braekers et al. (2014) modified the standard SA algorithm in a way that non-improving solutions are accepted if the variation in the objective function doesn't exceed a predefined threshold. Yu et al. (2018) also proposed a new SA algorithm to solve the share-a-ride problem that is similar to DARP, and showed that their algorithm outperformed the basic SA algorithm.

3. ORIGIN AND DESTINATION INFERENCE

PT ridership has been traditionally measured by regularly counting the number of boarding and alighting passengers at each stop. In order to validate and supplement these measurements, transit agencies periodically conduct surveys (Gordon, Koutsopoulos, and Wilson, 2018). With the increasing usage of Automated Data Collection (ADC) Systems, transit agencies have gained various transportation data that can be utilized for a wide range of applications. ADC systems have been shown to provide sufficient data for origin and destination (OD) inference. With the utilization of these technologies, the disadvantages of conventional data collection methods like small sample size, high cost, biased samples, etc., were eliminated.

The origin and destination matrix of public transportation users can be utilized in various ways. OD matrices can provide an insight into the performance of the bus routes. The number of passengers onboard, maximum loads over the route, interchange locations, average interchange duration, and various other information can be acquired from the inferred OD matrices. These metrics guide the transit agencies in service design, bus scheduling, route optimization, etc. Therefore, OD matrices provide valuable information for transit agencies to improve the service quality of public transportation systems.

To infer the origins of passengers, Automated Fare Collection (AFC) data and the Automated Vehicle Location (AVL) were used. The AFC and AVL data were matched by the time stamp to detect the boarding location of the passengers (Trépanier, Tranchant, and Chapleau, 2007; Cui, 2006). Many researchers aimed to infer the destinations of the passengers in their studies by setting up several rules and determining the trips that met these predefined rules. Barry et al. (2002) assumed that the destination of a rail trip is the nearest station to the origin of the next trip and the last trip of the day ends at the closest stop to the boarding of the first trip of the day. Zhao et al. (2007) and Wang et al. (2011) used similar assumptions to infer the rail destinations and bus destinations, respectively. Munizaga and Palma (2012) utilized a generalized cost function that uses the disutility of time and distance to infer bus destinations. Sánchez-Martínez (2017) extends the generalized-cost approach to rail networks and uses dynamic programming to infer transfers on multileg journeys. Jung and Sohn (2017) used a supervised machine learning model to infer the origins and destinations of bus passengers based on land-use characteristics and smartcard data.

OD inference studies in the literature offered a wide range of methods to infer the OD matrices of passengers. However, most of these studies have not utilized the OD matrices for further studies. In our research, OD pairs of the PT users were used in another problem, dynamic DARP. One of the aims of this study was to analyze the feasibility of AV fleet being an alternative to conventional public transportation bus routes. To achieve this, information on the PT trips of the passengers at an individual level must be known. With the inference of OD matrices at an individual level, each trip's origin and destination locations were gained. The OD matrices also contain the date and time of the trips. With the time, origin, and destination information of trips, it is possible to simulate the trips with different transportation modes.

The studied bus routes are operating near the university region. 9 different bus routes shown in Table 3.1 were analyzed. These trips were made between October 1, 2019, and December 31, 2019. However, OD pairs from four of the studied bus routes were used in further phases of this thesis because only these bus routes have the characteristics that suit the studied problem. Routes of the studied bus routes are shown in Figure 3.1.

Table 3.1. Number of trips and GPS data of the studied bus routes.

Bus Route ID	**Total Trips**	**Total GPS Data**
4/G	468,041	648,561
35/G	453,267	658,869
1/T	345,003	580,732
3/G	239,324	670,962
E/12	160,651	570,695
E/13	136,391	574,391
B/33-K	100,313	343,780
43/D	79,683	419,730
B/35	4,222	65,207
Total	**1,986,895**	**4,532,927**

Although the number of studied bus routes is low compared to the total number of bus routes operating in the city, searching certain information like the trips made before and after the studied trips, was necessary. Therefore, over 180 million GPS data and 60 million trip data were analyzed. After making the required analysis on big data, trips made on the bus routes operating around the university area were obtained. A total number of 1.9 million trips were determined. From the AVL data of the buses that were used in the related bus routes, over 4.5 million GPS data were obtained.

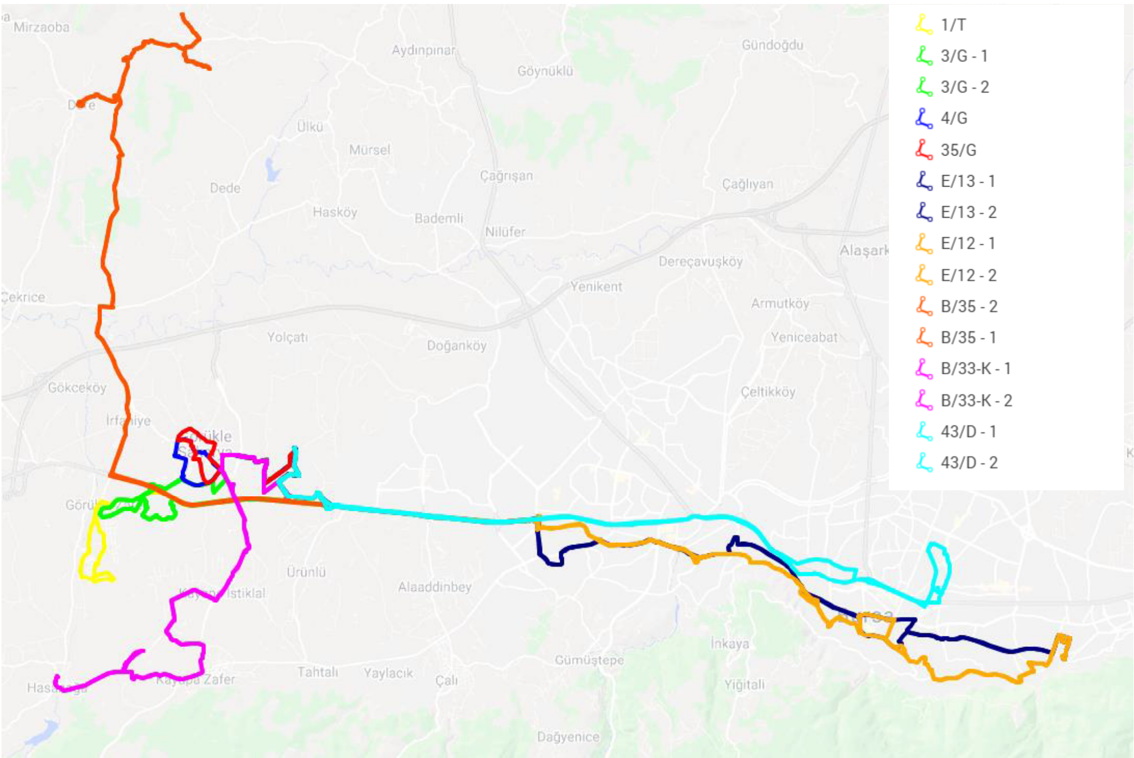

Figure 3.1. Routes of the studied bus routes.

3.1. Boarding Data

Boarding data analyzed in the study contains the trips from October 1, 2019, to December 31, 2019. These data were collected via Automated Fare Collection (AFC) Systems when the passengers use the smartcard while boarding the bus. The boarding data has the following information:

- Time and date of the trip

- The ID of the bus route
- Card ID of the passenger
- Vehicle ID
- Bus route ID of the previous trip
- Vehicle ID of the previous trip

3.2. GPS Data

The AVL system tracks the vehicles in the fleet dynamically. The locations of the vehicles are detected and recorded in a high frequency. Because of the need for precision on vehicle location data, the size of the GPS data reaches high numbers when the studied time becomes too long. Hence computational efforts for the operations made on GPS data might be considerably high when dealing with long periods. In our GPS data following information was available:

- Date and time of the data
- The ID of the bus route
- Vehicle ID
- Position of the vehicle by latitude and longitude

3.3. Origin inference

In most of the studies, AFC data contains the information of the boarding location of the passenger. This information can be achieved by matching the boarding data with the GPS data of the bus. The timestamp in the boarding data is determined and searched through the GPS data of the bus. The closest timestamp in the GPS data is selected, and the location information for that timestamp is taken.

However, in our study, origin inference required further analyses because of the data structure. The data acquired from the GPS data of the buses are the latitude and longitude of the buses. These GPS data are needed to be converted to bus stops along the related bus route. To achieve this, the positions of the bus stops should be known. After that, the nearest bus stop to the acquired vehicle position was assigned to the vehicle. This assignment could be done for only timestamps in the boarding data or all of the GPS data of the vehicle. In our study, GPS data don't contain the direction of the buses. To infer the correct boarding stop of the passenger, it is necessary to know the direction of the bus. To detect the directions of the buses, all GPS data of the buses were analyzed in our study.

It is not possible to determine the direction of the vehicle by a single location data. The previous and the following locations of the bus are also needed to infer the direction of the bus. From the sequence of the locations vehicle's movement can be detected. For this reason, GPS data were analyzed along the time on the studied day. For every GPS data, the closest bus stops in both directions were determined. Then, the direction which gives an increasing order was selected as the actual direction of the vehicle. This process was carried out for every vehicle that operated in the studied bus routes on the studied dates.

3.3.1. Errors in GPS Data

The boarding locations and the inferred destination locations are all based on the positions that were acquired from the GPS data. Therefore, even a small error in GPS data may result in an incorrect inference of the OD pairs. In our study, we have experienced various types of errors resulted from GPS data. Several methods were introduced to eliminate these errors. For simplicity, only one of them will be stated in this thesis.

The GPS data used in our study sometimes failed to give accurate locations of the vehicle. In some cases, the location data taken from the GPS data was wrong but considered close to the vehicle's actual location. This error can be detected after further analysis of these data points. The data points that were considered to be wrong were compared to previous and next GPS data of the vehicle. From the positions in the GPS data, the movement of the vehicle can be easily determined. However, these incorrect data break the increasing order and the movement of the vehicle. If the algorithm doesn't contain any method to correct these errors, it generates another route from the locations where the wrong data refers. To tackle this problem, instead of getting the nearest bus stops to the GPS data, all bus stops within a certain radius were determined. This process produces not an exact bus stop but possible bus stops that can be assigned to the studied GPS data. A bus stop in the possible stops that does not break the vehicle's movement is selected and assigned to the related GPS data as the boarding stop.

3.4. Destination inference

To infer the OD pairs of transit users, the location information of the vehicles must be known. For this reason, transit agencies all around the world equipped their vehicles with Global Positioning Systems. The equipment in the vehicles produces Automated Vehicle Location (AVL) data. The location data of the vehicles are matched with the boarding or alighting data of the users. Hence the OD locations of the passengers can be inferred. However, very few cities have transportation systems that record both boarding and alighting stops of the users (Jung and Sohn, 2017). In most PT systems, passengers only swipe the card when boarding the bus but don't give any information when getting off the bus. Therefore, there is a need for further inference study to get the alighting locations of the users.

Previous studies used similar assumptions to infer the destinations of the passengers (Zhao, Rahbee, and Wilson, 2007; Cui, 2006; Trépanier, Tranchant, and Chapleau, 2007; Barry et al., 2002; W. Wang, Attanucci, and Wilson, 2011). These assumptions can be summarized as follows:

- There is no other transportation mode used between two PT trips.
- The walking distance of the passengers between two PT trips cannot exceed a predefined distance.
- The destination of the last trip of the day is the boarding location of the first trip of the day.

Fidanoglu (2015) further proposed the assumption that if there is a missing leg in the chained trips of the passenger that can be easily determined, the destination inference should be made from the missing trip. This assumption is valid for some instances where the passenger cannot physically make the recorded trips without introducing the missing leg. To detect that the missing leg of the trips should be obvious. However, in our study, we did not assume any missing trip besides the recorded trip.

To determine the destination of a trip, all relevant trip data of the same passenger should be determined. In this study, the following trips of the passengers were also studied:

- Previous trip of the passenger
- Next trip of the passenger
- First trip of the passenger on the day of the studied trip
- Last trip of the passenger on the day of the studied trip.

All of the above trips serve for further inferences of destination. For some studied trips, the above trips can not be obtained due to the lack of information. It is also possible to acquire such trips that are irrelevant to studied trips. For example, a subsequent trip made after five days of the studied trip doesn’t give any direct information about the trip's destination.

The destination inferences based on the next trips are the most reliable method if the next trip is made within a short time. Therefore, at the first stage of the destination inferences, the trips made within 2 hours after the studied trip were analyzed. If the boarding location of the next trip is close enough to any bus stop on the route of the studied bus route, the nearest stop is taken as the destination of the trip. The maximum distance was limited to approximately 500 meters for this method.

Next, inferences from the first trip of the day were studied. If the trip is the last trip of the day, it is checked whether the passenger has a different trip that is the first trip of the passenger on that day. If the first trip is different from the last trip of the passenger, meaning that the passenger made at least two trips on the studied day, the boarding location of the first trip is determined. Then, the distance between the boarding location and the closest bus stop of the studied bus route is calculated. If the distance satisfies the requirements, the nearest bus stop is considered to be the destination of the last trip of the day. The maximum distance for this method was set to approximately 300 meters. The main logic behind these inferences was that at the end of the day, passengers return to the locations where they started their trips on that day. Therefore, the maximum distance was lower compared to the distance used in the previous destination inference method.

If the inferences from the interchange within 2-hours or the first trip of the day could not be achieved, next trips made within 24 hours after the studied trip are determined. The same procedure used in the first destination inference method is followed for these trips also.

Next, the proposed algorithm aims to infer destinations for the first trips of the day from the last trips of the day. If the boarding of the last trip is within a distance of 300-meters to any stop in the studied route, the closest stop to the boarding of the last trip is assigned as the destination of the first trip. In fact, the previous method includes destination inference from the last trip of the day if the next trip after the first trip of the day is the last trip of the day. However, in some cases, there are some other trips between the first and the last trips of the day. For some of those cases, destination inference can not be achieved from the interchanges. Therefore, this step is included in the proposed method.

For all of the destination inference methods, a stop correction method was introduced. This method aims to correct the inferred destination stop if a direction error occurs. For the bus routes that follow a circular route, some stops are very close to each other. When the closest stop to a certain location is determined for the destination inference, it is highly possible to select the wrong stop. In this case, a direction error will rise. To eliminate this problem, the introduced method searches the stops nearby and assigns the other bus stop if it resolves the direction problem.

In some of the proposed inference methods, the time between the studied trip and the trips that the destination inference was made from can be considered too long. However, for the PT users, this kind of trip pattern is not something unexpected. In addition, even if the time difference between the trips is too long, estimating the destination of the trip using other methods would require further assumptions. Therefore, it is assumed that if the passenger can be tracked within the routes of his trips, the destination inference is valid.

All of the above-mentioned methods that were used for the destination inference of the trips check the direction error. If the inferred boarding stop is behind the boarding stop of the studied trip in the direction of the bus, it is concluded that there is a direction error. In these cases, inferred destinations were not accepted.

3.5. Derived Destinations

At the end of destination inferences, some of the trips were still missing destinations. To infer destinations for these trips, inferred OD pairs were used. The OD matrix contains the ID for each cardholder. Therefore, it is possible to obtain the inferred OD pairs for each passenger. A data frame including the following information was generated:

- The ID of the bus route
- Card ID
- The direction of the bus route
- Inferred boarding stop
- Inferred alighting stop
- Number of trips

For every origin point, the probabilities of destination points were determined by the number of trips. For a trip missing a destination, the passenger Card ID is determined, and the above data frame is constructed. If the data frame contains the direction and the boarding stop of the studied trip, meaning that the passenger made a trip similar to the studied trip, the alighting stop is probabilistically inferred.

If the inference was not achieved from other trips of the cardholder, the overall distribution of OD pairs was used for destination inference. To achieve this, inferred OD pairs for the studied bus route were clustered into three groups of time zones. Time intervals of the categories were 06:00-10:59, 11:00-15:59, 16:00-23:59. This categorization was done because the patterns of the trips during the morning and evening hours were different.

Next, the time of the studied trip and the boarding stop was determined. From the inferred OD pairs, trips sharing the same time zone, direction, and boarding stop with the studied trip were extracted. A destination point from these pairs was probabilistically selected and assigned to the studied trip.

Previous studies evaluate the performance of the proposed algorithm by the inference rate. However, in our research, if the algorithm fails to infer a destination to a passenger from his or her trips, it searches for similar trips through the inferred OD pairs and assigns a destination to the trip. Therefore, at the end of the destination inference process, all trips have inferred destinations.

4. MODELING FRAMEWORK OF THE PROBLEM

4.1. Motivation and Problem Definition

The goal of the study is to design a transportation service that serves passenger requests most efficiently. The transportation service proposed in this study contains a fleet of AVs that is controlled by a centralized algorithm. Passengers make their trip requests with the origin and destination locations of their trips. These locations must be selected among the predefined bus stops. The request time of the passengers is also recorded and used in the model.

The fleet controller has stochastic information about the trips in the studied region. However, this information is used only for the initial positioning of the vehicles in the fleet. The system doesn't allow further actions to be made according to the origin and destination of the previous trips.

The main objectives of the system are to minimize the waiting time of the passengers and the total distance of vehicles. While total traveled distance by the fleet is directly related to the operational cost of the proposed service, minimizing the average waiting time of the passengers is the key factor that affects the performance of the service (Krueger, Rashidi, and Rose, 2016). However, it should be noted that these two parameters are related to each other. Increasing the total distance would increase the average waiting time of the passengers. Although these two parameters are not mutually exclusive, excluding one from the model would cause issues in service quality at an individual level.

In the proposed model, the vehicles are autonomous and shared by the passengers. Eliminating the driver has certain advantages, as described in previous sections. In the operation of the transportation service, the most significant advantage of AVs is that it enables a centralized decision-making system to be applied to the whole fleet. In this way, none of the vehicles can make their own decision thus, the overall efficiency can be increased considerably. The algorithm aims to optimize the overall system instead of focusing on specific vehicles or passengers.

Some variants of VRPs used the depots for the starting and ending points of vehicles. However, the introduction of a depot becomes highly impractical in dynamic routing problems. The vehicles travel throughout the studied region and don't use fixed depots. Besides, the related routing problem for the vehicle is needed to be resolved continuously. Due to the characteristics of our problem, vehicle depots are not included in the studied problem.

4.2. Characteristics of the Studied Model

The characteristics of the studied model can be summarized as follows:

- Pick-up and delivery locations must be specified by the passengers meaning that each passenger needs to have an origin and destination location.
- Origin and destination locations of the trip requests must be among the predefined locations. These locations are the stops of specific bus routes that operate in the studied area.
- The fleet controller has the trip history of the studied routes. However, this information can only be used to position the vehicles at the start of the operation. No other actions are made upon the stochastic information of trip requests.
- Centralized computing architecture is used for the studied problem. None of the vehicles or a bunch of vehicles can make their own decisions. A centralized algorithm optimizes the overall system with the information of vehicles and passengers.
- Time windows structure is designed as soft time windows that refer to the inclusion of the time window constraint into the objective function.
- Fleet size is fixed. The number of vehicles cannot be increased or decreased during the execution of the transportation service. If the demand exceeds the total capacities of the vehicles, passengers need to wait until some vehicles become available.
- Vehicles in the fleet are homogeneous. Characteristics and the capacities of the vehicles are assumed to be the same in terms of their functionality.

- The number of passengers assigned to a vehicle cannot exceed a predefined value. This capacity constraint for vehicles is imposed all the time. The capacity value cannot be increased or decreased during the operation.
- There is no maximum service time or distance for vehicles. Vehicles need to serve the passengers even if they need to carry out relatively long routes.
- The objective of the proposed model is to minimize the total traveled distance and the waiting time of the passengers.
- Rejection of a trip request is not allowed. All the trip requests must be served by the fleet eventually. The algorithm takes requests of the passengers in the very next decision epoch. Suppose a vehicle assignment to certain passengers couldn't be achieved in that decision epoch. In that case, the algorithm labels the passengers as "waiting" and tries to assign vehicles to them in the following decision epochs.
- All the trip requests are immediate. This means that passengers request the service as soon as possible. There is no reservation structure.
- Vehicles can be shared by passengers. The decision of sharing cannot be made by the passengers. The centralized algorithm decides which vehicles will be shared by passengers.
- En-route diversion is allowed. Passengers assigned to certain vehicles can be reassigned to different vehicles in the following decision epochs. However, reassignment cannot be done for the passengers onboard.
- A real road network is used. The real road network of the studied area is installed using the related libraries. The shortest paths between the nodes are determined on the real network considering the speed limits assigned to the roads.
- The system is reoptimized in predefined periods.

In order to simulate the defined problem, real OD pairs of bus passengers were used. OD pairs of bus trips were inferred by the algorithm explained in Chapter 3. This problem is considered to be a dynamic dial-a-ride problem.

4.3. Components of the Model

To model the proposed transportation service, a suitable framework should be constructed. Considering the fact that our problem is a highly dynamic routing problem, the modeling framework should capture the status of the vehicles and passengers over time and the evolution of these statuses during execution. To achieve this, the modeling framework should contain the following elements: state variables, decision epochs, exogenous information, decision variable, and transition function.

State variables refer to information needed to model the system. The status of vehicles, including the information of their availability and locations, should be tracked. Also, the state of passengers referring to the status of their trip request should be revised and recorded continuously.

Decision epoch can be defined as when the state variables are evaluated and the related decisions are made. In some studies, decision epochs are related to the system state. In these cases, the decision procedure is started when a predefined change is realized in the system. For other cases like ours, there is a certain number of decision epochs. When a predefined time is completed, the system collects the state variables and solves the related problem. This process is repeated until the predefined decision epochs are realized, or there are no unserved passengers.

The exogenous information becomes available at each decision epoch. It changes the state of the system. In our studied problem, the exogenous information only includes the trip requests made by the passengers between the previous decision epoch and the present decision epoch.

The fleet controller can control the operation using decision variables. These variables mainly contain the passenger-vehicle assignment and the determined route for the vehicle. However, these decisions should be made upon consideration of problem constraints.

Revision in the state of the system from the previous decision to the next one is defined by the transition function. State variables defined for the problem should be revised according to the decision made at the related epoch. The status of the passengers and the vehicles should be updated using the transition function.

4.4. Problem Formulation

The problem consists of a set M with m vehicles and a set N with n trip requests. Each vehicle $k \in M$ has an origin k^+ which is the location of a vehicle and capacity of q_k passengers. e_k denotes the earliest departure time of the vehicle that is also the end time of the decision epoch. Each trip request $i \in N$ has an origin i^+ and destination i^- with e_i being the request time.

$V = \{i^+, i^- | i \in N\}$ is set of all origins and destinations of all n trip requests. For trip request i which its origin is visited, i^+ is excluded from V. When a destination of a trip request is visited, i^- is excluded from V. $W = V \cup \{k^+ | k \in M\}$ is set of all points with total number $2n + m$. The points representing the same location in the network are denoted by distinct points. Let A denotes set of all arcs (u, v), $u \in W$, $v \in V$ with the condition of $u \neq v$.

p_w denotes the number of passengers getting in or off the vehicle at point $w \in W$. It gets a negative value in destination points. For the vehicle locations k^+, $p_w = 0$ since no passenger is being served at this location.

$R_k = \{u_1, u_2, \ldots, u_z\}$ is the route of a vehicle k. This is a sequence of points in the route sorted by the time of vehicles' arrival to the points. The first point in the route is the location of the vehicle k^+. Other points represent the origins and destinations of the passengers. If a trip request i is assigned to a vehicle k, then both origin i^+ and destination i^- must be in R_k, and i^- must be after i^+. $t^k_{u_{j-1},u_j}$ denotes the time of the shortest path between the points u_{j-1} and u_j. $c^k_{u_{j-1},u_j}$ denotes the cost of the shortest path between the points u_{j-1} and u_j which is the distance between these two points.

Let $B^k_{u_j}$ denotes the time of point u_j in the route R_k is served. $B^k_{u_1} = e_k$ and for the points in the route $B^k_{u_j} \geq B^k_{u_{j-1}} + t^k_{u_{j-1},u_j}$ for $j > 1$ where $t^k_{u_{j-1},u_j}$ denotes the time of the shortest path between the points u_{j-1} and u_j. The waiting time of a passenger is denoted by $s_i^k = B_{i^-}^k - e_i - t_{i^+,i^-}^k$ which can be defined as the gap between the service duration of the passenger and the travel time of the shortest path between the origin and destination of the passenger.

For the capacity constraint of a vehicle, the load of a vehicle $L_{u_j}^k$ cannot exceed the predefined vehicle capacity q_k. $L_{u_j}^k$ denotes the number of passengers in vehicle k after the point u_j. $L_{u_j}^k = 0$ for $j = 1$ in the first decision epoch, and for $j > 1$, $L_{u_j}^k = L_{u_{j-1}}^k + p_{u_j}$.

During the optimization, the following decision variables are used:

- $x_{u,v}^k$ is 1 if point v is just after point u in the route of vehicle k, 0, otherwise.
- y_i^k is 1 if request i is assigned to vehicle k.
- B_u^k is the time of the service at point u in the route of vehicle k. This variable is ignored in the case where u is not in R_k
- L_u^k is the number of passengers onboard after the service of point u in the route of vehicle k. This variable is ignored in the case where u is not in R_k
- α is the coefficient for the waiting time parameter.

Objective function:

$$min \sum_{k \in M} \left(\sum_{(u,v) \in A} x_{u,v}^k \, c_{u,v}^k + \alpha \sum_{i \in N} y_i^k s_i^k \right) \tag{1.1}$$

Subject to:

$$\sum_{k \in M} \sum_{(u,v) \in A} x_{u,v}^k \leq 1 \qquad \forall v \in V \tag{1.2}$$

$$\sum_{v \in V} x_{k^+,v}^k = 1 \qquad \forall k \in M \tag{1.3}$$

$$\sum_{v \in V} x_{i^+,v}^k - \sum_{v \in V} x_{i^-,v}^k = 0 \qquad \forall k \in M, \forall i \in N \qquad (1.4)$$

$$x_{u,v}^k = 1 \Rightarrow B_v^k \geq B_u^k + t_{u,v}^k \qquad \forall k \in M, \forall u, v \in W \qquad (1.5)$$

$$e_i \leq B_{i^+}^k \leq B_{i^-}^k \qquad \forall k \in M, \forall i \in N \qquad (1.6)$$

$$x_{u,v}^k = 1 \Rightarrow L_v^k \geq L_u^k + p_v \qquad \forall k \in M, \forall u, v \in W \qquad (1.7)$$

$$0 \leq L_v^k \leq q_k \qquad \forall k \in M, \forall v \in W \qquad (1.8)$$

To ensure that each trip request is visited once at most, constraint (1.2) is introduced. Constraint (1.3) guarantees that the position of vehicle k is in its route. Constraint (1.4) ensures that the origin and destination of a trip request will be visited by the same vehicle. Constraints (1.5) and (1.6) are introduced for the time constraints, while constraints (1.7) and (1.8) are used to guarantee that the number of passengers onboard does not exceed the capacity of the vehicle. Parameters used in the problem are summarized in Table 4.1.

Table 4.1. Parameters used in the problem formulation.

e_k	the earliest departure time of vehicle k
e_i	request time of request i
p_w	number of passengers getting in or off the vehicle at point $w \in W$
R_k	route of vehicle k
t_{u_{j-1},u_j}^k	time of the shortest path between the points u_{j-1} and u_j
c_{u_{j-1},u_j}^k	cost of the shortest path between the points u_{j-1} and u_j
B_u^k	time of the service at point u in the route of vehicle k
s_i^k	waiting time of a passenger i
L_u^k	number of passengers onboard after the service of point u in the route of vehicle
α	coefficient for the waiting time parameter

4.5. Solution Construction

The studied problem contains a vehicle fleet to serve the trip requests. Therefore, to construct a complete solution, two decisions need to be made: grouping and routing. Grouping refers to the assignment task in which the passengers are assigned to vehicles. Routing refers to finding the optimal routes for each vehicle according to the assignments made in the grouping phase.

4.5.1. Assignment

The very first step after getting the trip requests is to assign passengers to vehicles. In order to achieve this task, it is necessary to obtain the available capacities of vehicles. This information is acquired from the available seat column of the vehicle data frame. The locations of the vehicles are also needed since one of the major criteria for assigning a vehicle to a passenger is the distance between the passenger and the vehicle. Once the positions of the vehicles are taken from the related data frame, the SciPy python library (Virtanen et al., 2020) is used to determine the Euclidean distances between the vehicles and the passengers.

The assignment method aims to assign unassigned passengers to available vehicles by starting from the beginning of the passenger data frame ordered by the passengers' request times. This method ensures that the passengers with earlier request times will be assigned to the vehicles before the passengers who make their request later. Since available vehicles are revised after each assignment, earlier passengers have the advantage of being assigned to better vehicles. This strategy aligns with the objective function of the study because the waiting time of the passengers is the major component of the objective function and is considered to be the key factor that defines the service quality of the fleet.

Various strategies were tested for the vehicle assignment task in this study. At first, random assignment of vehicles to passengers was tested. The method assigns vehicles to each passenger according to the order of passengers' request time and revises the available seats of the vehicles. Since this method doesn't consider the distance between the vehicles and passengers when assigning the vehicles to passengers, the total distances of the vehicle routes were considerably high.

Another method used in this study was assigning passengers to the closest available vehicle. This strategy was tested with different values of the model parameters but failed to give satisfactory results. The main problem with this method was that it doesn't contain any randomness and doesn't allow any diversification. Therefore, the solutions were stuck in certain locations in the solution space. Even with improvement heuristics, the method could not outperform other assignment strategies.

To satisfy diversification in generated solutions, there must be some randomness in the assignment method. On the other hand, fully random assignments could not achieve good results. Therefore, the assignment method should contain some randomness and make wise assignments. To achieve that, a vehicle search within a distance that is incrementally increased, is introduced. In this method, available vehicles within a short distance to a passenger are determined. A random assignment among these vehicles is made. If there isn't any available vehicle within the distance, a search for the next passenger is carried out. This process continues until all requests in the iteration are examined. After that, the distance is increased by a predefined value and the search starts from the beginning of the passenger data frame. When all the searches are completed with all predefined distances, the assignment phase of the algorithm ends. An example of this passenger-vehicle assignment is shown in Figure 4.1. Red vehicles are the occupied vehicles that have no available seats, while blue vehicles have free seats. The search in the shortest radius gives only Vehicle 3, which is a fully occupied vehicle. Therefore, the radius is increased and the search is repeated. With the increased radius, the passenger has three vehicles that can be assigned. One of them, Vehicle 1, is occupied; hence is eliminated from the assignment list. Then, a random assignment is done from vehicles number 2 and 4. After random assignment Vehicle 2 is assigned to the passenger.

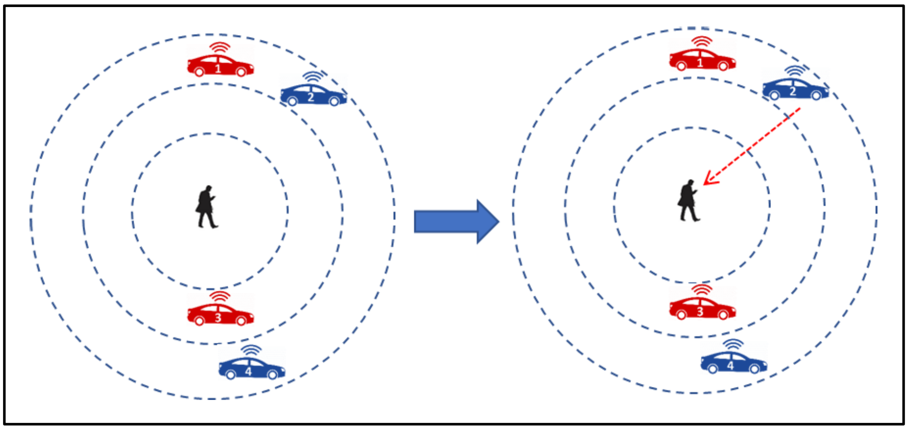

Figure 4.1. Passenger-vehicle assignment.

Some of the previous studies used the cheapest insertion method to generate the initial solution. In this method, the passenger is assigned to a vehicle with the least increase in the objective function. This method requires searching for the best vehicle and the best position for a passenger; thus, a considerable number of solutions are needed to be explored. Considering the problem size in our study, this assignment method would require high computational effort. Furthermore, the grouping and the routing tasks of the studied problem were carried out separately in this study. For this reason, at the assignment phase of the solution construction, passengers were assigned to the vehicles without determining the positions of their origins and destinations in the route.

In the initial trials of simulations, passengers are assigned to the vehicles according to their positions. However, while examining the results of the simulations, it was seen that even the average waiting times of the passengers were quite small and satisfactory, the maximum waiting time is found at an unacceptable level, over 40 minutes. When the simulation results were further analyzed it was observed that some passengers, mostly one or two passengers in every simulation, were being picked up by a vehicle immediately after his or her request but not delivered for a long time. Simulation data in each iteration was examined. It was found that the vehicles that these passengers boarded were assigned to other passengers with a destination point far away from the destination of the passengers onboard. Since the vehicle tries to optimize the total distance of the vehicle and the total waiting time of the passengers in each iteration, the route of the vehicle sometimes may cause huge delays in the delivery time of some passengers while it minimizes the travel time of other passengers. This situation keeps happening in every iteration because, in upcoming iterations, the vehicle is positioned far from the destination of the passengers onboard and accepts passengers with different destination locations. To solve this problem, the coefficient of the waiting time parameter in the objective function was increased. However, the problem was not fixed even with high values of the waiting time coefficient. This is mainly caused by the fact that these passengers having high waiting times are picked up by the vehicles very fast. After a passenger is in a vehicle, the status of the passenger becomes “onboard,” and the flexibility of removing the passenger from the vehicle and assigning him or her to other vehicles is no longer an option.

To eliminate this problem and limit the computational time, positions of stops which are possible origin and destination points in our problem, were clustered into clusters using k-means clustering, meaning that the studied region is clustered into sub-regions. After clustering, every trip had an origin and destination cluster along with origin-destination stops. K-means clustering was done by the related function in the scikit-learn python library (Pedregosa et al., 2011). In the assignment of vehicles to passengers, destination clusters of the passengers and the destination cluster of the vehicle should be met, or the vehicle assigned to a passenger should not have a destination cluster, meaning no passenger is assigned to that vehicle yet. In order to achieve this, destination clusters of the vehicles were updated in every assignment according to the assigned passengers. For further assignments, the destination cluster of the vehicle limits the possible assignments among passengers. Origin cluster information was not used in the assignment process because vehicles were already assigned to passengers based on their closeness to the passengers. On the other hand, for the improvement strategies, origin and destination cluster information were both used in order to eliminate bad neighborhood moves.

4.5.2. Routing

After the passengers are assigned to the available vehicles, the routes of the vehicles should be determined. The assignment phase of the solution construction deals with only grouping passengers into vehicles. However, to have a complete solution, routing of the vehicles also needed to be completed.

The vehicle capacities were limited to 4 passengers in this study. This constraint limits the number of possible solutions for a vehicle in an acceptable range. Hence, finding the exact solutions for the routes of the vehicles becomes feasible. To achieve this, all of the solution space is needed to be explored.

The analysis of this study was carried out by Pandas python library (Mckinney, 2011). Pandas library offers various functions for different data operations. However, to utilize the library efficiently, the code and the data should be constructed in a way that the library shows higher performance. Even for such operations that require high computational effort, the computational times were achieved to be kept short if the code and the data were constructed suitably. If a function can be applied to a whole column, the computational time is dramatically decreased compared to the operations carried out row by row. In this manner, the most challenging goal was to vectorize the operations.

All the functions used for the exact solution of the routing problem were designed to be vectorized. With this advantage, the functions that were developed to find the exact solutions complete their task in a short time. It should also be noted that even a small increase in the vehicle capacity disables the utilization of these functions for the search of the exact solutions. Before the exact solution algorithm runs, all possible route sequences were determined. These sequences were categorized by the number of passengers assigned to a vehicle and stored in different data frames. The exact solution algorithm recalls these data frames according to the number of passengers assigned to the vehicle.

One other parameter besides the number of assigned passengers was the number of passengers onboard. In DARPs, passengers have pick-up and delivery locations, meaning that for a passenger, two locations must be visited by the vehicle. During the execution of a route, a passenger might be onboard. This indicates that the vehicle assigned to that passenger visited the origin location of the passenger and picked the passenger. After this point, the vehicle has only one location related to that passenger which it needs to visit, and that is the destination of the passenger. Therefore, the size of the route for a particular vehicle is determined by the status of the passengers that are assigned to the vehicle. When the number of passengers onboard is determined, the data frame containing all possible sequences for a certain number of passengers is filtered according to this information.

The exact solution algorithm aims to find the optimum route for a vehicle according to a given objective function. In this sense, all necessary information regarding the parameters in the objective function should be present. The objective function in our study contains two main parameters, the total distance of the route and the total waiting time of passengers. The distance of a route can be determined by the sum of the distances of each edge in a route. On the other hand, the total waiting time for a route should be calculated by the summation of the individual waiting times of the passengers. Therefore, the request times of each passenger are also included in the data frame that is given to the exact solution algorithm as an input.

4.6. Improvement Heuristics

When a complete solution is generated with the exact solution for the related passenger-vehicle assignments, improvement heuristics take this solution and try to improve it. However, a complete search for a neighborhood of a solution requires high computational effort. Therefore, it is necessary to limit the number of considered moves. It is possible to restrict the neighborhood in the way of imposing geographical restrictions to eliminate the moves that consider two distant passengers (Johnson and Mcgeoch, 1995). In this study, the moves are limited to certain neighborhood structures. In this sense introduction of sub-regions is utilized for the solution improvements. Three different move operators and a SA algorithm are proposed for improvement heuristics.

4.6.1. Relocate Operator

This improvement heuristic method moves a trip request from a vehicle and assigns it to another vehicle. The origin and destination location of a request is removed from a route and inserted into another route. After the reassignment process, the routes of the vehicles are reoptimized. Insertion of the passenger can be applied only to the vehicles that have available seats. When a passenger is selected for this operator, all of the neighborhood is searched, and the move with best-improve in the objective function is accepted.

An example of the relocate operator is shown in Figure 4.2. Passenger 1 is selected from Vehicle1 for relocation, and Vehicle 2 is selected as the new assignment. Origin and destination points of Passenger 1 are removed from the route of Vehicle 1. With the new state of assignment, Vehicle 1 has Passenger 2, 3, and 7. The optimal route for these passengers is recalculated. It is highly possible that the remaining sequence of the route will not be the same, meaning that the positions of the remaining OD pairs might change. As seen in the example, positions of the 3^+ and 7^+ have changed after optimization. When Passenger 1 is assigned to Vehicle 2, the optimization of its route needs to be accomplished. After finding the optimal route for Vehicle 2 with the new assignment, OD pairs of Passenger 1 are inserted in the optimal positions in the route. It also changes the sequence of 4^- and 6^+ that are the destination of Passenger 4 and the origin of Passenger 6, respectively.

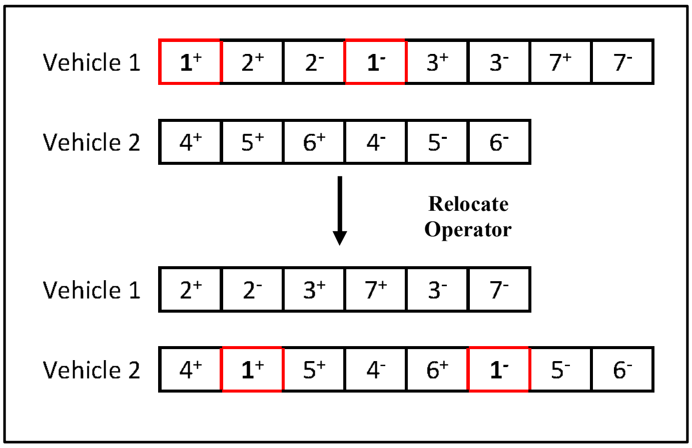

Figure 4.2. Relocate operator.

To limit the possible moves and ensure a good neighborhood structure, the neighborhood is restricted to the vehicles that go to the sub-region of the relocated passenger's destination. For example, if a passenger is selected for relocate operator and the destination of this passenger is located in the sub-region C_n, then the vehicles going to the same sub-region are selected for relocate operator. Among these vehicles, the one that gives the best improvement is selected. With this limitation, instead of exploring all neighborhoods, only promising areas are explored.

4.6.2. Exact Swap Operator

Another method used for solution improvement is to swap two passengers between two vehicles. An example of this method is shown in Figure 4.3. Passengers 1 and 5 are selected for the swap operator. Then Passenger 1 is removed from Vehicle 1 and reassigned to Vehicle 2. Similarly, Passenger 2 is removed from Vehicle 2 and reassigned to Vehicle 1. After reassignment is completed, the vehicles have different passengers assigned. With the new assignments, optimal routes for these vehicles are recalculated using the exact solution algorithm. As seen in the example, the sequence of the remaining passenger OD positions is changed due to the new assignments.

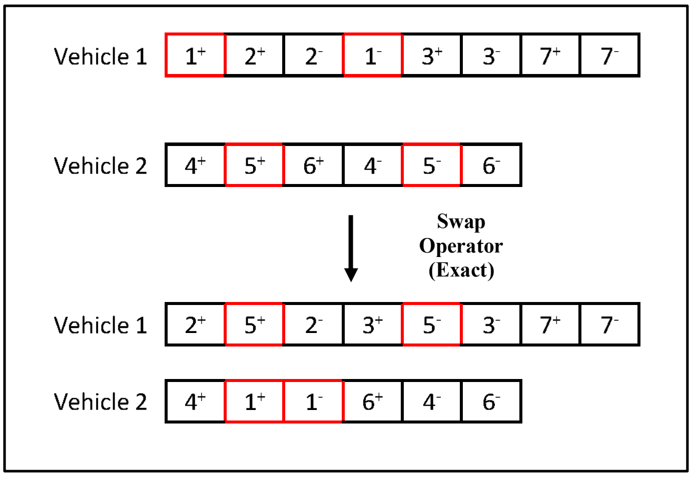

Figure 4.3. Swap operator (Exact).

Similar to relocate operator, restricted neighborhood structure is used for swap operator also. When a passenger is selected for swap operator, a second passenger can be selected among the passengers sharing the same sub-region for the destination. After exploring all possible swaps, the best-improve swap is implemented.

Although this method satisfies improvements in the solutions, it requires high computational effort. In this method, when two passengers are selected for swap operator, the exact solutions for the new assignments need to be found for each vehicle. For example, if Passenger 1 is selected for swap operator and the suitable pair for the swap is searched in the neighborhood, the exact solution for Vehicle 1 must be found for every trial of swaps.

4.6.3. Approximate Swap Operator

Due to the computational inefficiency of the exact swap operator, another method is utilized in this study. In this method, the passengers are swapped with the positions of their origins and destinations. An example of this operator can be found in Figure 4.4. Passenger 1 and Passenger 5 are selected for the swap operator. The origin of Passenger 1 is removed from Vehicle 1 and inserted into the location of Passenger 5's origin. Similarly, the origin of Passenger 5 is removed from Vehicle 5 and inserted into the position of Passenger 1's origin. The same operation is carried out for the destination points of the swapped passengers.

The same restriction used in previous improvement operators is utilized for the neighborhood. After searching for all possible swaps, the swap with the best improvement in the objective function is carried out. This method does not guarantee an optimal solution. However, it has computational efficiency since it does not require exact solutions.

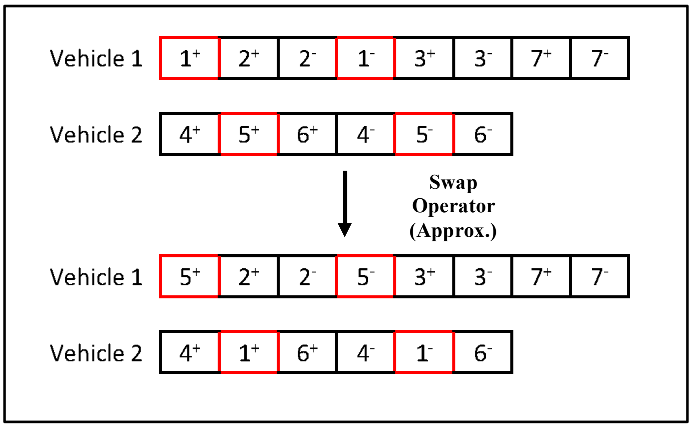

Figure 4.4. Swap operator (Approximate).

4.6.4. Simulated Annealing

Simulated annealing is widely used in solving combinatorial optimization problems. It is addressed as a local search metaheuristic since it explores a neighborhood of a solution (Nikolaev and Jacobson, 2010). The mechanism of accepting moves that worsen the solution provides an escape from local optima.

In this study, the SA algorithm is mainly used to diversify the solution. In order to improve diversification in the SA algorithm, we modified it and introduced the parameter of *similarity factor,* z. This factor is used to find the similarity rate between two solutions. The proposed SA algorithm is used after a certain improvement heuristic is applied to the solution. The solution acquired from an improvement heuristic is called the local optima of solution (s_b) of the solution (s) that the improvement heuristic is applied. These two solutions are given to the SA algorithm. The algorithm aims to find solutions around s with the decrease in similarity factor. The main purpose of it is to explore the neighborhood of s that is far from its local optima s_b. At the beginning of the algorithm, a similarity factor is calculated by using the given s and s_b. This value is taken as the reference value, and at upcoming iterations, it is revised with the new similarity factors calculated by using s' and s_b. If the factor is found to be higher than the reference value, then the algorithm moves to the next solution. If it is lower than the reference value, then steps for approval of a new solution are followed. If a solution is accepted, then the reference value for the similarity factor is revised. The similarity factor is calculated as the portion of common vehicle assignments. This is the ratio of the number of passengers sharing the same vehicle assignments in s' and s_b to the total number of passengers.

Similar to the previous SA algorithm, our proposed algorithm involves the parameters like initial temperature T_0 and the temperature reduction factor δ. To limit the number of iterations, two parameters are defined: I_t and I_n. I_t limits the number of iterations for temperature reduction. On the other hand, I_n limits the number of iterations for the neighborhood search of a passenger.

The algorithm starts by selecting a passenger p_1 among the passengers who are not onboard P_w. After the selection of candidates for a second passenger, P_{cand} is determined with predefined limitations. These limitations are similar to the ones utilized in the above-mentioned operators.

The algorithm uses the approximate swap operator for the neighborhood search. When the second passenger p_2 is selected from P_{cand}, the positions of selected passengers' origins and destinations are swapped between the vehicles that they are assigned to. As mentioned above, the approximate swap operator has a computational efficiency since it doesn't require optimal solutions. To limit the computational effort needed for the proposed SA algorithm, this operator is used.

Given an initial solution s and s_b $\{s_b$ is the local optima of s $\}$, initial temperature T_0, temperature reduction factor δ, iteration limit for temperature reduction I_t and for neighborhood search I_n
Find passengers who are not onboard P_w
Set temperature $T \leftarrow T_0$
Find similarity factor $z = Sim(s, s_b)$
for $(i = 0;\ i < I_t;\ i++)$ **do**
Select a random passenger p_1 in P_w
Generate a list of passengers P_{cand} in the neighborhood of p_1
for $(j = 0;\ j < I_n;\ j++)$ **do**
Select a random passenger p_2 in P_{cand}
Swap the passengers to get a new solution s'
if $Sim(s', s_b) > z$ **then**
continue
Find $\Delta = obj(s') - obj(s)$
if $(\Delta < 0)$ **then**
$s \leftarrow s'$
$z = Sim(s', s_b)$
else
$p = Random\ (0,1)$ {a random value between 0 and 1}
if $(p < e^{(-\Delta/T)})$ **then**
$s \leftarrow s'$
$z = Sim(s', s_b)$
$T \leftarrow T \times \delta$
Return s

Figure 4.5. Pseudocode of the proposed SA algorithm.

If the generated solution s' is better than the current solution s then the algorithm accepts the new solution. However, if it is worse than the current solution, then the probability of accepting this solution is calculated by $e^{(-\Delta/T)}$, where Δ refers to the change in the objective function. At the higher temperatures, the algorithm tends to accept the low-quality solutions more, while at the lower temperatures, the probability of acceptance of non-improving solutions becomes very small.

The pseudocode of the proposed SA algorithm is shown in

Figure 4.5. The performance of the algorithm is highly dependent on the choice of SA parameters like initial temperature, temperature reduction factor, and the number of iterations. Various combinations of these parameters are tested for the studied problem.

5. SIMULATION

5.1. Data Preparation

One of the main goals of this study is to explore the feasibility of a shared autonomous taxi system to replace conventional buses. In order to achieve this, the origins and destinations of the passengers in several bus lines in a university area were inferred using state-of-the-art algorithms. These origin and destination pairs are used in the dynamic dial-a-ride problem.

For origin and destination inference studies, it is highly possible that some portions of the OD pairs are missing. One of the main reasons behind this problem is the inaccurate GPS data of vehicles. Since the OD inference algorithms mainly rely on the position of the vehicle, origin and destination inferences cannot be made for the trips with inaccurate or missing GPS data. In order to simulate the actual trip load of the real-life application, a specific day with minimum data loss is chosen among the studied days.

Typical trip loads in PT vary during the day. Figure 5.1 shows the trip distribution of the studied bus routes by hours of a day. As seen in Figure 5.1, the maximum trip load is experienced during morning peak hours. To challenge the proposed model, a one-hour period (08:00 a.m. - 09:00 a.m.) that has the maximum trip load during the day was chosen. During this period of the day, studied bus routes carried 1996 passengers. The algorithms were tested for other periods of the day also and showed better performance. This was an expected result due to the decrease in trip load. Since the vehicle fleet should be designed to serve at the required performance level all day, the algorithm should be tested and perform well for the worst-case scenario.

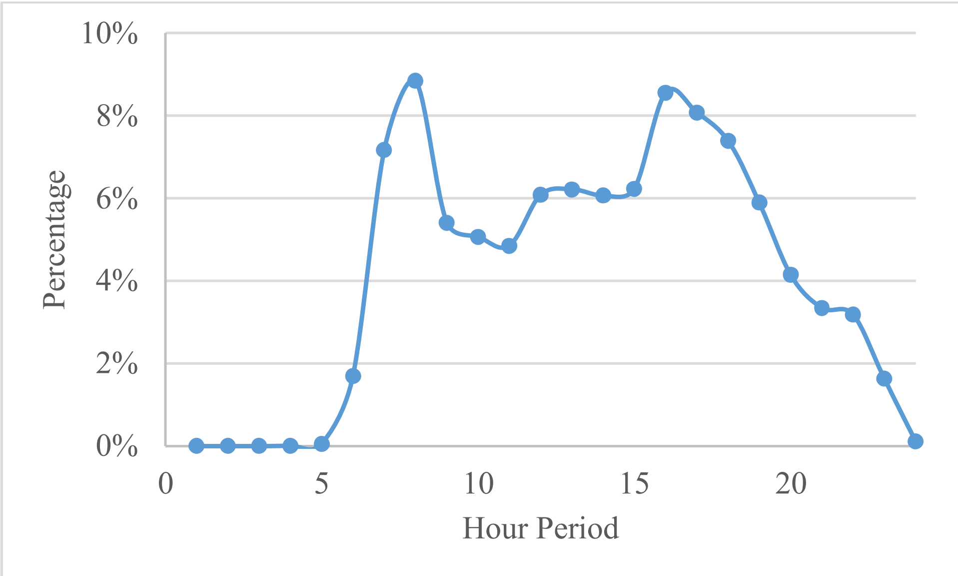

Figure 5.1. Percentages of daily trips by hours.

To decrease the computational time and problem size, the stops within an approximately 100-meter distance were consolidated into one reference stop.

The next step was to install the network, which contains the studied bus routes and the surrounding area. The network of the studied is imported using Open Street Map. In this network, road junctions are shown by the nodes, and edges between nodes represent road segments. The information on the position of the nodes by their coordinates and the several characteristics of edges, including lengths and speed limits, are given in the network.

To import and analyze the network from Open Street Map, we utilized the OSMnx Python package (Boeing, 2017). One can install two types of networks; simplified or detailed versions. To acquire a more precision detailed version of the network is installed. This increases the number of nodes hence computational times of operations. On the other hand, it is necessary to burden this problem in our studied case because the problem is solved using simulation methods, and in every iteration of the simulation, the position of vehicles should be determined in a more precise way. For our studied network, the number of nodes is over 5,000, and the number of edges is over 10,000. These nodes and edges should be connected to have a complete network. Otherwise, the routing functions can not operate correctly. However, in some examples, also in our case, it is seen that some nodes and edges are not connected properly. To solve this issue, the strongly connected component of the network is obtained by using osmnx.utils_graph.get_largest_component function in the OSMnx library. The installed network is shown in Figure 5.2.

Figure 5.2. Network of the studied area.

5.1.1. Distance Matrix

In most VRP studies, the distances between the nodes are assumed to be known in advance, while in practice, the routing problems are defined on the actual road network. If detailed information on the road network of the studied area is available, it is possible to generate a distance matrix by solving a series of shortest-path problems. However, if the network contains a high number of nodes, considering the fact that for a particular pair of nodes there could be different shortest, cheapest, and quickest paths, shortest path computations may require a significant amount of memory and time (Boyacı, Dang, and Letchford, 2021). In our study, the trips are defined on an actual road network, and the passengers are expected to make their trips between the predefined origin and destination locations. These locations were the stops of the studied bus routes. In heuristic solution methods, solution space is extensively explored to find the optimal solution. In our problem, this means that countless paths between origin and destination points should be found. It should be noted that finding the shortest path, distance, and duration between two points is extremely time-consuming when the number of different OD pairs increases. In order to eliminate this problem, OD locations were limited to the present bus stops in our study. In addition, setting the pick-up and delivery points at strategic points such as rail stations, bus stops, etc., increases the feasible passenger-vehicle assignments. The concept of meeting points has been shown to improve the efficiency of the proposed transportation system (Li et al., 2018; Stiglic et al., 2015).

A data frame containing all possible routes between the stops was generated. Since the network is not symmetric, the size of the distance matrix equals the square of the number of unique bus stops. In our studied bus routes, 93 unique bus stops are determined after the consolidation of very close bus stops into a single stop. Therefore 8,649 different paths are calculated with the nodes along the route, total distance, and total duration. If one intends to find all possible paths between the nodes in the network, the size of the distance matrix reaches an unacceptable range. For example, this figure can reach over 26 million for our studied network.

Although the origin and destination locations are fixed in our problem, the routes generated during the simulation are not limited to the routes in the distance matrix. Because our problem is defined as dynamic and the location of the vehicles is not necessarily a bus stop and can be any nodes along the routes when the simulation is stopped, new shortest paths between the locations of the vehicles and the origins and destinations of the passengers should be calculated. To achieve this, location of the vehicles at any time should be found first. In the distance matrix, apart from the shortest path, the distances and the durations, cumulative distance, and duration along the route were also calculated using the nodes in the shortest paths. An example is shown in Table 5.1. A function returning the cumulative distance and duration of the route was generated and used to find these for the routes in the distance matrix. With the help of cumulative duration over the route, simulation detects the location of the vehicle at any given time.

Table 5.1. Example of an instance in the distance matrix.

Origin Node	2584314936
Destination Node	3105982313
Distance (m)	486.703
Travel Time (sec)	58.1
Nodes in the Route	[2584314936.0, 6233190308, 6233190311, 2584314932, 6233190313, 470365556, 2584314929, 2584314930, 3262530838, 470365552, 3262530840, 2584314941, 3105982306, 3443350522, 3105982313]
Cumulative Distance	[112.58, 121.49, 144.37, 196.09, 259.53, 272.57, 282.25, 297.22, 317.52, 376.98, 425.32, 440.63, 461.93, 486.69]
Cumulative Time	[13.5, 14.6, 17.3, 23.5, 31.1, 32.6, 33.7, 35.4, 37.8, 44.9, 50.7, 52.5, 55.1, 58.1]

At the initial trials of the simulation, the distance matrix was kept unchanged, containing only the routes between the bus stops. When a new route was encountered because of the changing position of the vehicles, the shortest path was found using the related function. After a great number of simulations, it was observed that the trips are focused on a particular area of the network. Since the shortest path is calculated when an OD pair is not found in the distance matrix, it is logical to expand the distance matrix with possible paths if the size of the distance matrix can be kept in a reasonable range. For this purpose, when a new route is observed during the simulation, it is included in the distance matrix, and the distance matrix is revised. At the end of each simulation, the size of the distance matrix is increased. The increase in the size of the distance matrix keeps getting smaller in every iteration because some of the possible routes are already determined in previous simulations. By achieving this, the need for the calculation of the shortest path for new OD pairs and the additional computational effort is extensively reduced.

5.1.2. Initial Positions of the Vehicles

The inference of OD points of the passengers was the key step of this study. Information regarding the trips of the passengers is derived from the outputs of origin and destination inferences. Since the origin and destination distribution over time is acquired, it is highly effective to position the vehicle according to this information. The probabilistic distribution of the demand for origins is determined for every hour, and the initial positions of the vehicles were assigned probabilistically using the distribution in the related hour. If the inference study is conducted for a longer time and the data for origin-destination demand of the passengers is extended, the initial positioning of the vehicles is expected to increase the performance of the algorithms.

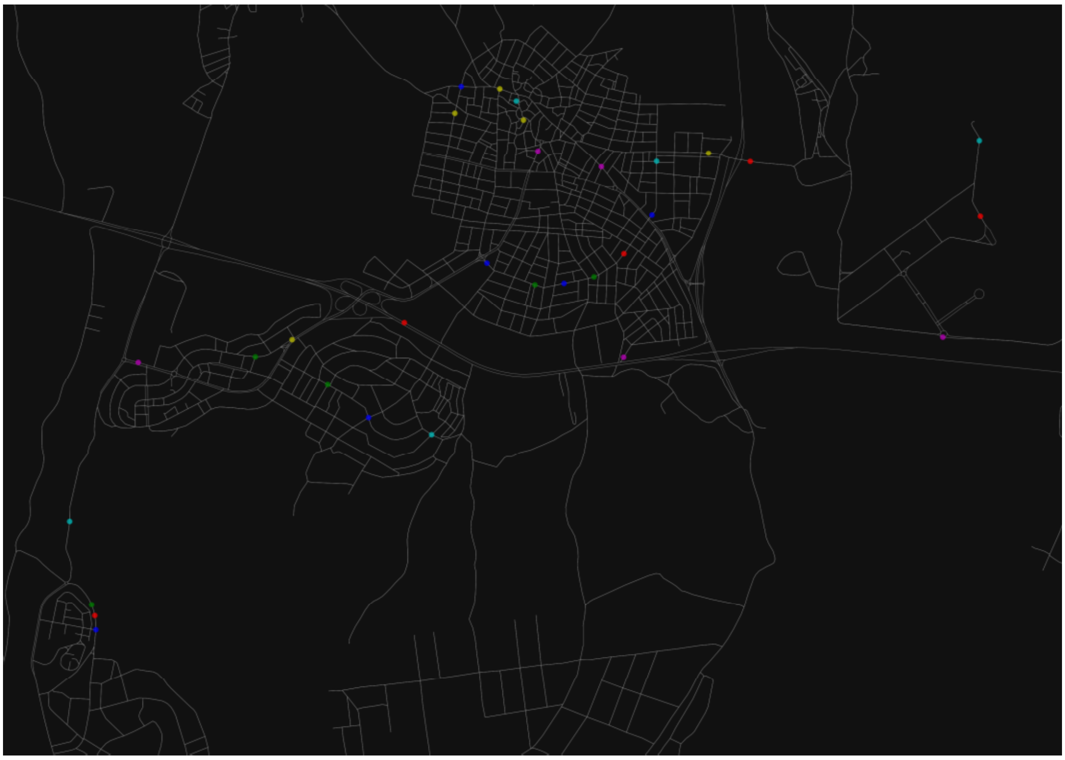

Figure 5.3. Initial positions of the vehicles (Colored circles represent vehicles).

In order to have diversity and understand the real performance of the algorithms, the initial positions of the vehicles are changed before each simulation starts. It is seen that the initial positions of the vehicles are affecting the performance of the solution algorithms considerably. However, it is not possible to determine the exact spatio-temporal distribution of the trips in real life. Therefore, in the autonomous shared taxi fleet operation, fleet controllers should determine the positions of the vehicles based on the trip history and revise the positions while the historical data grows.

5.2. Parameters Used in Simulation

5.2.1. Period Duration

In our problem, trip requests arrive dynamically, and the passengers expect to be served immediately. Therefore, the proposed algorithm should offer a proper framework to the fleet controller. One of the main parameters of the simulation study that affects the fleet's operation is the period duration representing the duration of the decision epoch. While some studies use the instant approach in which the problem is resolved when there is an update in the requests, we utilize the periodic approach in our study. The main reason behind this was the size of the problem in our study. For a 60 minutes time interval, there are about 2,000 trips in rush hours, meaning that in every 1.8 seconds, a new request arrives. It was highly inefficient to reoptimize the problem in every update of the requests when we consider the fact that even the simulation processes like revising the position of vehicles, status of the passengers, etc., need some computational time.

Since the requests in the decision epochs are solved at the end of each period duration, an increase in the period duration is expected to affect the waiting time of the passengers. Therefore, in the initial simulations, the period duration was kept as small as possible to ensure that the requests of the passengers are solved, and the vehicles are assigned to those passengers immediately. However, after the simulations with different period durations, it is seen that a decrease in period duration doesn't necessarily guarantee less waiting time. When the period duration increases, the number of requests being optimized in a decision epoch increases. The optimization algorithm shows higher performance when the size of the problem increases. On the other hand, with higher values of period duration, the gain from the optimization cannot cover the increase in the waiting times of the passengers. As seen in Figure 5.4, the optimum period duration is found to be around 120 seconds for our study.

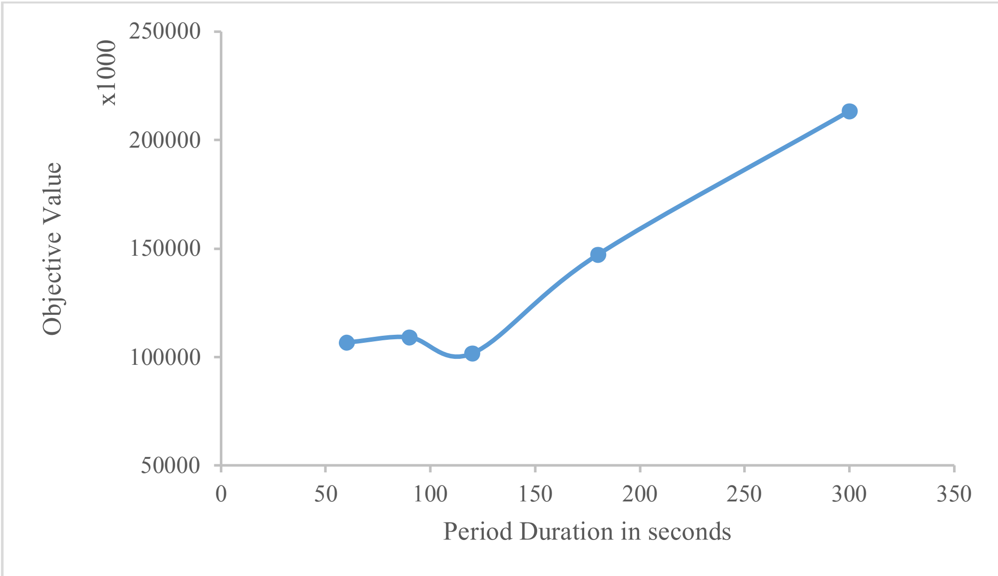

Figure 5.4. Objective values for different period durations.

5.2.2. Vehicle Capacity

There are mainly three types of vehicle capacity constraints; imposed all the time, imposed at certain times, not imposed (Bodin and Golden, 1981). If the vehicles are small and the shared rides are offered as a service option, the capacity constraints for vehicles should be imposed all the time. Since our study deals with a problem with shared rides, the capacity constraint was imposed during the whole operation of the fleet.

In systems with shared rides, capacity constraints prevent the vehicles from being overused. In some studies using relatively big vehicles, the minimum number of passengers for shared rides is introduced in the model as the lower bound of the capacity constraint (Kaan and Olinick, 2013). We used relatively small vehicles in our study hence didn't impose a minimum number of passengers for shared rides. Furthermore, the immediate pick-up and delivery of the passengers are quite crucial in our study, and imposing such a constraint would increase the waiting time dramatically.

The maximum number of passengers is set to be four in this study. Since the proposed model aims to offer a taxi-like service, the capacity of the vehicles is kept small. The capacity constraint is imposed not only for the passengers onboard; the number of passengers assigned to a vehicle in the grouping phase cannot exceed four. This limitation eliminates the need for a capacity check along the route of the vehicle. Otherwise, the algorithm should check whether the capacity constraint is violated during the execution of pick-ups and deliveries. Another advantage of small capacities is the feasibility of using exact solution methods for routing problems. If the number of passengers is kept in a small range, the computational time for exact solution methods becomes suitable for such a dynamic problem. The routing problem in our study contains pick-up and delivery locations. This means that the number of locations to be visited by the vehicle is two times the number of passengers assigned to that vehicle. For example, a vehicle having four passengers assigned to it will visit 8 locations. The total number of possible route sequences is 2,520 for four passengers. If we increase the capacity with only one passenger, a vehicle with five would have 113,400 different routes. As seen in this example, the problem size increases considerably even with a small increment.

5.2.3. Number of Vehicles

In some versions of the VRPs, the objective function aims to minimize the number of vehicles to be used. Some studies set the minimization of the number of vehicles as the primary objective, then optimize the total distance and duration. In our case, the number of vehicles is predefined, and the solution algorithm is free to assign passengers to available vehicles during the simulation. On the other hand, the number of vehicles should be minimized without violating the problem constraints since the fleet operators would prefer less investment cost. Therefore, the number of vehicles is incrementally decreased while the algorithm converges its performance limit for that number of vehicles. If the objective function value is no longer decreasing in a meaningful range and the constraints of the problems are not violated, further simulations are carried out with a smaller fleet size. In this study, the number of vehicles decreased from 120 to 90. Considering the fuel consumption and the investment cost of the conventional bus, this figure seems acceptable. In comparison, during the studied hour, the total 1996 trips were carried by 20 unique buses on the studied day.

5.3. Simulation Framework

The simulation starts with acquiring the trip requests N_p between start time t_s and end time t_e of the related iteration. The start and end times of each iteration depend on the period time d defined before the simulation starts. At the end of each iteration, start and end time values are increased by the period time. The following information about the trip requests are taken:

- Time of request
- Passenger number (This is assigned to each passenger by the algorithm according to their request time)
- Coordinates of the OD pairs
- OSMIDs of the OD pairs
- Clusters of OD pairs
- Distance and duration of the shortest path between the origin and destination of the passenger. This data is previously determined and stored in the distance matrix

This information generates a data frame that contains all necessary information about the passengers and their trips. After acquiring the information from the passenger side, the data frame is extended with columns filled by the algorithm over the simulation steps. These columns contain the following information:

- The vehicle number assigned to the passenger
- The pick-up and delivery time of the passenger
- The state of the passenger

There are three different states for passengers. The first one is the “waiting” state meaning that the passenger request is taken, but no vehicle has picked up the passenger yet. Another state is the “onboard” state, referring to the passengers that are taken by the vehicles and on a route to their destinations. The final state for the passengers is the “delivered” state that stands for the passengers being delivered to their destinations. These passengers are considered to be served, and the information about these passengers is no longer carried on next iterations.

Apart from the passenger information, the simulation needs vehicle states also. As previously mentioned, the vehicles are positioned in certain locations according to the demand history of studied routes. This information is written in a data frame that contains other relevant information on vehicles. The following information is stored and revised in the vehicle data frame at each iteration:

- Vehicle number
- Vehicle capacity
- Passengers onboard
- Number of passengers onboard
- Available seats
- To which cluster the vehicle is going to
- Total traveled distance of the vehicle
- The total duration that the vehicle has been actively serving
- Location of the vehicle in terms of OSMID
- Passengers assigned to the vehicle
- Number of passengers assigned to the vehicle

The simulation has two major tasks in each iteration: passenger-vehicle assignment and routing. For the assignment task, the proposed algorithm aims to assign the passengers to the vehicles. The assignment algorithm always prioritizes the passengers with earlier request times. When the demand exceeds the capacity of the fleet, no more assignments can be done. This results in passengers without assigned vehicles. It is also possible not to be able to assign vehicles to some passengers who don’t have suitable vehicles nearby.

After the assignment task is completed, the routes of the vehicles are generated by the exact algorithm that finds the optimal route for the vehicles with given passengers. The algorithm aims to find an optimal solution for each vehicle that has passengers assigned to it according to the proposed objective function. As discussed in Chapter 4, the objective function contains the total traveled distance and the waiting time of the passengers. In order to satisfy the service quality, the waiting time parameter is included in the objective function. The coefficient of waiting time α is needed to be determined to balance the effects of the objective function parameters. With a given α, the optimal route for each vehicle is determined. The algorithm tries to minimize the total distance of the route while the total waiting time of the passengers is aimed to be kept short.

The passenger-vehicle assignments and the routes generated by the exact algorithm form a complete solution. This solution is improved by the selected improvement heuristics. In our study, several improvement heuristics are proposed and tested. In some cases, these heuristics are jointly utilized for the improvement task.

After the solution improvement is achieved, the simulation takes the routes and moves the vehicles along the routes for a period duration d. During the implementation of the routes, some passengers might get on the vehicles while some might be delivered. The status of trip requests is revised according to the states of the passengers. Delivered passengers are eliminated from the requests N_p. The vehicle positions and the passengers onboard are revised according to moves carried out by the vehicles. The revised variables are defined as the state variables. The functions utilized for the revision in state variables can be considered as the transition functions.

At last, the start time t_s and the end time t_e of the period is increased by the predefined period duration d. At the next iteration, the trip requests are captured between the increased t_s and t_e. This process is carried out until there is no unserved trip request left. The pseudocode for the simulation is shown in

Figure 5.5.

Not only the solution quality but also the computational time of the simulation runs are highly dependent on the parameters used in the simulation. Therefore, various settings are tested to get suitable parameters. However, with some settings, the computational effort needed to solve the problem reaches a high level. These settings were not used for further simulation runs since it is impractical to get results in an unreasonable time.

Initial Positions of Vehicles
A set N with n trip requests
Choose values for period duration d and waiting time coefficient α
Set start time t_s and end time t_e
while stopping criteria is not met **do** {criteria: there is no unserved request}
    Collect the trips requests N_p between t_s and t_e
    Revise N_p and find unserved requests {by appending $N_p = N_p \cup N_{p-1}$}
    for $i \in N_p$ **do**
        if i^+ is not visited {For each passenger who is not onboard} **then**
            Make vehicle assignment
    for $k \in M$ {For each vehicle} **do**
            Find optimal route
    Improve Solution {Using improvement heuristics}
    Update status of requests N_p {exclude the served requests}
    Update status of vehicles {vehicles' locations and the passengers onboard}
    $t_s = t_s + d$
    $t_e = t_e + d$

Figure 5.5. Pseudocode for the simulation.

6. SIMULATION RESULTS

The simulations were run on a 64-bit computer with 128GB of RAM, and a 2.20 GHz processor.

The performance of the solution methods is evaluated upon the results obtained from the simulation runs. However, it is important to set the parameters used in the algorithms since the results are highly dependent on these parameters.

The period duration d is chosen as 120 seconds. With this period duration, the proposed solution methods produce good results in a reasonable time. The number of vehicles m are changed during the study. If satisfactory results are obtained with a certain number of vehicles, then it is decreased to a certain value to challenge the algorithm further. This process is carried out until the value of 90 for the fleet size, and most of the simulation runs used this value.

As stated before, the objective function of the optimization problem contains two parameters, total traveled distance and the total waiting time. Even it is not included in the objective function, the maximum waiting time is taken as an output from the simulation. It is seen that for some cases, even average waiting time is quite low, some passengers might suffer from long waiting times. Therefore, the solution methods that produce low maximum waiting times are preferred. This is a significant concern since the proposed transportation service should serve all passengers at certain service quality.

The simulation runs carried out with different waiting time coefficient values. Table 6.1 summarizes the results of 30 simulation runs. The value of 200 gives promising results in terms of both waiting times and traveled distances. To recognize the importance of the waiting time parameter, the coefficient for the waiting time α in the objective function is chosen as 200.

Table 6.1. Waiting time coefficient trials.

		Average Waiting Time (min)	Average Total Traveled Distance (mt)
Waiting Time Coefficient	0.5	4.57	3701557.87
	5	4.52	3728996.59
	10	4.55	3701475.13
	30	4.70	3734726.26
	50	4.49	3740575.09
	100	4.48	3691138.88
	200	4.37	3669086.35
	300	4.41	3705927.09
	500	4.40	3688062.58
	1000	4.44	3710244.67

In our proposed SA algorithm, four parameters need to be determined before the algorithm runs. These are initial temperature T_0, the temperature reduction factor δ, I_t the number of iterations for temperature reduction, and I_n the number of iterations for every temperature value.

Typically temperature reduction factor ranges between 0.8 and 0.99 (Hosny, 2010). Various reduction factor values are tested, but none of them makes a significant difference, and we set the reduction factor δ as 0.90. We also tried several initial temperature values, and choose 10^6 as T_0. Table 6.2 summarizes 16 simulation runs with different initial temperatures. 10^6 has given the best performance. This value is high because our objective function produces high values, hence the changes in the objective function are also high. To ensure that some non improving solutions are also accepted, the temperature needs to be high enough. To limit the search, we use I_t and I_n parameters. We used the values of 20 and 40 for I_t and I_n respectively.

Table 6.2. Initial temperature trials.

Initial Temperature	Objective Value
1000	5605047.4
10000	5554370.6
100000	5567771.9
1000000	5530567.3

6.1. Computational Time

The biggest challenge of this study was to limit the computational efforts needed to solve the related problem. Due to the dynamism of the problem, the solutions need to be generated in a reasonable time. Because the passengers require immediate service for taxi-like transportation systems. It is also important to have a short computational time because the simulation runs for a certain time, and it requires several iterations. To see the performance of a proposed solution method, the output of the simulation needs to be acquired. If the computational effort is not limited to a reasonable range, it would be impractical to find the optimal parameters due to a need for various trials of simulation runs. Therefore, besides the performance parameters related to the operation of such a system, the proposed solution methods are always evaluated by their computational time.

As discussed in Chapter 4, the exact solution algorithm is used to find the routes of vehicles according to given passenger-vehicle assignments. Among the improvement heuristics, relocate operator and exact swap operator also utilizes the exact solution algorithm. In the exact swap operator, the routes of two vehicles are reoptimized for every move, while in relocate operator, the route of the vehicle that is the current vehicle of the passenger who will be relocated, is reoptimized once. Thus, the exact swap operator requires much higher computational time. Due to this inefficiency, the exact swap operator is only used a couple of times. In the approximate swap operator, the routes are not found optimally; hence, the computational time required for this method is in a reasonable range. Table 6.3 summarizes the average computational time required for a simulation run. When no improvement strategy is applied to the solution, simulation is completed in around 10 minutes which includes the operations needed for solution construction and the simulation. The total CPU time required for the improvement heuristics should be evaluated considering the fact that they also contain these operations.

Table 6.3. The computational time required for improvement operators.

Improvement Type	Total CPU (min)
No Improvements	10.06
Relocate Operator	21.05
Approximate Swap Operator	27.64
Exact Swap Operator	173.98

6.2. Radius Ranges in Passenger-Vehicle Assignment

Another parameter that needs to be determined is the radiuses used in the passenger-vehicle assignment task. As indicated before, various strategies are tested for passenger-vehicle assignments. Assigning the closest available vehicles is one of the strategies that is tested and failed to give satisfactory results. Another method used for the assignment is to make the assignment in a predefined radius first, then assign the unassigned passengers to the closest vehicles. This method ensures that if there are available vehicles, every single passenger is assigned to a vehicle. However, this method also couldn’t outperform our proposed assignment method. The main reason is believed to be that, this method tries to make assignments for all passengers. Therefore, in some cases, it assigns vehicles to passengers far away from each other.

We proposed a vehicle search in certain radiuses. These radiuses are incrementally increased to expand the area for the vehicle search. To limit the distance between vehicle and passenger for assignment, the maximum radius is set to a certain value. Among the various set of radiuses, most promising results are taken by the following set of radiuses: [0.001, 0.002, 0.003, 0.004, 0.005, 0.006, 0.007, 0.008, 0.009, 0.01, 0.02, 0.03]. The first radius refers to approximately 100 meters, while the last one refers to 3 km.

6.3. Number of Sub-regions

As stated before, the introduction of sub-regions considerably improves the solution quality. The origin and destination points of the trip requests are clustered into a predefined number of clusters for generating the sub-regions. The sub-region information is used both in the passenger-vehicle assignment and improvement tasks. For the assignment task, the vehicles moving towards a sub-region can be assigned to the passengers having the same sub-region for the destination points. Similarly, relocate operator can be applied to the passengers and vehicles sharing the same destination cluster. For the swap improvement heuristics, the neighborhood structure is restricted with the passengers having the same origin and destination clusters.

Before the simulation starts, the number of sub-regions needs to be determined. The solution quality is affected by the number of sub-regions. Therefore, there is a need for a fine-tuning of this parameter. To find the suitable range for the number of sub-regions, over 90 simulations are carried out, and the average objective results are shown in Figure 6.1.

The value 1 refers to the case where there is no sub-region restriction. All the studied area is considered to belong to the same cluster. As expected, this value produces low-quality results. While the clusters are increased, the objective value gradually decreases. However, after the 10-15 range, the objective value starts to increase again. Therefore, a value in this range should be selected. For most of the simulation runs, a value of 11 is chosen for the number of sub-regions.

It should also be noted that when the number of sub-regions is kept small, the computational efforts needed for the solution generation and improvement heuristics increase. When the number of sub-regions is small, it requires a larger neighborhood search for the solution improvements.

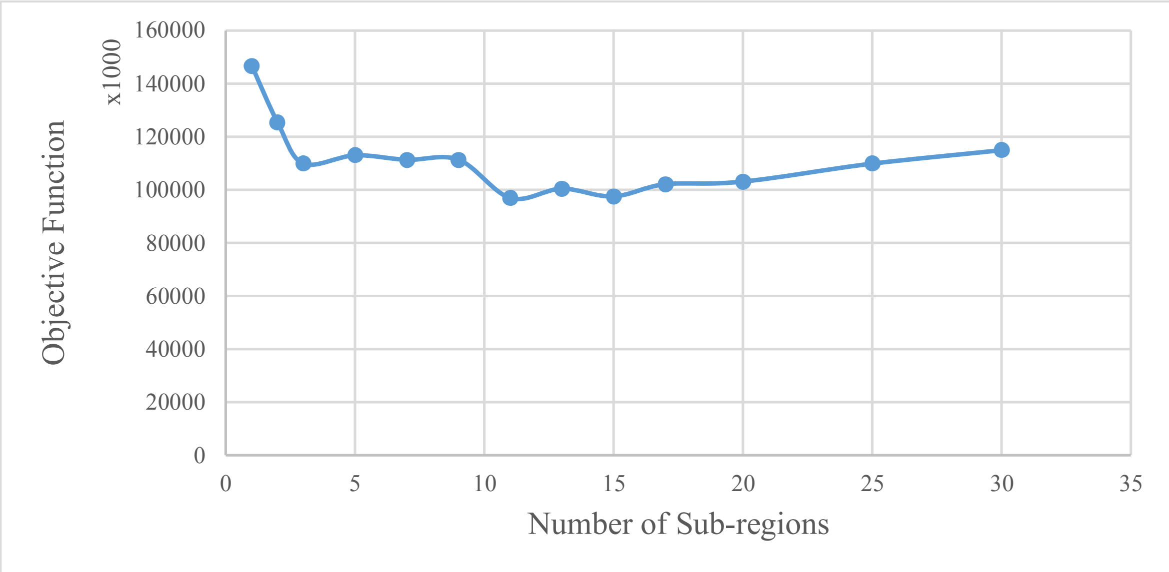

Figure 6.1. Average objective values vs. the number of sub-regions.

6.4. Improvement Types

As discussed in Chapter 4, four different improvement heuristics are proposed for this study. We analyzed the individual performance of the three operators. The proposed SA algorithm is mainly used for diversification purposes. Therefore, it is utilized jointly with other operators. The following improvement types are tested:

- Improvement Type 1: Relocate operator
- Improvement Type 2: Approximate swap operator
- Improvement Type 3: Relocate operator + SA algorithm.
- Improvement Type 4: Relocate operator + SA algorithm + Relocate operator.
- Improvement Type 5: Approximate swap operator + SA algorithm
- Improvement Type 6: Approximate swap operator + SA algorithm + Relocate operator
- Improvement Type 7: Relocate operator + Approximate swap operator
- Improvement Type 8: Exact swap operator
- Improvement Type 9: Relocate operator + Exact swap operator
- Improvement Type 10: Approximate swap operator + Relocate operator
- Improvement Type 11: Exact swap operator + Relocate operator

In some improvement types, the improvement heuristics are jointly used. For example, in improvement type 4, relocate operator is applied to the current solution just after the exact solution algorithm. After the relocate operator, the SA algorithm is applied to the same solution generated by the exact solution algorithm. The SA algorithm also takes the solution that is produced by the relocate operator as the local optima of the current solution. The solution produced by the SA algorithm is taken as the current solution, and the relocate operator is used again and applied to that solution. The proposed SA algorithm requires two solutions, the current solution and the local optima of that solution. Therefore, in improvement types including the SA algorithm, the solution produced by the exact solution algorithm is used. The solution generated by the operators is taken as the local optima by the SA algorithm.

The improvement types without the SA algorithm don't consider any local optima. The current solution is revised with the operators, then the solution generated by the operators is taken as the new current solution. The next operator uses only this solution for further improvements. In the improvement types including the SA algorithm, keep the solutions produced by each improvement step and take the best solution as the final solution.

6.5. Performance of Improvement Types

The proposed solution algorithms aim to find the solutions with the minimum objective function. Therefore, it is logical to evaluate the performance of the solution methods according to the objective values. For the passenger-vehicle assignment previously mentioned method is used for all solution methods. Then the optimal routes of the vehicles are found by the exact algorithm. The solution methods differentiate each other with the improvement types used for the solution improvement.

To evaluate the performance of the improvement types, several simulation runs for each improvement type were carried out. Figure 6.2 shows the average objective values obtained at the end of the various simulations. Improvement type 8 outperforms the rest of the improvement types in terms of objective value. However, as stated before, the exact swap operator, which is used in improvement type 8, requires high computational effort. Therefore, the improvement methods including the exact swap operator are computationally inefficient. Improvement type 6 performs well and doesn't require high computational time. In this method, the approximate swap operator, the SA algorithm, and the relocate operator are applied to the solution in order. The proposed SA algorithm satisfies the diversification and leads to high-quality results. When the approximate swap operator is used before the SA algorithm, the solution quality is increased like in improvement types 5 and 6. However, when the relocate operator is utilized before the SA algorithm, the improvement phase produces low-quality solutions like improvement types 3 and 4. This may have resulted from the fact that the individual performance of the approximate swap operator for solution improvement is higher than the relocate operator. One other reason could be the neighborhood structure used in the SA algorithm. The SA algorithm and the approximate swap operator share the same neighborhood structure. Since the SA algorithm uses the improved solution acquired from the predecessor improvement heuristic as the local optima of the current solution, sharing the same neighborhood structure might improve the success of the diversification.

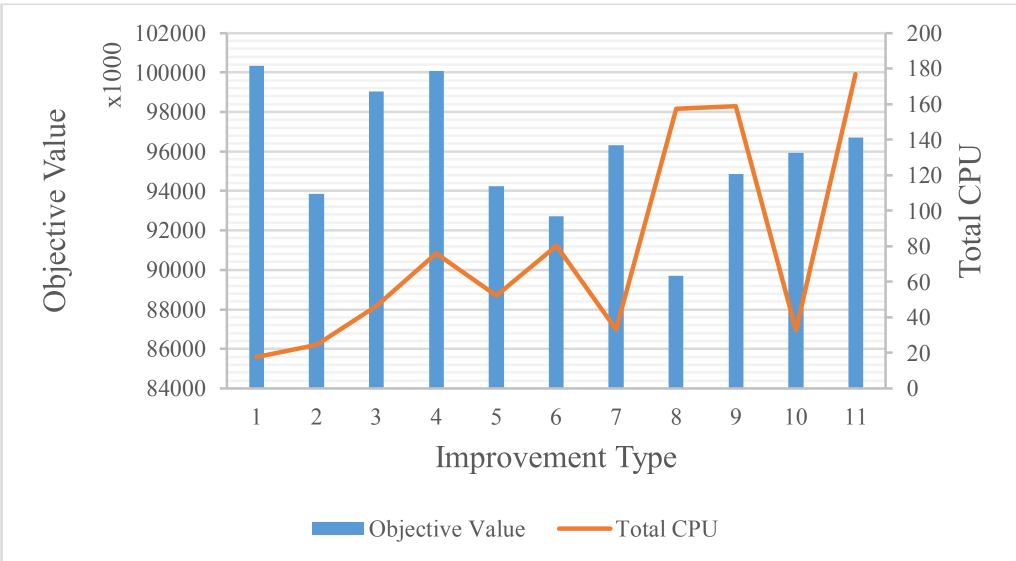

Figure 6.2. Objective values and total CPU of simulations vs. improvement types.

The objective function contains two main parameters, the total traveled distance of the fleet and the waiting time of the passengers. Figure 6.3 summarizes the performances of the improvement types on both traveled distance and the average waiting time. Except for the improvement type 6, all improvement types including the relocate operator produce results with less traveled distance. The reason behind this could be the fact that the relocate operator can reduce the number of vehicles being on the operation while the swap operators can only replace passengers between vehicles. By reducing the number of vehicles, the optimization algorithm can offer solutions with less traveled distance. However, when the number of vehicles is reduced, it is possible to have a high waiting time. There seems to be an inverse relationship between the traveled distance and the waiting times.

Figure 6.3. Total traveled distance and average waiting time vs. improvement types.

6.6. Operational Efficiency

To evaluate the operational efficiency of the fleet, we further analyzed whether the fleet is utilized properly. Figure 6.4 and Figure 6.5 show the result of an instance where improvement type 9 is used. As seen in both figures, the number of used vehicles reaches the size of the fleet in a short time and continues to stay at that point for almost the entire operation. This means that for most of the studied period, there are not idle vehicles. This shows that the fleet is not underutilized. Figure 6.4 also shows the occupancy rate of the moving vehicles and the whole fleet. Since almost all vehicles are assigned to passengers and on move, the occupancy rates of the moving vehicles and the whole fleet become equal. The average occupancy rate of the moving vehicles is 55%. Considering the fact that the vehicles need to move towards the origins of the passengers with available seats, this rate is found quite satisfying. The assignment rate in Figure 6.4 refers to the rate of assigned passengers to the capacity of the fleet. The average assignment rate is around 80%.

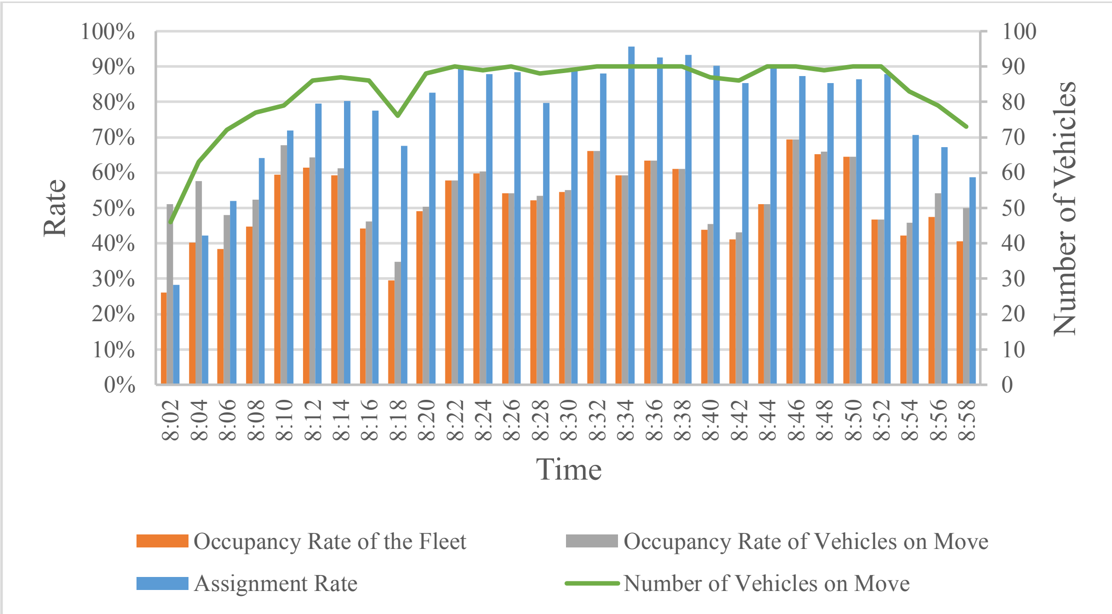

Figure 6.4. Occupancy and assignment rate of the fleet during the studied period.

Figure 6.5 shows the number of passengers onboard, the total number of passengers, and the number of passengers assigned to a vehicle during the studied period. The total number of passengers refers to the number of passengers who are not delivered to their destination points yet. The total number of passengers exceeds the capacity of the fleet at certain times of the execution. This shows that the demand of the studied period challenges the proposed system in terms of capacity. However, it is observed that even there is available capacity in the fleet, the algorithm doesn't make passenger-vehicle assignments. As discussed before, this is caused by the fact that there are certain geographical restrictions in the assignment phase. A vehicle cannot be assigned to a distant passenger even though no vehicle is assigned to that passenger.

The improvement heuristics contain the relocation of a passenger or swapping two passengers between vehicles. Therefore, these heuristics can only be applied to the passengers who are not onboard since passenger transfers are not allowed in our study. The gap between the number of passengers onboard and the number of assigned passengers Figure 6.5 represents these passengers. As seen in the figure, in some decision epochs, there are not that many passengers to whom the improvement heuristics can be applied.

Figure 6.5 also shows that the problem size for each decision epoch is considerably large. In some cases, the total number of passengers waiting for the delivery reaches over 450 passengers. This problem is repeatedly solved in every decision epoch.

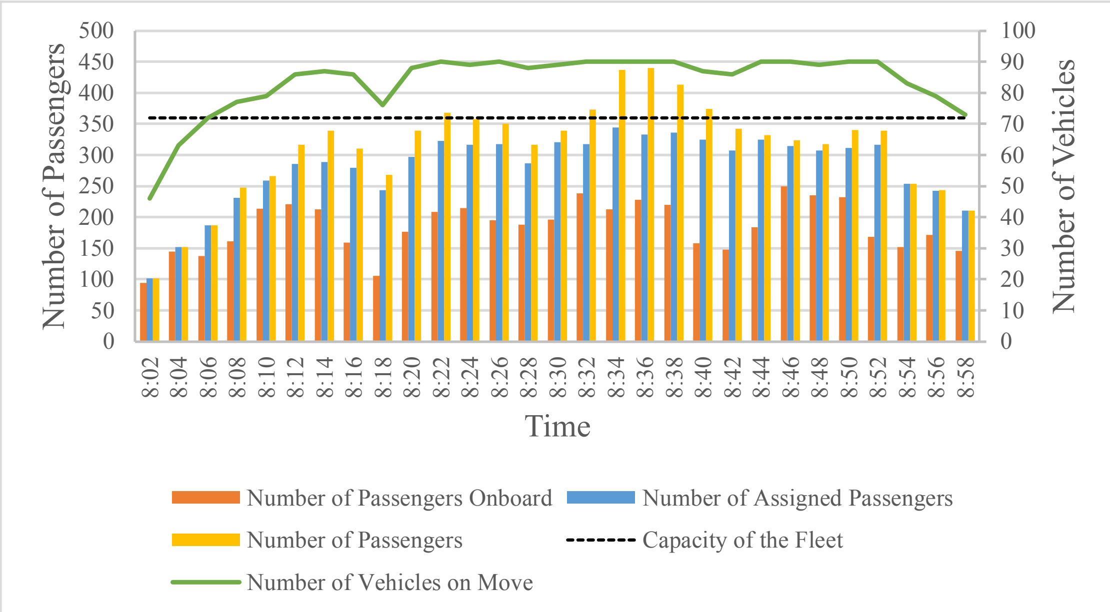

Figure 6.5. Passenger statistics.

7. CONCLUSION

In this study, we investigated the feasibility of a shared AV fleet to replace conventional bus routes. To achieve this, OD pairs of bus trips are inferred at an individual level. Then the related vehicle routing problem is formulated. The proposed transportation system is considered to be a dynamic dial-a-ride problem, which is a common variant of VRP. Then the solution algorithms are generated for both solution construction and improvement. The construction of a solution mainly contains two steps, grouping and routing. For the grouping of passengers into vehicles, we designed a new method that gives promising results in reasonable computational time. For the routing part, we developed an exact solution algorithm. With the help of capacity constraints of the vehicles in the proposed transportation system, the exact solution algorithm performs well and doesn't require high computational effort. After a complete solution is acquired, several improvement heuristics are applied to the solution. The improvement heuristics are applied to the problem individually and jointly to obtain the performance of all combinations of improvement heuristics. While some methods are not suitable for a highly dynamic environment, others achieved produce good results in a short time.

We have faced several challenges during this study. However, some of them deserve to be emphasized. One of the challenging tasks to infer the OD pairs of PT users was to eliminate the GPS errors. The algorithms used for the OD inferences mainly rely on the location data of the boarding. If incorrect data is not eliminated or corrected, both origin and destination points are inferred incorrectly. We developed several methods to correct the location data and infer the correct positions of the OD pairs.

Another challenge was to develop simulation software to obtain the performance of the proposed solution methods. Since our problem is a highly dynamic vehicle routing problem, it was important to apply this dynamism to the problem. To fully appreciate the dynamism, a simulation needs to be carried out with the inferred OD pairs. All the codes were developed specifically for this problem. The simulation also used a real network, and all settings of the network are designed to reflect the real-world situation.

We have tried countless methods and settings to solve the related problem. Some of them produced promising results, while others couldn't achieve to generate good results. Although some solution methods or settings were good at generating high-quality solutions, it was impractical to use them due to the high computational effort they required. Among all of these methods and settings, some affect the solution quality considerably. Therefore, we need to mention them for future works.

At the initial trials of the simulation, the passenger-vehicle assignments were carried to the following iterations. This means that if a passenger is assigned to a vehicle, it is impossible to change this assignment. However, due to the high dynamism of the studied problem, it is possible to have better options for the passenger-vehicle assignment with the new trip requests arrived in the following decision epochs. After we dismiss all assignments in the upcoming iterations, except for the passengers onboard, the solution quality and performance metrics were considerably improved.

Another important factor that greatly affected the solution quality was the introduction of sub-regions. With the help of sub-regions, passengers having OD pairs close to each other are grouped into the same vehicles. By achieving this, the solution construction phase produces high-quality solutions. The introduction of sub-regions also serves for the restrictions of neighborhood structure used in the solution improvement phase. Therefore, clustering the studied region into sub-regions not only helps to acquire promising solutions but also limits the computational effort needed to improve the current solutions.

As repeatedly stated before, one of the major concerns of the study was to limit the computational time. To generate the shortest path between the origins and the destinations requires some computational time in real networks. If the number of paths increases, the computational effort needed to find these paths and generate a distance matrix might reach unbearable levels. It is also impractical to generate all possible paths before the execution since even for a small network, the size of the distance matrix reaches high numbers. In this study, we first generate a distance matrix only for the bus stops. However, after a close look into the results of the simulations, it is realized that the routes of vehicles are concentrated in certain areas. After that point, we add a new path to the distance matrix when a new path is obtained during the simulation. From the point of this decision, countless simulation runs are completed, and the size of the distance matrix is converged to a reasonable level.

7.1. Investment and Operation Costs

One of the main goals of this study is to evaluate the feasibility of using a shared AV fleet instead of conventional buses. To achieve this, the operation and the capital cost of the proposed transportation service must be determined. In this study, we assume a fleet of AVs that are electric vehicles. Therefore, the comparison is made between electric vehicles and buses. Since autonomous cars and buses have not been widely used in the transportation domain and it is hard to find any information regarding the operation and the investment costs of such vehicles, the analysis is restricted to electric vehicles.

Quarles et.al. (2020) used 550,000 USD purchase price for a 12-meter electric bus in their study based on the statistics of various electric buses. On the other hand, the average price for an electric car was 55,600 USD in 2019, down from 64,300 USD in 2018, according to Cox Automotive (Crothers, 2020). In terms of purchase prices, electric buses are ten times more expensive than electric cars.

The proposed system can solve the problem with 90 vehicles while in the studied hour, the total number of 1996 trips were carried by 20 unique buses. In terms of investment cost, the proposed system has an obvious advantage over the conventional bus routes. This comparison is valid for such fleets containing autonomous vehicles or autonomous buses. If the vehicles have drivers, the costs related to the drivers should be included in the economic analysis. On the other hand, AV technology is expected to occupy the transportation domain in the near future. Therefore, compared to the conventional bus routes, the proposed transportation system would benefit the service providers at a significant level in terms of investment cost.

The average energy consumption for a 12-meter electric bus is found to be 1.15 kWh/km by a Dutch company ViriCiti upon a 10-months data collection of over 100 electric vehicles (ViriCiti, 2020). The type of buses used in the bus routes studied in this thesis is also 12-meter buses. The average energy consumption of electric cars is concentrated in the range between 0.14 to 0.2 kWh/km (Hao et al., 2020). This comparison shows us that electric buses consume 6-8 times more energy than electric cars.

The total traveled distance by the buses in the studied period and routes is shown in Table 7.1. The buses traveled about 674 km in the fixed routes to serve the trip requests in the related period. When the energy consumption rates of electric buses and electric cars are considered, a total traveled distance up to about 5,000 km is acceptable. Over this value, the proposed system would lose operational advantages. As shown in Chapter 6, the total distance of the AV fleet is almost always below 4,000 km for the studied period.

Table 7.1. Total traveled distance of the studied routes during the studied period.

Bus Route	Number of Tours	Total Distance of a Tour in km	Total Distance in km
1/T	11	22	242
3/G	8	22	176
35/G	10	16	160
4/G	6	16	96
		Total	**674**

The proposed system is designed to serve PT users a more comfortable transportation service. On the other hand, it also benefits the service providers in terms of investment in the fleet and operational costs. Ultimately both parties, users and service providers, would gain a lot from the implementation of such a system.

One of the highest operating costs of the current PT system is related to the drivers. Therefore, decision-makers would immediately utilize it when the AV technology is ready to be implemented in the PT domain. The proposed service eliminates the need for drivers and offers a more advantageous service economically. This finding is believed to be highly crucial for transit agencies.

7.2. Limitations

Although this study aimed to simulate the real-world application of the proposed transportation service as much as possible, there are certain limitations. Some of these limitations are related to the fact that we needed to solve the problem in a reasonable time to test the solution methods. Therefore, the problem should be restricted to some extent. Also, the hardware used for this study is a standard computer, while the hardware utilized in real life for such a system has high computational power. For this reason, our proposed model and the solution methods can be applied in real life with less computational time.

The methods used in our study for OD inferences are taken mostly from the previous studies. These studies validated their methods by various validation methods. Most of the studies used surveys to validate their OD inference algorithms. Since the OD matrix algorithm was not the core of this study, the validation process was not conducted. Even the methods used in OD inferences are validated methods by the previous studies, it is possible to have different results caused by the characteristics of the studied transit network.

In our study, passengers are not allowed to make a trip from an origin to a destination other than the predefined locations. Since the study was aimed to explore the feasibility of AVs' utilization in transportation networks instead of conventional busses, these origin and destination locations were selected as the present stops of the studied bus lines. This limits the problem size, especially in routing problems hence decrease the computational time. It is important to note that finding the shortest path, distance, and duration are extremely time-consuming if heuristics are used to find the optimal routing since the heuristics aim to find the optimum route by exploring the solution space. This means that countless paths should be calculated to converge the good results, causing an increase in computational efforts. However, it is highly possible that passengers using a shared taxi-like system prefer to decide where they are picked up and delivered. If the passengers are forced to come to the locations for their trips, they might expect further advantages from the proposed system like cost, comfort, efficiency, etc.

Depending on the assignment strategies and improvement heuristics, grouping and the routing steps in each iteration take some time. Even in some cases, computational time can reach 500 seconds per iteration due to the exhaustive searches in the improvement phase. In our study, the computational time needed to solve the problem is not taken into account, meaning that results of the grouping of the passengers and the routing of the vehicles are immediately executed in each iteration. Therefore, the simulation doesn't add the computational time to its time before the execution. The vehicle routing problem in each iteration is considered to be solved immediately. In real life, this cannot be realized since every step of the algorithm requires some computational time. On the other hand, specialized software and hardware are used in the real-life application of this kind of problem by the fleer operators. The performance of these systems is far better than the hardware that we used in our study. It should also be noted that even in our study, some of the improvement strategies require low computation time.

The proposed study was aimed to propose an autonomous shared taxi model for the transportation of PT users. For this purpose, the capacity of the vehicles was kept small. With the advantage of small capacities, exact solution methods are utilized for solving routing problems in the proposed algorithm. If the capacity of vehicles is increased, exact solution methods may need to be excluded from the algorithm.

7.3. Future Works

To reflect the last-mile problem, this study is conducted on a specific area of the public transportation network. When the size of the service area is increased, the problem size is also increased. Hence the computational effort needed to solve the related problem becomes higher. On the other hand, with the increased problem size, solution methods can produce solutions with higher quality. It would be much more practical to analyze the whole system since the fleet is sometimes underutilized, and the idle vehicles can serve in other service areas where the demand is high.

To simplify the problem, the congestion on the roads is not considered in this study. However, in real-life applications, congestion causes considerable delays in transportation services. To reflect the real-life situations more, future works should consider the congestion on the network, especially for rush hours.

We proposed a taxi-like transportation service for the studied problem. Therefore, we used vehicles with small capacities in this study. The number of possible routes for a vehicle is directly related to the number of passengers assigned to it. Even a small increase in the number of passengers expands the solution space dramatically. The limitation on the capacity enabled us to use the exact solution methods. However, in some cases, using small-capacity vehicles might be inefficient. Future works should improve the computational performance of the solution methods. By achieving this, the vehicles with different capacities can be used for the related problem.

In this study, the stochastic information of the bus trips was only used for determining the initial positions of the vehicles. It is highly possible to forecast the trip requests during the day if there is sufficient trip data. Future works can utilize the trip history of related bus routes and acquire the possible OD pairs for different periods of a day.

REFERENCES

Barry, J.J., R. Newhouser, A. Rahbee, and S. Sayeda, 2002, "Origin and Destination Estimation in New York City with Automated Fare System Data." *Transportation Research Record* 1817 (1): 183–87.

Battarra, M., J.-F. Cordeau, and M. Iori, 2014, "Chapter 6: Pickup-and-Delivery Problems for Goods Transportation." In *Vehicle Routing*, 161–91. Society for Industrial and Applied Mathematics, Philadelphia, PA.

Beaudry, A., G. Laporte, T. Melo, and S. Nickel, 2010, "Dynamic Transportation of Patients in Hospitals." *OR Spectrum* 32 (1): 77–107.

Bent, R.W., and P. Van Hentenryck, 2004, "Scenario-Based Planning for Partially Dynamic Vehicle Routing with Stochastic Customers." *Operations Research* 52 (6): 977–87.

Bodin, L., and B. Golden, 1981, "Classification in Vehicle Routing and Scheduling." *Networks* 11 (2): 97–108.

Boeing, G., 2017, "OSMnx: New Methods for Acquiring, Constructing, Analyzing, and Visualizing Complex Street Networks." *Computers, Environment and Urban Systems* 65 (September): 126–39.

Boyacı, B., T.H. Dang, and A.N. Letchford, 2021, "Vehicle Routing on Road Networks: How Good Is Euclidean Approximation?" *Computers and Operations Research* 129 (May): 105197.

Braekers, K., A. Caris, and G.K. Janssens, 2014, "Exact and Meta-Heuristic Approach for a General Heterogeneous Dial-a-Ride Problem with Multiple Depots." *Transportation Research Part B: Methodological* 67 (September): 166–86.

Branke, J., M. Middendorf, G. Noeth, and M. Dessouky, 2005, "Waiting Strategies for Dynamic Vehicle Routing." *Transportation Science* 39 (3): 298–312.

Brownell, C.K., 2013, *Shared Autonomous Taxi Networks: An Analysis of Transportation Demand in NJ and a 21 st Century Solution for Congestion*. Bachelor Thesis, Princeton University.

Chiang, W.C., and R.A. Russell, 1996, "Simulated Annealing Metaheuristics for the Vehicle Routing Problem with Time Windows." *Annals of Operations Research* 63 (1): 3–27.

Chong, Z.J., B. Qin, T. Bandyopadhyay, T. Wongpiromsarn, E.S. Rankin, M.H. Ang, E. Frazzoli, D. Rus, D. Hsu, and K.H. Low, 2011, "Autonomous Personal Vehicle for the First- and Last-Mile Transportation Services." In *Proceedings of the 2011 IEEE 5th International Conference on Cybernetics and Intelligent Systems, CIS 2011*, 253–60. IEEE.

Clarke, G., and J.W. Wright, 1964, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points." *Operations Research* 12 (4): 568–81.

Cordeau, J.F., and G. Laporte, 2003, "A Tabu Search Heuristic for the Static Multi-Vehicle Dial-a-Ride Problem." *Transportation Research Part B: Methodological* 37 (6): 579–94.

Cordeau, J.F., G. Laporte, M.W.P. Savelsbergh, and D. Vigo, 2007, "Chapter 6 Vehicle Routing." In *Handbooks in Operations Research and Management Science*, 14:367–428.

Cortés, C.E., M. Matamala, and C. Contardo, 2010, "The Pickup and Delivery Problem with Transfers: Formulation and a Branch-and-Cut Solution Method." *European Journal of Operational Research* 200 (3): 711–24.

Crothers, B., 2020, *Prices Are Coming Down For Electric Cars: The Most Affordable EVs Fall 2020 — Tesla Model 3 Vs The Rest*. 2020, https://www.forbes.com/sites/brookecrothers/2020/10/04/plan-to-buy-an-electric-car-the-most-affordable-evs-fall-2020---tesla-model-3-vs-the-rest/?sh=78fabf2d7be7, accessed in May 2021.

Cui, A., 2006, *Bus Passenger Origin-Destination Matrix Estimation Using Automated Data Collection Systems*. M.Sc. Thesis, Massachusetts Institute of Technology.

Dantzig, G., R. Fulkerson, and S. Johnson, 1954, "Solution of a Large-Scale Traveling-Salesman Problem." *Journal of the Operations Research Society of America* 2 (4): 393–410.

Detti, P., F. Papalini, and G.Z.M. de Lara, 2017, "A Multi-Depot Dial-a-Ride Problem with Heterogeneous Vehicles and Compatibility Constraints in Healthcare." *Omega (United Kingdom)* 70 (July): 1–14.

Doerner, K.F., and J.-J. Salazar-González, 2014, "Chapter 7: Pickup-and-Delivery Problems for People Transportation." In *Vehicle Routing*, edited by Paolo Toth and Daniele Vigo, Second, 193–212. Society for Industrial and Applied Mathematics, Philadelphia, PA.

Eksioglu, B., A.V. Vural, and A. Reisman, 2009, "The Vehicle Routing Problem: A Taxonomic Review." *Computers and Industrial Engineering* 57 (4): 1472–83.

Elshaer, R., and H. Awad, 2020, "A Taxonomic Review of Metaheuristic Algorithms for Solving the Vehicle Routing Problem and Its Variants." *Computers and Industrial Engineering* 140 (February).

Federal Highway Administration, 2011, "Our Nation's Highways: 2011."

Fernandes, P., and U. Nunes, 2010, "Platooning of Autonomous Vehicles with Intervehicle Communications in SUMO Traffic Simulator." In *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 1313–18. IEEE.

Fidanoglu, A., 2015, *Origin and Destination Inference of Bus Passengers: Istanbul Case Study*. M.Sc. Thesis, Boğaziçi University.

Flood, M.M., 1956, "The Traveling-Salesman Problem." *Operations Research* 4 (1): 61–75.

Glover, F., 1986, "Future Paths for Integer Programming and Links to Artificial Intelligence." *Computers and Operations Research* 13 (5): 533–49.

Golden, B.L., A.A. Assad, and E.A. Wasil, 2002, "Routing Vehicles in the Real World: Applications in the Solid Waste, Beverage, Food, Dairy, and Newspaper Industries." In *The Vehicle Routing Problem*, 245–86. Society for Industrial and Applied Mathematics.

Gordon, J.B., H.N. Koutsopoulos, and N.H.M. Wilson, 2018, "Estimation of Population Origin–Interchange–Destination Flows on Multimodal Transit Networks." *Transportation Research Part C: Emerging Technologies* 90 (May): 350–65.

Guerriero, F., M.E. Bruni, and F. Greco, 2013, "A Hybrid Greedy Randomized Adaptive Search Heuristic to Solve the Dial-a-Ride Problem." *Asia-Pacific Journal of Operational Research* 30 (1).

Häll, C.H., and A. Peterson, 2013, "Improving Paratransit Scheduling Using Ruin and Recreate Methods." *Transportation Planning and Technology* 36 (4): 377–93.

Hao, X., H. Wang, Z. Lin, and M. Ouyang, 2020, "Seasonal Effects on Electric Vehicle Energy Consumption and Driving Range: A Case Study on Personal, Taxi, and Ridesharing Vehicles." *Journal of Cleaner Production* 249 (March): 119403.

Heinrichs, D., 2016, "Autonomous Driving and Urban Land Use." In *Autonomous Driving: Technical, Legal and Social Aspects*, 213–31. Springer Berlin Heidelberg.

Ho, S.C., and D. Haugland, 2011, "Local Search Heuristics for the Probabilistic Dial-a-Ride Problem." *OR Spectrum* 33 (4): 961–88.

Ho, S.C., W.Y. Szeto, Y.H. Kuo, J.M.Y. Leung, M. Petering, and T.W.H. Tou, 2018, "A Survey of Dial-a-Ride Problems: Literature Review and Recent Developments." *Transportation Research Part B: Methodological* 111 (May): 395–421.

Hosny, M.I., 2010, *Investigating Heuristic and Meta-Heuristic Algorithms for Solving Pickup and Delivery Problems*. Ph.D. Thesis, Cardiff University.

Howard, D., and D. Dai, 2014, "Public Perceptions of Self-Driving Cars: The Case of Berkeley, California." *93rd Annual Meeting of the Transportation Research Board*. Berkeley, California.

Hyland, M.F., 2018, *Real-time Operation of Shared-use Autonomous Vehicle Mobility Services: Modeling, Optimization, Simulation, and Analysis*. Ph.D. Thesis,.

Ilani, H., E. Shufan, T. Grinshpoun, A. Belulu, and A. Fainberg, 2014, "A Reduction Approach to the Two-Campus Transport Problem." *Journal of Scheduling* 17 (6): 587–99.

Iqbal, S., M. Kaykobad, and M.S. Rahman, 2015, "Solving the Multi-Objective Vehicle Routing Problem with Soft Time Windows with the Help of Bees." *Swarm and Evolutionary Computation* 24 (October): 50–64.

Irnich, S., P. Toth, and D. Vigo, 2014, "Chapter 1: The Family of Vehicle Routing Problems." In *Vehicle Routing*, 1–33. Society for Industrial and Applied Mathematics.

Jaw, J.J., A.R. Odoni, H.N. Psaraftis, and N.H.M. Wilson, 1986, "A Heuristic Algorithm for the Multi-Vehicle Advance Request Dial-a-Ride Problem with Time Windows." *Transportation Research Part B* 20 (3): 243–57.

Johnson, D.S., and L.A. Mcgeoch, 1995, "The Traveling Salesman Problem: A Case Study in Local Optimization."

Jung, J., and K. Sohn, 2017, "Deep-learning Architecture to Forecast Destinations of Bus Passengers from Entry-only Smart-card Data." *IET Intelligent Transport Systems* 11 (6): 334–39.

Kaan, L., and E. V. Olinick, 2013, "The Vanpool Assignment Problem: Optimization Models and Solution Algorithms." *Computers and Industrial Engineering* 66 (1): 24–40.

Kockelman, K., P. Bansal, S.D. Boyles, M. Levin, P. Bujanovic, J. Li, T. Choudhary, et al., 2016, "Implications of Connected and Automated Vehicles On The Safety and Operations of Roadway Networks: A Final Report."

Krueger, R., T.H. Rashidi, and J.M. Rose, 2016, "Preferences for Shared Autonomous Vehicles." *Transportation Research Part C: Emerging Technologies* 69 (August): 343–55.

Kuo, Y., 2010, "Using Simulated Annealing to Minimize Fuel Consumption for the Time-Dependent Vehicle Routing Problem." *Computers and Industrial Engineering* 59 (1): 157–65.

Lahyani, R., M. Khemakhem, and F. Semet, 2015, "Rich Vehicle Routing Problems: From a Taxonomy to a Definition." *European Journal of Operational Research* 241 (1): 1–14.

Laporte, G., S. Ropke, and T. Vidal, 2014, "Chapter 4: Heuristics for the Vehicle Routing Problem." In *Vehicle Routing*, 87–116. Society for Industrial and Applied Mathematics, Philadelphia, PA.

Lenz, B., and E. Fraedrich, 2016, "New Mobility Concepts and Autonomous Driving: The Potential for Change." In *Autonomous Driving: Technical, Legal and Social Aspects*, edited by Markus Maurer, J. Christian Gerdes, Barbara Lenz, and Hermann Winner, 173–91. Springer Berlin Heidelberg.

Li, X., S. Hu, W. Fan, and K. Deng, 2018, "Modeling an Enhanced Ridesharing System with Meet Points and Time Windows." *PLoS ONE* 13 (5).

Lin, S., 1965, "Computer Solutions of the Traveling Salesman Problem." *Bell System Technical Journal* 44 (10): 2245–69.

Lorini, S., J.Y. Potvin, and N. Zufferey, 2011, "Online Vehicle Routing and Scheduling with Dynamic Travel Times." *Computers and Operations Research* 38 (7): 1086–90.

Martin, S., 2019, *The Edge of Large-Scale Optimization in Transportation and Machine Learning*. Ph.D. Thesis, Massachusetts Institute of Technology.

Masson, R., F. Lehuédé, and O. Péton, 2014, "The Dial-a-Ride Problem with Transfers." *Computers and Operations Research* 41 (1): 12–23.

Mauri, G.R., L. Antonio, and N. Lorena, 2009, "Customers' Satisfaction in a Dial-a-Ride Problem." *IEEE Intelligent Transportation Systems Magazine* 1 (3): 6–14.

Mckinney, W., 2011, "Pandas: A Foundational Python Library for Data Analysis and Statistics." *Python for High Performance and Scientific Computing* 14 (9): 1–9.

Molenbruch, Y., K. Braekers, A. Caris, and G. Vanden Berghe, 2017, "Multi-Directional Local Search for a Bi-Objective Dial-a-Ride Problem in Patient Transportation." *Computers and Operations Research* 77 (January): 58–71.

Mourad, A., J. Puchinger, and C. Chu, 2019, "A Survey of Models and Algorithms for Optimizing Shared Mobility." *Transportation Research Part B: Methodological* 123: 323–46.

Munizaga, M.A., and C. Palma, 2012, "Estimation of a Disaggregate Multimodal Public Transport Origin-Destination Matrix from Passive Smartcard Data from Santiago, Chile." *Transportation Research Part C: Emerging Technologies* 24 (October): 9–18.

NHTSA, 2008, "National Motor Vehicle Crash Causation Survey: Report to Congress."

Nikolaev, A.G., and S.H. Jacobson, 2010, "Simulated Annealing." In , 1–39. Springer, Boston, MA.

Osman, I.H., 1993, "Metastrategy Simulated Annealing and Tabu Search Algorithms for the Vehicle Routing Problem." *Annals of Operations Research* 41 (4): 421–51.

Pandi, R.R., S.G. Ho, S.C. Nagavarapu, T. Tripathy, and J. Dauwels, 2018, "GPU-Accelerated Tabu Search Algorithm for Dial-A-Ride Problem." In *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2018-November:2519–24. Institute of Electrical and Electronics Engineers Inc.

Paquette, J., J.F. Cordeau, G. Laporte, and M.M.B. Pascoal, 2013, "Combining Multicriteria Analysis and Tabu Search for Dial-a-Ride Problems." *Transportation Research Part B: Methodological* 52 (June): 1–16.

Parragh, S.N., K.F. Doerner, and R.F. Hartl, 2008, "A Survey on Pickup and Delivery Problems." *Journal Fur Betriebswirtschaft* 58 (2): 81–117.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, et al., 2011, "Scikit-Learn: Machine Learning in Python." *Journal of Machine Learning Research* 12: 2825–30.

Pillac, V., M. Gendreau, C. Guéret, and A.L. Medaglia, 2013, "A Review of Dynamic Vehicle Routing Problems." *European Journal of Operational Research* 225: 1–11.

Potvin, J.-Y., and J.-M. Rousseau, 1995, "An Exchange Heuristic for Routeing Problems with Time Windows." *The Journal of the Operational Research Society* 46 (12): 1433.

Prosser, P., and P. Shaw, 1996, "Study of Greedy Search with Multiple Improvement Heuristics for Vehicle Routing Problems." *University of Strathclyde*.

Psaraftis, H.N., 1995, "Dynamic Vehicle Routing: Status and Prospects." *Annals of Operations Research* 61 (1): 143–64.

Pureza, V., and G. Laporte, 2008, "Waiting and Buffering Strategies for the Dynamic Pickup and Delivery Problem with Time Windows." *INFOR* 46 (3): 165–75.

Quarles, N., K.M. Kockelman, and M. Mohamed, 2020, "Costs and Benefits of Electrifying and Automating Bus Transit Fleets." *Sustainability (Switzerland)* 12 (10).

Rayle, L., D. Dai, N. Chan, R. Cervero, and S. Shaheen, 2016, "Just a Better Taxi? A Survey-Based Comparison of Taxis, Transit, and Ridesourcing Services in San Francisco." *Transport Policy* 45 (January): 168–78.

Reinhardt, L.B., T. Clausen, and D. Pisinger, 2013, "Synchronized Dial-a-Ride Transportation of Disabled Passengers at Airports." *European Journal of Operational Research* 225 (1): 106–17.

Ritzinger, U., J. Puchinger, and R.F. Hartl, 2016, "A Survey on Dynamic and Stochastic Vehicle Routing Problems." *International Journal of Production Research* 54 (1): 215–31.

Saint-Guillain, M., C. Solnon, and Y. Deville, 2017, "The Static and Stochastic VRP with Time Windows and Both Random Customers and Reveal Times." In *20th European Conference on Applications of Evolutionary Computation (EvoApplications)*, 110–27. Amsterdam.

Sánchez-Martínez, G.E., 2017, "Inference of Public Transportation Trip Destinations by Using Fare Transaction and Vehicle Location Data: Dynamic Programming Approach." *Transportation Research Record* 2652 (January): 1–7.

Savelsbergh, M., 1992, "The Vehicle Routing Problem with Time Windows: Minimizing Route Duration." *INFORMS Journal On Computing* 4 (2): 146–54.

Savelsbergh, M.W.P., and M. Sol, 1995, "The General Pickup and Delivery Problem." *Transportation Science* 29 (1): 17–29.

Scheltes, A., and G.H. de Almeida Correia, 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.

Seow, K.T., N.H. Dang, and D.H. Lee, 2010, "A Collaborative Multiagent Taxi-Dispatch System." *IEEE Transactions on Automation Science and Engineering* 7 (3): 607–16.

Shen, Y., H. Zhang, and J. Zhao, 2018, "Integrating Shared Autonomous Vehicle in Public Transportation System: A Supply-Side Simulation of the First-Mile Service in Singapore." *Transportation Research Part A: Policy and Practice* 113 (July): 125–36.

Sleator, D.D., and R.E. Tarjan, 1985, "Amortized Efficiency of List Update and Paging Rules."

Solomon, M.M., 1987, "Algorithms for The Vehicle Routing and Scheduling Problems with Time Window Constraints." *Operations Research* 35 (2): 254–65.

Stiglic, M., N. Agatz, M. Savelsbergh, and M. Gradisar, 2015, "The Benefits of Meeting Points in Ride-Sharing Systems." *Transportation Research Part B: Methodological* 82 (December): 36–53.

Torgal, M., T.G. Dias, and T. Fontes, 2021, "A Multi Objective Approach for DRT Service Using Tabu Search." In *Transportation Research Procedia*, 52:91–98. Elsevier B.V.

Trépanier, M., N. Tranchant, and R. Chapleau, 2007, "Individual Trip Destination Estimation in a Transit Smart Card Automated Fare Collection System." *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations* 11 (1): 1–14.

Union of Concerned Scientists, 2017, "Maximizing the Benefits of Self-Driving Vehicles: Principles for Public Policy."

ViriCiti, 2020, "E-Bus Performance."

Virtanen, P., R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, et al., 2020, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python." *Nature Methods* 17 (3): 261–72.

Voelcker, J., 2014, *1.2 Billion Vehicles On World's Roads Now, 2 Billion By 2035: Report*. 2014, https://www.greencarreports.com/news/1093560_1-2-billion-vehicles-on-worlds-roads-now-2-billion-by-2035-report, accessed in November 2020.

Wang, H., and A. Odoni, 2016, “Approximating the Performance of a ‘Last Mile’ Transportation System.” *Transportation Science* 50 (2): 659–75.

Wang, W., J.P. Attanucci, and N.H.M. Wilson, 2011, “Bus Passenger Origin-Destination Estimation and Related Analyses Bus Passenger Origin-Destination Estimation and Related Analyses Using Automated Data Collection Systems Background and Purpose.” *Journal of Public Transportation* 14 (4).

Wen, J., Y.X. Chen, N. Nassir, and J. Zhao, 2018, “Transit-Oriented Autonomous Vehicle Operation with Integrated Demand-Supply Interaction.” *Transportation Research Part C: Emerging Technologies* 97 (January): 216–34.

WHO, 2018, “Global Status Report On Road Safety.” Geneva.

World Energy Council, 2019, “World Energy Scenarios.”

Xu, Z., K. Zhang, H. Min, Z. Wang, X. Zhao, and P. Liu, 2018, “What Drives People to Accept Automated Vehicles? Findings from a Field Experiment.” *Transportation Research Part C: Emerging Technologies* 95 (July): 320–34.

Yan, X., J. Levine, and X. Zhao, 2019, “Integrating Ridesourcing Services with Public Transit: An Evaluation of Traveler Responses Combining Revealed and Stated Preference Data.” *Transportation Research Part C: Emerging Technologies* 105 (August): 683–96.

Yu, V.F., S.S. Purwanti, A.A.N.P. Redi, C.C. Lu, S. Suprayogi, and P. Jewpanya, 2018, “Simulated Annealing Heuristic for the General Share-a-Ride Problem.” *Engineering Optimization* 50 (7): 1178–97.

Zhao, J., A. Rahbee, and N.H.M. Wilson, 2007, “Estimating a Rail Passenger Trip Origin-Destination Matrix Using Automatic Data Collection Systems.” *Computer-Aided Civil and Infrastructure Engineering* 22 (5): 376–87.