

MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

The Potential Impact of Autonomous Passenger Ferry Services on Urban Mobility

verfasst von / submitted by
Emily Johanna Louise Weidt, B.Sc.

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of
Master of Science (MSc)

Wien, 2023 / Vienna 2023

Studienkennzahl lt. Studienblatt /
degree programme code as it appears on
the student record sheet:

UA 066 915

Studienrichtung lt. Studienblatt /
degree programme as it appears on
the student record sheet:

Masterstudium Betriebswirtschaft

Betreut von / Supervisor:

Univ.-Prof. Dr. Jan Fabian Ehmke

Abstract

Numerous cities are situated in coastal areas or along rivers. Despite the demand for bridges and tunnels that arises from this geographical layout, high construction costs and limited space constrain the options for crossing waterways in many regions. The continuously growing urbanization makes integrating waterways into city infrastructure crucial for efficient mobility, viewing them as potential assets rather than obstacles. This thesis explores autonomous ferry systems as innovative solutions for crossing waterways and reducing route distances. Beyond a comprehensive literature review, this work seeks to propose a research approach for evaluating the potential of autonomous ferry systems by introducing a re-usable Python application to facilitate a ferry service. This application encompasses the generation of fictional passenger requests and the subsequent routing of ferries within the system to meet all demands as effective as possible compared to routes entailing conventional river crossings. The underlying optimization model corresponds to an adapted version of the Dial-A-Ride Problem with Penalty Costs. It is investigated by a case study involving various configurations.

Zusammenfassung

Trotz des Bedarfs an Brücken und Tunneln in Städten, die an Flüssen oder in Küstengebieten liegen, schränken hohe Baukosten und begrenzter Raum oft die Möglichkeiten zur Überquerung von Wasserwegen in vielen Regionen ein. Mit wachsender Urbanisierung wird die Integration von Wasserstraßen in die städtische Infrastruktur aber unerlässlich, um die Mobilität effizient zu verbessern und Wasserwege als potenzielle Vorteile anstelle von Hindernissen zu betrachten. Diese Arbeit untersucht autonome Fährsysteme als innovative Lösungen für die Überquerung von Wasserstraßen und die damit verbundene Verkürzung von Routen. Neben einer umfassenden Literaturübersicht zielt die Arbeit darauf ab, einen Forschungsansatz für die Bewertung des Potenzials autonomer Fährsysteme bereitzustellen, indem eine wiederverwendbare Python-Anwendung vorgestellt wird. Diese Anwendung umfasst die Generierung fiktiver Fahrgastanfragen und das anschließende Routing von Fähren innerhalb des Systems, sodass Passagieranfragen so effektiv wie möglich erfüllt werden können, insbesondere im Vergleich zu Routen mit herkömmlichen Flussüberquerungen. Das zugrunde liegende Optimierungsmodell entspricht einer angepassten Version des Dial-A-Ride-Problems mit Strafkosten und wird anhand einer Fallstudie unter verschiedenen Konfigurationen untersucht.

Table of Contents

1	INTRODUCTION.....	1
1.1	RESEARCH QUESTION.....	1
1.2	STRUCTURE OF THESIS.....	2
2	THEORETICAL BACKGROUND.....	3
2.1	AUTONOMOUS TECHNOLOGIES ON ROADS AND WATER.....	3
2.2	AUTONOMOUS FERRIES.....	6
2.2.1	The Emergence of Autonomous Shipping.....	6
2.2.2	Small Autonomous Passengers Ferries: Status Quo.....	7
2.3	SERVICE DESIGN AND OPTIMIZATION MODELING.....	11
2.3.1	Introducing the Dial-A-Ride-Problem	13
3	URBAN-AUTONOMOUS-FERRY-SERVICE.....	17
3.1	PROBLEM DESCRIPTION.....	18
3.1.1	Customer Satisfaction	21
3.2	MATHEMATICAL FORMULATION	21
3.2.1	Objective Function	25
3.2.2	Constraints	25
4	IMPLEMENTATION OF THE UAF-SERVICE.....	27
4.1	APPLICATION ENVIRONMENT.....	27
4.1.1	Parameter Settings.....	28
4.1.2	Technological Infrastructure and Visualization.....	29
4.2	APPLICATION PROCESS	29
4.2.1	Demand Generation.....	30
4.2.2	Station Assignment.....	31
4.2.3	Clean Requests.....	32
4.2.4	Optimize Routes	33
4.2.5	Result Evaluation	35
5	COMPUTATIONAL STUDIES	36
5.1	PERFORMANCE INDICATORS	36
5.2	SPECIFYING THE AREA OF OPERATION	37
5.3	EXPERIMENTAL SETUP.....	38
5.4	RESULTS AND DISCUSSION	39
5.4.1	Single-Instance Analysis.....	39
5.4.2	Multi-Instance Analysis.....	51
6	CONCLUSION.....	56
	APPENDIX.....	57
A	GITHUB REPOSITORY	57
B	EXEMPLIFYING RESULT SET	57
	BIBLIOGRAPHY	61

List of Figures

FIGURE 1. ILLUSTRATIONS OF AUTONOMOUS PASSENGER FERRIES BY ZEABUS	9
FIGURE 2. 3D-PRINTED FERRY AND ON-DEMAND INFRASTRUCTURE	10
FIGURE 3. ILLUSTRATIONS OF AUTONOMOUS PASSENGER FERRIES	11
FIGURE 4. CONVENTIONAL ROUTE AND FERRY ROUTE IN COMPARISON	17
FIGURE 5. ILLUSTRATION OF THE FERRY ROUTE CONSISTING OF THREE SUB ROUTES	18
FIGURE 6. PROCESS DIAGRAM OF THE UAF-SERVICE	19
FIGURE 7. ILLUSTRATION OF GENERATED GRAPH G	20
FIGURE 8. APPLICATION MODULES	29
FIGURE 9. SPECIFIED USE CASE AREA	30
FIGURE 10. COMPARING STATION ASSIGNMENTS	31
FIGURE 11. RESULTING TIME-WINDOWS	34
FIGURE 12. DEPLOYED APPLICATION FRONT-END	41
FIGURE 13. TIME SAVINGS IN % (SINGE-INSTANCE ANALYSIS WITH PF = 1)	43
FIGURE 14. DELAYS IN MINUTES (SINGE-INSTANCE ANALYSIS WITH PF = 1)	43
FIGURE 15. AVG. DISTANCE TRAVELLED PER FERRY (SINGE-INSTANCE ANALYSIS WITH PF = 1)	44
FIGURE 16. TIME SAVINGS IN % (SINGE-INSTANCE ANALYSIS WITH PF=100)	46
FIGURE 17. DELAYS IN MINUTES (SINGE-INSTANCE ANALYSIS WITH PF = 100)	46
FIGURE 18. AVG. DISTANCE TRAVELLED PER FERRY (SINGE-INSTANCE ANALYSIS WITH PF = 100)	47
FIGURE 19. TIME SAVINGS IN % (SINGE-INSTANCE ANALYSIS WITH PF=1000)	49
FIGURE 20. DELAYS IN MINUTES (SINGE-INSTANCE ANALYSIS WITH PF = 1000)	49
FIGURE 21. AVG. DISTANCE TRAVELLED PER FERRY (SINGE-INSTANCE ANALYSIS WITH PF = 1000)	50
FIGURE 22. AVG. TIME AND DISTANCE SAVINGS OF 5 HOMOGENEOUS INSTANCES (N = 10)	51
FIGURE 23. AVG. TIME-BASED RESULTS OF 5 HOMOGENEOUS INSTANCES (N = 10)	52
FIGURE 24. AVG. TIME AND DISTANCE SAVINGS OF 5 HOMOGENEOUS INSTANCES (N = 15)	53

List of Tables

TABLE 1. UAF-DARF MODEL NOTATION	23
TABLE 2. PARAMETERS TO BE CONFIGURED IN THE UAF-SERVICE-APPLICATION	28
TABLE 3. CONFIGURATION FOR SINGE-INSTANCE ANALYSIS	40
TABLE 4. TIME-RELATED RESULTS (SINGE-INSTANCE ANALYSIS WITH PF = 1)	43
TABLE 5. DISTANCE-RELATED RESULTS (SINGE-INSTANCE ANALYSIS WITH PF=1)	45
TABLE 6. TIME-RELATED RESULTS (SINGE-INSTANCE ANALYSIS WITH PF=100)	46
TABLE 7. DISTANCE-RELATED RESULTS (SINGE-INSTANCE ANALYSIS WITH PF=1)	47
TABLE 8. TIME-RELATED RESULTS (SINGE-INSTANCE ANALYSIS WITH PF = 1000)	49
TABLE 9. DISTANCE-RELATED RESULTS (SINGE-INSTANCE ANALYSIS WITH PF=1000)	50
TABLE 10. AVG. DISTANCE-BASED RESULTS OF 5 HOMOGENEOUS INSTANCES (N = 10)	52
TABLE 11. AVG. TIME-BASED RESULTS OF 5 HOMOGENEOUS INSTANCES (N = 15)	54
TABLE 12. AVG. DISTANCE-BASED RESULTS OF 5 HOMOGENEOUS INSTANCES (N = 15)	54

1 Introduction

Presently, 8 of the world's 10 largest cities are located in coastal areas or areas along rivers (Aslaksen et al., 2021) with approximately 40% of the global population residing in these regions (Reddy et al., 2019). Despite the resulting demand for bridges and tunnels, high construction costs and limited space keep the number of options for traversing waterways low in many areas. The absence of bridges increases the travel distance between locations separated by water, despite their geographical proximity. Travel distances are stretched far beyond what the eye perceives on a map, encouraging people, especially commuters, to use cars instead of biking or walking their way, letting bridges and tunnels become bottlenecks for road-based mobility. Rather than perceiving water ways as obstacles to traffic flow, they should be recognized as untapped potential for public transport (Guo et al., 2023). Also, the continuously growing urbanization further underscores the necessity of integrating waterways into urban infrastructure. Passenger ferry services are envisioned to improve urban infrastructure by bridging the barriers of water for bicycles and pedestrians. Especially the promising development of autonomous technologies in the field of mobility services opens new opportunities to replace missing bridges with ferry services, while reducing the limitations of traditional ferry lines. In this work, an approach for designing an autonomous ferry service is presented. The motivational goal behind the service is to reduce distances to the point where bicycling becomes even faster than driving and to create modal shifts towards foot travel and bike riding. Fewer people commuting by car result in external benefits such as reduced emissions and congestion on busy roads leading over bridges or even an improvement in public health. Also, poorly connected suburbs benefit from growing accessibility when expanding transportation networks by ferry services, making certain areas more attractive (Tarkowski & Puzdrakiewicz, 2021).

1.1 Research Question

In this paper, the research question of whether autonomous ferry services can serve as an efficient extension of urban infrastructure is investigated from an operations research perspective. Therefore, the objective is to make the potential quantifiable through the definition of performance indicators. The introduction of a new ferry service is only

considered successful if it brings improvements to passengers compared to the status quo. But how can the best service conditions be determined to provide the greatest possible savings despite many simultaneous requests? To tackle this question, a mathematical optimization problem based on the Dial-A-Ride-Problem (DARP), should be integrated into the ferry service to assist in determining the best possible routing within a given resource constraint. To obtain quantifiable results of the presented service, an application was written in the Python programming language that integrates the Gurobi math solver along with other APIs and libraries. The custom application developed for the introduced autonomous ferry service allows the analysis of different service configurations within a simulated demand environment. It is particularly investigated how parameters like fleet size, objective functions, and time-windows impact the quality of service.

1.2 Structure of Thesis

To find answer to the research question, Chapter 2 lays the theoretical groundwork by reviewing literature covering autonomous driving on roads and water (see Section 2.1), ongoing projects related to autonomous ferries (see Section 2.2), as well as foundations about the underlying optimization problem arising from the autonomous ferry service (see Section 2.3), which is introduced in Section 3. After describing the service, Section 3 continues by presenting the mathematical formulation rooted in the Dial-A-Ride problem before diving into the implementation of the optimization model. Therefore, before concluding Section 3, an overview of the custom-developed application, its processes, modules, and functions is provided. Section 4 then explores the computational results obtained through the application before summarizing the findings and results of this work in Section 5. The application repository published on GitHub serves as an important complement to this work, enabling the reuse of code through detailed documentation. Furthermore, the Appendix of this work includes an exemplifying result set and provides access to the application code.

2 Theoretical Background

In preparation for the computational studies of an autonomous ferry service, this chapter examines relevant literature to convey fundamentals concerning autonomous technologies on road and water, ongoing projects associated with autonomous ferry services, and an in-depth discussion of the associated optimization problem. The fact that the ferries are not operated manually but autonomously is crucial for three essential benefits: 24/7 availability, high flexibility, and geographic accessibility that should be guaranteed regardless of the general demand. Only through these factors can a ferry service be considered a complete substitute for bridges and tunnels. Autonomous driving services are quantified with higher initial investments, but the ongoing operating costs can be reduced significantly. An analogous example is provided by a study conducted by Deutsche Bahn, revealing that the personnel costs of a bus service account for around 60% of the total running costs (Canzler & Knie, 2023). Reducing personnel requirements is particularly important as the transport sector is struggling to find qualified staff: according to current estimates, there will be a deficit of 87,000 bus drivers in Germany in 2030 (Deutsche Bahn, 2023). Due to regulations, a ferry in Stockholm for example, usually requires two or more operators. In addition, the salary of an operator at sea is usually higher than that of his counterpart on land (Cheemakurthy, 2017). The high scalability associated with reduced personnel cost might in turn have positive effects on availability and connectivity to less populated areas.

2.1 Autonomous Technologies on Roads and Water

As media-present industry players like Google and Tesla are continuously working on the commercialization of self-driving cars, autonomous technology is primarily linked to land-based transportation. Although the number of publications on autonomous technology on water has grown in the last decade, the focus of many studies remains at a technical level, rather than on the analysis of potentials (Gu & Wallace, 2021). On both, roads and water, *Autonomous Driving* is a term used to describe transport systems able to move purposefully without the intervention of a human driver. The idea of self-driving vehicles has been around for decades, but technological realization used to face expensive challenges. Nowadays, the rapid development of (communication) technology set new

standards and opportunities, which continuously triggered the emergence of autonomous vehicle (AV) technologies at new (Bagloee et al., 2016). This is why autonomous vehicles are now perceived as a potential technological disruption, although barriers to mass-market penetration remain (Fagnant & Kockelman, 2015). Once these barriers are overcome, AVs are likely to dramatically change the transportation network. Fully-autonomous vehicles have the potential to democratize mobility by eliminating labor costs and personnel requirements (Hyland & Mahmassani, 2018). Especially when considering growing opportunities for sharing economies which might correlate with growing dispensability of personal vehicles. Transferring control from humans to vehicles is a stepwise process. Classifying this process aims to bring clarity to the role and responsibility of a human driver (Serban et al., 2018). Against this background, the National Highway Traffic Safety Administration (NHTSA)¹ differs between six different levels of automation. Only vehicles whose automation level is high (level 4) or complete (level 5) are considered self-driving cars (Arena et al., 2020). In analogy to the aforementioned classification, a similar classification has been adopted for autonomous ships (Felski & Zwolak, 2020). For both concepts, vehicles of lower levels might already have certain functions which support the human driver while still requiring the driver's attention. **Level 0** applies to vehicles demanding human drivers to take full control, resulting in only five levels that cope with autonomous functions. **Level 1** involves function-specific automation where at least one function in a vehicle is fully automated. **Level 2** is further extended by combining automated functions. To exemplify the cooperation of multiple functions in level 2, Bagloee et al. (2016) mention adaptive cruise control paired with an automated lane centering function for road-based vehicles. In the case of autonomous vessels, the operator does not have to be aboard for controlling the vessel from level 2 onwards. **Level 3** is related to limited self-driving automation, which entails the ability of autonomous driving, while expecting the driver or operator to be available and stay cautious. Under certain traffic and environment conditions, the driver or operator is required to take over control with sufficient time enabling a smooth transition. Cars of level 3 are especially suited for the relatively uncomplicated traffic on highways. Decision-making for autonomous vessels is shared between the operator and the Unmanned Vessel. **Level 4** refers to full automation but only under certain conditions.

¹ the U.S. civilian federal highway and vehicle safety agency.

If all conditions are met, the driver or operator is not required to be available for control at any time but may have the option to control the vehicle. Automation functions are limited in extreme weather conditions. **Level 5** vehicles can perform all driving functions under all conditions, making the driver become a pure passenger (Bagloee et al., 2016). Unmanned vessels of level 5 invoke autonomous functions by the on-board systems without notifying any external units or operators (Felski & Zwolak, 2020). Summing up, Level 0, level 1, and level 2 requires the human driver to carefully monitor the vehicles environment, while from level 3 onwards, this task is taken over by the vehicle itself (Serban et al., 2018).

Level 4 and 5 cars thus fulfill the image of futuristic mobility that has been drawn for years: People can devote themselves to completely different tasks while being driven safely to the destination of their choice (Anderson et al., 2014; Bagloee et al., 2016; Becker et al., 2016; NHTSA, 2021). Vessels from level 2 onwards can already be operated from another location. In 2020, car manufacturers were planning to test the market for autonomous vehicles of level 3 or possibly level 4 in 2021 and 2022. In some regions, vehicles capable of Level 3 autonomy reached the market, while only few countries have legal frameworks permitting their sale and use. Major market players like Tesla and Volkswagen forecast to bring autonomous vehicles of levels higher than 3 on the market in the next couple of years (Arena et al., 2020). In 2022, market available cars reach level 2. Recent developments for market-ready autonomous shipping have lagged behind their road-driven counterparts, as the current focus is more on the electrification of vessels and boats. However, there are efforts and projects on autonomous shipping as well, mostly by academic institutions, as described in Section 2.2.2.

Communication between vehicles and infrastructure is an essential component of autonomous driving. Therefore, it is not the individual AV itself that brings the greatest benefit, but rather an AV as part of an autonomous fleet. Connected vehicles are not only able to communicate with relevant agents in traffic but also to share and receive traffic data via satellite navigation systems in order to avoid queues (Arena et al., 2020). By this means, connectivity amongst vehicles has the potential to heavily increase efficiency of vehicle routing. Bagloee et al (2016) categorizes two traffic patterns in either cooperative or non-cooperative patterns. When seeking the best route for getting to a destination, it is often concentrated on the shortest path only. This entails that every vehicle in a traffic

system opportunistically chooses its own shortest path, disregarding the choices of other traffic participants. The resulting equilibrium is known as non-cooperative since knowledge about other vehicles' routes is limited. In contrast, connected vehicles may benefit from cooperative traffic pattern, as vehicles can access information about routings and destinations of other vehicles. Therefore, routing methods are not only based on shortest paths, but also on how many other vehicles are already using certain routes to prevent identical routes and thus avoid congestion (Bagloee et al., 2016).

2.2 Autonomous Ferries

The concept of autonomous ships is gaining momentum in the maritime industry as it can improve safety, efficiency, and sustainability in maritime transport. But still, ferry systems are rarely included in public transportation planning despite large potentials. Low demand and a lack of scalability often limit the growth of (urban) ferry systems. Against this background, research by Tarkowski & Puzdrakiewicz (2021) suggest a practical approach for triggering demand through using small ferries with busy and flexible schedules instead of larger vessels with strict departure times. This approach is adapted in this paper. Therefore, it should be mentioned that the term "ferry" used throughout this work refers to small, flexible passenger ferries. Small ferries can be used for river crossings in urban areas, but they can also be particularly beneficial in coastal areas with shallow waters. The operating times of large ferries often depend on the tides, since usual major ferries can only operate at high tide. Also, the higher visibility of operating ferries resulting from higher frequencies might lead to an increase in public acceptance, as potential walkability improvements constantly brought to the attention of the target public moving through the area (Tarkowski & Puzdrakiewicz, 2021).

The following sections intend to provide an overview of ongoing projects addressing autonomous ferry systems for supplementing urban transport. In line with the approach described and the ferry service introduced in Chapter 3, projects evaluated in Section 2.2.2 are rather dealing with small passenger ferries instead of larger vessels.

2.2.1 The Emergence of Autonomous Shipping

The introduction of autonomy in the shipping industry is likely to lead to profound change, as the operation and management of vessels and ferries will change drastically,

resulting in new business models emerging. Currently, over 80% of the global trade in goods (measured by volume) is transported via maritime routes (Nguyen et al., 2022). The reduction in personnel costs associated with autonomous shipping is just one possible benefit. More importantly, autonomous shipping has the potential to eliminate the need for the physical presence of humans, which is particularly important when considering long sea routes. One of the first remote and autonomous ferries called Falco and was tested in 2018 in the South of Finland (Munim et al., 2022). The project named SVAN (Safer Vessel with Autonomous Navigation) started in 2015 as a research partnership between Rolls Royce and the state-owned ferry operator Finnferries and resulted in an autonomously navigated ferry, including maneuvers to avoid collisions and docking without human intervention (Felski & Zwolak, 2020; Kidd & McCarthy, 2019). The ferry's functionality is based on two core elements. The (1) intelligent awareness system, also referred to as *eyes of the vessel*, is installed at the bow of the ferry. It enables the collection of data from a variety of long- and short-range sensors, which are then combined to create situational awareness of the ship's environment. Results exceed the capabilities of the human eye. (2) The autonomous navigation system is fitted in a plug-and-play container which mitigates required modifications on the vessel, as the container includes a major of technology. All operations are observed and may even be controlled from a remote-control station built at the Finferries' headquarter, which is located 45 km away from the starting location of the ferry route (Bratić et al., 2019). Kongsberg Maritime follows the development of the SVAN project by introducing the world's first fully autonomous container ship called Yara Birkeland in 2018. The autonomous vessel was designed to transport cargo (mainly fertilizer) on the Glomma River in Norway between the Yara company's plants. Based on its fully battery powered solution, the project aims at delivering a zero-emission solution for container shipping while targeting to save up to 40,000 truck trips per year. The ship is 80 meters in length and has a top speed of 15 knots (Yara International, 2018).

2.2.2 Small Autonomous Passengers Ferries: Status Quo

According to Wang et al. (2020) ongoing innovations for autonomous vessels are mostly developed for open waters, as the application of autonomous boats in urban environments faces even more challenges considering narrow and crowded waterways (Wang et al., 2020). Yet, there are several projects dealing with autonomous ferries for

urban environments, that indicate a progressing development in the near future. Also, it should be noted that ferries only operate in a limited area, allowing the autonomously completed tasks to be better constrained than for large vessels in diverse environments (Brekke et al., 2022). The idea of autonomous passenger ferries in urban areas is based on the vision of making inner-city mobility more efficient, safe, and sustainable. Autonomous ferries could help create a seamless connection between different parts of the city via waterways (Goerlandt & Pulsifer, 2022). The resulting reduced distances could relieve street-based traffic by increasing citizens' willingness to bike or walk through shorter routes. Tarkowski und Puzdrakiewicz (2021) also revealed first developments towards stronger usage of electric vehicles on urban waterways in a dozen major cities worldwide, and also growing ambitions to enable those vehicles to operate autonomously. This section intends to present ongoing projects dealing with autonomous ferry passengers in urban environment, which served as inspirational reference for the ferry service presented in Chapter 3.

MilliAmpère and Zeabus

The first autonomous ferry was developed at the Norwegian University of Science and Technology (NTNU). In order to traverse a 60-meter-wide canal in Trondheim, the construction of a new bridge was discussed, which would have impaired shipping traffic (Kirschgens, 2022). In 2016, a research team from Trondheim-based NTNU tried to construct the vision of an on-demand mobile bridge and the idea of an autonomous ferry emerged. After developing algorithms for motion planning, motion control, collision avoidance, docking, multi-target tracking, and localization, the autonomous passenger ferry MilliAmpère 2 was successfully tested in 2021 (Brekke et al., 2022). MilliAmpère 2 serves as the technical foundation for the startup Zeabus, which emerged from the NTNU research project. In Stockholm, July 2023, the company launched the first autonomous and fully electric commercial ferry with a capacity of up to 24 passengers, about 7 years after the idea of an autonomous ferry emerged. The self-driving ferry is powered by four electric motors and its battery is said to be sufficient for 15 hours of operation. Radar, lidar, GPS, infrared sensors and video cameras provide data which is constantly updated ten times a second. As a result, situations such as crossing boats are detected in less than a second, ensuring appropriate course corrections can be applied. From 2024, the ferry

is expected to be able to operate unmanned, allowing the complete monitoring of the autonomous ferry to take place onshore (Zeabus, 2023a).



Figure 1. Illustrations of autonomous passenger ferries by Zeabus

Roboat

Likewise, the research project "Roboat" had the vision of harnessing the potential of waterways through autonomous technologie. By developing self-driving watercrafts for the city of Amsterdam, the goal of the project was determined by exploring how self-driving technology could transform both urban areas and their water networks. The project was conducted from 2015 to 2022 as a collaboration between the Massachusetts Institute of Technology (MIT) and Advanced Metropolitan Solutions (AMS), an Amsterdam-based research initiative focused on developing sustainable and technology-driven solutions to urban challenges. The most recent outcome of this project is Roboat III, a fully electric autonomous watercraft with capacity for 6 passengers. Using lidar and camera sensors, the Roboat III operates autonomously by collecting data about its surroundings for up to 10 hours. Roboats are not only intended to serve as a solution for water transportation, but also for other purposes such as package delivery, water quality monitoring, and waste collection (van Toor, 2020; Wang et al., 2020). In addition, Roboat vessels can coordinate as a swarm by locking together. New latching mechanisms also allow robots to interact and coordinate as a swarm to form bridges and other formations (Benson, 2021). In this way, the boats become part of a programmable infrastructure as shown in Figure 2. In 2023, Roboat spins off as a start-up company to continue the development of the project (Roboat, 2023). The autonomous electric, and 3D-printed ferry is expected to be in operation in Q2 2024.



Figure 2. 3D-printed ferry and On-Demand infrastructure.

CAPTN

In 2018, Christian Albrecht University of Kiel (CAU) initiated the CAPTN initiative, an abbreviation for Clean Autonomous Public Transport, which pursues the goal of creating flexible, emission-free, and autonomous mobility in Kiel. Initially, the focus of the project was to develop an autonomous and clean passenger ferry that would connect the eastern and western shores of the Kiel Fjord, i.e. the university campuses. Today, an alliance of various partners from science and industry is working on projects that are essential for the deployment of autonomous ferries. These include the expansion of 5G mobile networks, a digital test field for autonomous shipping, distribution and transport of renewable energies, and a smart city project exploring AI methods and architectures for data collection. In 2023, the experimental carrier "MS Wavelab" was completed, a 21-meter-long and eight-meter-wide catamaran on which autonomous shipping is being researched in order to be able to deploy autonomous passenger ferries in the Kiel Fjord in the long term. From now on, autonomous shipping in the Kiel Fjord will be researched on the 21-meter-long and eight-meter-wide catamaran. The foundation has thus been laid for the future use of autonomous ferries in the Kiel Fjord (CAPTN, 2023). For the future application area of the CAPTN initiative, Aslaksen et al. (2012) explored the potential of combining fixed schedules with on-demand ferry services. In order to meet a large portion of the demand at busy ports, fixed schedules are determined by maximizing user utility based on estimated demand. A part of the ferry fleet is used to serve the fixed schedules, while the remaining ferries serve the on-demand component. The optimal allocation of ferries is a fundamental subject of the paper, which is measured by the percentage of requests fulfilled and the excess service time per passenger as two key

indicators. The on-demand component is based on the dial-a-ride problem and aims to additionally increase the number of served requests (Aslaksen et al., 2021)

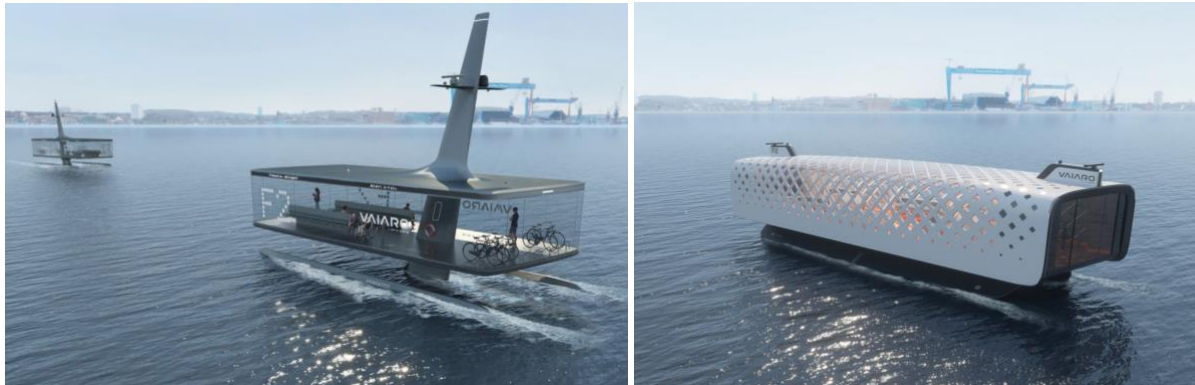


Figure 3. Illustrations of autonomous passenger ferries

2.3 Service Design and Optimization Modeling

The projects on autonomous ferries outlined in the previous chapter raise expectations for efficient mobility through the inclusion of waterways in the future. In this work, the operational level dealing with efficient routing of ferry services are in focus. To obtain the best possible routing for a given demand, an optimization model is to be integrated into the planning of the service. To better contextualize this approach, the concept of optimization modeling is further outlined before examining the roots of the optimization problem arising in the ferry service presented in this work. Therefore, this section intends to lay the foundation for the service presented in Chapter 3, which is considered a shared-ride service as the number of passengers exceeds the number of ferries available and therefore, multiple transportation requests are to be fulfilled simultaneously.

Routing

Especially shared-ride services require detailed organization and reliable management on various levels. The process of finding the best and most efficient routes, is denoted as routing problem. Routing problems are solved to optimize the routing design so that customer demand for goods (here: transportation service) is fulfilled without breaking any constraints (Caceres-Cruz et al., 2014). According to Eldrandaly & Ahmed (2008) routing problems can be divided into two different categories. One category involves path finding problems, where the objective is to find the shortest path between destinations.

The second and more complex category involves the aim of constructing one or multiple tours in given networks (Eldrandaly & Ahmed, 2008). The routing problem treated in this work belongs to the second category.

Optimization

Optimization models are used in a wide range of industries: from the allocation of beds in hospitals to the pricing of aircraft seats or the routing of vehicle fleets to minimize transportation costs like money and time. The optimization model treated in this work, corresponds to a variant of the Vehicle Routing Problem. Whenever multiple processes compete for resources, optimization aims to prevent an imbalanced allocation of those resources. Edmonds (2008) identifies three hierarchical ingredients that define such problem formulations: (1) *instances* are defined as inputs requiring an optimized output. Usually, every instance has a large set of (2) *solutions* (here: possible routings) but only those that satisfy all constraints are valid. The quality of solutions varies as every solution is related to certain cost and value. This value is defined as the (3) *measure of success*, which represents the third ingredient. Measuring success is either related to minimizing or maximizing an objective function (Werners, 2013). There are two techniques for solving optimization problems: exact approaches and approximate approaches (heuristics). Since many optimization problems are classified NP-hard, complexity and computational cost increase exponentially with the size of the problem (Pearl, 1984). Therefore, heuristics are used to find the best possible solution through a systematic approach. For illustration, Pearl (1984) uses the analogy of testing whether a cantaloupe fruit is ripe: if pressing the spot that was attached to the plant releases a certain smell, it can be strongly assumed that the fruit is ripe. This rule of thumb is an effective method but can still not guarantee that only ripe fruit will be detected. In conclusion, exact approaches guarantee to find the optimal solution of a problem whereas heuristics may find solutions *close* to optimality while optimality cannot be proven (Werners, 2013). Time complexity is the biggest trade-off: heuristics have a shorter computational time because they eliminate several steps in the computational process based on assumptions. Exact approaches, on the other hand, calculate every feasible solution to find the best and optimal solution. Due to longer computational times, exact approaches are more suitable for smaller problems (Eldrandaly & Ahmed, 2008). The ferry application presented in this paper aims to find exact solutions to the underlying optimization problem, which is since

the size of the instances is relatively small at this stage. However, with an increasing number of service users, heuristic methods should be additionally applied.

2.3.1 Introducing the Dial-A-Ride-Problem

The optimization problem integrated into the ferry service application presented, is based on the Dial-A-Ride-Problem (DARP) presented by Cordeau and Laporte (2007). The DARP originates from the family of Vehicle Routing Problems (VRP) but differs in considering the human factor. The main objective of the VRP is to keep travel costs as minimal as possible while delivering goods to a set of customer locations. As the DARP deals with the transportation of passengers the objective balances user inconvenience (e.g. waiting times, maximum ride-times) against minimizing operating costs.

Origins: The VRP

The Vehicle Routing Problem (VRP) deals with the efficient allocation of vehicles to deliver goods to a set of customer locations. It is rooted in the Travelling Salesman Problem (TSP), which involves determining the shortest route for a given set of locations, where each location must be visited exactly once. The tour must end at the same node where it started and finally forms a so-called Hamilton cycle, which describes a cycle that traverses each node of a given graph exactly once (Jünger et al., 1995). The VRP is considered a generalization of the TSP, since the entire demand of a set exceeds the capacities of a single delivery vehicle. Therefore, multiple sub-routes must be assigned to a fleet of vehicles. Dantzig and Ramser (1959) were the first to suggest a mathematical model (labelled as “truck dispatching problem”) for allocating delivery trucks to gas stations in such a way that the demand at each station is met while minimizing the total fleet distance. Since all routes have to start and end at the same depot while covering different clusters, the problem may also be referred to as “Clover Leaf Problem” (Dantzig & Ramser, 1959). The objective of the VRP is either to “(1) find optimal routes to be operated by available vehicles so as to supply the customer requirements at minimum total variable cost or (2) find the smallest possible number of vehicles and their routes which can supply all customer requirements” (Christofides, 1976, p. 3). Due to the wide range of application scenarios, classifications and namings of VRPs may vary across the literature while the basic structure of the VRP remains the same. The VRP with Time

Windows (VRPTW) deals with customers who have a specific time window in which delivery must occur (Beck et al., 2003). It is distinguished between soft time windows and hard time windows. Hard time windows do not allow any violation, customers must be visited in their desired time window. Soft time windows, on the other hand, may be violated at the expense of customer inconvenience (Cordeau et al., 2001). Not every VRP requires goods to be picked up at depots only, as a VRP with pickup and delivery (VRPPD) allows goods to be picked up and delivered at various locations along the route. VRPPD with unpaired locations consider a homogenous good, wherein each picked-up good can fulfill the demand of any customer. VRPPD with paired locations consist of pickup locations associated with specific delivery locations (Parragh et al., 2008).

From VRP to DARP

According to Parragh et al. (2008), the DARP belongs to the subclass of VRPPD with paired locations, as transportations requests consist of an individual start with an associated destination. In some literature the DARP is labeled as a generalization of the Pick-up and Delivery Problem with Time Windows (PDPTW) (Detti et al., 2017). A lot of research considering the DARP has been influenced by the transportation of elderly or disabled people, but the model can also be useful for the transportation of perishable goods, since maximum travel times apply to them as well (Parragh et al., 2008). Dial-A-Ride-Problems are either of static or dynamic nature. A static operation mode allows new customer requests to be considered immediately, even if the vehicles' operation already started. In turn, static problems require all passengers' requests to be known before the solving algorithm is implemented and the corresponding vehicles are put into operation. A purely dynamic operating mode is rather rare, since requests are often made in advance. Cordeau and Laporte (2007) identify two similar, but slightly differing objectives of the DARP: in the first case, every request must be accepted, which results in an objective minimizing the number of vehicles required to fulfill the entire demand. In the second case, vehicle availability is limited and therefore, the objective is to maximize demand satisfaction for a given fleet capacity.

Dial-A-Ride and Ride-Sharing

To allocate vehicles and capacities as efficient as possible, dial-a-ride-services are intertwined with ride-sharing. As the name implies, ride-sharing systems aim at bundling different passengers' requests with similar time schedules and itineraries by combining routes. Each additional rider involves detours, i.e., additional costs that are only worthwhile if the benefits (like shared fuel costs or cumulative time savings) outweigh the costs. Ride-Sharing can significantly reduce the number of vehicles required to satisfy the mobility needs of passenger but individual ride times, waiting times and detours must be constrained in order to minimize the inconvenience for passengers (Stiglic et al., 2015). Referring to street-based ride-sharing systems, Agatz et al. (2012) point out the key difference between DARP and Ride-Sharing to be the number of depots: while DARP systems usually operate out of a fixed set of depots, drivers in ride-sharing systems can start or end their service at unique locations (Agatz et al., 2012). The authors further specify three different objectives of ride-sharing systems, regardless of whether they are private (profit-oriented) or public (welfare-oriented²). The objective is either to (1) minimize system-wide vehicle miles, (2) minimize the system-wide travel time, or (3) maximize the number of participants. In every case, time efficiency plays a major role. The less time customers must wait for their service, the more are willing to use the service. The more customers use the service, the higher the success-rate of matching rides and the greater the social welfare (Agatz et al., 2012). According to Stiglic et al. (2015) a high matching rate is to be considered a critical success factor but can be challenging if demand is not high enough to attract supply. At the same time, the amount of rejected requests is to be mitigated as successfully matching increases positive user experience and often result in a continuous use of the service (Stiglic et al., 2015).

Across the literature, different concepts are applied. For example, (Mourad et al., 2019) refer to the shared-taxi problem as a variant of the Ride-Sharing Problem. Here, passengers can specify their desired pick-up and drop-off locations and a maximum travel time. Solving the underlying routing problem ensures customers a fast transfer from A to B, while using scarce resources (such as energy, distance, and time) as efficient as possible. Yao et al., (2021) developed a ride-sharing model which included the incorporation of passengers' efforts. In order to reduce the amount of detour costs,

² Societal objectives may be to provide affordable transportation or to reduce congestion and pollution (Agatz et al., 2012)

passengers are encouraged to walk up to 300m to an assigned station. They found that, especially in densely populated areas and during peak hours, assigning stations to passengers instead of picking them up at individual locations can dramatically increase the matching rate (Yao et al., 2021). Stiglic et al. (2018) developed a model which combines public transit and ride-sharing. Here, ride-sharing can either be (1) used conventionally from a requested pick-up to destination or (2) ride-sharing is only used for certain parts of a requested route and includes a drop-off/pick-up at a station, so that passengers can easily switch to public transport. The authors were able to demonstrate benefits for both parties - particularly when associated to the last mile and suburban areas (Stiglic et al., 2018).

3 Urban-Autonomous-Ferry-Service

After establishing the theoretical foundations, this section will proceed to introduce the developed autonomous ferry service and assess its performance from an operations research perspective. A use case is deployed in a city having its urban center separated by a river. There are bridges for crossing the river, but they are located several kilometers apart and thus, crossing the river leads to detours. Ferry stations are set up along the riverbanks, which are served by an autonomous fleet of ferries. The ferry service introduced is designed for users whose starting point is on the opposite shore of their destination and provides alternative routes that minimize detours by using ferry crossings. From now on, the service will be referred to as the "UAF-Service" (Urban-Autonomous-Ferry-Service). The efficiency of the UAF-Service is gauged by the time savings and distance savings it provides compared to the conventional route. The term *conventional route* between a passenger's start and destination describes the regular route leading over existing infrastructure like bridges and can be retrieved via route planning software like Google Maps. Analogously, *conventional arrival time* defines the time a passenger would arrive at his or her destination when traveling via the conventional route. The alternative route provided by the UAF-Service is referred to as *ferry route* and describes the route involving the use of a ferry to traverse the river. Figure 4 illustrates two routes from A to B in comparison for the purpose of highlighting achievable distance savings.

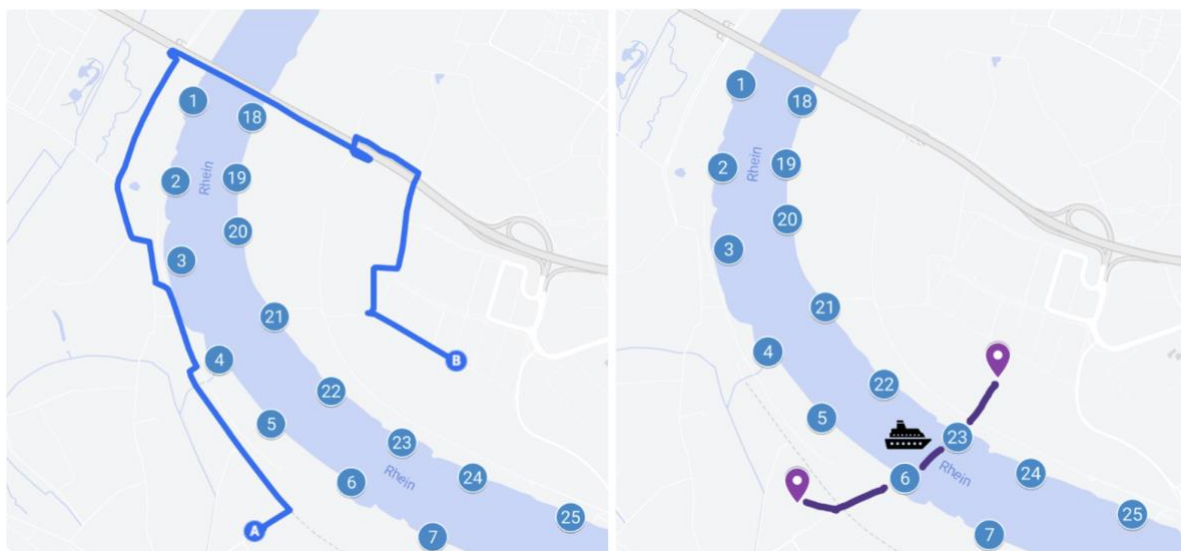


Figure 4. Conventional route and Ferry route in comparison

The ferry stations are denoted by blue numbered circles. The service is intended for cyclists, as bicycles are considered a realistic substitute for cars. All subsequent calculations and analyses are conducted and compared with this consideration in mind. The average speed on the conventional route is therefore equivalent to that on the land segments of the ferry route.

3.1 Problem Description

The UAF-Service aims to provide ferry routes to passengers, which consist of three sections as illustrated in Figure 5.

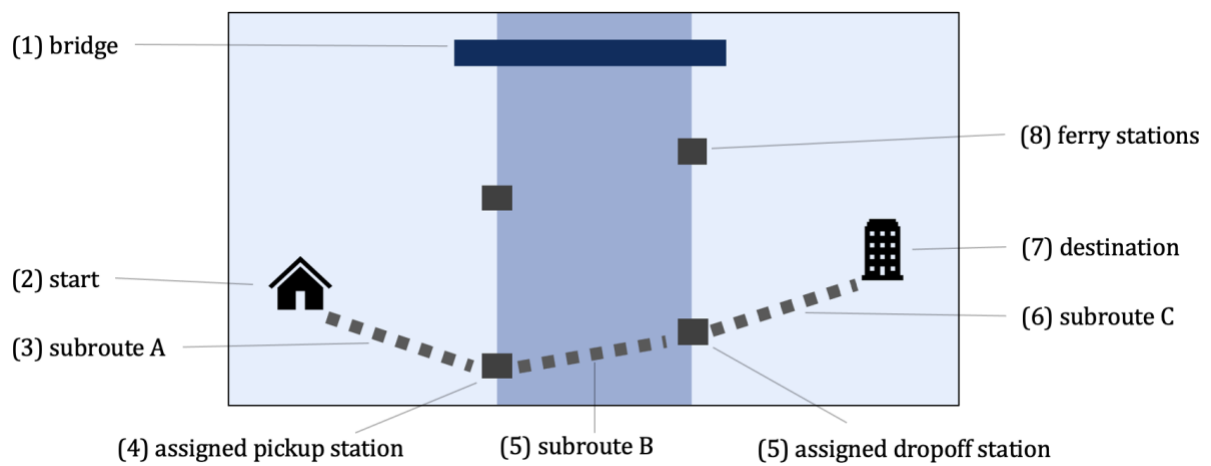


Figure 5. Illustration of the ferry route consisting of three sub routes.

Section A includes the route from the start to the pick-up station, section B includes the route of a ferry from the pick-up station to the drop-off station, and section C includes the route from the destination station to the final destination. Although distances of the conventional route might be longer, the corresponding ferry route can still consume more time, as rides are usually shared with other passengers and waiting times are incurred. How long it takes a passenger to travel the ferry route depends on the routing of the ferry fleet. Therefore, the routing of the ferry fleet is the major challenge for the UAF-Service, as simultaneous requests for a limited number of available ferries lead to various constraints in order to maintain a high level of service. Figure 6 illustrates the service process.

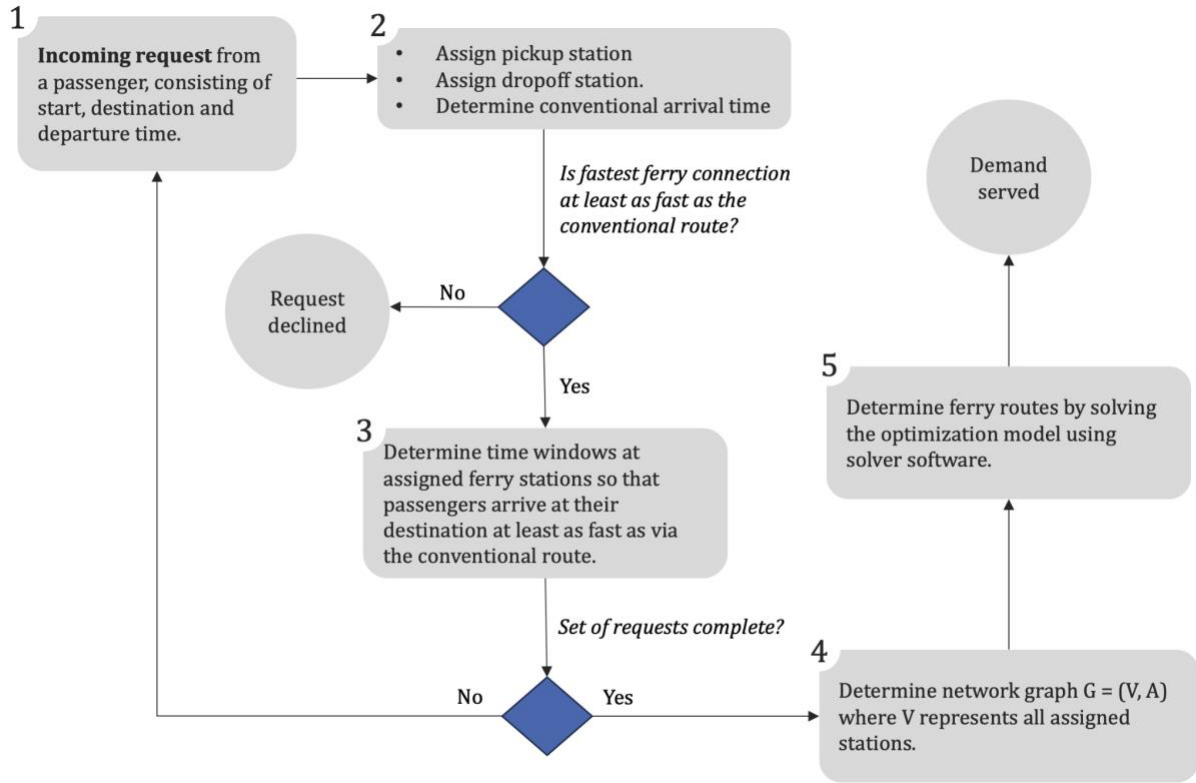


Figure 6. Process diagram of the UAF-Service

A request consists of the passenger's initial start and destination (e.g., home and office) and the desired departure time from the initial start. The closest ferry stations are assigned in step two, while retrieving the conventional route via route planning software. Given the passenger's departure time, obtaining the conventional route also implies obtaining the conventional arrival time. If even the fastest possible travel time for the ferry route exceeds that for the conventional route, those requests will be denied before service is commenced. This situation can for instance arise when the starting or ending point is in close proximity to the bridge, minimizing the detour caused by the river crossing. In step three, the time windows at the assigned ferry stations are determined in such a way that passengers arrive at their destination no later than they would by choosing their conventional route. The assignment of pickup and drop-off stations to passenger locations, resulting in sub-routes A and C (including time windows) are treated as *input data* for the optimization model as it constructs graph G, the underlying network for the optimization model. Once all requests have been received, step four composes the final graph $G = (V, A)$, which does not include all available stations of the ferry system, but

only those that are assigned to the requests of an instance. In step five, optimization is performed to provide a ferry route for every passenger in the request set. Figure 7 serves for illustrating the resulting graph with nodes in V denoting all the assigned stations within the demand set, and edges in A linking these nodes. From now on, the focus of this chapter is on sub-route B as subject of the optimization problem discussed in this work. The initial destination and departure points of the passengers are not relevant and the input for solving the model is limited to the assigned stations per customer request, as well as the corresponding time windows. The station assignment process is further described in Section 4.2.2, when outlining the service implementation. In most similar applications, the underlying graph G consists of nodes representing unique locations. However, in this specific scenario the same set of assignable stations is available to all passengers, and each station can function both as a pickup and drop-off point for multiple requests. To formulate this problem, each station assignment corresponds to an individual node in the graph, particularly since time windows can vary. As a result, distinct nodes in $V \in G$ can represent the same physical location, resulting in a distance equal to zero between those stations.

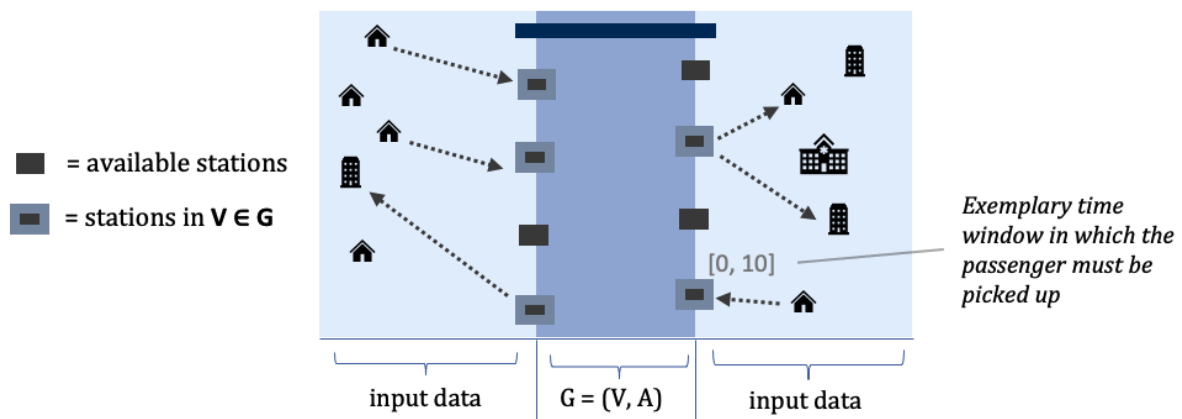


Figure 7. Illustration of generated graph G

The UAF-Service is a static on-demand service, as route calculation is performed only after a set of customer requests has been received. The on-demand aspect is not necessarily associated with dynamic services but serves as a means to flexibly adjust routes and schedules in response to changing demand.

3.1.1 Customer Satisfaction

Given the finite capacity of the ferry fleet, the service's primary challenge lies in efficiently serving multiple concurrent customer requests while ensuring a convenient experience. Since the passengers' pickup and destination points might differ, shared rides are generally more time-consuming than individual rides. Passengers pursue the objective of arriving at their destination as fast as possible. From the operator's perspective ride-sharing and pooling demand is inevitable in order to maximize the number of passengers using the service. Avoiding idle capacity is essential for publicly financed services since only those used by a great portion of citizens are worth maintaining in the long term. The primary goal of the UAF-Service is to maximize the number of users, which is achieved by ensuring the highest possible level of customer convenience. Convenience, in this context, is characterized by ensuring that passengers are transported to the opposite riverbank within their specified time windows. If a time-window is met, a passenger arrives at his or her destination at least at the same time or earlier compared to the conventional route. This implicitly prevents long waiting and travel times, but tight time windows result in an increase in mathematical complexity. Exceeding time-windows should be avoided, but at the same time all customers should be served. In many applications of the Dial-a-ride-problem, the measurement of customer satisfaction involves the total travel time, including waiting times (Pfeiffer & Schulz, 2022). The greater the difference between the conventional arrival time and the arrival time via the ferry route, the higher the customer satisfaction and vice versa.

3.2 Mathematical Formulation

To determine the optimal ferry routes under a given demand, the three-index DARP formulation of Cordeau and Laporte (2007) serves as a basis. The model applied in this paper is referred to as UAF-DARP (Urban-Autonomous-Ferry-Dial-A-Ride-Problem) as it deviates from the standard DARP formulation by permitting time-window violations, which are penalized in the objective function. This approach was selected due to the ferry service handling numerous requests that may overlap in time. Consequently, with hard time-windows in place, the underlying optimization model rapidly becomes infeasible

when faced with capacity constraints. A high penalty factor strongly penalizes any constraint violations, potentially causing the model to prioritize meeting constraints closely, even at the expense of a less favorable overall objective (Salani & Battarra, 2018). To accommodate all passengers, some may need to tolerate delays, acknowledging that their service time exceeds the upper limit of their time window.

The UAF-DARP is based on a directed graph $G = (V, A)$ where the vertex set $V = P \cup D \cup \{0, 2n + 1\}$ represents a set of nodes including two copies of a pseudo-depot as well as pickup nodes $P = \{1, \dots, n\}$ and drop-off nodes $D = \{n + 1, \dots, 2n + 1\}$. Each request consists of a pair of nodes $(i, n + 1)$ where $i \in P$ represents a pickup station and $i + n \in D$ represents the corresponding drop-off station. Non-identical nodes in V can refer to the same physical location (i.e. station) but have individual indices. In this way, two different requests can involve identical stations, but they are treated as individual nodes. Distance and travel time between such nodes are equal to zero. Travel times between stations in $V = P \cup D \cup \{0, 2n + 1\}$ are calculated from a distance matrix by dividing the respective distances between two nodes by the average speed of a ferry. A denotes a set of arcs $a = (i, j)$ connecting the set of vertices V . In this sense, the graph does not cover all available station but only those stations assigned to the set of requests. It should be stressed that $(i, j) \neq (j, i)$ since all requests represent ordered pairs with a well-defined pickup location and an associated drop-off location. The set of all arcs A is defined as

$$A = \{(i, j) | (i = 0 \wedge j \in P) \vee (i, j \in P \cup D \wedge i \neq j \wedge i \neq j + n) \vee (i \in D \wedge j = 2n + 1)\}$$

which implies that a directed arc either leads from the first depot $= 0$ to a pickup node $j \in P$ or connects two points of the set $P \cup D$ or leads from a drop-off point $j \in D$ back to the depot. The binary variable $x_{i,j}^k$ is equal to 1 if arc (i, j) is traversed by vehicle k , and 0 otherwise. The complete notation of variables and parameters used in the UAF-DARP model can be found in Table 1.

$G = (V, A)$	Directed graph on which the problem is defined
$V = P \cup D \cup \{0, 2n + 1\}$	Set of vertices in the graph V
$A \subset V \times V$	Set of directed arcs
Sets and Indices	
$0, 2n + 1$	Pseudo depots, start and end of each tour
$i, j \in V$	Nodes in $V = \{0, 2n+1\} \cup P \cup D$
P	Pickup nodes, $P = \{1, \dots, n\}$
D	Dropoff nodes, $D = \{n + 1, \dots, 2n\}$
n	Number of passenger requests
$(i, n + i)$	Passenger request with Pickup $i \in P$ and Dropoff $n + i \in D$
$k \in K$	Set of ferries
$t_{i,j}$	Travel time from i to j according to distance matrix D
$c_{i,j}$	Distance from i to j resulting from distance matrix D
Parameters	
d_i	Service duration at node i
q_i	Load of request $(i, n + i)$
$[ei, li]$	Time windows of request $(i, n + i)$
Q_k	Capacity of the ferry k
T_k	Maximal duration of route $k \in K$
L	Maximal ride time of passengers
PF	Penalty factor to multiply with time window violations
Variables	
$x_{i,j}^k$	= 1 if j is direct successor of i , 0 otherwise
u_i^k	Time at which vehicle k starts servicing node i
wk_i	Load of ferry k upon leaving node i
r_i	Ride time of request $(i, n + 1)$
$pvar_i^k$	Time window violation at pickup node of request $(i, i + n)$
$dvar_{i+n}^k$	Time window violation at dropoff node of request $(i, i + n)$

Table 1. UAF-DARF Model Notation

The UAF-DARP is formulated as follows.

$$(1a) \min \sum_{i \in P} \sum_{k \in K} u_{i+n}^k - u_{i,k}^k + \sum_{i \in D} \sum_{k \in K} dvar_i^k \cdot PF$$

$$(1b) \min \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} c_{i,j}^k \cdot x_{i,j}^k + \sum_{i \in D} \sum_{k \in K} dvar_i^k \cdot PF$$

Subject to:

$$\begin{aligned}
(2) \quad & \sum_{k \in K} \sum_{j \in V} x_{i,j}^k = 1 & (i \in P) \\
(3) \quad & \sum_{i \in V} x_{0,i}^k = \sum_{i \in V} x_{i,2n+1}^k = 1 & (k \in K) \\
(4) \quad & \sum_{j \in V} x_{i,j}^k - \sum_{j \in V} x_{n+i,j}^k = 0 & (i \in P, k \in K) \\
(5) \quad & \sum_{j \in V} x_{j,i}^k - \sum_{j \in V} x_{i,j}^k = 0 & (i \in P \cup D, k \in K) \\
(6) \quad & u_j^k \geq (u_i^k + d_i + t_{ij})x_{i,j}^k & (i, j \in V, k \in K) \\
(7) \quad & w_j^k \geq (w_i^k + q_j)x_{i,j}^k & (i, j \in V, k \in K) \\
(8) \quad & r_i^k \geq u_{n+i}^k - (u_i^k + d_i) & (i \in P, k \in K) \\
(9) \quad & u_{2n+1}^k - u_0^k \leq T_k & (k \in K) \\
\hline
(10) \quad & u_i^k \geq e_i & (i \in V, k \in K) \\
(11) \quad & u_i^k - pvar_i^k \leq l_i & (i \in P, k \in K) \\
(12) \quad & u_{i+n}^k - dvar_{i+n}^k \leq l_{i+n} & (i \in P, k \in K) \\
(13) \quad & pvar_i^k \leq dvar_{i+n}^k & (i \in P, k \in K) \\
(14) \quad & u_{i+n}^k \geq u_i^k & (i \in P, k \in K) \\
\hline
(15) \quad & t_{i,n+i} \leq r_i^k \leq L & (i \in P, k \in K) \\
(16) \quad & \max\{0, q_i\} \leq w_i^k \leq \min\{Q_k, Q_k + q_i\} & (i \in V, k \in K) \\
(17) \quad & x_{i,j}^k = 0 \text{ or } 1 & (i, j \in V, k \in K)
\end{aligned}$$

3.2.1 Objective Function

When designing a mobility service, it is essential to find a balance between the objectives of both passengers and operators. Passengers pursue the objective of reaching their destination as soon as possible. Operators may implicitly share the same objective for maximizing the user rates, but their primary focus lies in minimizing operational costs. Therefore, there are two different objective functions introduced, one aligned with the preferences of passengers (1a) and the other tailored to the priorities of operators (1b). In the pursuit of optimal outcomes, computational procedures employ only one objective function. Both objective functions deviate from the conventional DARP formulation by integrating the minimization of penalty costs. These costs stem from the multiplication of delays at drop-off nodes ($dvar$) by a specified penalty factor PF . Due to the minimization objective, the model strives to keep time windows violations equal or close to zero. The objective function (1a) aims to reduce the time passengers spend on the ferry service, which is determined by the difference between the start of service u_i at a passenger's pickup node and the start of service at the corresponding drop-off node u_{i+n} . The variable u_i is composed of the service time d_i used for docking and loading and the travel time between u_i and u_{i+n} . The total travel time is at least as long as the direct travel time $t_{i,i+n}$. If additional passengers are served along the route, the total travel time is composed of the respective sub-routes. Since objective function (1a) minimizes travel times, the best routing is likely to result in routes without ridesharing, if capacity permits. Objective function (1b) minimizes travel costs determined by distance. The variable $c_{i,j}$ describes the costs occurring if an arc (i, j) is travelled via vehicle k .

3.2.2 Constraints

The objective function is subject to flow constraints (2) - (5), time and capacity constraints (6) - (16) and the binary constraint (17). Flow constraints ensure a continuous flow into and out of nodes: Constraint (2) ensures that each pickup node in V is visited exactly once. Thus, every passenger requesting a ride is to be picked up. (3) Depots are defined as the first and last vertex of a route: Each ferry leaving the first depot must return to the second depot. The UAF-DARP model uses pseudo depots for this to start and end the tours. Travel costs from and to depots are close to 0. Therefore, ferries can start their service at any station in V . In analogy, ferries can be charged or found idle

at any node in V . Constraint (4) ensures that the pickup node of a request is served by the same ferry as the corresponding drop-off node of the request, meaning that passengers are not required to transfer. Moreover, (5) any ferry k entering a node $i \in P \cup D$ is constrained to continue its route to another point in V . This guarantees that no ferry is stuck on the route since each ferry $k \in K$ must end its route at the depot. The order of the indices of x_{ji}^k and x_{ij}^k further implies directed arcs. Constraints (6) to (15) ensure that time and capacity constraints are met: (6) The time u_j at which a service can start at node j corresponds to the sum of the start of the service time at node i , the service duration at the previous node i , and the travel time t_{ij} . In analogy, condition (7) determines the number of passengers leaving node j to be either equal to 0 or equal to the previous ferry load at node i and the change in load w_j at node j . Constraint (8) defines the travel time of passenger i on ferry $k \in K$. If the model would only consider direct trips, the condition would entail an equal sign. The (9) maximum travel time T_k assigned to the ferries (which may result from charging or refueling requirements) must not be exceeded. Constraints (10) to (14) impose time limitations on the model. In the standard DARP formulation (Cordeau & Laporte, 2007), u_i is primarily required to adhere to the time windows corresponding to request resulting in the constraint $e_i \leq u_i^k \leq l_i$. The UAF-DARP extends this approach by introducing the additional variables $pvar_i^k$ at pickup nodes and $dvar_i^k$ at drop-off nodes for allowing, but penalizing, time window violations. This enables to pick up or drop off passengers later than the latest start of service l_i if necessary, but never before the earliest start of service e_i associated with their request nodes. Since time window violations are penalized in the objective function, the variable is expected to be equal or close to 0. (13) The variable $pvar_i^k$ is implicitly minimized by constraining it to be less or equal than $dvar_{i+n}^k$. Constraint (14) mandates the servicing of passengers' pickup nodes before their corresponding drop-off nodes, leveraging the overlap in time-windows at both nodes. (15) Further, a passenger's ride-time must be equal or greater than the direct travel time, but less than the specified maximum ride time. Constraint (16) provides a two-sided constraint for loading a ferry. When leaving any point in V , there is at least no passenger on the ferry. The upper limitation corresponds to the maximum capacity Q_k .

4 Implementation of the UAF-Service

To apply the mathematical formulation presented in the previous chapter for the ferry service, an appropriate implementation environment is required and will be elucidated in this chapter. The UAF-Service application was developed to assess the potential benefits of introducing an autonomous ferry service from an operations research perspective. The custom application simulates the entire process from route request to ferry schedule creation to obtain quantifiable results without empirical data. Since there is no off-the-shelf software for calculating alternative routes based on the newly introduced ferry service, the configurable UAF-Service application forms an essential part of this thesis. The complete application code is provided in a repository hosted on the version control platform Github. Python docstrings are used throughout the application code to describe the functionality of a method, the expected input parameters, and the return value. Python docstrings appear immediately after the definition of a function, method, class, or module. Further, this chapter intends to (1) outline the process structure of the application, (2) provide other users with an overview of the deposited project for convenient usage or further development.

4.1 Application Environment

For the sake of reusability, "hard-coded" elements should not occur in the program. Therefore, the file *config.py* was integrated as a central element for the configuration. This file serves as a kind of control center in which all important parameters, settings, variables, and configurations are centralized. Running processes in the application can then access the configuration file to extract relevant values. This is intended to make the application adaptable to different scenarios, environments, and requirements without changing the source code of the application itself. The application consists of five steps, which are described in detail in the following sections. It is important to mention that each presented module can be executed independently using its respective module file (e.g. *_1_demand_generation*). However, to run the complete application process, which sequentially includes every module, including demand generation, route optimization, and result evaluation, it is necessary to run the *main.py* file located in the application folder.

4.1.1 Parameter Settings

Before running the application, parameters and file paths for data input and data output must be defined in *config.py*. Parameters to be set either correspond to the parameters included in the mathematical formulation presented in Section 4.2 or to parameters required for defining the run process e.g. how many request instances are to be generated and optimized per run. An instance consists of n passengers. A set consists of m homogeneous instances. Homogeneous implies that instances of a set differ in data, but not in size or parameter settings. In each iteration, i.e. for each instance, the five process steps described in Section 5.2.1 to Section 5.2.5 are executed. Table 2 entails the most relevant parameters to be configured in the config file of the application.

Parameter	Description
dt_DEPARTURE_TIME_SPAN	Upper limit of the time span, starting from 0, within which passenger departure times occur.
FERRY_VELOCITY	Ferry velocity in meter per second (12km/h)
BIKING_VELOCITY	Cycling speed in km/h
BENCHMARK_CLEAN_REQUESTS	Required time savings threshold to consider a request valid
<i>Model Parameters:</i>	
n_NUMBER_OF_PASSENGERS	Size of instance
d_SERVICE_TIMES	Berthing and embarking time
K_FLEET_SIZE	Fleet size
Q_CAPACITY	Number of seats per ferry
q_LOAD_PER_REQUEST	Number of passengers traveling per request
PENALTY_FACTOR	Factor to determine weight of delays
TK_MAX_RT_VEHICLE	Maximal duration of route per ferry
L_MAX_RT_PASSENGER	Maximal ride-time per passenger

Table 2. Parameters to be configured in the UAF-Service-Application

4.1.2 Technological Infrastructure and Visualization

The application was developed using Python 3.9.5 within the PyCharm IDE. For obtaining conventional routes along with travel times, the application utilized the Google Maps API. The input of demand polygons was managed through the use of Shapely. Data management was facilitated by the Panda framework and the SQLite3 database. Gurobipy, a Python-based API provided by Gurobi, served as the mathematical solver for route optimization. To visualize passenger requests and their corresponding station assignments, a frontend component was integrated into the application. This component can be locally hosted by executing the file *helper/visualize_requests.py* after running the application, irrespective of the optimization success. The visualization involves an HTML file incorporating a comprehensive JavaScript script, leveraging the Google Maps API to dynamically display ferry stations and requests including station assignments on a map. After successful optimization runs, the application generates tables and line plot graphics to visualize obtained results in the IDE console. This visualization is achieved using the Matplotlib library.

4.2 Application Process

The application comprises five phases, each of which can function as standalone modules. Phases 1, 2, and 3 are designed for generating test data for the optimization model. Consequently, these phases can be omitted from the process when the application is utilized with readily available real-world data. Figure 8 illustrates the process design, while the individual phases are briefly described in the following sections.

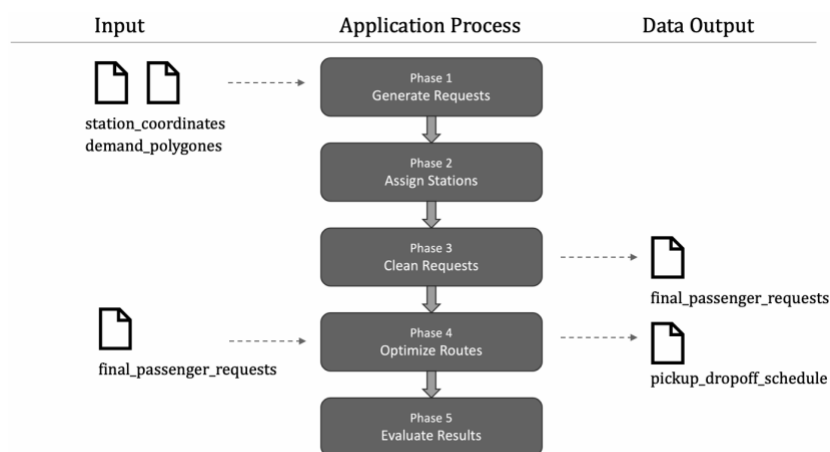


Figure 8. Application Modules

4.2.1 Demand Generation

To simulate a specific application scenario, the initial input shall consist of (1) the coordinates of the ferry stations available in the service and (2) a minimum of two demand polygons. These polygons specify the areas where the starting or ending points of requests can be found, and they are defined by the coordinates of their vertices. Figure 9 Illustrative excerpts of the use case area in Duesseldorf before applying the initial input on the left and after implementing coordinates of ferry stations and two demand polygons, marked in green.

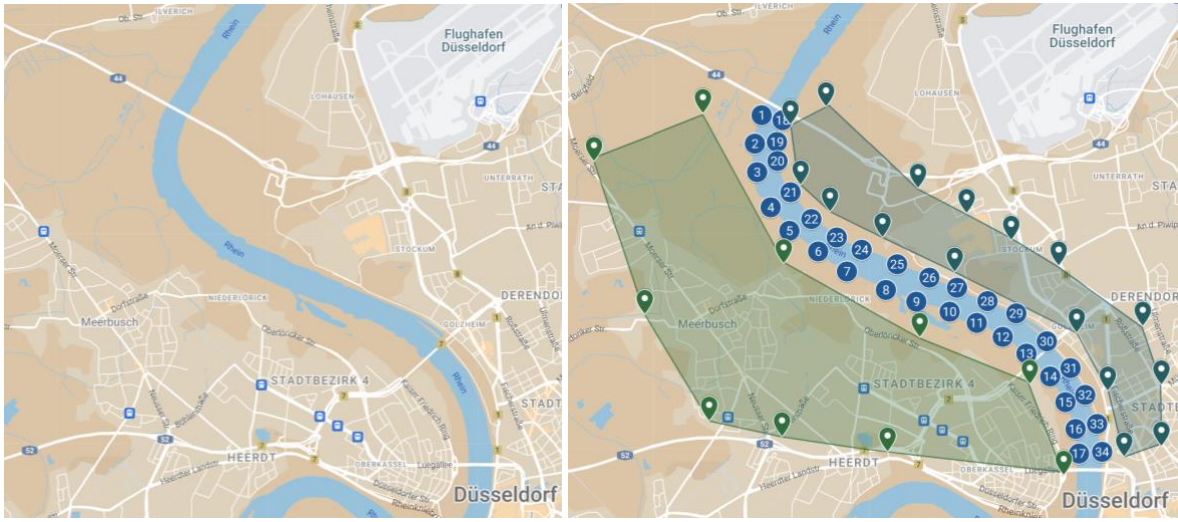


Figure 9. Specified Use Case Area

The parameter `n_NUMBER_OF_PASSENGERS` hosted in the `config.py` file of the application determines the quantity of passenger requests produced for each instance. In this process, n random coordinates are selected from each of the two polygons. Subsequently, a coordinate from the right-side polygon is matched with a coordinate from the left-side polygon. This yields a request in which the starting point is situated in a distinct polygon from the destination point, signifying a request that involves crossing a river. Moreover, each request is assigned a random value within the time frame commencing at 0 and constrained by the configured value `dt_DEPARTURE_TIME_SPAN`. Every passenger of an instance leaves his or her starting point within this specified range. The smaller the time span is set, the more simultaneous requests are likely to occur in an instance of passenger requests. Further, the Google Maps API is used to obtain the conventional route including the corresponding travel time for

each request, which is later used as a benchmark in determining the passenger's time-windows for the ferry service. To simulate targeted scenarios in the absence of empirical data more accurately, demand polygons could also be clustered, as it is also possible to entail more than two demand polygons.

4.2.2 Station Assignment

Finding the best ferry stations to assign to start and destination location has a significant impact on the efficiency of the UAF-Service. For every request, there are two stations to assign: a pickup station, and a drop-off station. The assignment is entirely based on distance. However, there are two possible approaches: (1) minimizing the distance on land, i.e. the distance between stations and passenger locations or (2) minimizing the distance on water. Figure 10 illustrates both approaches in comparison with (a) representing station assignment with minimized land distance.

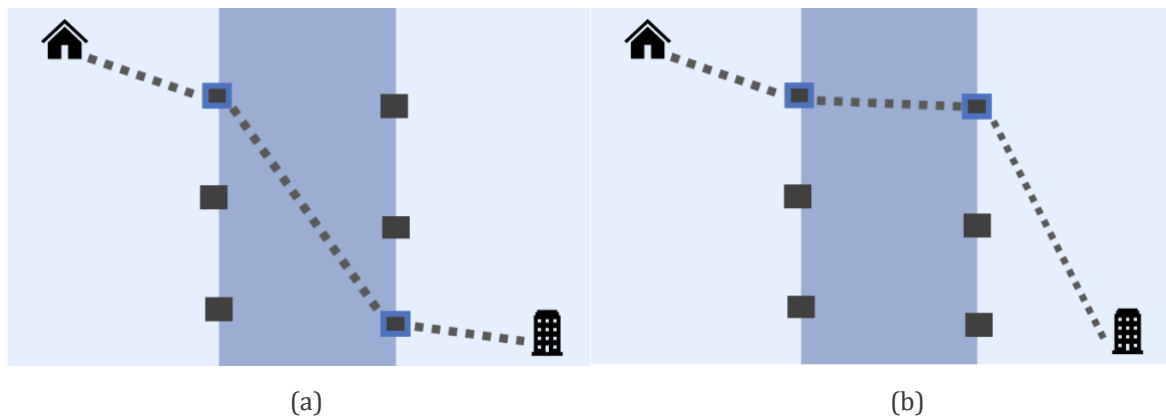


Figure 10. Comparing Station Assignments

For the specific application in this study, it has been observed that variant (b) consistently produces significantly improved results. Nevertheless, the application code provided also offers the flexibility to implement variant (a), which might be more appropriate for different geographical scenarios. The primary objective of this assignment process is to minimize the overall distance between the request's start and destination points, considering the route as a whole rather than individual segments. If the selection of the nearest pickup station is based solely on the shortest distance to the initial starting point, there is a possibility that this chosen pickup station may be farther away from the final destination than another station (such as the second closest station) that is closer to the

destination. Therefore, the station assignment is conducted in a manner that compromises the distances of the sub-routes: Let point A be the starting point of a request. To assign a pickup station to point A, it is initially determined in which polygon (on which side of the river) Point A is located. If point A is located on the right side, a distance matrix is generated, which connects point A to all stations located on the same right side. From this one-dimensional matrix, the indices of the three smallest values are declared as *choice of pickup stations*. The selection of the pickup station should strike a balance between being as close as possible to the starting point while also being situated in the direction of the destination. To accomplish this, a one-dimensional distance matrix is established between the passenger's destination point and the *choice of pickup stations*. The pickup station with the shortest distance among the three preselected options is designated as the final choice. The determination of the destination station follows a similar procedure.

Route distances are typically obtained using route planning software throughout this work. However, since many stations in this use case are located in areas without dedicated pedestrian or bike infrastructure, distances calculated by Google Maps may be higher than actual distances, as it considers only existing bike lanes. Consequently, subroutes A and C are based on Euclidean distances in the current stage.

4.2.3 Clean Requests

This step is used to exclude dispensable requests. All previously created requests (including the station assignments) are now checked for validity. A request is only considered valid if the minimum possible time for the ferry route (minimum ferry time) is shorter than the travel time of the conventional route. The minimum ferry time results from a scenario in which a passenger is picked up immediately after arrival at their pick-up station and transported directly, i.e. without ride-sharing, to their drop-off station. The benchmark for this procedure is configurable and determines the size of the time-windows. For example, if the benchmark parameter is set to 3, the minimum ferry route, including the loading time, must be at least 3 minutes shorter than the conventional route.

4.2.4 Optimize Routes

An initial distance matrix, encompassing all stations within the ferry service, is provided. For each request instance, a new distance matrix is generated based on this initial matrix, with its indices aligning with the order of the stations in $V = \{\text{depot}, P, D, \text{depot}\}$. To guarantee that the model treats each point within V as unique, even in cases where stations may overlap, the model employs $V_{_}$ as an encapsulation of V . $V_{_}$ replaces the actual stations with a numerical sequence ranging from 0 to n . Consider a scenario where $n = 2$ passengers are picked up from Station 3, with one of them requiring transportation to Station 7 and the other to Station 5. As a result, the sequence $V = [0, 3, 3, 7, 5, 0]$ is obtained. The indices in the newly created distance matrix correspond to this V sequence, with identical stations having a distance of 0 from each other. Subsequently, each index enables access to the corresponding position in the newly formed distance matrix, where the order aligns with the actual stations in V .

Determining of Time-Windows at Stations

A time window, represented as $[e, l]$, specifies the allowable time frame at during which a passenger must be served, with e indicating the earliest service start time and l as the latest service end time. Each pickup and drop-off station assigned to a request is associated with a specific time window, ensuring that the ferry route's arrival time is at least as early as that of the arrival time resulting from the conventional route. When using common route planning software, either the departure or arrival time is typically defined. This holds true in this scenario as well: the passenger specifies a departure time when they leave their original starting point. Based on the departure time, the conventional arrival time at the destination is determined. This also enables the determination of when the passenger will arrive at their pickup station at the earliest, which is referred to as the earliest start of service at the pickup station.

For this use-case, the time window of a request's pickup station and drop-off station are identical, although in many applications of the DARP, time windows at pickup stations differ from those at their associated drop-off stations. In the context of medical transport, there is often a critical need to pick up passengers within a designated time frame to ensure they receive timely assistance. Similarly, in the transportation of goods, variations in opening hours at different locations result in unique time constraints for both pickup

and drop-off points. In the context of this service however, passengers do not benefit from having smaller pickup time windows compared to their drop-off time windows, as it could lead to potentially spending more time on board of the ferries without any added advantages. Given this context, the two time windows are defined as follows: The earliest start of service e corresponds to the earliest time at which a passenger reaches their pickup station and is determined by adding the departure time d_p to the travel time t_A of subroute A (the time required to travel by bicycle from the passenger's starting point to the assigned pickup station). The latest start of service l is obtained by subtracting the conventional arrival time a_c from the travel time t_c of subroute C (the time required to travel from the assigned drop-off station to the passenger's destination). Summing up, the resulting time window $TW = [e, l]$ is determined by

$$(1) \quad e = d_p + t_A \quad \text{and} \quad (2) \quad l = a_c - t_c$$

Figure 11 illustrates this procedure, where each t is represented in minutes, while d_p and a_c denote actual time points. TW_p and TW_d are the resulting time windows at the assigned stations.

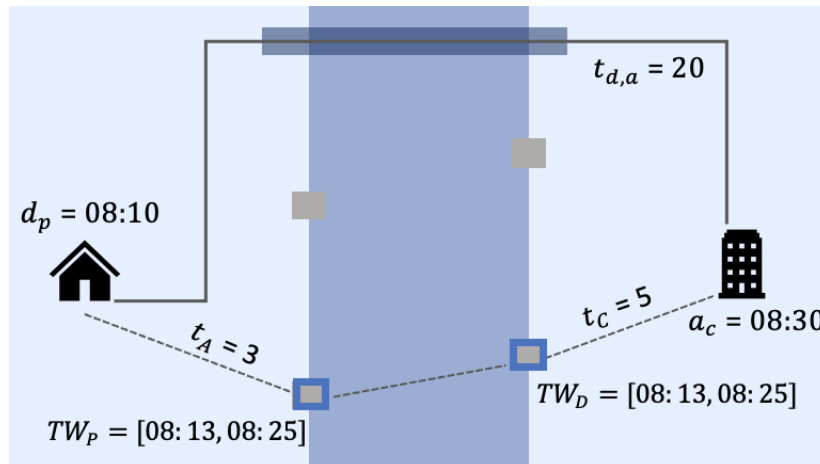


Figure 11. Resulting Time-Windows

In alignment with the mathematical formulation presented in Section 3.2, the service must not commence under any circumstances before the earliest start time of the service e . Assigned ferries are required to wait until the time window opens in case of early arrival at the pickup station. Delays, or violations of the latest start time of service l , however, are permissible but may incur penalties.

4.2.5 Result Evaluation

At the end of each optimization, the data series are appended to a csv file. Subsequently, at the conclusion of each application run, this csv file is transferred into an SQLite3 database, enabling targeted querying of results in subsequent analyses. An illustrative data table is provided in the appendix, along with the corresponding SQL queries that generate the average results depicted in Section 5.4. In addition to the tables presented in Section 5.4, the appendix also includes figures that visually enhance the interpretation of the results through line plots. The Matplotlib library, in conjunction with Panda, was employed for this purpose. The detailed process of result obtainment is outlined in the *main.py* file of the application. In this file, configurations are first set before initiating the application run, which includes optimization steps and the retrieval of results.

5 Computational Studies

The following chapter is dedicated to a computational study to evaluate how the UAF-DARP model performs as a basis for route optimization of a ferry service. The outcome quality is analyzed across different parameter settings such as fleet size, instance sizes, and choice of objective function. After detailing the performance indicators and explicating the application area that inspired this service, the experimental setup is outlined, followed by discussing optimization results.

5.1 Performance Indicators

The objective is to establish quantifiable metrics that gauge the effectiveness and potential of the ferry service. The concept of "potential" encompasses the efficiency and overall value of the service for both passengers and operators. For operators, the profitability of introducing a ferry service depends on a high utilization rate, which makes the substantial initial investments in infrastructure and vehicles worthwhile. Simultaneously, operators aim to minimize the distance traveled, which is inherently associated with achieving higher ride-sharing rates. Passengers, on the other hand, are inclined to adopt the service if it improves their mobility by saving time and/or distance compared to their previous travel options. As a result, differences in time and distance between the resulting ferry routes and conventional routes are used as performance indicators. This consistency is based on the assumption that only the savings in land-based distance (determined by station assignment) matter to passengers since they travel by bike or on foot, requiring physical effort. The distance covered on water, from the pick-up to the drop-off station, is not part of this consideration.

(1) *Time savings* refer to the deviation between the arrival time obtained from the conventional route and the arrival time obtained from the ferry route. It does not only result from reduced travel distance on land, but mainly from the routing of the ferry fleet, which determines the travel time for subroute B (route on water). The total travel time for the ferry route is the cumulative sum of the travel times for subroute A, subroute B (over water), and subroute C. The arrival time of the ferry route is determined based on the model's output, which specifies the time when a passenger is both picked up and

dropped off at their designated station. The delay $dvar_i^k$ is greater than 0, if the drop-off does not occur within the associated time window.

(2) *Distance savings* can refer to both, the (a) deviation between the passengers' travel distance obtained from the conventional route and their travel distance obtained from the ferry route, but also to the (b) deviations of distance traveled per ferry depending on the size of the ferry fleet.

5.2 Specifying the Area of Operation

The UAF-DARP Service is applied to the Northern area of Duesseldorf, where a river separates a growing residential area on the left from the city center on the right side of the river, including commercial districts, the airport, and the city's largest trade fair center. In the bounded section, three bridges cross the river, which are exposed to a high volume of traffic, especially during rush hour. All bridges serve as extended highway routes from the surrounding area and thus form necessary connections to the city center.

The introduction of a flexible ferry service is intended to shorten distances and thus create an incentive to replace private cars with bicycles or walking. Currently, the distances between the bridges are between 2 and 6 km. This range is likely to increase as the deteriorating condition of the central bridge leads to serious consideration of whether it should be demolished in the near future (Keß, 2021). The described application scenario merely serves as an example since the described problem is generally valid and thus transferable to any comparable scenario. Time savings resulting from a ferry system in this area would typically lead to single-digit time savings since bridges are already available. The fewer crossing options exist in an area, the greater the impact of a new service. Therefore, the prospect of an extended deployment of ferry services becomes even more attractive since many bridges throughout Europe reveal significant deficits (Fischer, 2017).

5.3 Experimental Setup

The passenger requests are generated randomly. The starting point of a request is not found within the same demand polygon as the destination point of that request. Clustering is not employed, and the entire application area is stretched over 8 km along a river. For each request, it is possible to identify a ferry route that is at least as efficient as the conventional route. All requests of an instance occur within a pre-defined time span ranging between 15 and 30 minutes. Delays, i.e. violations of the latest service time l_i , are allowed, but penalized in the objective function by means of a penalty factor. Each passenger is transported without transfers from the pickup station to the corresponding drop-off station. Considering the computational effort of the UAF-DARP, the runtime is limited to maximum value between 1 and 4 hours, depending on the instance size. This is especially beneficial given that a minimum of 12 optimizations are conducted for each instance, enabling a comparison of different configurations on the same instance. For every generated instance, the performance indicators are observed under varying (1) fleet size, (2) penalty factor and (3) the choice of objective function. If the fleet size of K is equal to the number of requests n , the ferry routes correspond to the minimum possible time, as there are no waiting times or detours due to ride-sharing. Each ferry can accommodate a maximum of 15 passengers. With a higher penalty factor, the focus on sticking to constraints becomes stronger, sometimes at the expense of achieving a better overall objective. Only one of two objective functions is selected. Objective function (1a) focuses on minimizing the distance traveled, which tends to result in a higher ride-sharing rate. On the other hand, objective function (1b) aims to minimize the service times of the passengers. The former is considered more beneficial for the operator, while the latter prioritizes the passengers' convenience. All experiments were conducted on a computing system equipped with an Intel(R) Core (TM) i7-9750H CPU, which runs at 2.60 GHz. This CPU boasts 6 physical cores and 12 logical processors, allowing for the utilization of up to 12 threads during the optimization process.

5.4 Results and Discussion

This section is dedicated to the analysis of ferry routes based on specified performance indicators to identify optimal parameter settings for meeting the demand of a ferry service. It commences with a micropective examination, where one generated instance involving 10 passengers is analyzed. This approach aims to enhance understanding of the application process before the assessment scope broadens across multiple instances of varying sizes in the subsequent parts of the chapter. Due to computational constraints and resource limitations, the number of passengers per instance is limited to 15 passengers throughout the entire evaluation.

5.4.1 Single-Instance Analysis

The subsequent analysis involves a request instance of $n = 10$ passengers. This analysis demonstrates the process of station assignment and the creation of network graph G using JavaScript-based visuals, which comprises a set of nodes and edges resulting from the assigned pickup and drop-off stations. A series of optimization iterations is then carried out to explore how the choice of objective function, fleet size and penalty factor impacts the results. The iteration is conducted as depicted below.

Algorithm 1: Optimization run

```
generate instance;
modes = ["operator", "passenger"];
PF = [1, 100, 1000];
K = [1, 2, 4, 6, 8, 10];
for mode in modes do
    for pf in PF do
        for k in K do
            optimize(mode, pf, k) ;
            append results to csv file
        end
    end
end
write csv file to database;
query results;
```

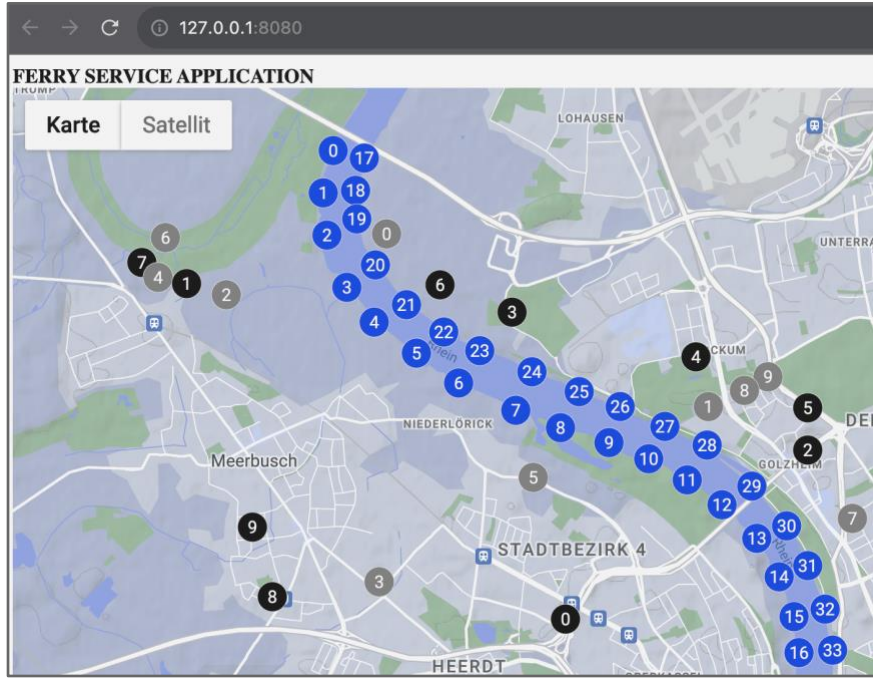
There are two different modes (operator and passenger) to choose from, which define the objective function of the model. The penalty factor is either 1, 100, or 1000. The fleet

size takes a value of 1, 2, 4, 6, 8, or 10. All configurations for this specific scenario are documented in Table 3, extracted from the application's config file.

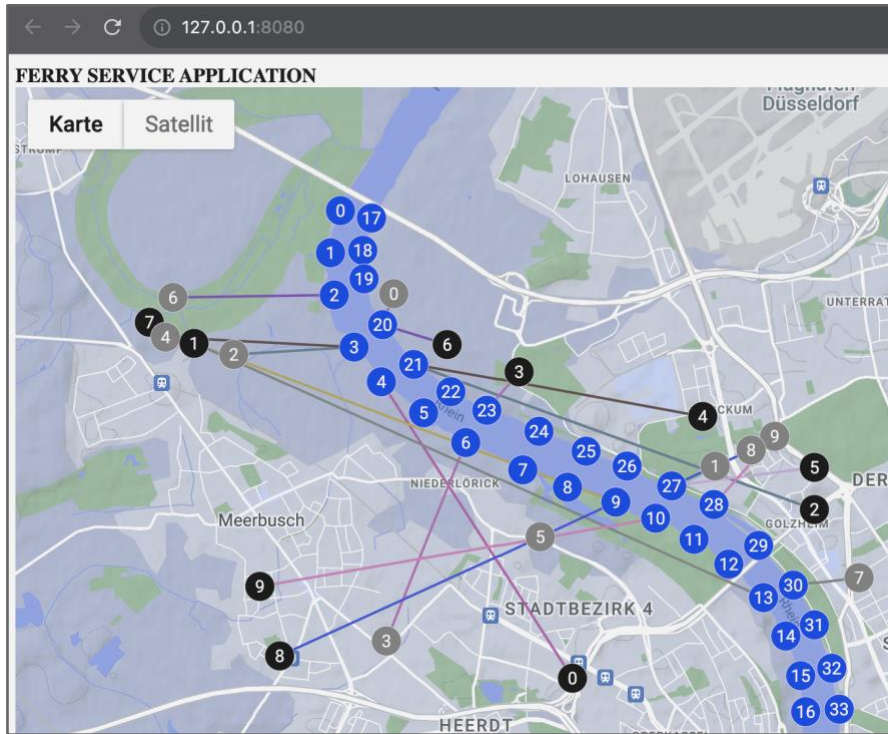
Parameter	Value	Description
dt_DEPARTURE_TIME_SPAN	15	All passenger requests for this instance fall within the time range from 0 to 15
FERRY_VELOCITY	3.33	Ferry velocity in meter per second (12km/h)
BIKING_VELOCITY	15	Cycling speed in km/h
<i>Model Parameters:</i>		
n_NUMBER_OF_PASSENGERS	10	Size of instance
d_SERVICE_TIMES	1	Berthing and embarking time
K_FLEET_SIZE	1, 2, 4, 6, 8 10	Fleet size
Q_CAPACITY	15	Number of seats per ferry
q_LOAD_PER_REQUEST	1	Number of passengers traveling per request
PENALTY_FACTOR	1, 100, 1000	Factor to determine weight of delays
TK_MAX_RT_VEHICLE	600	Maximal duration of route per ferry (in line with Roboats, see Section 2.2.2)
L_MAX_RT_PASSENGER	30	Maximal ride-time per passenger

Table 3. Configuration for single-instance analysis

Figure 12 (a) depicts available ferry stations and passenger requests visualized in the application's html file. Blue circles indicate ferry stations, gray circles pinpoint the starting points of requests, and black circles represent the corresponding destinations of the requests. Figure 12 (b) depicts the station assignments, carried out in accordance with the description in Section 4.2.2. To enhance clarity, each line connecting request locations to stations is color-coded with a unique color. All data pertaining to the instance that has been visualized and evaluated in this section can be found in the data table included in the Appendix as an exemplifying result of the comprehensive application run with penalty factor PF set to 100.



(a)



(b)

Figure 12. Deployed Application Front-End

The analyzed time span is 30 minutes, since the earliest pickup time in the instance is set to 2, and the latest return time is set to 32. The application run includes a total of 36 optimizations resulting from 6 iterated fleet sizes $\{1, 2, 4, 6, 8, 10\}$, each multiplied by 3 iterated penalty costs PF from $\{1, 100, 1000\}$, and 2 iterated objective functions. If the

objective function (1a) is selected, the optimization occurs in what is commonly referred to as *passenger mode*, focusing on minimizing passenger service time without considering the operator's costs. When solving the model with (1b) as the objective, it is termed the *operator mode*, as it prioritizes the minimization of the distance traveled by the ferries. It's important to note that both objectives also encompass the minimization of penalty costs.

$$(1a) \min \sum_{i \in P} \sum_{k \in K} u_{i+n}^k - u_{i,k}^k + \sum_{i \in D} \sum_{k \in K} dvar_i^k \cdot PF$$

$$(1b) \min \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} c_{i,j}^k \cdot x_{i,j}^k + \sum_{i \in D} \sum_{k \in K} dvar_i^k \cdot PF$$

When the objective function is set to passenger mode, it demands a significantly extended runtime. As a result, the maximum runtime for each optimization was limited to 15 minutes. It's worth mentioning that the Optimality Gap for the Passenger's objective consistently remained at 100%, given that the minimized variable u offers greater flexibility, allowing for violations due to penalty costs. Nevertheless, it still yielded results comparable to those of the operator's objective, with an optimality gap fluctuating between 75% (for 1 ferry) and 6% (for 10 ferries). The presented results showcase the outcomes obtained by running the exact same instance with different settings. Figure 13 visualizes the time savings in percentage, while Figure 14 depicts delays in minutes resulting from time-window violations at passengers' drop-off nodes. The results in Table 4 and Table 5 are analyzed for time and distance discrepancies compared to the conventional route.

(1a) Time-related results with Penalty Factor PF = 1

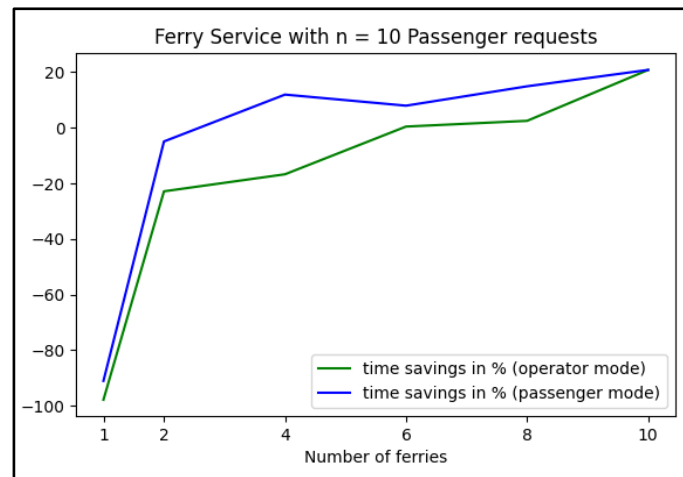


Figure 13. Time savings in % (singe-instance analysis with PF = 1).

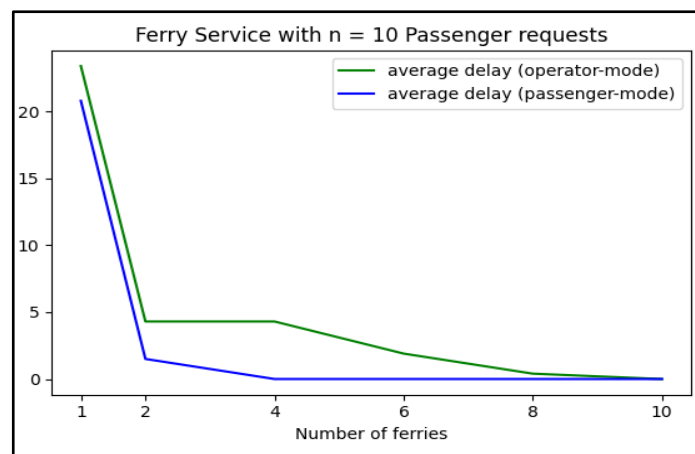


Figure 14. Delays in minutes (singe-instance analysis with PF = 1).

PF = 1				OPERATOR MODE			PASSENGER MODE		
Fleet Size	Time savings in %	Time savings in min	Delay in min	Time savings in %	Time savings in min	Delay in min	Time savings in %	Time savings in min	Delay in min
1	-97.84	-24.4	23.4	-91.05	-21.0	20.8	-91.05	-21.0	20.8
2	-22.87	-5.1	4.3	-4.94	-1.2	1.5	-4.94	-1.2	1.5
4	-16.74	-3.2	4.3	11.92	3.0	0.0	11.92	3.0	0.0
6	0.42	0.5	1.9	7.93	2.1	0.0	7.93	2.1	0.0
8	2.48	0.5	0.4	14.9	3.9	0.0	14.9	3.9	0.0
10	20.79	5.4	0.0	20.79	5.4	0.0	20.79	5.4	0.0

Table 4. Time-related results (singe-instance analysis with PF = 1)

With just one ferry ($k=1$), and a penalty factor of $PF=1$, the average time savings is above -90% (negative percentages indicate time loss) in both modes, passenger mode and operator mode. This means that, on average, passengers would take at least 20 minutes longer in both modes if they choose the ferry route instead of the conventional route. Each of the 10 passengers would be affected by delays. However, delays are not evenly distributed: five of the passengers would experience delays of less than 10 minutes, while the other 5 passengers would face delays of more than 30 minutes. As the number of ferries increases, time savings also increase. The most pronounced change in values can be observed between 1 or 2 ferries, indicating a substantial improvement in passenger time savings in this range. From 2 ferries onwards, the delay per passenger only ranges from 0 to 10 Minutes, keeping the average amount of delays below 4 minutes. To achieve time savings that surpass delays, it takes 4 ferries in passenger mode and 5 ferries in operator mode. With a fleet of 10 ferries, passengers travel directly from their pick-up station to their drop-off station since no ride-sharing is required. This results in a potential average time saving of up to 20.79% or 5.4 minutes.

(1b) Distance-related results with Penalty Factor $PF = 1$

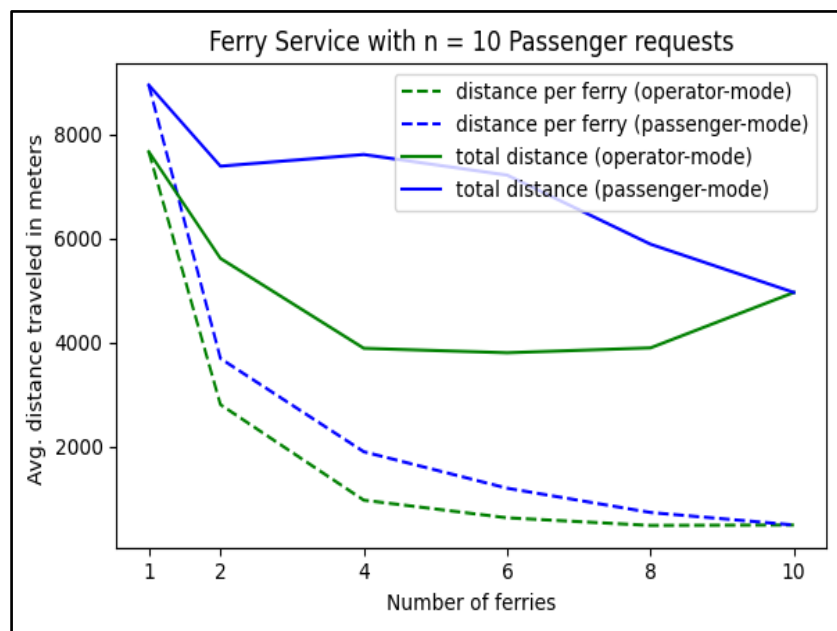


Figure 15. Avg. distance travelled per ferry (single-instance analysis with $PF = 1$).

PF = 1			OPERATOR MODE		PASSENGER MODE	
Fleet Size	Distance in meter traveled per Ferry	Total Distance in meter	Distance in meter traveled per Ferry	Total Distance in meter	Distance in meter traveled per Ferry	Total Distance in meter
1	7662	7662	8942	8942		
2	2807	5614	3692	7384		
4	972	3888	1902	7608		
6	634	3802	1202	7212		
8	487	3896	736	5886		
10	496	4960	496	4960		

Distance savings in % per passenger:	
47,23	
Distance savings in meter per passenger:	
3670	

Table 5. Distance-related results (singe-instance analysis with PF=1)

In analogy to the time savings the number of traveled distances naturally decreases as the fleet size grows. Once the fleet size exceeds four ferries, the total distance traveled remains relatively stable in operator mode. However, in passenger mode, the total distance depends more on the number of ferries, as efforts are made to optimize passenger time, even if this leads to longer travel distances. The average distance savings per passenger remain consistent at a 47.23% savings compared to the conventional route, which equates to a fixed 3.7 km reduction in land distance, and this consistency is independent of the fleet size.

(2a) Time-related results with Penalty Factor PF = 100

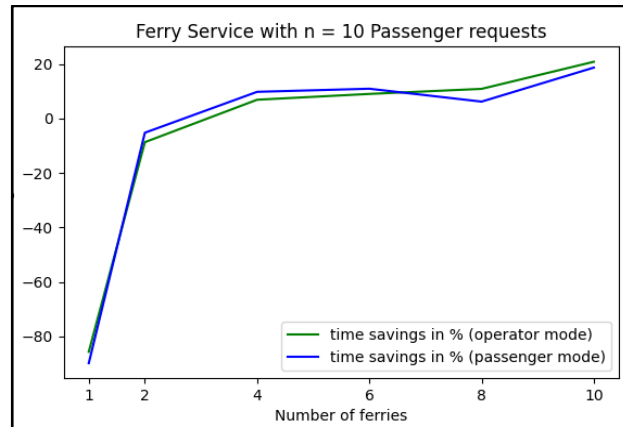


Figure 16. Time savings in % (single-instance analysis with PF=100).

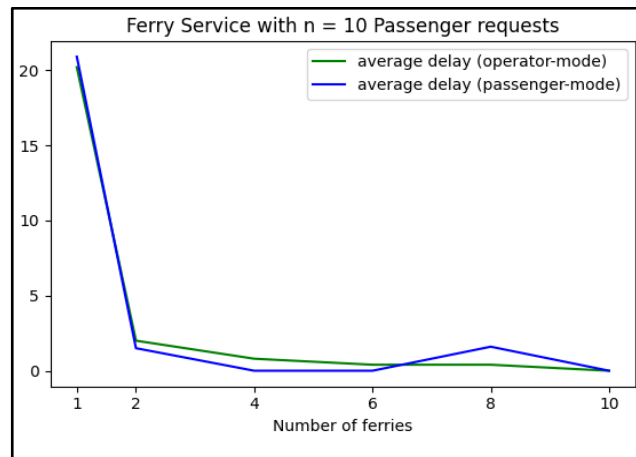


Figure 17. Delays in minutes (single-instance analysis with PF = 100).

PF = 100		OPERATOR MODE		PASSENGER MODE		
Fleet Size	Time savings in %	Time savings in min	Delay in min	Time savings in %	Time savings in min	Delay in min
1	-85.69	-20.8	20.2	-89.87	-21.5	20.9
2	-8.73	-2.1	2.0	-5.26	-1.3	1.5
4	6.85	1.8	0.8	9.75	2.4	0.0
6	8.98	2.5	0.4	10.89	2.6	0.0
8	10.84	2.6	0.4	6.16	1.6	1.6
10	20.79	5.4	0.0	18.65	4.8	0.0

Table 6. Time-related results (single-instance analysis with PF=100)

When comparing the results obtained from optimization with a Penalty Factor (PF) of 100 to the results presented in the previous section with $PF = 1$, overall higher time savings for passengers are achieved in both modes while maintaining the same runtime. The Operator mode exhibits significant enhancements in every fleet configuration. In the passenger mode, a slight deterioration is only observed when the ferry fleet consists of 8 ferries. Overall, with $PF=100$, the results in both modes are more closely aligned compared to when PF is equal to 1.

(2b) Distance-related results with Penalty Factor $PF = 100$

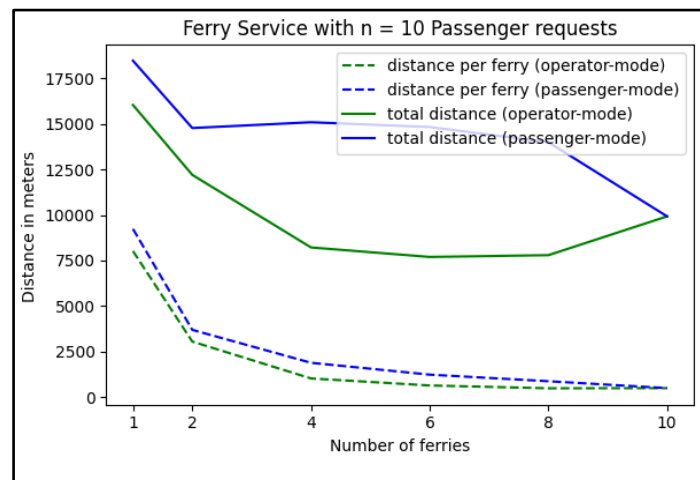


Figure 18. Avg. distance travelled per ferry (singe-instance analysis with $PF = 100$).

PF=100			PASSENGER MODE	
OPERATOR MODE				
Fleet Size	Distance in meter traveled per Ferry	Total Distance in meter	Distance in meter traveled per Ferry	Total Distance in meter
1	8372.0	8372.0	9522.0	9522.0
2	3292.0	6584.0	3692.0	7384.0
4	1082.0	4328.0	1869.5	7478.0
6	648.67	3892.0	1268.67	7612.0
8	487.0	3896.0	1013.25	8106.0
10	496.0	4960.0	496.0	4960.0

Table 7. Distance-related results (singe-instance analysis with $PF=1$)

Again, the distance covered per ferry diminishes as the fleet size expands. In operator mode, the total distance steadily decreases until a fleet size of 6, while it increases again with a growing number of ferries. This behavior may be attributed to the arc structure of the model, which necessitates each ferry to depart from the depot. In contrast, the passenger mode exhibits continuous growth in distance traveled from a fleet size of 2 until a fleet size of 8, as opposed to a decrease. This suggests that the operator mode proves advantageous for both operators and passengers since the distance covered in passenger mode is partially 50% higher than in operator mode while time savings from the respective routes remain closely aligned. If the penalty factor is set to 100 instead of 1, this leads to longer distances traveled, particularly noticeable in passenger mode. The trends in distance per ferry and total distance are consistent between the two modes, but the magnitudes differ due to the penalty factor.

(3a) Time-related results with Penalty Factor PF = 1000

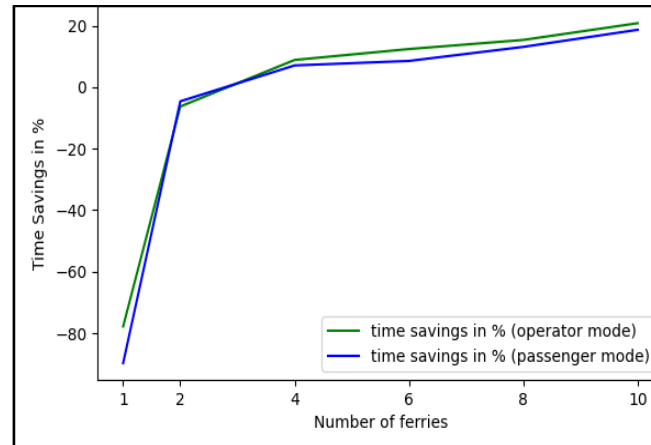


Figure 19. Time savings in % (singe-instance analysis with PF=1000).

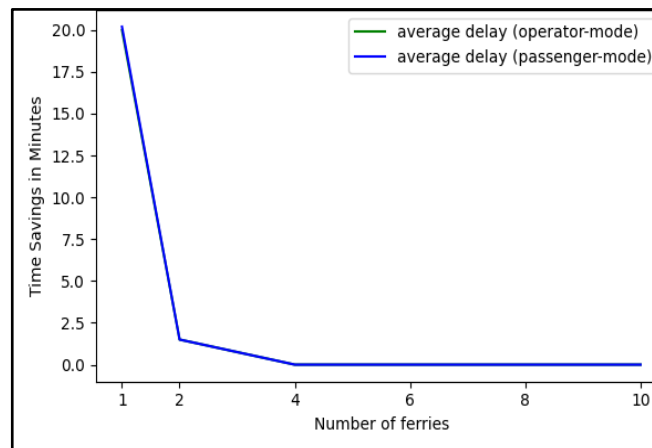


Figure 20. Delays in minutes (singe-instance analysis with PF = 1000).

PF = 1000 OPERATOR MODE				PASSENGER MODE		
Fleet Size	Time savings in %	Time savings in min	Delay in min	Time savings in %	Time savings in min	Delay in min
1	-77.81	-20.1	20.0	-89.73	-21.0	20.2
2	-6.29	-1.5	1.5	-4.61	-1.1	1.5
4	8.84	2.4	0.0	7.06	1.7	0.0
6	12.39	3.5	0.0	8.52	1.8	0.0
8	15.35	4.2	0.0	13.09	3.1	0.0
10	20.79	5.4	0.0	18.65	4.8	0.0

Table 8. Time-related results (singe-instance analysis with PF = 1000)

In the case of a Penalty Factor $PF = 1000$, the trend persists, with better results across all configurations compared to $PF = 1$ and $PF = 100$, while maintaining the same runtime. A high penalty factor leads to the optimization algorithms giving greater emphasis on minimizing delays. This is particularly advantageous, as time savings outweigh delays with a fleet size of only two ferries. The delays in minutes are almost identical in both modes.

(3b) Distance-related results with Penalty Factor $PF = 1000$

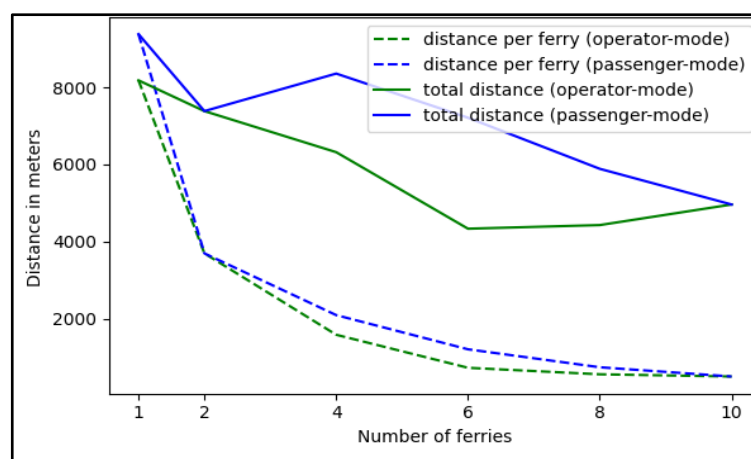


Figure 21. Avg. distance travelled per ferry (sing-instance analysis with $PF = 1000$).

PF=1000		OPERATOR MODE		PASSENGER MODE	
Fleet Size	Distance in meter traveled per Ferry	Total Distance in meter		Distance in meter traveled per Ferry	Total Distance in meter
1	8182.0	8182.0		9382.0	9382.0
2	3692.0	7384.0		3692.0	7384.0
4	1579.5	6318.0		2089.5	8358.0
6	722.0	4332.0		1202.0	7212.0
8	553.25	4426.0		735.75	5886.0
10	496.0	4960.0		496.0	4960.0

Table 9. Distance-related results (sing-instance analysis with $PF=1000$)

While the traveled distance remains relatively constant with a penalty factor of 100, a more pronounced decrease in the overall distance traveled is evident in both modes when the ferry size increases up to 6 vehicles with a penalty factor of 1000. In the operator mode, this distance increases slightly again from a size of 6 vehicles onward. In the Operator Mode, distances are consistently higher with a penalty factor of 1000 compared to PF=100 or PF=1. In the Passenger Mode, stronger reductions in distances are observed with increasing fleet size.

5.4.2 Multi-Instance Analysis

To validate the results presented above, the optimization run depicted in Figure 22 was conducted for 5 instances of the same size. The resulting outcomes are presented in the following section. If not stated otherwise, the experimental settings aligned with the configurations presented in Table X from the previous section.

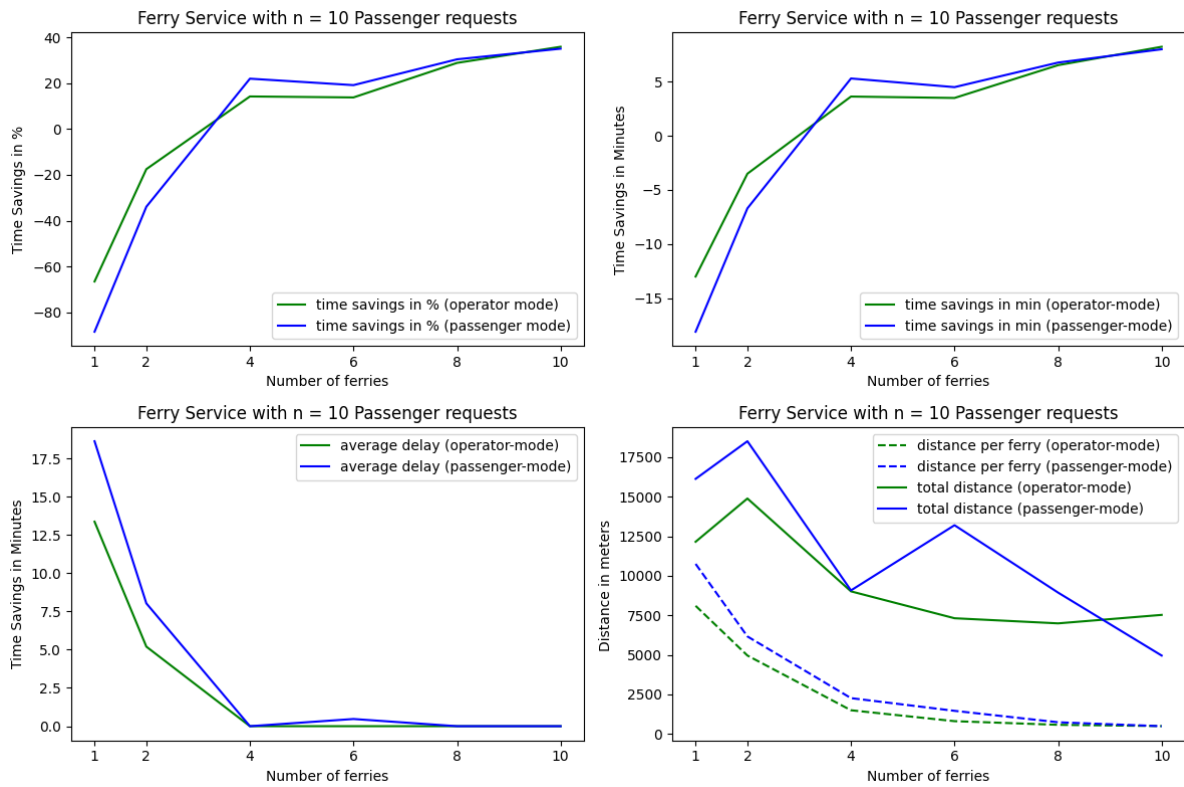


Figure 22. Avg. time and distance savings of 5 homogeneous instances ($n = 10$).

(1) The first part covers results for 5 instances each including 10 passenger requests. A Penalty Factor of 1000 was chosen, as it yielded the best results based on the preceding section. The maximum runtime was set to 20 minutes, which was found to be sufficient to obtain feasible results for every iterated configuration.

PF = 1000 OPERATOR MODE					PASSENGER MODE			
Fleet Size	Time savings in %	Time savings in min	Delay in min	Optimality Gap	Time savings in %	Time savings in min	Delay in min	Optimality Gap
1	-66.51	-13.0	13.37	92%	-88.44	-18.1	18.63	99%
2	-17.5	-3.5	5.2	79%	-33.9	-6.7	8.03	99%
4	14.22	3.63	0.0	0%	21.98	5.3	0.0	40%
6	13.79	3.5	0.0	0%	19.16	4.5	0.47	60%
8	28.86	6.53	0.0	0%	30.42	6.77	0.0	100%
10	35.9	8.23	0.0	0%	35.06	8.0	0.0	0%

Figure 23. Avg. time-based results of 5 homogeneous instances (n = 10).

PF=1000 OPERATOR MODE			PASSENGER MODE	
Fleet Size	Distance in meter traveled per Ferry	Total Distance in meter	Distance in meter traveled per Ferry	Total Distance in meter
1	8102.0	8102.0	10752.0	10752.0
2	4962.0	9924.0	6168.67	12337.33
4	1502.83	6011.33	2269.5	6052.0
6	813.11	4878.66	1466.44	8798.66
8	582.83	6994.0	744.92	8939.0
10	502.0	7530.0	496.0	4960.0

Distance savings in % per passenger:	
60.31	
Distance savings in meter per passenger:	
3944	

Table 10. Avg. distance-based results of 5 homogeneous instances (n = 10).

(2) The second of this analysis then reveals results of 5 instances, each including 15 passenger requests. The Penalty Factor PF remained at a value of 1000. The maximum runtime was set to 2 hours, as trials with smaller runtimes did not find feasible solutions. In comparison, for 10 passengers it was found to be sufficient to set the maximum runtime to 20 minutes. The iterated fleet size ranged from 5, 10, 12 to 15 ferries. After several trials failed to find results for 12 ferries but for 15, a condition was removed from the arc structure that specifies that vehicles may only travel from 0 to one point in P or D. A ferry could now travel from depot 1 to depot 2. This means that not all ferries can, but do not have to, be used for passenger requests.

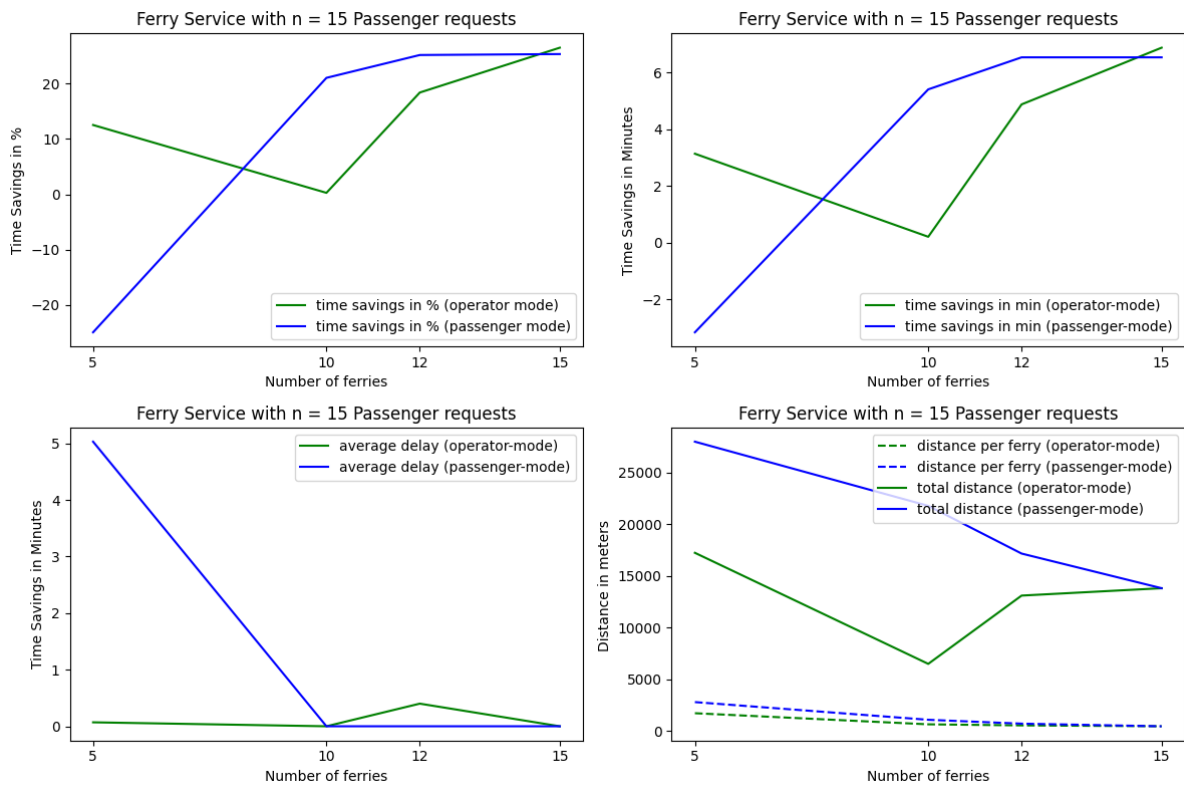


Figure 24. Avg. time and distance savings of 5 homogeneous instances (n = 15).

PF = 1000 MODE					PASSENGER MODE			
Fleet Size	Time savings in %	Time savings in min	Delay in min	Optimality Gap	Time savings in %	Time savings in min	Delay in min	Optimality Gap
5	12.5	3.13	0.07	44.5%	-24.91	-3.17	5.03	100%
10	0.26	0.2	0.0	14%	21.01	5.4	0.0	87%
12	18.36	4.87	0.4	43%	25.13	6.53	0.0	98%
15	26.45	6.87	0.0	0%	25.3	6.53	0.0	100%

Table 11. Avg. time-based results of 5 homogeneous instances (n = 15)

PF=1000 OPERATOR MODE			PASSENGER MODE	
Fleet Size	Distance in meter traveled per Ferry	Total Distance in meter	Distance in meter traveled per Ferry	Total Distance in meter
5	1722.0	8610.0	2796.0	13980.0
10	649.0	3245.0	1088.0	10880.0
12	545.33	6544.0	714.5	8574.0
15	460.0	6900.0	460.0	6900.0

Distance savings in % per passenger:	
52,11	
Distance savings in meter per passenger:	
3798	

Table 12. Avg. distance-based results of 5 homogeneous instances (n = 15)

5.5 Limitations and Outlooks

The number of passengers in this computational study remains relatively small, with a maximum of $n = 15$. It should be emphasized, that the primary goal of this work is to present a practical solution approach. However, to generate solutions for larger instances, heuristics and computationally more intensive resources should be incorporated. Moreover, there are additional approaches that could contribute to an enhanced version of the application.

(1) The application process and the model could be designed to provide the model with multiple possible pickup and drop-off stations per passenger, rather than a single, definitive station assignment. This would afford the model greater flexibility and potentially enhance demand bundling, thus increasing the ride-sharing rate.

(2) The optimization of penalty costs could be enhanced by assigning varying weights to delays for specific passengers. Currently, time windows and, consequently, the objective function are grounded in conventional routes as a benchmark. This might result in more significant increases in the objective function for short conventional routes in the event of delays. Passengers with larger time windows derive a greater benefit from the service due to longer routes, hence smaller time windows should be penalized more. To achieve this, the model implementation could employ an array of individual penalty factors instead of a unique penalty factor for all.

(3) Furthermore, the application process could introduce an option allowing passengers to actively cancel a trip after the initial schedule has been provided. Subsequently, routes could be further optimized due to a reduction in demand. It should be mentioned that the UAF-Service is a static on-demand service, as route calculation is performed only after a set of customer requests has been received. Extending the on-demand aspect may yield better results by flexibly adjusting routes and schedules in response to changing demand.

(4) Regarding objective functions, it is worth considering whether the ride-sharing rate could be increased more efficiently within the operator function than through the minimization of travel costs. Since the proposed ferry service is intended to be autonomous and electrically operated, alternative approaches, such as integrating the minimization of ferry usage, should be considered for increasing the ride-sharing rate rather than relying solely on the minimization of travel costs.

6 Conclusion

This thesis aimed to explore the possibilities of autonomous ferry systems in urban settings by reviewing relevant literature on analogous projects and the principles of relevant optimization models before introducing a developed and implemented ferry service named UAF-Service. The literature review highlighted the underutilization of ferry systems in public transportation planning, while emphasizing the potential benefits of flexible ferries. Projects driven by academic institutions showcased ongoing advancements in autonomous ferry technology before discussing the fundamentals of the Dial-A-Ride-Problem as the foundation for the ferry service presented. The UAF-Service application was developed to assess the potential benefits of introducing an autonomous ferry service from an operations research perspective. Interested parties can access the python-based application on GitHub for further usage or development. In a computational analysis, various configurations were examined for time and distance savings for passengers and operators. Noteworthy differences between operator and passenger objectives underscored the significance of objective function selection in ferry system optimization. While the computational study only scratched the surface of potential parameter settings, the fundamental observation is that ferry systems based on the Dial-A-Ride-Problem can achieve substantial time and distance savings but do require optimized usage. Against this background, the potential of autonomous ferries discussed in this thesis encourages further exploration by academic and private sectors, fostering possibilities for innovative waterway utilization in response to the challenges of urbanization.

Appendix

A Github Repository

https://github.com/emilywe170694/FERRY_SERVICE_APPLICATION.git

B Exemplifying Result Set

Request Coordinates of Single Instance Analysis (PF=100) (see Section 5.4)

Start of Request	Destination of Request
(51.26999717159475, 6.709110597661836)	(51.23662691128537, 6.73381485348564)
(51.2550047279871, 6.75348971507969)	(51.26570544862253, 6.681464101694137)
(51.2647314391159, 6.6869342724008805)	(51.25124534289744, 6.767211443103097)
(51.23984402756247, 6.707946212791647)	(51.26322547096908, 6.726391823252583)
(51.26621960413196, 6.677457614432723)	(51.25935670860031, 6.751813642180755)
(51.24892904830692, 6.729220438557718)	(51.25495625799842, 6.767175117876903)
(51.26967327815839, 6.678443264998846)	(51.265530748179344, 6.716365585039993)
(51.24540089782636, 6.773376917978012)	(51.26745634040095, 6.675276148810144)
(51.25643354274168, 6.758545285397966)	(51.23861088954723, 6.69328671633926)
(51.25761911216152, 6.761765022730833)	(51.24457031999798, 6.690487609424903)

Result Data

Column abbreviations:

dep = departure; **k** = fleet size; **ct** = conventional travel time; **tw** = time-window; **pi** = pickup time; **dr** = drop-off time; **fa** = ferry assignment; **ast** = absolute time savings; **rts** = relative time savings; **cd** = conventional distance; **fdl** = ferry route distance on land; **ads** = absolute distance savings; **rds** = relative distance savings

stations	dep	mode	k	ct	tw	pi	dr	fa	del	ft	ats	rts	cd	fdl	ads	rds
(20, 4)	13	operator	1	24	[14, 24]	61	68	0	43	68	-44	-183,33	7121	3729,79	3391,21	47,62
(27, 9)	10	operator	1	27	[11, 20]	23	34	0	13	41	-14	-51,85	8525	4815,84	3709,16	43,51
(3, 21)	15	operator	1	27	[19, 26]	53	64	0	37	65	-38	-140,74	8638	5280,8	3357,2	38,87
(6, 23)	8	operator	1	31	[16, 38]	41	44	0	5	37	-6	-19,35	9275	2555,49	6719,51	72,45
(3, 21)	2	operator	1	28	[9, 19]	52	65	0	45	74	-46	-164,29	8845	4665,83	4179,17	47,25
(9, 27)	11	operator	1	16	[14, 22]	17	21	0	0	15	1	6,25	4546	2197,78	2348,22	51,65
(2, 20)	1	operator	1	18	[7, 17]	56	60	0	42	61	-43	-238,89	5483	2202,38	3280,62	59,83
(30, 13)	2	operator	1	32	[4, 9]	5	8	0	0	31	1	3,12	10336	7128,49	3207,51	31,03
(27, 9)	4	operator	1	28	[7, 18]	22	33	0	14	43	-15	-53,57	7921	4417,51	3503,49	44,23
(28, 10)	13	operator	1	28	[16, 26]	26	30	0	3	32	-4	-14,29	8651	4763,9	3887,1	44,93
(20, 4)	13	operator	2	24	[14, 24]	15	23	1	0	23	1	4,17	7121	3729,79	3391,21	47,62
(27, 9)	10	operator	2	27	[11, 20]	16	22	0	1	29	-2	-7,41	8525	4815,84	3709,16	43,51
(3, 21)	15	operator	2	27	[19, 26]	20	27	1	0	28	-1	-3,7	8638	5280,8	3357,2	38,87

(6, 23)	8	operator	2	31	[16, 38]	33	36	1	0	29	2	6,45	9275	2555,49	6719,51	72,45
(3, 21)	2	operator	2	28	[9, 19]	19	26	1	6	35	-7	-25	8845	4665,83	4179,17	47,25
(9, 27)	11	operator	2	16	[14, 22]	23	27	0	4	21	-5	-31,25	4546	2197,78	2348,22	51,65
(2, 20)	1	operator	2	18	[7, 17]	8	16	1	0	17	1	5,56	5483	2202,38	3280,62	59,83
(30, 13)	2	operator	2	32	[4, 9]	5	8	0	0	31	1	3,12	10336	7128,49	3207,51	31,03
(27, 9)	4	operator	2	28	[7, 18]	17	21	0	2	31	-3	-10,71	7921	4417,51	3503,49	44,23
(28, 10)	13	operator	2	28	[16, 26]	30	34	0	7	36	-8	-28,57	8651	4763,9	3887,1	44,93
(20, 4)	13	operator	4	24	[14, 24]	15	23	2	0	23	1	4,17	7121	3729,79	3391,21	47,62
(27, 9)	10	operator	4	27	[11, 20]	12	16	1	0	23	4	14,81	8525	4815,84	3709,16	43,51
(3, 21)	15	operator	4	27	[19, 26]	20	27	2	0	28	-1	-3,7	8638	5280,8	3357,2	38,87
(6, 23)	8	operator	4	31	[16, 38]	17	20	3	0	13	18	58,06	9275	2555,49	6719,51	72,45
(3, 21)	2	operator	4	28	[9, 19]	18	26	2	6	35	-7	-25	8845	4665,83	4179,17	47,25
(9, 27)	11	operator	4	16	[14, 22]	17	22	1	0	16	0	0	4546	2197,78	2348,22	51,65
(2, 20)	1	operator	4	18	[7, 17]	8	12	2	0	13	5	27,78	5483	2202,38	3280,62	59,83
(30, 13)	2	operator	4	32	[4, 9]	5	8	0	0	31	1	3,12	10336	7128,49	3207,51	31,03
(27, 9)	4	operator	4	28	[7, 18]	8	18	1	0	28	0	0	7921	4417,51	3503,49	44,23
(28, 10)	13	operator	4	28	[16, 26]	25	29	1	2	31	-3	-10,71	8651	4763,9	3887,1	44,93
(20, 4)	13	operator	6	24	[14, 24]	21	25	3	0	25	-1	-4,17	7121	3729,79	3391,21	47,62
(27, 9)	10	operator	6	27	[11, 20]	12	17	5	0	24	3	11,11	8525	4815,84	3709,16	43,51
(3, 21)	15	operator	6	27	[19, 26]	20	25	2	0	26	1	3,7	8638	5280,8	3357,2	38,87
(6, 23)	8	operator	6	31	[16, 38]	17	20	4	0	13	18	58,06	9275	2555,49	6719,51	72,45
(3, 21)	2	operator	6	28	[9, 19]	10	24	2	4	33	-5	-17,86	8845	4665,83	4179,17	47,25
(9, 27)	11	operator	6	16	[14, 22]	18	23	5	0	17	-1	-6,25	4546	2197,78	2348,22	51,65
(2, 20)	1	operator	6	18	[7, 17]	8	12	3	0	13	5	27,78	5483	2202,38	3280,62	59,83
(30, 13)	2	operator	6	32	[4, 9]	5	8	0	0	31	1	3,12	10336	7128,49	3207,51	31,03
(27, 9)	4	operator	6	28	[7, 18]	13	19	5	0	29	-1	-3,57	7921	4417,51	3503,49	44,23
(28, 10)	13	operator	6	28	[16, 26]	17	21	1	0	23	5	17,86	8651	4763,9	3887,1	44,93
(20, 4)	13	operator	8	24	[14, 24]	15	19	2	0	19	5	20,83	7121	3729,79	3391,21	47,62
(27, 9)	10	operator	8	27	[11, 20]	12	16	6	0	23	4	14,81	8525	4815,84	3709,16	43,51
(3, 21)	15	operator	8	27	[19, 26]	20	27	0	0	28	-1	-3,7	8638	5280,8	3357,2	38,87
(6, 23)	8	operator	8	31	[16, 38]	17	20	5	0	13	18	58,06	9275	2555,49	6719,51	72,45
(3, 21)	2	operator	8	28	[9, 19]	10	24	0	4	33	-5	-17,86	8845	4665,83	4179,17	47,25
(9, 27)	11	operator	8	16	[14, 22]	15	19	4	0	13	3	18,75	4546	2197,78	2348,22	51,65
(2, 20)	1	operator	8	18	[7, 17]	8	12	3	0	13	5	27,78	5483	2202,38	3280,62	59,83
(30, 13)	2	operator	8	32	[4, 9]	5	10	7	0	33	-1	-3,12	10336	7128,49	3207,51	31,03
(27, 9)	4	operator	8	28	[7, 18]	8	19	6	0	29	-1	-3,57	7921	4417,51	3503,49	44,23
(28, 10)	13	operator	8	28	[16, 26]	17	27	1	0	29	-1	-3,57	8651	4763,9	3887,1	44,93
(20, 4)	13	operator	10	24	[14, 24]	15	19	1	0	19	5	20,83	7121	3729,79	3391,21	47,62
(27, 9)	10	operator	10	27	[11, 20]	12	16	9	0	23	4	14,81	8525	4815,84	3709,16	43,51
(3, 21)	15	operator	10	27	[19, 26]	20	24	5	0	25	2	7,41	8638	5280,8	3357,2	38,87
(6, 23)	8	operator	10	31	[16, 38]	17	20	7	0	13	18	58,06	9275	2555,49	6719,51	72,45
(3, 21)	2	operator	10	28	[9, 19]	10	14	6	0	23	5	17,86	8845	4665,83	4179,17	47,25

(9, 27)	11	operator	10	16	[14, 22]	15	19	2	0	13	3	18,75	4546	2197,78	2348,22	51,65
(2, 20)	1	operator	10	18	[7, 17]	8	12	3	0	13	5	27,78	5483	2202,38	3280,62	59,83
(30, 13)	2	operator	10	32	[4, 9]	5	8	0	0	31	1	3,12	10336	7128,49	3207,51	31,03
(27, 9)	4	operator	10	28	[7, 18]	8	12	4	0	22	6	21,43	7921	4417,51	3503,49	44,23
(28, 10)	13	operator	10	28	[16, 26]	17	21	8	0	23	5	17,86	8651	4763,9	3887,1	44,93
(20, 4)	13	passenger	1	24	[14, 24]	55	61	0	36	61	-37	-154,17	7121	3729,79	3391,21	47,62
(27, 9)	10	passenger	1	27	[11, 20]	22	33	0	12	40	-13	-48,15	8525	4815,84	3709,16	43,51
(3, 21)	15	passenger	1	27	[19, 26]	52	58	0	31	59	-32	-118,52	8638	5280,8	3357,2	38,87
(6, 23)	8	passenger	1	31	[16, 38]	41	44	0	5	37	-6	-19,35	9275	2555,49	6719,51	72,45
(3, 21)	2	passenger	1	28	[9, 19]	64	74	0	54	83	-55	-196,43	8845	4665,83	4179,17	47,25
(9, 27)	11	passenger	1	16	[14, 22]	17	21	0	0	15	1	6,25	4546	2197,78	2348,22	51,65
(2, 20)	1	passenger	1	18	[7, 17]	67	71	0	53	72	-54	-300	5483	2202,38	3280,62	59,83
(30, 13)	2	passenger	1	32	[4, 9]	5	8	0	0	31	1	3,12	10336	7128,49	3207,51	31,03
(27, 9)	4	passenger	1	28	[7, 18]	23	34	0	15	44	-16	-57,14	7921	4417,51	3503,49	44,23
(28, 10)	13	passenger	1	28	[16, 26]	26	30	0	3	32	-4	-14,29	8651	4763,9	3887,1	44,93
(20, 4)	13	passenger	2	24	[14, 24]	15	21	0	0	21	3	12,5	7121	3729,79	3391,21	47,62
(27, 9)	10	passenger	2	27	[11, 20]	17	21	1	0	28	-1	-3,7	8525	4815,84	3709,16	43,51
(3, 21)	15	passenger	2	27	[19, 26]	24	28	0	1	29	-2	-7,41	8638	5280,8	3357,2	38,87
(6, 23)	8	passenger	2	31	[16, 38]	36	39	0	0	32	-1	-3,23	9275	2555,49	6719,51	72,45
(3, 21)	2	passenger	2	28	[9, 19]	11	18	0	0	27	1	3,57	8845	4665,83	4179,17	47,25
(9, 27)	11	passenger	2	16	[14, 22]	23	27	1	4	21	-5	-31,25	4546	2197,78	2348,22	51,65
(2, 20)	1	passenger	2	18	[7, 17]	8	14	0	0	15	3	16,67	5483	2202,38	3280,62	59,83
(30, 13)	2	passenger	2	32	[4, 9]	5	8	1	0	31	1	3,12	10336	7128,49	3207,51	31,03
(27, 9)	4	passenger	2	28	[7, 18]	16	22	1	3	32	-4	-14,29	7921	4417,51	3503,49	44,23
(28, 10)	13	passenger	2	28	[16, 26]	30	34	1	7	36	-8	-28,57	8651	4763,9	3887,1	44,93
(20, 4)	13	passenger	4	24	[14, 24]	15	19	2	0	19	5	20,83	7121	3729,79	3391,21	47,62
(27, 9)	10	passenger	4	27	[11, 20]	12	16	0	0	23	4	14,81	8525	4815,84	3709,16	43,51
(3, 21)	15	passenger	4	27	[19, 26]	20	24	1	0	25	2	7,41	8638	5280,8	3357,2	38,87
(6, 23)	8	passenger	4	31	[16, 38]	36	39	3	0	32	-1	-3,23	9275	2555,49	6719,51	72,45
(3, 21)	2	passenger	4	28	[9, 19]	10	14	1	0	23	5	17,86	8845	4665,83	4179,17	47,25
(9, 27)	11	passenger	4	16	[14, 22]	19	23	0	0	17	-1	-6,25	4546	2197,78	2348,22	51,65
(2, 20)	1	passenger	4	18	[7, 17]	8	12	2	0	13	5	27,78	5483	2202,38	3280,62	59,83
(30, 13)	2	passenger	4	32	[4, 9]	7	10	3	0	33	-1	-3,12	10336	7128,49	3207,51	31,03
(27, 9)	4	passenger	4	28	[7, 18]	11	17	0	0	27	1	3,57	7921	4417,51	3503,49	44,23
(28, 10)	13	passenger	4	28	[16, 26]	17	21	3	0	23	5	17,86	8651	4763,9	3887,1	44,93
(20, 4)	13	passenger	6	24	[14, 24]	15	19	3	0	19	5	20,83	7121	3729,79	3391,21	47,62
(27, 9)	10	passenger	6	27	[11, 20]	17	21	1	0	28	-1	-3,7	8525	4815,84	3709,16	43,51
(3, 21)	15	passenger	6	27	[19, 26]	23	27	2	0	28	-1	-3,7	8638	5280,8	3357,2	38,87
(6, 23)	8	passenger	6	31	[16, 38]	28	31	1	0	24	7	22,58	9275	2555,49	6719,51	72,45
(3, 21)	2	passenger	6	28	[9, 19]	10	14	2	0	23	5	17,86	8845	4665,83	4179,17	47,25
(9, 27)	11	passenger	6	16	[14, 22]	16	20	0	0	14	2	12,5	4546	2197,78	2348,22	51,65
(2, 20)	1	passenger	6	18	[7, 17]	8	12	5	0	13	5	27,78	5483	2202,38	3280,62	59,83

(30, 13)	2	passenger	6	32	[4, 9]	7	10	4	0	33	-1	-3,12	10336	7128,49	3207,51	31,03
(27, 9)	4	passenger	6	28	[7, 18]	8	12	1	0	22	6	21,43	7921	4417,51	3503,49	44,23
(28, 10)	13	passenger	6	28	[16, 26]	23	27	0	0	29	-1	-3,57	8651	4763,9	3887,1	44,93
(20, 4)	13	passenger	8	24	[14, 24]	15	19	4	0	19	5	20,83	7121	3729,79	3391,21	47,62
(27, 9)	10	passenger	8	27	[11, 20]	17	21	3	0	28	-1	-3,7	8525	4815,84	3709,16	43,51
(3, 21)	15	passenger	8	27	[19, 26]	20	24	2	0	25	2	7,41	8638	5280,8	3357,2	38,87
(6, 23)	8	passenger	8	31	[16, 38]	17	20	5	0	13	18	58,06	9275	2555,49	6719,51	72,45
(3, 21)	2	passenger	8	28	[9, 19]	16	20	0	0	29	-1	-3,57	8845	4665,83	4179,17	47,25
(9, 27)	11	passenger	8	16	[14, 22]	15	19	7	0	13	3	18,75	4546	2197,78	2348,22	51,65
(2, 20)	1	passenger	8	18	[7, 17]	13	17	2	0	18	0	0	5483	2202,38	3280,62	59,83
(30, 13)	2	passenger	8	32	[4, 9]	5	8	6	0	31	1	3,12	10336	7128,49	3207,51	31,03
(27, 9)	4	passenger	8	28	[7, 18]	8	12	1	0	22	6	21,43	7921	4417,51	3503,49	44,23
(28, 10)	13	passenger	8	28	[16, 26]	39	43	2	16	45	-17	-60,71	8651	4763,9	3887,1	44,93
(20, 4)	13	passenger	10	24	[14, 24]	15	19	5	0	19	5	20,83	7121	3729,79	3391,21	47,62
(27, 9)	10	passenger	10	27	[11, 20]	12	16	4	0	23	4	14,81	8525	4815,84	3709,16	43,51
(3, 21)	15	passenger	10	27	[19, 26]	20	24	7	0	25	2	7,41	8638	5280,8	3357,2	38,87
(6, 23)	8	passenger	10	31	[16, 38]	17	20	2	0	13	18	58,06	9275	2555,49	6719,51	72,45
(3, 21)	2	passenger	10	28	[9, 19]	10	14	6	0	23	5	17,86	8845	4665,83	4179,17	47,25
(9, 27)	11	passenger	10	16	[14, 22]	15	19	3	0	13	3	18,75	4546	2197,78	2348,22	51,65
(2, 20)	1	passenger	10	18	[7, 17]	8	12	9	0	13	5	27,78	5483	2202,38	3280,62	59,83
(30, 13)	2	passenger	10	32	[4, 9]	5	8	0	0	31	1	3,12	10336	7128,49	3207,51	31,03
(27, 9)	4	passenger	10	28	[7, 18]	8	12	8	0	22	6	21,43	7921	4417,51	3503,49	44,23
(28, 10)	13	passenger	10	28	[16, 26]	23	27	1	0	29	-1	-3,57	8651	4763,9	3887,1	44,93

Bibliography

- Agatz, N., Erera, A., Savelsbergh, M., & Wang, X. (2012). Optimization for dynamic ride-sharing: A review. *European Journal of Operational Research*, 223(2), 295–303.
<https://doi.org/10.1016/j.ejor.2012.05.028>
- Anderson, J. M., Nidhi, K., Stanley, K. D., Sorensen, P., Samaras, C., & Oluwatola, O. A. (2014). *Autonomous Vehicle Technology: A Guide for Policymakers*. Rand Corporation.
- Arena, F., Pau, G., & Severino, A. (2020). An Overview on the Current Status and Future Perspectives of Smart Cars. *Infrastructures*, 5(7), Article 7.
<https://doi.org/10.3390/infrastructures5070053>
- Aslaksen, I. E., Svanberg, E., Fagerholt, K., Johnsen, L. C., & Meisel, F. (2021). A combined dial-a-ride and fixed schedule ferry service for coastal cities. *Transportation Research Part A: Policy and Practice*, 153, 306–325.
<https://doi.org/10.1016/j.tra.2021.09.004>
- Bagloee, S., Tavana, M., Asadi, M., & Oliver, T. (2016). Autonomous Vehicles: Challenges, Opportunities and Future Implications for Transportation Policies. *Journal of Modern Transportation*, 24, 284–303. <https://doi.org/10.1007/s40534-016-0117-3>
- Beck, J. C., Prosser, P., & Selensky, E. (2003). Vehicle Routing and Job Shop Scheduling: What's the difference? 9.
- Becker, F., Bösch, P. M., Ciari, F., & Axhausen, K. W. (2016). Entwicklung konsistenter Szenarien für die Einführung autonomer Fahrzeuge. *Arbeitsberichte Verkehrs- und Raumplanung*, 1168. <https://doi.org/10.3929/ethz-b-000249260>

- Benson, T. (2021, September 25). Programmable Cities: Using Roboat to Create a responsive Autonomous Infrastructure in Amsterdam | by Urban AI | Urban AI | Medium. <https://medium.com/urban-ai/programmable-cities-using-roboat-to-create-a-responsive-autonomous-infrastructure-in-amsterdam-1894bae64f91>
- Bratić, K., Pavić, I., Vukša, S., & Stazić, L. (2019). Review of Autonomous and Remotely Controlled Ships in Maritime Sector. *Transactions on Maritime Science*, 8(2), 253–265. <https://doi.org/10.7225/toms.v08.n02.011>
- Brekke, E. F., Eide, E., Eriksen, B.-O. H., Wilthil, E. F., Breivik, M., Skjellaug, E., Helgesen, Ø. K., Lekkas, A. M., Martinsen, A. B., Thyri, E. H., Torben, T., Veitch, E., Alsos, O. A., & Johansen, T. A. (2022). milliAmpere: An Autonomous Ferry Prototype. *Journal of Physics. Conference Series*, 2311(1), 12029-. <https://doi.org/10.1088/1742-6596/2311/1/012029>
- Caceres-Cruz, J., Arias, P., Guimarans, D., Riera, D., & Juan, A. A. (2014). Rich Vehicle Routing Problem: Survey. *ACM Computing Surveys*, 47(2), 32:1-32:28. <https://doi.org/10.1145/2666003>
- Canzler, W., & Knie, A. (2023). The Next Big Thing: Autonome Shuttles als neuer öffentlichen Nahverkehr. Perspektiven und Probleme der Implementierung. In D. Sack, H. Straßheim, & K. Zimmermann (Hrsg.), *Renaissance der Verkehrspolitik: Politik- und mobilitätswissenschaftliche Perspektiven* (S. 383–404). Springer Fachmedien. https://doi.org/10.1007/978-3-658-38832-4_15
- CAPTn. (2023, August 21). CAPTN. <https://Captn.Sh/>. <https://captn.sh/>
- Cheemakurthy, H. (2017). Urban waterborne public transport systems: An overview of existing operations in world cities. <https://www.diva-portal.org/smash/record.jsf?pid=diva2:1168873>

- Cordeau, J.-F., Desaulniers, G., Desrosiers, J., Solomon, M. M., & Soumis, F. (2001). VRP with Time Windows. In *The vehicle routing problem* (S. 157–193). Society for Industrial and Applied Mathematics.
- Christofides, N. (1976). The vehicle routing problem. *Revue Française d'automatique, Informatique, Recherche Opérationnelle. Recherche Opérationnelle*, 10(V1), 55–70. <https://doi.org/10.1051/ro/197610V100551>
- Dantzig, G. B., & Ramser, J. H. (1959). The Truck Dispatching Problem. *Management Science*, 6(1), 80–91.
- Detti, P., Papalini, F., & Lara, G. Z. M. de. (2017). A multi-depot dial-a-ride problem with heterogeneous vehicles and compatibility constraints in healthcare. *Omega*, 70, 1–14. <https://doi.org/10.1016/j.omega.2016.08.008>
- Deutsche Bahn. (2023). On-Demand-Verkehre starten im Pilotbetrieb - Mobilität auf Abruf für eine smarte Stadt. Deutsche Bahn. https://www.deutschebahn.com/resource/blob/10397974/11ba197c7db25432db0bed24ca7b5dc5/20230222_PI-On-Demand-Verkehre-data.pdf
- Eldrandaly, K., & Ahmed, A. (2008, Dezember 22). Routing Problems: A Survey.
- Fagnant, D. J., & Kockelman, K. (2015). Preparing a nation for autonomous vehicles: Opportunities, barriers and policy recommendations. *Transportation Research Part A: Policy and Practice*, 77, 167–181. <https://doi.org/10.1016/j.tra.2015.04.003>
- Felski, A., & Zwolak, K. (2020). The Ocean-Going Autonomous Ship—Challenges and Threats. *Journal of Marine Science and Engineering*, 8(1), 41. <https://doi.org/10.3390/jmse8010041>
- Fischer, O. (2017). Do old bridges last longer than expected? TUM. <https://www.tum.de/en/about-tum/news/press-releases/details/33959>

- Goerlandt, F., & Pulsifer, K. (2022). An exploratory investigation of public perceptions towards autonomous urban ferries. *Safety Science*, 145, 105496.
<https://doi.org/10.1016/j.ssci.2021.105496>
- Gu, Y., & Wallace, S. W. (2021). Operational benefits of autonomous vessels in logistics — A case of autonomous water-taxis in Bergen. *Transportation Research Part E: Logistics and Transportation Review*, 154, 102456.
<https://doi.org/10.1016/j.tre.2021.102456>
- Guo, C., Haugen, S., & Utne, I. B. (2023). Risk assessment of collisions of an autonomous passenger ferry. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 237(2), 425–435.
<https://doi.org/10.1177/1748006X211050714>
- Hyland, M., & Mahmassani, H. S. (2018). Dynamic autonomous vehicle fleet operations: Optimization-based strategies to assign AVs to immediate traveler demand requests. *Transportation Research Part C: Emerging Technologies*, 92, 278–297.
<https://doi.org/10.1016/j.trc.2018.05.003>
- Jünger, M., Reinelt, G., & Rinaldi, G. (1995). Chapter 4 The traveling salesman problem. In *Handbooks in Operations Research and Management Science* (Bd. 7, S. 225–330). Elsevier. [https://doi.org/10.1016/S0927-0507\(05\)80121-5](https://doi.org/10.1016/S0927-0507(05)80121-5)
- Keß, M. (2021, Februar 3). Marode Rheinquerung: Düsseldorf will noch 2021 über Abriss der Theodor-Heuss-Brücke entscheiden. RP ONLINE. https://rp-online.de/nrw/staedte/duesseldorf/theodor-heuss-bruecke-duesseldorf-stadt-will-noch-2021-ueber-abriss-entscheiden_aid-56029889
- Kidd, R., & McCarthy, E. (2019). MARITIME EDUCATION IN THE AGE OF AUTONOMY. 221–230. <https://doi.org/10.2495/MT190201>

- Kirschgens, L. (2022, Dezember 21). Autonome Schifffahrt. Kapitän über Bord. Sueddeutsche Zeitung.
<https://www.sueddeutsche.de/projekte/artikel/wissen/autonome-faehre-elektrisch-schifffahrt-e852478/>
- Mourad, A., Puchinger, J., & Chu, C. (2019). A survey of models and algorithms for optimizing shared mobility. *Transportation Research Part B: Methodological*, Volume 123(123), 323–346. <https://doi.org/10.1016/j.trb.2019.02.003>
- Munim, Z. H., Imset, M., Faury, O., Sukke, M., & Kim, H. (2022). Public Perception on Safety of Autonomous Ferry in the Norwegian Context. 2022 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), 1027–1032. <https://doi.org/10.1109/IEEM55944.2022.9989557>
- Nguyen, G. T. H., Ruzaeva, D., Góez, J. C., & Guajardo, M. (2022). Insights on the introduction of autonomous vessels to liner shipping networks. *Journal of Shipping and Trade*, 7(1), 12. <https://doi.org/10.1186/s41072-022-00113-w>
- NHTSA. (2021). Automated Vehicles for Safety | NHTSA [Text].
<https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety#topic-road-self-driving>
- Parragh, S. N., Doerner, K. F., & Hartl, R. F. (2008). A survey on pickup and delivery problems: Part II: Transportation between pickup and delivery locations. *Journal Für Betriebswirtschaft*, 58(2), 81–117. <https://doi.org/10.1007/s11301-008-0036-4>
- Pearl, J. (1984). *Heuristics: Intelligent search strategies for computer problem solving*. Addison-Wesley.

- Pfeiffer, C., & Schulz, A. (2022). An ALNS algorithm for the static dial-a-ride problem with ride and waiting time minimization. *OR Spectrum*, 44(1), 87–119.
<https://doi.org/10.1007/s00291-021-00656-7>
- Reddy, N. P., Zadeh, M. K., Thieme, C. A., Skjetne, R., Sorensen, A. J., Aanondsen, S. A., Breivik, M., & Eide, E. (2019). Zero-Emission Autonomous Ferries for Urban Water Transport: Cheaper, Cleaner Alternative to Bridges and Manned Vessels. *IEEE Electrification Magazine*, 7(4), 32–45.
<https://doi.org/10.1109/MELE.2019.2943954>
- Roboat—Autonomy on the waterways. (2023, September 5). Roboat.
<https://roboat.tech/>
- Salani, M., & Battarra, M. (2018). The opportunity cost of time window violations. *EURO Journal on Transportation and Logistics*, 7(4), 343–361.
<https://doi.org/10.1007/s13676-018-0121-3>
- Serban, A. C., Poll, E., & Visser, J. (2018). A Standard Driven Software Architecture for Fully Autonomous Vehicles. 2018 IEEE International Conference on Software Architecture Companion (ICSA-C), 120–127. <https://doi.org/10.1109/ICSA-C.2018.00040>
- Stiglic, M., Agatz, N., Savelsbergh, M., & Gradisar, M. (2015). The benefits of meeting points in ride-sharing systems. *Transportation Research Part B: Methodological*, 82, 36–53.
- Stiglic, M., Agatz, N., Savelsbergh, M., & Gradisar, M. (2018). Enhancing urban mobility: Integrating ride-sharing and public transit. *Computers & Operations Research*, 90, 12–21. <https://doi.org/10.1016/j.cor.2017.08.016>
- Tarkowski, M., & Puzdrakiewicz, K. (2021). Connectivity Benefits of Small Zero-Emission Autonomous Ferries in Urban Mobility—Case of the Coastal City

of Gdańsk (Poland). *Sustainability*, 13(23), Article 23.

<https://doi.org/10.3390/su132313183>

van Toor, H. (2020). Waste Collection in Amsterdam Centrum Using Autonomous Vessels: Optimization of container type selection and location for household waste collection in Amsterdam Centrum with Autonomous Vessels (Roboats). <https://repository.tudelft.nl/islandora/object/uuid%3A8d5e0ef2-63d1-42ea-a69e-1f8fc89f4b10>

Wang, W., Shan, T., Leoni, P., Fernández-Gutiérrez, D., Meyers, D., Ratti, C., & Rus, D. (2020). Roboat II: A Novel Autonomous Surface Vessel for Urban Environments. 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 1740–1747. <https://doi.org/10.1109/IROS45743.2020.9340712>

Werners, B. (2013). *Grundlagen des Operations Research*. Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-40102-2>

Yao, D., He, S., & Wang, Z. (2021). A new ride-sharing model incorporating the passengers' efforts. *Naval Research Logistics (NRL)*, 68(4), 397–411. <https://doi.org/10.1002/nav.21965>

Yara International. (2018, Mai 24). Yara None. <https://www.yara.com/corporate-releases/yara-and-kalmar-to-develop-worlds-first-fully-digitalized-and-zero-emission-cargo-solution-for-yara-birkeland/>

Zeabus. (2023a, Juli 3). Torghatten and Zeabuz Make History in Stockholm. Zeabuz. <https://www.zeabuz.com/torghatten-and-zeabuz-make-history-in-stockholm/>

Zeabus. (2023b, August 12). Zeabuz. <https://www.zeabuz.com/>