Dynamic Centralised Fleet Management of Waterborne Vessels for Heterogeneous Services

Heisuke Miyoshi

Delft University of Technology

Dynamic Centralised Fleet Management of Waterborne Vessels for Heterogeneous Services

by

Heisuke Miyoshi

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday August 1, 2024.

Student number:	5733693	
Project duration:	February 12, 2024 – August 1, 2024	
Thesis committee:	Dr. S. Sharif Azadeh ,	TU Delft, supervisor
	Prof. Dr. Oded Cats,	TU Delft, chair
	Dr. Yimeng Zhang,	TU Delft, advisor

Style: TU Delft Report Style, with modifications by Daan Zwaneveld

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Acknowledgement

This thesis closes my chapter of life as a master's student at TU Delft and brings me to the next chapter of my life book. Two years ago, I decided to grab a one-way ticket from Japan to the Netherlands. This twist in my life surprised many people, including myself. Getting out of the comfort zone is always scary, but at the same time, it brings a lot of excitement and joy. My time in Delft has been filled with enriching experiences: learning new skills and knowledge through study, meeting and bonding with new people, and living in a different culture are all invaluable experiences I had here. They have brought so much entertainment to my life story. I would like to express my gratitude to those who have been part of this chapter, and I would like to thank some people specifically for contributing to my life book.

To my committee, thank you for providing me the opportunity to work on an exciting topic and for supporting me throughout the last six months. This project made me grow more as an engineer through technical challenges and critical thinking, and this project has built the fundamentals for me to apply my knowledge and skills in the future.

To my friends, I cannot express how grateful I am for having all of you in my life in Delft. Going through the academic journey full of ups and downs, despite how flat this country is, was impossible without you all. I will miss all the technical, cultural, political, philosophical, financial, or whatever discussions we had on campus and in pubs. Every single conversation and interaction I have had with you has shaped who I am now, and it will be the most important part of a chapter in my life.

最後に父さん、母さん、家族のみんな、とても貴重な経験をさせてくれて、そして気にかけてくれて本当にありがとう。例え遠く離れていても僕のことを応援してくれている人がいることが僕のデルフトでの二年間を支えてくれました。この二年間のみならず、生涯を通して僕の好きなようにさせてくれたこと、そして僕をいつでも見守り、困ったときには助けくれたこと、感謝してもしきれません。みんなが僕に与えてくれたすべてのものが僕の財産で、僕の誇りです。

Heisuke Miyoshi
Delft, July 2024

Abstract

Urban areas have been facing the challenge of increased demand for urban transportation due to the expansion of opportunities for activities and the growth of e-commerce. These areas constantly face congestion, leading to more emissions and noise/air pollution. These circumstances have led to a strong interest in sustainability in mobility and logistics in urban areas. In particular, integrating mobility and logistics in a multimodal transport network instead of conventional transportation in which mobility and logistics are handled separately has caught interest. It is expected that the utilisation of water transit as a major means of urban transportation is important for the realisation of a sustainable multimodal urban transport network where abundant waterways are available, and combining mobility and logistics in the urban ferry system has the potential to provide more efficient urban transportation. To investigate the potential of such a system, a dynamic centralised fleet management model that optimises the operation of waterborne vessels is developed in this thesis, considering an electric waterborne vessel system for heterogeneous on-demand service that serves stochastic passenger travel and parcel delivery requests. The model dynamically optimises the operation of vessels by applying a rolling horizon and updates the operation plan every time a new request is inserted. Two solving algorithms, the exact method and the insertion heuristic, are proposed. The computational experiments are conducted by taking the city of Fredrikstad in Norway as the case location to evaluate the solving capability of the proposed solving algorithms and to assess the efficiency and service level of the transport system by comparing the performance with the conventional fixed purpose vessels under different demand scenarios. The demand scenarios are generated using a stochastic approach, which applies a non-homogeneous Poisson process for passenger requests and a probabilistic approach for parcel requests. The results suggest that the proposed insertion heuristic is capable of being applied to this model by providing good solutions in a significantly shorter computational time. Also, combining mobility and logistics results in higher efficiency and service levels for all demand scenarios than those of the fixed purpose vessels. To conclude, a model that determines the operation of electric waterborne vessels for on-demand heterogeneous services considering the stochasticity of the demand is developed in this thesis. The results show the capability of the proposed model to dynamically optimise the operation and the benefits of combining mobility and logistics in vessels for heterogeneous on-demand service.

Contents

List of Figures

List of Tables

1 Introduction

1.1. Research Context

Mobility and logistics are essential systems for human society which base the opportunities for human activities. After the invention and spread of automobiles, mobility and logistics drastically expanded its reachable area in a short period, which has provided more and more opportunities for human activities. However, the expansion of human activities has led to enormous consumption of resources as well as emissions from transport, and sustainability in mobility and logistics has been a vital challenge worldwide. In fact, the European Green Deal sets a goal of achieving a 90% reduction in greenhouse gas emissions from transport by 2050 (EuropeanCommision, 2021), and promotes the realisation of connected and automated multimodal mobility. In order to realise sustainable multimodal mobility, providing highly efficient service is essential to decrease the emissions per capita for each service line and reduce the aggregated emissions compared to transport modes for individuals. Efficient multimodal mobility will also contribute to reducing traffic congestion and noise caused by traffic as well by enhancing modal shifts and improving the urban environment. Therefore, it is necessary to design the most efficient service that satisfies the transport demand so that it does not hinder human activities while ensuring the sustainability of the transport systems.

Although many urban areas are already facing traffic congestion and air and noise pollution from vehicles, the rapid growth of e-commerce activities is leading to more necessity of urban freight transportation, and this growth in freight transportation in urban areas aggravates the traffic situation. One reason for these issues is that people and freight transport are operated separately in urban areas. In fact, it is rarely the case that a transport system in urban areas, such as buses and trams, carries parcels in their vehicles, as well as freight vehicles, such as trucks or vans, provide transportation service for passengers. This situation raised the idea of combining passengers and freight transportation in urban areas to accommodate the growth of demand for urban freight transportation while maintaining the traffic conditions and urban environment (Cleophas et al., 2019). However, Cleophas et al. (2019) states that the low capacity of vehicles when combining passengers and freight transport leads to longer travel distances to satisfy the demand. This implies the necessity of largely capacitated vehicles to combine passenger and freight transport for more efficient urban transport.

When it comes to multimodal transport networks, land transport such as road and railway has been the major means of urban public transportation. However, water transit also has a big potential to be a major mode of transport in regions with abundant waterways. Urban ferry systems are implemented in many cities with attractive rivers and canals, and they usually have a strong character as tourist attractions rather than contributing to mobility in the region. By reflecting on the situation in which road transport is congested and uncomfortable, interest in urban ferry systems is rising (Kamen and Barry, 2011). Despite their unreliability against weather conditions and high initial and maintenance costs (Kamen and Barry, 2011), ferry systems can be an important part of multi-modal transport networks in urban areas with water due to their capability to solve the hindrance between areas divided by water.

Implementing adequate ferry systems can provide better accessibility in the area and, consequently, promote the use of multi-modal transportation rather than private cars.

Another factor of inefficient transport systems is the discrepancy between the demand and the supply for transport services. In contrast to the homogeneity of the vehicles and the inflexibility of conventional fixed-schedule transport services that are currently provided in the real world, the demand for them drastically varies spatially and temporarily. This variety of demands provokes the inefficiency of the transport system by creating unused capacity of the vehicles because of the over-provision of the service compared to the demand, or insufficient transport supplies force more users to wait longer. The solution to this mismatching of demand and supply has been studied widely by optimising the transport service in terms of the capacity of vehicle units, scheduling of services, or designing a demand-responsive transport (DRT). DRT is a type of transportation service that corresponds to demand in real time (Alonso-González et al., 2018). Unlike the predefined scheduling and capacity, DRT dispatches the vehicles to demand by collecting the demand information in advance. By DRT, the vehicles are assured to have a high occupancy rate, which results in efficient transport services and better accessibility to users. The flexibility in terms of routing and interior space allocation in waterborne vessels contains the potential for the application of ride-hailing services as well as combining passengers and freight transportation. In contrast to cars, buses or trains, which require physical infrastructure to define the routes, waterway transits do not need predefined infrastructure for routing. This enables the agile reaction to pick up requests by ferries. In addition, vessels have larger capacities than ride-hailing services by car and more freedom in capacity allocation. This freedom of interior space distribution allows us to arrange the capacity allocation by corresponding to the proportion of demand for passengers and freight.

1.2. Research Problem

Considering the aspects mentioned above, on-demand ferry systems that correspond to the demand for mobility and logistics have a big potential to be part of the multimodal network in urban environments to achieve higher efficiency in urban mobility and logistics. However, only a few studies have studied the dynamic management of fleets by considering how to handle passengers and parcels simultaneously. By integrating the mobility of people with the transportation of goods, the transport services will achieve higher efficiency, and developing a model framework to optimise the operation of this service will contribute to realising transport provides services to both passengers and parcels. In this thesis, an electric ferry system for heterogeneous on-demand service will be considered, and a model that dynamically determines the operation of vessels will be developed. The impact of combining passenger mobility and logistics will be assessed by comparing the operations of mixed purpose vessels, which allows passengers and parcels to be on board simultaneously, and conventional fixed purpose vessels.

As mentioned before, on-demand services solve the inefficiency of fixed-schedule public transport by removing the discrepancy between the demand and the supply. In order to achieve an efficient urban ferry system for heterogeneous services, some decisions have to be made in operation, such as the matching of the travel requests and the available vessels to satisfy the service and avoid inefficient matching (e.g. matching a vessel far away from the pickup point while another vessel is closer to the pickup point), and the route plan of vessels to serve the matched travel requests. These decisions can be modelled by the well-known pick up and delivery problem and have been studied widely (Berbeglia et al., 2007). While most studies consider road transport modes with small capacity, ferries are largely capacitated in that they can simultaneously serve multiple passenger and parcel requests. The large capacity of vessels is expected to provide more opportunities for merging requests to achieve more efficient transportation.

1.3. Research Scope and Objective

The previous sections described the benefits and the essential aspects of operating an on-demand urban ferry system for heterogeneous services. The realisation of a well-designed ferry system as such requires the decisions mentioned above, which raises the importance of modelling a decision framework to serve the demand in the most efficient way.

Therefore, this thesis aims to develop a model to determine the dispatching and routing of vessels for heterogeneous on-demand service, considering the stochasticity of the demand. Moreover, it aims to investigate the efficiency and service level of the system by comparing the capacity allocation for mixed purpose with the conventional fixed purpose vessel system. This thesis only considers the stochasticity in demand. The demand will be represented by a stochastic occurrence of requests from passengers and parcels throughout the time period, and other uncertainties, such as travel time, will not be considered in this thesis.

1.4. Research Question

The research objective and scope are narrowed down to the following research questions, which will be answered throughout this thesis.

1) **How can the dispatching of vessels be determined so that the efficiency of the waterborne vessels system for heterogeneous on-demand service is maximised, considering the stochasticity of the demand?**

   When a request from a passenger or parcel pops up, the dispatching of vessels will be decided immediately. A model of a dynamic pick up and delivery problem that considers heterogeneous service in a waterborne vessel system will be proposed in this thesis. The stochasticity of the demand for the service will be incorporated to consider the real-life situation where travel/delivery requests randomly pop up by applying a stochastic approach in generating the requests.

2) **To what extent does the mixture of capacity in vessels improve the efficiency and the service level of the waterborne vessels system?**

   The impact of the capacity allocation on the efficiency and the service level of the service will be investigated by comparing the performance of the mixed purpose vessels with that of the conventional fixed purpose vessels. KPIs such as total travel distance, total empty travel distance, and request met ratio will be analysed to assess the transportation performance of each vessel type.

1.5. Outline

The remainder of this thesis is organised as follows. In Chapter 2, the literature review is provided, and the methodology of this thesis is described in Chapter 3. Chapter 4 describes the experiments to apply the proposed method and their results. Then, the results and the limitations are discussed in Chapter 5. Finally, a conclusion is drawn in Chapter 6.

2
Literature Review

In this chapter, a literature review is presented to identify the research gap in the field. This thesis considers the dynamic operation of an on-demand waterborne vessel system for heterogeneous services. This topic is closely related to the well-known pick up and delivery problem (PDP) and one of its variants the share-a-ride problem (SARP) (Li et al., 2014) which integrates passengers and freight transportation. These topics have been studied widely and in different variants.

2.1. Pickup and Delivery Problem

The problem considered in this thesis is a variant of PDP. PDP is an extension of the vehicle routing problem (VRP), which considers the “pickup” as well as the “delivery” of travel requests. In contrast, the VRP only considers the destination since the vehicles depart from the depot. This problem has been popularly studied.

PDP has many variants on the settings, objectives, and solving methods. Berbeglia et al. (2007) classified the static PDP into three categories, “Many-to-many”, “One-to-many-to-one”, and “One-to-one”, which varies by the number of origins and destinations of the commodities. In a “Many-to-many” problem, any node can be an origin or a destination of the commodities. ”One-to-many-to-one” problems are considered when some commodities are transported from the depot to customers while other commodities are transported back to the depot. In contrast, the ”one-to-one” problem gives each commodity a single origin and destination. This type of problem consists of the well-known “Vehicle routing problem with pick up and delivery” (Savelsbergh and Sol, 1995) and “Dial-a-ride problem” (Cordeau and Laporte, 2007). In these problems, the routing and the schedules of vehicles for a number of customers are constructed in the best way for the objective. Cordeau (2006) introduced a mixed integer programming formulation of a Dial-a-ride problem and a solving algorithm called branch-a-cut algorithm. Although the time windows of each travel request are present, this information is known in advance, so the whole routing is constructed by one decision step. These problems are similar to the problem considered in this thesis in a way that a ”one-to-one” problem for each passenger or parcel will be solved. However, the aforementioned static problems require the travel information of each customer in advance to build the route. One of the main aspects of the variants is whether or not the dynamicity is considered. PDP arose as a static problem, in which all the information is known before the routing of vehicles is determined. In this thesis, the travel information from customers would arrive gradually and be unknown beforehand, and thus, the problem must be solved dynamically.

The emergence of DRT services in transportation, such as Uber, led to strong research attention to incorporate the dynamicity in PDP. In the dynamic PDP (DPDP), the request information is gradually revealed over time. Cai et al. (2023) stated the decision types in a DPDP as to wait/leave and accept/reject. When the request information is revealed at a time step, the system decides whether to accept the request or not, and each vehicle needs to decide if to wait at the current node or leave the node and head to the pickup node. They classified the DPDP based on aspects such as the degree of dynamism and the objective function.

The degree of dynamism is measured by the granularity of the time step in which new information is revealed (frequency) and the time between a new request's arrival and the serving of the request (urgency) (Cai et al., 2023). Ulmer et al. (2021) developed a DPDP for restaurant meal deliveries and conducted a case study in Iowa City. The time horizon was set to 7 hours, considering 12:00-19:00 for delivery requests, and an order could arrive every minute. They considered the cooking time and the service time when serving requests. Pan and Liu (2023) developed a model for a dynamic vehicle routing problem for the morning peak or evening peak, considering the uncertainty in demand during these time periods. For this thesis, the time horizon will be set as a typical operation during a weekday (6:00-22:00) with discrete time steps of every minute, and the urgency of requests will differ between passengers and parcels. Passengers' requests should be served with less waiting time, while parcel requests are more lenient on waiting at the origins if they can arrive at the destination before the requirement.

The objective functions can vary by the characteristics of the services and purpose of the study. The objective functions mostly involve either minimising or maximising an indicator, and most of them are related to either efficiency or the costs of transportation services. Sun et al. (2019) set the objective to minimise the total cost of vehicle scheduling, which includes the travel cost, the penalty for not meeting time windows, and the fixed costs for scheduling vehicles in the DPDP. Similarly, Arslan et al. (2019), developed a DPDP with Ad Hoc drivers aiming to minimise the total cost. When it comes to costs, several studies also aimed to maximise the profit of services (Su et al., 2022) (Bertsimas et al., 2019). In terms of the efficiency of the services, Sitek et al. (2021) and Geiser et al. (2020) aimed to minimise the total length of the route for their DPDP in parcel deliveries. Apart from objectives related to costs and efficiency of the services, Aslaksen et al. (2021) proposed and optimised a combined dial-a-ride and fixed schedule ferry service from the user perspective. It assigns the travel demand to either fixed schedule services or an on-demand service so the user utility is maximised. In this thesis, the waterborne vessel system will focus on the efficiency of the service. It will aim to minimise the total travel distance since one of the interests of this thesis is to investigate the improvement of the efficiency of mixed purpose vessels compared to the fixed purpose of vessel usage.

2.2. Combined Passengers-Goods Urban Transport

Collaborative urban transportation is a concept of collaborating urban logistics and urban transportation in order to increase their efficiencies in the urban area and lead to less congestion and less pollution in the urban areas. Collaborative urban transportation can be classified into two types: vertical and horizontal (Cleophas et al., 2019). Vertical collaboration often refers to logistic systems in which different partners/levels of the supply chain, such as suppliers, carriers, and even customers, work together (Martin and Tom, 2016). Vertical collaboration in logistics mostly remains within the logistics sectors and rarely involves urban passenger transport systems. In contrast, horizontal collaboration involves multiple stakeholders at the same level in the transport system (Francesco et al., 2013). It used to remain within the supply chain of goods, but the concept has expanded to involve the urban passenger transport system and collaboration considers sharing the same mode, infrastructure, and information (Cleophas et al., 2019). Collaboration of passenger transport and logistics in a dense urban environment is expected to increase the efficiency of transport and contribute to reducing congestion, air pollution, and emissions in transportation.

Crowd shipping is one of the concepts of combining passenger mobility and urban logistics by outsourcing the delivery of urban freight to the "crowd", mostly urban citizens (Le et al., 2019). The occasional couriers for delivering parcels to the destination by making a small detour from the original travel route and the couriers would receive a small compensation. This system allows freight vehicles to make less travel by only requiring them to travel between distribution centres to small hubs in the neighbourhoods instead of transporting the goods to every door. Therefore, it is expected to reduce traffic congestion and air pollution and increase safety while satisfying the growing demand for last-mile delivery in urban areas (Sina Mohri et al., 2023). Crowd shipping requires interaction between the logistics supplier and occasional couriers, who are not dedicated to transporting goods, and developing a crowd shipping system involves different levels of decisions. Strategic decisions involve determining the transport network so that the potential crowd shippers would effectively be involved in the logistics. Stokkink and Geroliminis (2023) developed a model that locates depots for a crowd shipping system by simulating the daily operational decision of assigning parcels to crowd shippers based on historical parcel delivery records. In addition, tactical and operational decisions such as timetabling, matching of parcels and couriers, are necessary.

The concept of crowd shipping can be extended to allow parcels to be carried by public transport systems. Pimentel and Alvelos (2018), Masson et al. (2017) proposed models which distribute goods in the existing passenger bus system to ensure last-mile delivery in urban environments. The model considers the assignment of freight requests to bus fleets and also the synchronisation of micro-logistic operation which ensures the delivery of freights to customers from bus stops. The multi-purpose vehicles in urban transport will simultaneously enable the mobility of persons and freight. Chen et al. (2022) investigated the impact of implementing multi-purpose vehicles for passengers and freight on the PDP by comparing it with vehicle sets composed of only singular-purpose vehicles. The numerical experiment showed that fewer vehicles are required to serve all passengers and freight requests, thus making urban transport more efficient. Similarly, Hatzenbühler et al. (2024) analysed the effect of different operations of multi-purpose modular vehicles by solving a pickup and delivery problem. The scenario study showed that modular vehicles saved the operation cost significantly, satisfying the demand of passengers and freight, and enabling the consolidation (allowing passengers and freight to be in the same platoon) service, which saved even more cost. Also, the distance of vehicles driving without any passengers/freight was significantly reduced. This result shows the potential for higher efficiency by combining passengers and freight in urban transportation. The collaboration not only remains in road transport but also in the rail network. Behiri et al. (2018) addressed aspects to be considered for integrating urban freight transport into the passenger rail network. In addition, the authors developed a model for scheduling freight rail transport in an existing passenger rail network without distracting passenger demand. Li et al. (2022) introduced a model to obtain the optimal timetable for trains and freight vehicles for last-mile from station hubs. In other words, this model is a capacity matching model that ensures the dynamic passenger demand to be served and assigns parcel demand to the excess capacity of trains by adjusting the trains' schedule so that the total freight transport time is minimised. The numerical results and case study suggest the capability of train services to handle freight without influencing passenger transport, especially during the off-peak hours of passenger demand, and the integration will contribute to more efficient and environmentally friendly urban transport.

In the operational level of decision in crowd shipping, it is necessary to decide which transport requests (demand) to exchange and plan a route to satisfy the requests. In other words, passengers and freight ”share rides”, and several models and approaches have been studied to optimise the sharing. Li et al. (2014) proposed a ”Share-a-Ride Problem” (SARP), which considers taxi service handling both passengers and parcel delivery requests. The authors formulated a mixed-integer linear problem that matches person and parcel requests and taxis to maximise the taxi company's profit. The numerical results showed that the performance of the system has high potential by considering the spatial distribution of requests beforehand and ensuring enough capacity to serve the requests in every region of the network. The emergence of SARP has brought attention to the field and has been studied widely. Li et al. (2016) extended the SARP by considering stochastic travel times and stochastic delivery locations, respectively. The authors developed a two-level stochastic programming model with recourse decisions after the information on travel times or delivery locations was revealed. The computational results suggested the improvement of the profit by considering the stochasticity compared to deterministic solutions. Yu et al. (2023) extended the SARP to which there are multiple depots for the taxis to incorporate the complexity and scattering of pickup and delivery locations for passengers and parcels, and real life situation which taxis departing from multiple depots instead of one centralised depot. the computational study shows an improvement in the objective function values by approximately 15% by setting multiple depots compared to a single depot case. The SARP does not only remain within studies that consider taxi services. Zhan et al. (2023) developed a model which considers a ride-hailing sharing between passengers and parcels by ride-hailing vehicles (RHV) and electric motorcycles(EM).

The case study of performing the model showed a significant improvement on the profit of the platform and drivers, as well as the number of matched requests by allowing parcels to be delivered by not only EMs but also RHV and share a ride with passengers. Electric vehicles have also been considered in SARP. Gao et al. (2024) introduced a stochastic SARP with electric vehicles and developed a reinforcement learning approach. The model decides not only the assignment of requests to vehicles and the routing of vehicles but also the EV charging decisions. The case study showed the capability of EVs as a mode for sharing passengers and parcel transportation and also sharing the capacity of vehicles improved the rewards by up to 10% compared to serving requests individually in the case study.

Table 2.1: Literature overview

Reference	Problem	Mode	Objective	Dynamicity	Solution Method	Case study
(Arslan et al., 2019)	PDP	Ad-hoc drivers	Minimise total cost	✓	Exact	Illustrative example
(Aslaksen et al., 2021)	PDP	Ferry	Maximise user utility	✓	Exact	Kiel, Germany
(Bertsimas et al., 2019)	PDP	Taxis	Maximise total profit	✓	Heuristics	New York, USA
(Cordeau, 2006)	PDP	N/A	Minimise routing cost		Exact	Illustrative example
(Gao et al., 2024)	SARP	Electric vehicles	Maximise net profit	✓	Reinforcement Learning	New York, USA
(Geiser et al., 2020)	PDP	Cars	Minimise total travel	✓	Heuristics	Illustrative example
(Hatzenbühler et al., 2024)	SARP	Modular electric vehicles	Minimise travel cost		Heuristics	Stockholm, Sweden
(Karami et al., 2020)	PDP	Cars	Minimise tardiness, overtime, travel	✓	Heuristics	Illustrative example
(Li et al., 2014)	SARP	Taxis	Maximise profit for taxi company	✓	Heuristics	San-Francisco, USA
(Li et al., 2016)	SARP	Cars	Maximise total profit		Stochastic programming	Illustrative example
(Mitrović-Minić and Laporte, 2004)	PDP	Cars	Minimise total travel	✓	Heuristics	Illustrative example
(Sitek et al., 2021)	PDP	Cars	Minimise total travel	✓	Heuristics	Illustrative example
(Su et al., 2022)	PDP	Heterogeneous vehicles	Maximise carrier profit	✓	Heuristics	Illustrative example
(Sun et al., 2019)	PDP	Cars	Minimise vehicle scheduled cost	✓	Heuristics	Illustrative example
(Ulmer et al., 2021)	PDP	Meal delivery vehicle	Minimise expected cost	✓	Heuristics	Iowa, USA

Table 2.1: Literature overview

Reference	Problem	Mode	Objective	Dynamicity	Solution Method	Case study
(**Vonolfen and Affenzeller, 2016**)	PDP	Cars	Minimise fleet size, travel	✓	Heuristics	Illustrative example
(Yu et al., 2023)	SARP	Taxis	Maximise profit for taxi company		Heuristics	Illustrative example
(Zhan et al., 2023)	SARP	Ride-hailing vehicles, electric motorcycles	Maximise profit, minimise cost	✓	Bi-level mixed integer programming	Chengdu, China
This study	SARP	Electric ferry	Minimise total distance	✓	Exact/heuristic	Fredrikstad, Norway

Exact: Exact method, PDP: Pick up and delivery problem, SARP: Share-a-ride problem

2.3. Summary

The review shows the variety of studies conducted in the PDP. Due to the emergence of on-demand transport services, implementing dynamicity in PDP has caught research interest, and it has been studied in various settings. The dynamicity in the PDP has been implemented by making the travel requests arrive gradually in the model and solving the problem iteratively throughout the time horizon. After the emergence of SARP by Li et al. (2014), combining passengers and parcels in a ride sharing transportation has become a popular problem in the field. This problem has also been studied widely in different settings and by adding aspects such as dynamicity and stochasticity.

However, most of these studies consider ride-sharing systems for road transport when it comes to on-demand services. Combining passengers and parcels in a transport service has the potential to be expanded to different modes. Because of their large capacity, urban ferry systems have the ability to accelerate urban logistics without interfering with passenger transport. In conclusion, the contribution of this thesis can be summarised as follows.

- **Considering a dynamic pick up and delivery problem for a waterborne vessel transport system for heterogeneous service.**
  A model that dynamically determines the dispatching of waterborne vessels to serve requests from passengers and parcels will be developed in this thesis. While most of the previous studies considered road transport such as taxis, waterborne vessels have a larger capacity than them, and therefore, the optimal operations of the fleet are expected to differ from small capacitated road transport.

- **Considering the impact of capacity allocation of waterborne vessels for heterogeneous service.**
  The flexibility of capacity allocation derived from the modularised interior of waterborne vessels brings the possibility of dynamic capacity allocation in a vessel. In this thesis, the impact of capacity allocation on the efficiency and service level of the ferry system will be investigated by comparing the performance of mixed purpose vessels with that of conventional fixed purpose vessels. Mixed purpose vessels are able to serve passengers and parcels simultaneously, while

conventional fixed purpose vessels are only able to serve one of the types of requests. As previous studies suggest in different modes, it is expected that mixed purpose vehicles provide more efficient operation of the fleet than fixed purpose vehicles, and this hypothesis will be assessed for a waterborne vessel system by performing the proposed dispatching model in this thesis given the demand scenarios.

- **Considering the different characteristics for passengers and parcels.**
  The characteristics of the demand from passengers and parcels are different. They differ in aspects such as waiting time and the temporal distribution of demand. These aspects will be modelled to make the problem more realistic.

3 Methodology

In this chapter, the methodology of this research is described. This thesis focuses on developing a model of dynamic centralised fleet management of a waterborne vessel transport system which combines passenger mobility and parcel delivery. Also, the benefits of combining passengers and parcels will be investigated by comparing the transport performance between the cases in which mixed purpose vessels are available and the case of conventional fixed purpose vessels. In the remaining of this chapter, first, the theoretical benefits of the mixed purpose vessel are explained. Secondly, the problem description and the model are presented. Afterwards, the solution methods are presented, and finally, the indicators of the transportation performance are defined.

3.1. Proof of concept

The theoretical benefits of implementing mixed purpose vessels are explained in this section. Figure 3.1 and 3.2 illustrate simple examples of the routing of conventional fixed purpose vessels and mixed purpose vessels, respectively. Each request has a type, either passenger or parcel. In Figure 3.1, each vessel can only serve one of the types of requests, while the vessels in Figure 3.2 are able to handle both types of requests. Both cases decide the routing of vessels to minimise the total distance, but because the mixed purpose vessels can merge passenger and parcel requests in the same vessel, it requires less total travel distance than that of conventional vessels. In the example, the total distance of the conventional vessels is 9.4km and it is 5.8km for the mixed purpose vessels even though they both handle the same set of requests. In addition, the total empty travel distance for the conventional vessels is 0.8+0.5+2.0 = 3.3km, while it is 0.5+0.5 = 1.0km for the mixed purpose vessels. It is expected that the mixed purpose vessels provide better efficiency of operations than conventional fixed purpose vessels while serving the same demand.

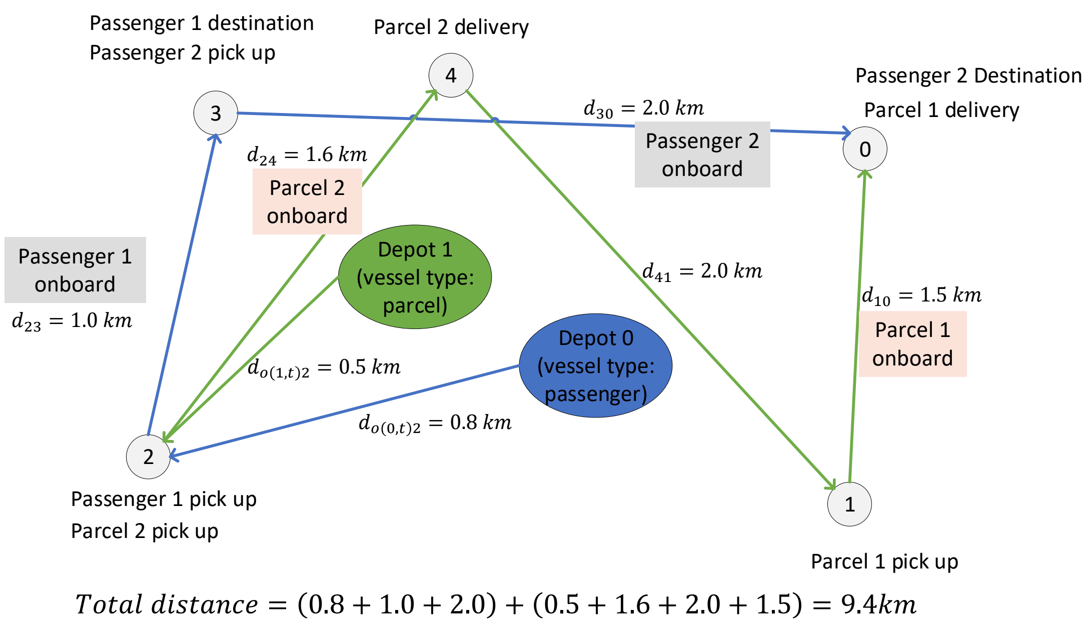


Figure 3.1: Routing solution with conventional vessels

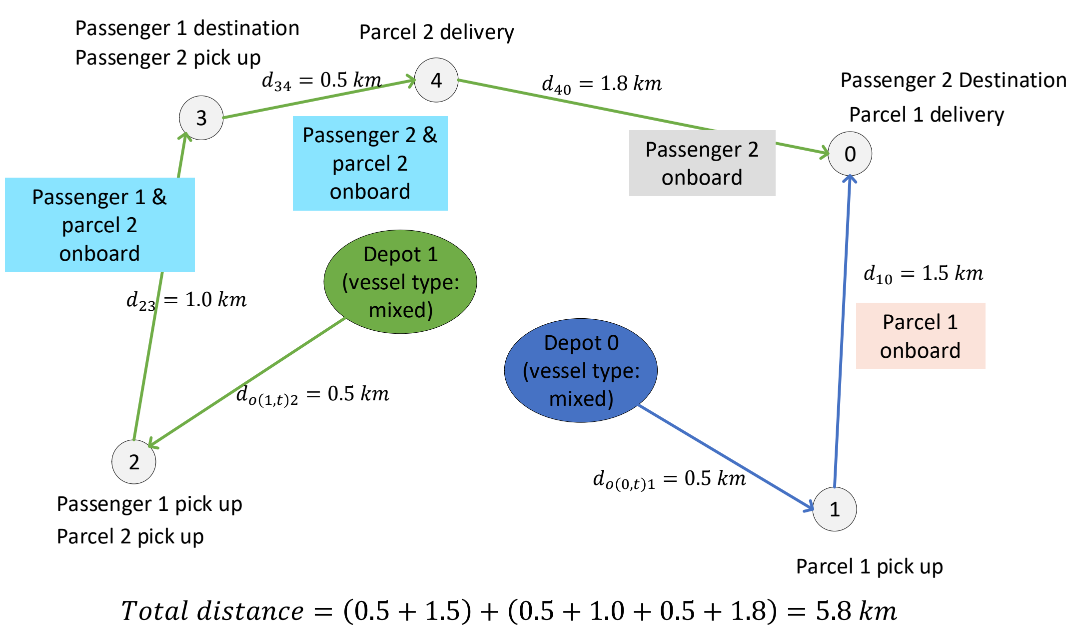


Figure 3.2: Routing solution with mixed purpose vessels

3.2. Problem description

The investigated problem is an extension of the pickup and delivery problem, which considers the problem dynamically and for heterogeneous service. Travel requests will be gradually inserted into the model, and the vessels will update their routing plan immediately when the new request arrives. All the requests in the model are served by dispatching a vessel. A rolling horizon approach is applied to incorporate this dynamicity. Assuming each interval is τ, the time horizon T is divided into p intervals, $TS = \{t_1, t_2, ...t_p\}$, where $p = T/\tau$.

A situation in which a given number of homogeneous vessels serve on-demand travel requests from passengers and parcels will be considered. Each vessel has one of the types of "mixed", "passenger", or "parcel". Mixed vessels are able to serve passenger and parcel requests simultaneously, while passenger and parcel vessels are dedicated to serving the requests that the type matches. The type of vessel is pre-defined. In a series of experiments, different vessel type combinations are experimented to analyse the impact of the mixed purpose vessels compared to the conventional fixed purpose vessels. Each vessel has a homogeneous total capacity and battery capacity. The route of vessels must be made so the loading level on the vessel does not exceed the total capacity and the vessel has enough battery level to complete the route. In addition, it is assumed that the vessel travels at a constant speed. Therefore, it is assumed the travel time and the battery consumption between two nodes are deterministic. Also, the battery consumption is assumed to have a positive linear correlation with the distance between the two nodes.

Requests are inserted gradually into the model. Each request has a pick up node, delivery node, time window, size, and type. The request type is either passenger or parcel. The time window consists of the earliest pickup time, the maximum waiting time, and the latest delivery time. Passenger requests will have a stricter time window than parcel requests because passengers make requests when they want to travel to their destination, while what is important for parcel requests is to be delivered to the delivery node before the required delivery time and the waiting time does not influence the service level. The size and dimensions of the passengers and parcels differ in real life. In order to calculate the occupied spaces in each vessel, the size of passengers and parcels will be converted into the same unit. Specifically, parcels will be translated into passenger units. The problem can be defined as follows:

Let $G(t) = \{N(t), A(t)\}$ be the complete graph, where $N(t) = \{0, 1, ..., n, ..., o(k,t), o'\}$ is the set of nodes at time step $t \in TS$ and $A(t) = \{(i,j), i, j \in N(t), i \neq j\}$ is the set of arcs at time step t. The graph has nodes $\{0, 1, ...n\}$ which are fixed to represent the ferry terminals. For each vessel k in the set of vessels K, it has the current location of vessel k as the depot $o(k,t)$ at time t, as well as the dummy terminal o' as arrival depot so the vessel is able to stay at the final destination of the route plan. $N_c \subset N(t)$ represents the set of nodes which has a charging facility; thus, when a vessel is dwelling at a terminal in this set, it is possible to charge the battery of the vessel.

At time t, the set of all requests $R(t)$ is defined as $R(t) = R_a(t) \cup R_u(t) \cup R_s(t)$, where $R_a(t)$ is the set of assigned requests, $R_u(t)$ is the set of unassigned requests, and $R_s(t)$ is the set of serving requests. When a request is assigned to be served by a vessel, the assignment will be fixed and will not change later. Each request input in the static problem is part of only one of the sets, R_u, R_a, or R_s. Each request r has its type $s(r)$, which is a binary parameter: 1 if the request is a passenger request, and 0 if the request is a parcel request. Also, they have a pickup and delivery terminal $p(r), d(r)$, the earliest pickup time $a(r)$, the latest delivery time $b(r)$, the maximum waiting time $\alpha(r)$, and the size of the request $q(r)$. For assigned and serving requests, the assigned vessels are represented as $k(r)$, and this is applied to formulate the constraints that once a vessel is assigned to a request, it is not allowed to change the assignment of the vessel to the request.

Each vessel has a type pre-defined, mixed, passenger, or parcel. The set of mixed vessels, passenger vessels, and parcel vessels are represented as K_m, K_p, K_f, respectively. The set of all vessels is represented as $K = K_m \cup K_p \cup K_f$. The vessel type limits the type of requests the vessel can serve. Mixed vessels can serve both passenger and parcel requests, while passenger vessels can only serve passenger requests, and parcel vessels can only serve parcel requests. Apart from the type of vessel, each vessel has a homogeneous property regarding the total loading capacity and the battery. The loading capacity of a vessel and the battery capacity are represented as parameters C and B, respectively. In this model, vessels are assumed to travel at a constant speed of v. Because of this assumption, the battery consumption is assumed to have a positive linear relationship with the travel distance of the vessel. δ_{ij} represents the battery consumption between two nodes $i, j \in N(t)$ and is calculated by $r_c d_{ij}$, where d_{ij} is the distance between two nodes and r_c is the consumption rate of battery per distance. At a terminal with a charging facility, vessels always charge their batteries at a constant rate of η. At each time step, the battery levels of vessels are input as the initial battery level of the static problem and are represented as $e(k, t)$ for each vessel k at time t.

The decision variables of this problem are related to the routing of the vessels. x_{ij}^k is a binary variable, 1 if vessel $k \in K$ travel between node $i, j \in N(t)$, otherwise 0. y_{ij}^{kr} is also a binary decision variable which is 1 if vessel $k \in K$ travel between node $i, j \in N(t)$ serving request $r \in R(t)$, otherwise 0. z_{ij}^k is a binary variable 1 if node $i \in N(t)$ precedes (not necessarily immediately) node $j \in N(t)$ in the route of vessel $k \in K$, otherwise 0 and is used for sub-tour elimination.

In addition to the decision variables, variables related to the time domain and the battery level are defined. t_i^k, $t_i^{'k}$, $\bar{t}_i^k$, $t_i^{''k}$ are the arrival time, service start time, service end time, necessary service time at node $i \in N(t)$ by vessel $k \in K$, respectively. Similarly, t_i^{kr}, $t_i^{'kr}$, $\bar{t}_i^{kr}$, $t_i^{''kr}$ are the arrival time, service start time, service end time, necessary service time for request $r \in R(t)$ at node $i \in N(t)$ by vessel $k \in K$, respectively. The battery level of vessel $k \in K$ upon arrival at node $i \in N(t)$ is defined as β_i^k and $\beta_i^{'k}$ on departure. The service time in this problem is representing the time passengers and parcels need to board and get off the vessel.

Table 3.1 is the list of all the notations defined for the model. It is worth noting that the sets are time-dependent, which is because the network graph $G(t)$ changes by including the current locations of vessels as temporal nodes. Also, the requests are possible to transfer between sets or to be newly input, or to be removed after complete serving, so at each time step the sets related to requests are also different. After the notations, the mathematical formulation of the problem is proposed.

Table 3.1: Notations of the model

Notation	Description
Sets	
K_m	Set of mixed purpose vessels
K_p	Set of passenger dedicated vessels
K_f	Set of parcel dedicated vessels
K	Set of all vessels. $K = K_m \cup K_p \cup K_f$
$N(t)$	Set of nodes at time t
N_c	Set of nodes with charging facilities
$A(t)$	Set of arcs at time t
$R(t)$	Set of all requests at time t. $R_a(t) \cup R_u(t) \cup R_s(t)$.
$R_a(t)$	Set of assigned requests at time t
$R_u(t)$	Set of unassigned requests at time t
$R_s(t)$	Set of currently serving requests at time t
$O(t)$	Set of departing depots for vessel $k \in K$ at time t

Parameters

C	Capacity of each vessel
B	Battery capacity of each vessel
η	Charging rate of the battery per time
v	Speed of each vessel
o'	dummy arrival depot
t	Time step in the scheduling horizon
d_{ij}	Travel distance from node $i \in N(t)$ to node $j \in N(t)$
δ_{ij}	Battery consumption when travelling from node $i \in N(t)$ to node $j \in N(t)$
$s(r)$	Binary parameter, 1 if the request is a passenger request, 0 if the request is a parcel request.
$k(r)$	Assigned vessel to request $r \in R_a(t) \cup R_s(t)$
$p(r)$	Pickup terminal of request $r \in R(t)$
$d(r)$	Delivery terminal of request $r \in R(t)$
$a(r)$	The earliest pickup time of request $r \in R(t)$
$b(r)$	The latest delivery time of request $r \in R(t)$
$\alpha(r)$	Maximum waiting time for request $r \in R(t)$
$q(r)$	Size of request $r \in R(t)$
$o(k,t)$	The location of vessel $k \in K$ at time t, which is the departing depot of vessel k.
$e(k,t)$	Battery level of vessel $k \in K$ at time t

Variables

t_i^k	Arrival time at node $i \in N(t)$ of vessel $k \in K$
$t_i^{'k}$	Service start time at node $i \in N(t)$ by vessel $k \in K$
$\bar{t}_i^k$	Service end time at node $i \in N(t)$ by vessel $k \in K$
$t_i^{''k}$	Necessary service time at node $i \in N(t)$ by vessel $k \in K$
t_i^{kr}	Request arrival time at node $i \in N(t)$ for request $r \in R(t)$ by vessel $k \in K$
$t_i^{'kr}$	Service start time for request $r \in R(t)$ at node $i \in N(t)$ by vessel $k \in K$
$\bar{t}_i^{kr}$	Service end time for request $r \in R(t)$ at node $i \in N(t)$ by vessel $k \in K$
$t_i^{''kr}$	Necessary service time for request $r \in R(t)$ at node $i \in N(t)$ by vessel $k \in K$
β_i^k	Battery level of vessel $k \in K$ when it arrived at node $i \in N(t)$
$\beta_i^{'k}$	Battery level of vessel $k \in K$ when it departed from node $i \in N(t)$

Decision variables

x_{ij}^k	Binary variable, 1 if vessel $k \in K$ travel using arc $(i,j) \in A(t)$, otherwise 0.
y_{ij}^{kr}	Binary variable, 1 if vessel $k \in K$ travel using arc $(i,j) \in A(t)$ serving request $r \in R(t)$, otherwise 0.

z_{ij}^k	Binary variable, 1 if node $i \in N(t)$ precedes (not necessarily immediately) terminal $j \in N(t)$ in the route of vehicle k, otherwise 0

3.3. Mathematical model

For each time step, the following mathematical programming model is proposed. The model is built by modifying the study from Zhang et al. (2023).

Objective function

$$min \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} d_{ij} x_{ij}^k \tag{3.1}$$

Subject to:

$$\sum_{j \in N(t)} x_{o(k,t)j}^k = 1 \qquad \forall k \in K \tag{3.2}$$

$$\sum_{j \in N(t)} x_{o(k,t)j}^k = \sum_{j \in N(t)} x_{jo'}^k \qquad \forall k \in K \tag{3.3}$$

$$x_{ij}^k \leq z_{ij}^k \qquad \forall (i,j) \in A(t), k \in K \tag{3.4}$$

$$z_{ij}^k + z_{ji}^k = 1 \qquad \forall (i,j) \in A(t), k \in K \tag{3.5}$$

$$z_{ij}^k + z_{jp}^k + z_{pi}^k \leq 2 \qquad \forall i,j,p \in N(t), k \in K \tag{3.6}$$

$$\sum_{k \in K} \sum_{j \in N(t)} y_{p(r)j}^{kr} = 1 \qquad \forall r \in R_u(t) \tag{3.7}$$

$$\sum_{k \in K} \sum_{i \in N(t)} y_{id(r)}^{kr} = 1 \qquad \forall r \in R_u(t) \tag{3.8}$$

$$s(r) y_{ij}^{kr} = 0 \qquad \forall (i,j) \in A(t), k \in K_f, r \in R(t) \tag{3.9}$$

$$(1 - s(r)) y_{ij}^{kr} = 0 \qquad \forall (i,j) \in A(t), k \in K_p, r \in R(t) \tag{3.10}$$

$$\sum_{i \in N(t)} x_{ip(r)}^{k(r)} = 1 \qquad \forall r \in R_a(t) \tag{3.11}$$

$$\sum_{j \in N(t)} y_{p(r)j}^{k(r)r} = 1 \qquad \forall r \in R_a(t) \tag{3.12}$$

$$\sum_{i \in N(t)} y_{id(r)}^{k(r)r} = 1 \qquad \forall r \in R_a(t) \cup R_s(t) \tag{3.13}$$

$$\sum_{j \in N(t)} y_{o(k,t)j}^{k(r)r} = 1 \qquad \forall r \in R_s(t) \tag{3.14}$$

$$\sum_{i \in N(t)} y_{id(r)}^{kr} - \sum_{j \in N(t)} y_{p(r)j}^{kr} = 0 \qquad \forall k \in K, r \in R_a(t) \cup R_u(t) \tag{3.15}$$

$$\sum_{j \in N(t)} x_{ij}^k - \sum_{j \in N(t)} x_{ji}^k = 0 \qquad \forall k \in K, i \in N(t) \backslash O(t), o' \tag{3.16}$$

$$\sum_{j \in N(t)} y_{ij}^{kr} - \sum_{j \in N(t)} y_{ji}^{kr} = 0 \qquad \forall k \in K, r \in R_a(t) \cup R_u(t), i \in N(t) \backslash p(r), d(r) \tag{3.17}$$

$$\sum_{j\in N(t)} y_{ij}^{kr} - \sum_{j\in N(t)} y_{ji}^{kr} = 0 \qquad \forall k \in K, r \in R_s(t), i \in N(t)\backslash p(r), d(r), O(t) \tag{3.18}$$

$$y_{ij}^{kr} \leq x_{ij}^{k} \qquad \forall (i,j) \in A(t), k \in K, r \in R(t) \tag{3.19}$$

$$t_i^k \leq t_i^{'kr} \qquad \forall i \in N(t), k \in K, r \in R(t) \tag{3.20}$$

$$t_i^{'kr} \leq t_i^{'k} \qquad \forall i \in N(t), k \in K \tag{3.21}$$

$$\bar{t}_i^{kr} \leq \bar{t}_i^{k} \qquad \forall i \in N(t), k \in K, r \in R \tag{3.22}$$

$$t_i^k \leq t_i^{kr} \qquad \forall i \in N(t), k \in K, r \in R(t) \tag{3.23}$$

$$t_i^{''k} = \sum_{r\in R(t)} t_i^{''kr} \qquad \forall i \in N(t), k \in K \tag{3.24}$$

$$\bar{t}_i^k = t_i^{'k} + t_i^{''k} \qquad \forall i \in N(t), k \in K \tag{3.25}$$

$$t_i^{''kr} + t_i^{'kr} \leq \bar{t}_i^{kr} + M(1 - y_{ij}^{kr}) \qquad \forall i,j \in N(t), k \in K, r \in R(t) \tag{3.26}$$

$$a(r)y_{ij}^{kr} \leq t_i^k \qquad \forall (i,j) \in A(t), k \in K, r \in R(t) \tag{3.27}$$

$$\bar{t}_i^{kr} \leq b(r)(y_{ij}^{kr} + M(1 - y_{ij}^{kr})) \qquad \forall (i,j) \in A(t), k \in K, r \in R(t) \tag{3.28}$$

$$\bar{t}_i^k + d_{ij}/v - t_j^k \leq M(1 - x_{ij}^k) \qquad \forall (i,j) \in A(t), k \in K \tag{3.29}$$

$$\bar{t}_i^k + d_{ij}/v - t_j^k \geq M(1 - x_{ij}^k) \qquad \forall (i,j) \in A(t), k \in K \tag{3.30}$$

$$t_{p(r)}^{'kr} - a(r) \leq \alpha(r)y_{p(r)j}^{kr} + M(1 - y_{p(r)j}^{kr}) + M(1 - s(r)) \qquad \forall j \in N(t), k \in K, r \in R_a(t) \cup R_u(t) \tag{3.31}$$

$$\sum_{r\in R(t)} y_{ij}^{kr} q(r) \leq C x_{ij}^k \qquad \forall (i,j) \in A(t), k \in K \tag{3.32}$$

$$\beta_{o(k,t)}^k = e(k,t) \qquad \forall k \in K \tag{3.33}$$

$$\beta_i^{'k} = \beta_i^k \qquad \forall i \in N(t)\backslash N_c, k \in K \tag{3.34}$$

$$\beta_i^{'k} = min(B, b_i^k + \eta(\bar{t}_i^k - t_i^k)) \qquad \forall i \in N_c, k \in K \tag{3.35}$$

$$\beta_i^k \geq 0.2B \qquad \forall i \in N(t), k \in K \tag{3.36}$$

$$\beta_i^{'k} - \delta_{ij} - \beta_j^k \leq M(1 - x_{ij}^k) \qquad \forall (i,j) \in A(t), k \in K \tag{3.37}$$

$$\beta_i^{'k} - \delta_{ij} - \beta_j^k \geq -M(1 - x_{ij}^k) \qquad \forall (i,j) \in A(t), k \in K \tag{3.38}$$

$$x_{ij}^k, y_{ij}^{kr}, z_{ij}^{kr} \in \{0,1\} \qquad \forall (i,j) \in A(t), k \in K, r \in R(t) \tag{3.39}$$

The objective function (3.1) is to minimise the total travel distance of all vessels. The objective of this system is to serve the travel requests in the most efficient way, and minimising the total travel distance indicates the maximum efficiency of the system.

Constraints (3.2) - (3.15) are the typical PDP constraints. Constraints (3.2), (3.3) ensure a vessel will leave the depot and arrive at the end depot. Constraints (3.4) - (3.6) are the subtour elimination constraints. Constraints (3.7) - (3.15) ensure that a request is either only served by one vessel or not served, and if the request is served, it is picked up and delivered by the same vessel. In the case of assigned and serving requests, the vessel is already determined as $k(r)$. Constraints (3.9) and (3.10) restrict the assignment of requests to vessels if the request and vessel type do not match.

The flow conservation constraints are set as well. Constraints (3.16) and (3.17), (3.18) are the flow conservation constraints in terms of the vessel's route and request's route. Constraints (3.19) ensure if a request is being served between two nodes, the vessel travels between the two nodes.

Constraints (3.20) - (3.31) are related to the time constraints. Constraints (3.20) ensure the service start time for a request is after the vessel arrives at the terminal. Constraints (3.21) define the vessel's last service start time. Constraints (3.22) ensure the vessel only departs after all services are completed. Constraints (3.23) maintain the arrival time of a vessel and requests at the terminal. Constraints (3.24) determine the necessary service time for a vessel by summing all requests' necessary service time. Constraints (3.25) determine the service end time of a request by adding the necessary service time to the service start time. Constraints (3.27) and Constraints (3.28) are the time windows of requests. Constraints (3.29) and Constraints (3.30) define the deterministic travel time between two nodes. Constraints (3.31) are the maximum waiting time for a request. For parcel requests, the maximum waiting time constraint is loosened by big M because the waiting time for parcels does not matter as long as the delivery time window is respected in constraint (3.28). This is implemented by the parameter $s(r)$, which specifies the type of request, and $s(r)$ is equal to 0 for parcel requests.

Constraints (3.32) are the capacity constraints. Constraints (3.33) - (3.38) are related to the battery level of the vessels. Constraints (3.33) conserve the initial battery level at each time step. Constraints (3.34) keep the battery level when arriving if the node is not a charging terminal, and constraints (3.35) describe the charging of vessels but never exceed the battery capacity. Constraints (3.36) limit the battery level of vessels to never go under 20% of the battery capacity, and constraints (3.37) and (3.38) describe the deterministic battery consumption between two nodes. Finally, Constraints (3.39) are the domain of the decision variables.

3.4. Dynamic framework

Given the demand from passengers and parcels and the service of the ferries, the dispatching of the fleet will be determined dynamically. The demand is described as a set of individual travel requests. This is because an on-demand service is assumed for both passenger and parcel requests in this thesis. The dynamicity of the model is incorporated by applying a rolling horizon approach. Figure 3.3 shows the flow of the dynamic optimisation. During the rolling horizon, the model will solve a static subproblem explained in Section 3.2 to update the routing and scheduling of the vessels by taking into account the new request when a new request is inserted. The new route plans will be pursued by the vessels until the next time step when a new request appears. In the next time step, the status of vessels, such as their locations, loading levels, and battery levels, will be updated based on the route plan. Also, the request statuses will be updated accordingly. Requests can be failed, serve-complete, serving, assigned, or unassigned. Failed means a request was rejected and will not be served, serve-complete means the request was served, serving indicates the request is currently being served on a vessel, assigned has a vessel assigned and is waiting to be served at the origin, and unassigned means the request does not have a vessel assigned yet. It is assumed that assigned requests cannot be newly assigned to a different vessel once the assignment is determined, despite the fact that there might be some cases in which the reassignment of requests to vessels leads to better efficiency. These updated statuses of the vessels and requests will be inherited in the next time step and will act as constraints to determine the next route plan. This process continues until the time steps reach the end of the time horizon. In this way, this model is able to optimise the full day of the operation of waterborne vessels for heterogeneous services.

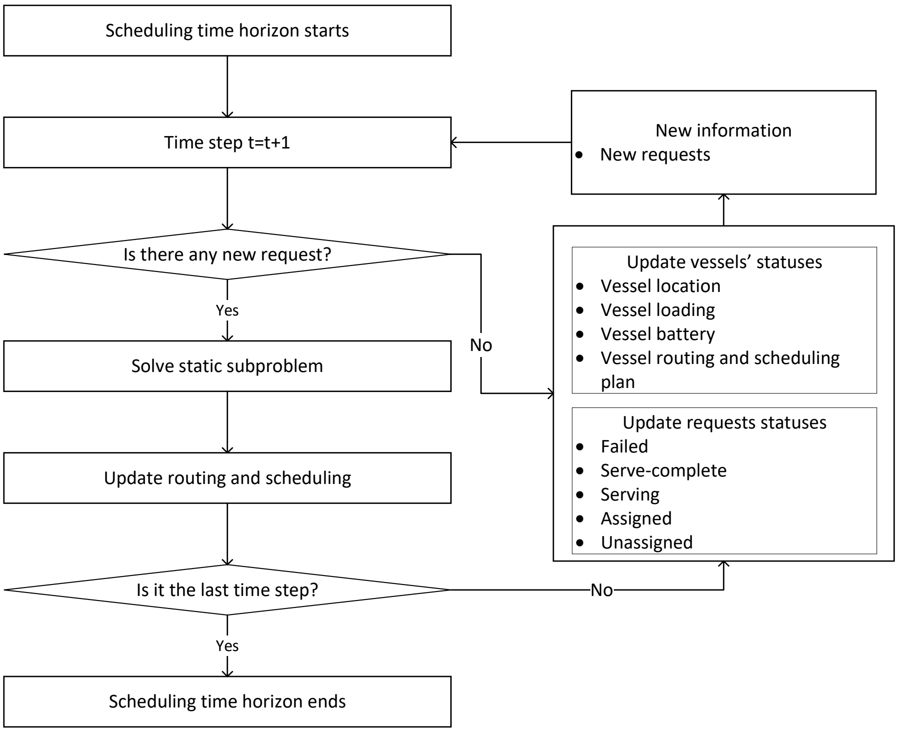


Figure 3.3: Flow chart for the dynamic optimisation

3.4.1. Status updating

Before executing the optimisation of the static subproblem at a time step, the vessels and requests status is updated by pursuing the route plan obtained in the previous time step. The vessels' status consists of their location, loading levels, and battery levels. For convenience, the methods of status updating are explained by using an example. Consider a simple example as shown in Figure 3.4. Let τ be a time step in the time horizon, and vessel k obtains a route plan departing from its current location $o(k, \tau)$. The vessel is at location $(\varphi_\tau^k, \theta_\tau^k)$ at time τ and node 0 and node 1 has locations of (φ^0, θ^0) and (φ^1, θ^1), respectively. Two requests r_0 and r_1 will be served by this vessel. Request r_0 is already on board at time τ and must be delivered to node 1. Request r_1 is assigned to the vessel and is waiting to be picked up at node 0 and to be delivered to node 1 by the vessel. The vessel departs the current location immediately with a battery level of $e(k, \tau)$ at time τ. It arrives at node 0 by travelling at a constant speed so the arrival time at node 0 is $t_0^k = \tau + d_{o(k,\tau)0}/v$, where $d_{o(k,\tau)0}$ is the distance between the current location and node 0. On arrival, the vessel has a battery level of β_0^k. At node 0, request r_1 boards on the vessel so it requires a service time $t_0^{''k}$ to complete the boarding. Since node 0 has a charging facility, the battery of the vessel is being charged at a constant rate η while dwelling at node 0. In the next trip between node 0 and node 1, both requests are on board. Similar to the first trip, the arrival time to node 1 is the summation of the departure time from node 0 and the deterministic travel time between node 0 and node 1. Battery consumption is also assumed to be deterministic and has a positive linear correlation with travel distance. Therefore, the battery level upon arrival to node 1 is $\beta_1^k = \beta_0^{'k} - r_c d_{01}$, where r_c is the consumption rate of battery per kilometre.

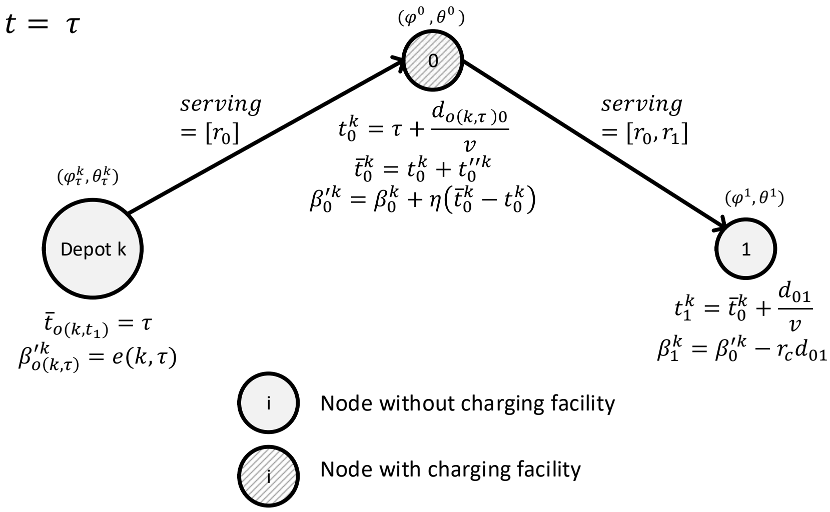


Figure 3.4: Example of route plan at time τ

When a new request pops up at time $\tau^{'}(> \tau)$, the vessel and request statuses are updated before deciding the new route including the new request based on the assumption that the vessel is pursuing the route in Figure 3.4 until $\tau^{'}$. There are three possibilities to consider when updating the statuses.

Vessel is in the middle of a trip

The first case is when the vessel is in the middle of a trip, such as the trip between depot and node 0 or from node 0 to node 1. In this case, the location of the vessel at time $\tau^{'}$ is obtained by linear interpolation between the origin and the destination of the trip. For instance, when $\tau < \tau^{'} < t_0^k$, the origin and the destination of the trip is $(\varphi_\tau^k, \theta_\tau^k)$ and (φ^0, θ^0). Figure 3.5 shows the example of when a vessel is in the middle of a trip, using the same example from Figure 3.4. Therefore, the location of the vessel at time $\tau^{'}$, $(\varphi_{\tau^{'}}^k, \theta_{\tau^{'}}^k)$ is calculated as,

$$\varphi_{\tau^{'}}^k = \varphi^0 + \frac{\tau^{'} - \tau}{t_0^k - \tau}(\varphi^0 - \varphi_\tau^k)$$
$$\theta_{\tau^{'}}^k = \theta^0 + \frac{\tau^{'} - \tau}{t_0^k - \tau}(\theta^0 - \theta_\tau^k)$$

The battery level $e(k, \tau^{'})$ is calculated by obtaining the distance between the origin and the current location. By taking the same trip as an example, the distance between the origin and the current location represented as $d_{o(k,\tau^{'})o(k,\tau)}$ is applied to obtain the battery level $e(k, \tau^{'})$,

$$e(k, \tau^{'}) = e(k, \tau) - r_c d_{o(k,\tau^k)o(k,\tau)}$$

The same approach for location and battery level is applied for the trip between node 0 and node 1.

In terms of the request, when the request is on board during the trip, its status is ”serving”. Other requests are either ”serve-complete” if the delivery is completed before the trip or ”assigned” if the request is still waiting at its pick up node.

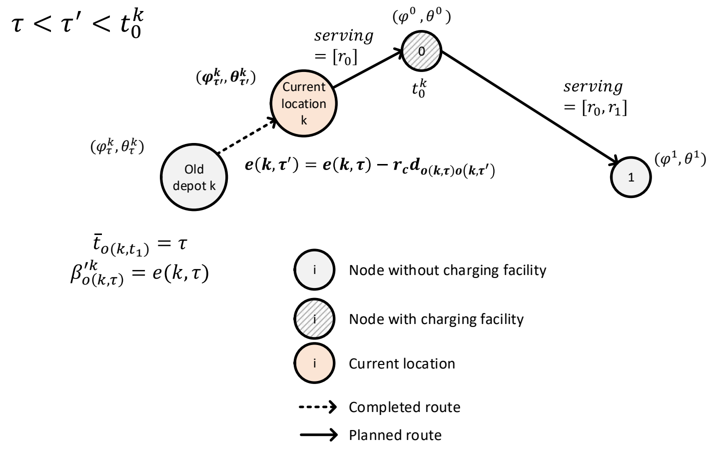


Figure 3.5: Example of when a vessel is in the middle of a trip at time $\tau^{'}$

Vessel is dwelling at a node

The second case is when the vessel is dwelling at a node and still has trips to complete. If $t_0^k \leq \tau^{'} \leq \bar{t}_0^k$, the vessel is dwelling at node 0 to pick up request r_1 (Figure 3.6). In this case, the location of the vessel is the same as node 0, $(\varphi^k_{\tau^{'}}, \theta^k_{\tau^{'}}) = (\varphi^0, \theta^0)$. The battery level depends on whether the dwelling node has a charging facility. If the node does not have a charging facility, the battery level remains the same as when the vessel arrived at the node. If the node has a charging facility, as in node 0 in the example, the battery is charged at a constant rate η in time until the battery level reaches the battery capacity. The battery level at time $\tau^{'}$ is calculated as follows in this case.

$$e(k, \tau^{'}) = min(B, \beta_0^k + \eta(\tau^{'} - t_0^k))$$

Request r_0 is still in "serving" and request r_1 is in "assigned" when the vessel is dwelling at node 0.

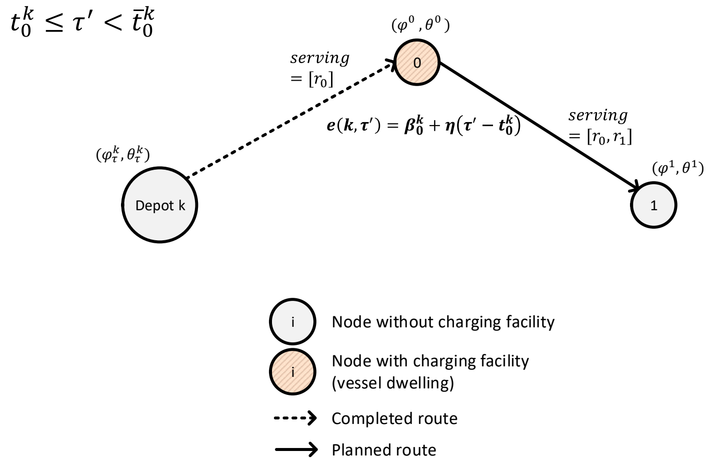


Figure 3.6: Example of when a vessel is dwelling at a node at time $\tau^{'}$

Vessel completed the route

The last case is when the vessel has completed the route. Figure 3.7 shows the example of route completion. When $\tau' \geq t_1^k$, the vessel has completed the route plan obtained in time τ, and is staying at the final destination of the route, which is node 1. Therefore, the location of the vessel $(\varphi^k_{\tau'}, \theta^k_{\tau'}) = (\varphi^1, \theta^1)$. In the example, the battery level of the vessel at time τ' remains the same as when it arrived at node 1 since node 1 does not have a charging facility. Therefore, $e(k, \tau') = \beta_1^k$. In case the final destination has a charging facility, the battery is assumed to be charged every time a vessel is dwelling at the node so the battery will be charged as explained in the case of dwelling at node 0. As the entire route plan for the vessel is pursued in this case, both requests are in ”serve-complete” status.

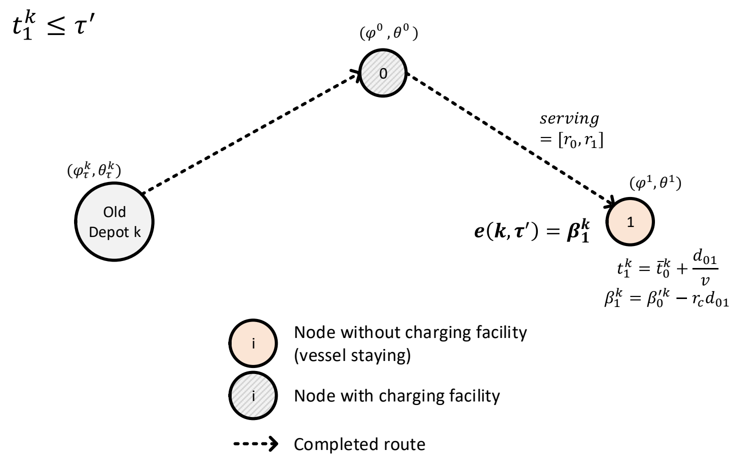


Figure 3.7: Example of when a vessel has already completed the route at time τ'

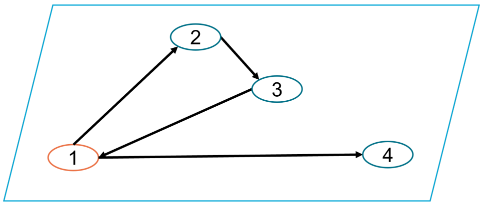
3.5. Solving methods

The formulated optimisation problem will be solved in two approaches in this thesis: the exact method using Gurobi and the insertion heuristics. The exact method is a method which tries to find the global optimum every time a problem is solved. However, it requires a lot of computational resources and time to find the global optimal solution since the problem is a combinatorial problem. In contrast, heuristics are methods that provide solutions with good quality in a short time but are not guaranteed to be the global optimum. Some techniques suitable for the problem are implemented to find a good solution in a short time to prevent heavy computation time, as the exact method requires for complex problems. In this thesis, a simple insertion heuristic is developed to determine the route plans of vessels.

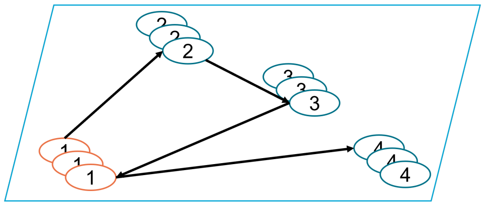
3.5.1. Exact method

Exact methods in optimisation are those which ensure that they give the optimal solutions that are proven to be global optimum for a problem. Gurobi Optimiser is a commercial solver that uses mathematical optimisation to find the optimal solutions to problems. Gurobi provides a package in Python, and this package is used for the exact method for the problem in this thesis.

In principle, a node can only be visited once. However, it is possible to encounter a situation in which a vessel needs to visit a particular terminal multiple times, especially in the patterns in which the vessel's purpose is fixed to one service. Figure 3.8a illustrates an example in which a vessel requires multiple visits to the same terminal. A vessel departs Terminal 1 to deliver a request to Terminal 2, and afterwards, it picks up two other requests from Terminal 3, which is requesting Terminal 1 and Terminal 4 as the destinations, respectively. In this case, the vessel needs to visit Terminal 1 again to deliver the new request. These situations cannot be handled by a MIP with only one node per terminal. In order to allow multiple visits to a terminal, the network nodes can be duplicated in the model (see Figure 3.8b) (Koyuncu and Yavuz, 2019). In this way, the vessel is able to visit the same terminal by visiting the duplicated nodes, which have the same attribute as the original node but different unique IDs. This approach still maintains a node to be visited only once, while in reality, the terminal can be visited multiple times through different nodes. It is important to note that the duplication of nodes amplifies the network size. For instance, if there are 4 nodes in the original network and then allow 3 visits, the network consists of 12 nodes. It is expected that this expansion of the network requires more computational time to obtain the solution. The model duplicates the nodes three times in the model.



(a) Example of a terminal needs multiple visits



(b) Example of the duplicated nodes

Figure 3.8: Example of multiple visits to a node and duplication of nodes

In a dynamic optimisation, it is important to react to the changes in the situation and provide routing plans to the fleet in a quick manner so the fleet is quickly able to adjust its operations. Therefore, in dynamic optimisation, the solving time of each time step is limited to a short time. Since the problem is a complex combinatorial problem, it is possible that the solver encounters situations in which it cannot find a feasible solution within the time limit. In this case, the model will keep the previous route plan, and the newly inserted requests' assignment will be postponed to the next time step until the requests become infeasible. The vessels are assumed to keep pursuing the previous route plan until the next step when the solver manages to find a feasible solution.

3.5.2. Insertion heuristic

The proposed linear programming is a variant of PDP and is widely studied. It is known that the complexity of these combinatorial optimisation problems is proven to be NP-hard (Savelsbergh and Sol, 1995) and requires computational resources to solve large instances. Therefore, solving algorithms and heuristics/metaheuristics for similar problems have been widely studied. Insertion heuristic is a popular method for solving routing problems. Insertion heuristics considers the routes of vehicles as a sequence of nodes, such as $(0, 1, 2, 3)$. This sequence indicates the order of the nodes to visit. For instance, in this case, node 0 is visited first, and then node 1 and node 2 are visited sequentially, and the route ends at node 3. Given the route of a vehicle and the set of new requests with the origin and the destination, it is possible to construct the next route plan by inserting the origin and destination nodes in the current sequence of nodes. By taking the example from before for the current route and assuming a new request with the origin and the destination of $(4, 5)$, node 4 and node 5 can be inserted in the sequence, such as $(0, 1, 4, 2, 5, 3)$. It is important to note that the origin node must always be positioned before the destination node. There are several algorithms that effectively insert the nodes to obtain a reasonable route plan. Greedy insertion is a simple algorithm that tries all possible solutions for insertion (Ghilas et al., 2016). It investigates all possible combinations of insertion, and therefore, it is more computationally expensive than other insertion methods such as random insertion, nearest insertion, or cheapest insertion (Zhang et al., 2023). However, since greedy insertion provides all possible combinations of the sequence, it is effective to find a feasible solution for a problem in a reasonable computation time, depending on the size of the problem.

In this thesis, the greedy insertion heuristic will be applied to accelerate the optimisation of the vessel operation at each time step of the rolling horizon. A greedy insertion is applied for several reasons. The first reason is its exploration ability. Greedy insertion investigates all the possible insertions in the current route. Therefore, there is no risk of not finding a better sequence regarding the objective function. Since the objective function of the problem is to minimise the total travel distance, the greedy insertion will find the best insertion position of the nodes so the total travel distance is minimised. The second reason is the size of the problem. This thesis considers a dynamic pick up and delivery problem for heterogeneous service in a rolling horizon manner. It determines the routing and scheduling of the vessels every time (a) new request(s) popup. Therefore, at every time step, the amount of travel requests inserted into the system is expected to be smaller than the traditional static pick up and delivery problem. In a traditional static pick up and delivery problem, it is expected to provide a good feasible solution in a short time for large instances since all demand is known a-priori. Therefore, more efficient insertion algorithms such as nearest insertion and Adaptive Large Neighbourhood Searching (Ghilas et al., 2016) have been studied in different cases. However, the problem in this thesis is less likely to encounter large instances, and therefore, it is possible to obtain a good, feasible solution by greedy insertion in a short computational time.

The insertion heuristic is developed based on the following assumptions.

- Vessels serve the requests as soon as possible. It does not wait to serve a request, even if there is a margin until the delivery time window.
- Related to the above assumption, vessels do not charge their batteries longer if they are already above 20% of the capacity and will pursue the route. It does not wait to serve a request in order to charge the battery.
- Once a request is assigned to a vessel, the assignment of the vessel does not change.

These assumptions allow the scheduling of visits to each node to be simplified and to be easier to assess the feasibility of the route. Based on the sequence of node visits, the route plan can be constructed, including the time domain, the battery level, the loading of vessels, and the serving requests.

The insertion algorithm for the dynamic pickup and delivery problem solved in a rolling horizon is illustrated in Algorithm 1. When a new request pops up while rolling the time horizon, the algorithm creates a new sequence and its associated route plan for each vessel trying to serve the new requests. Firstly, the locations of the vessels and the request status are updated based on the assumption that the vessels are pursuing the previous route plan. The locations are updated in the way mentioned in Section 3.4.1. Also, the requests in the previous route plan can have different statuses from when the previous route plan was generated. The previous requests can be in one of the statuses assigned, serving, or serve-complete. The next step is to limit the positions of insertion of the new requests' origin and destination nodes in the sequence.

Each new request has a pickup and delivery time window, as well as a request type of either passenger or parcel. This information is applied to limit the position of insertion in the previous route plan to make at least the new requests feasible. Regarding the type of request, the request can only be served by either a mixed purpose vessel or a fixed purpose vessel with the same request type assigned. When a fixed purpose vessel is present, requests that do not match the type with the vessel type cannot be served by the vessel; thus, the insertion of the origin and destination nodes in the sequence of this vessel is prohibited. This constraint does not apply to mixed purpose vessels.

The route plan contains the time information about when the vessel arrives and departs each node of the route. This information provides the minimum requirement of by which position in the sequence the request has to be picked up and delivered. For instance, consider an example of a vessel with a sequence of $(0, 1, 2, 3)$ with a route plan such as in Table 3.2. Assume that two requests r_0 and r_1, which r_0 is at serving status and r_1 is at assigned status. Let a new request r_2 with pick up node and a delivery node of $(p(r_2), d(r_2)) = (4, 5)$ and a pickup time window of $25 \leq a(r_2) \leq 40$, and a delivery time window of $b(r_2) \leq 65$. In this case, the request has to at least be picked up before the vessel visits node 2 and has to be delivered before the vessel visits node 3. Therefore, the origin node 4 has to be inserted before node 2 appears in the sequence, and node 4 has to be inserted before node 3 and after node 4. As a result, the possible new sequences are $(0, 4, 5, 1, 2, 3)$, $(0, 4, 1, 5, 2, 3)$, $(0, 4, 1, 2, 5, 3)$,

Table 3.2: Example of the route plan of a vessel

Trip	Origin	Destination	Departure time	Arrival time	Serving requests
Trip 1	node 0	node 1	25	30	$[r_0]$
Trip 2	node 1	node 2	35	50	$[r_0, r_1]$
Trip 3	node 2	node 3	60	70	$[r_1]$

$(0, 1, 4, 5, 2, 3)$, $(0, 1, 4, 2, 5, 3)$. This information enables us to identify the maximum index of the insertion position of the new requests' origins and destinations into the current sequence. These constraints are generated for each new request before the model generates all possible sequences with the new nodes inserted. After the constraints are generated, all the possible sequences and the corresponding route plans are generated. At the same time, the objective value, which is the total travel distance, is calculated by simply adding the distances between the two nodes appearing in the sequence sequentially. The set of possible sequences and routes are sorted by the objective value for the next step to check the feasibility of the routes. Although the time feasibility of the new requests is secured in the constraints by the insertion positions, the new route plan is generated based on the aforementioned assumptions and it is necessary to check the feasibility of the new routes considering all the requests being served by the vessel, as well as the loading and battery capacity of the vessel. The constraints are the same as what is described in Section 3.3. The loading of a vessel must not exceed the capacity of the vessel, the battery level of the vessel must not become below 20% of the battery capacity, and the time windows of all requests must be respected. If one or more than one of the constraints are violated, the route becomes infeasible and will not be kept. The feasibility check starts from the route plan with the lowest objective value, and it continues until a feasible route plan is found. This route plan will be the next route plan for the vessels and will be pursued until the next time step a new request pops up. This process continues until the time horizon is completed.

Algorithm 1 Insertion algorithm with new requests

```
for t in TS do
    if new request arrives then
        Input: new_requests, route_plan

        // Update current location of vessels and requests' states
        current_locations, requests_status ← update_status(previous_route_plan)

        // Limit insertion positions of new requests
        feasible_positions ← []
        for r in new_requests do
            limit_r ← find_limit_position(r)
            feasible_positions.append(limit_r)
        end for

        // Create route plans for possible insertions and calculate the objective value
        possible_routes ← generate_possible_routes(feasible_positions)
        objective_values ← calculate_objective_values(possible_routes)

        // Check feasibility of routes and remove infeasible ones
        feasible_routes ← []
        for γ in possible_routes do
            if capacity(γ) and time(γ) and battery(γ) then
                feasible_routes.append(γ)
            end if
        end for

        // Retrieve the route plan with the best objective value
        best_route_plan ← get_best_route(feasible_routes, obj)

        // Update the route plan
        route_plan ← best_route_plan
    end if

    Go to the next time step
end for
```

Algorithm 2 and Algorithm 3 shows the flow of the route plan construction. The route plan construction takes a sequence, requests' information, and the tracking of requests and vessels as input. The tracking keeps track of the assignment of requests to vessels. The first step is to identify in which trips the requests are being served. Each request has an origin and destination node, and the sequence should contain these nodes appearing in order. The first time the origin node appears in the sequence is when the request boards, and the first time the destination node appears after the appearance of the origin node is when the request is getting off the vessel. The indices which the request boards and gets off the vessel are identified in Algorithm 2. It searches for the first index which the value matches with the origin node and the destination node for each request. The service for unassigned and assigned requests has not started at the time step, based on the definition of request status. Therefore, Both the boarding and getting off indices for these requests are searched in this algorithm (line 3-17). However, the serving requests are already on board when in this time step, which means only the getting off index for each serving request is necessary to search (line 18-27).

The boarding information obtained in Algorithm 2 is utilised to construct the new route plan. Based on the boarding information, it is possible to identify which requests are involved in each trip (boarding, on board, getting off). A trip is built between two consecutive nodes in the sequence. Each trip has a departure time from the origin, arrival time at the destination, the battery level upon departure and arrival, the serving requests, and the loading level. At time tt, the new sequence for a vessel is fixed to start from the current location and will depart the current location at time tt. The travel time between the current location and the next terminal is calculated from the distance between the two nodes and is deterministic. In the case in which the current location is the same as one of the terminals, which indicates the vessel is dwelling at the terminal, a trip from the current location node and the terminal node is generated with a travel time of zero. It is modelled in this way to satisfy the constraint that the vessels must leave the current location.

Between the arrival at a terminal and the departure from a terminal, requests originating from the terminal must board. Therefore, the service time of the requests must be considered before the next trip. The service time for each request is assumed to be dependent on the request size, and the necessary service time for a vessel at a terminal is assumed to be the sum of the service time of individual requests boarding or getting off at the terminal. While a vessel is dwelling at a terminal, the battery will be charged if the terminal has a charging facility. When the terminal does not have a charging facility, the battery level upon departure will be the same as when the vessel arrives at the terminal. However, if the terminal has a charging facility, the battery would be charged at a constant rate for the duration of the vessel's dwell at the terminal. As mentioned before, it is assumed that the vessel will leave the terminal as early as possible. Therefore, the vessels will not dwell and charge their battery for more than necessary to pursue the route plans. Also, the battery level will never be above its capacity B. The battery consumption between two nodes are assumed to be deterministic and has a positive correlation with the distance between the two nodes, which is represented as δ_{ij}, when the vessel travels from node i to node j.

Since the boarding information is available, it is known in which trips each request is served. Therefore, the total loading as well as the loading per request type are calculated by summing the size of requests on the trips. The loading information will be applied to assess the feasibility of the route later, as explained in Algorithm 1.

Algorithm 2 Boarding position identification for insertion heuristic

Input $seq, R_s, R_a, R_u, track$
Initialise the boarding position in the sequence $board_r \leftarrow (0, -1) \forall r \in R = R_s \cup R_a \cup R_u$
for all $r \in R_a \cup R_u$ **do**
Assigned vessel to each request $k(r) = track(r)$
Pick up and delivery node $(p(r), d(r))$
// Find the indices of boarding and getting off
for $ind \in range(length(seq[k(r)]))$ **do**
if $seq[(k(r)][ind] = p(r)$ **then**
$board_r[0] = ind$
end if
end for
for $ind \in range(board_r[0], len(seq[k(r))$ **do**
if $seq[(k(r)][ind] = d(r)$ **then**
$board_r[1] = ind$
end if
end for
end for
for all $r \in R_s$ **do**
Assigned vessel to each request $k(r) = track(r)$
delivery node $d(r)$
// Find the index of getting off
for $ind \in length(seq[k(r)])$ **do**
if $seq[(k(r)][ind] = d(r)$ **then**
$board_r[1] = ind$
end if
end for
end for

Algorithm 3 Route plan construction for insertion heuristic

function ConstructRoutes($n_k, seq, board, dict_{rk}, R_s, R_u, R_a, tt, b_{temp}, Traveltime, charge\_loc, B$)
    Routes $\leftarrow$ [0 **for** _ **in** **range**(n_k)]
    for all $k \in n_k$ **do**
        route $\leftarrow$ zeros((length(seq[k])-1, 8), dtype=object)
        $R \leftarrow$ [r **in** $R_a \cup R_s \cup R_u$ **if** $dict_{rk}[r] = k$]
        for all $i \in range(length(seq[k]) - 1)$ **do**
            serving_requests $\leftarrow$ [r **for** r **in** r_serving **if** $dict_{rk}[r] = k$]
            $origin \leftarrow seq[k][i]$
            $destination \leftarrow seq[k][i+1]$
            $travel\_time \leftarrow Traveltime[seq[k][i], seq[k][i+1]]$
            service_time $\leftarrow$ 0
            for $r \in R$ **do**
                if i $\geq board_r[0]$ **and** i < $board_r[1]$ **then**
                    if r **not in** serving_requests **then**
                        Append r to serving_requests
                        service_time += $service\_time_r$
                    end if
                end if
                if i $\geq board[r][1]$ **then**
                    if r **in** serving_requests **then**
                        Remove r from serving_requests
                        service_time += $service\_time_r$
                    end if
                end if
            end for

            // If it is the first trip
            if i = 0 **then**
                $dep\_time[i] \leftarrow tt$
                $arr\_time[i+1] \leftarrow dep\_time[i] + travel\_time$
                $b_d^{seq[k][i]}$ $\leftarrow$ b_temp[k]
                $b^{seq[k][i+1]}$ $\leftarrow$ b_temp[k] - $\delta_{seq[i]seq[i+1]}$
                $loading\_all \leftarrow$ sum([$size_r$ **for** r **in** serving_requests])
                $loading\_parc \leftarrow$ sum([$size_r$ **for** r **in** serving_requests **if** $type_r$ is parcel])
                $loading\_pass \leftarrow$ sum([$size_r$ **for** r **in** serving_requests **if** $type_r$ is passenger])
                // Other trips
            else
                $dep\_time[i] \leftarrow arr\_time[i-1] + service\_time$
                $arr\_time[i+1] \leftarrow dep\_time[i] + travel\_time$
                if $seq[k][i]$ **in** charge_loc **then**
                    $b_d^{seq[k][i]}$ $\leftarrow$ min($b^{seq[k][i-1]}$ + r_c (route[i, 2] - route[i-1, 3]), B)
                else
                    $b_d^{seq[k][i]} \leftarrow b^{seq[k][i-1]}$
                end if
                $b^{seq[k][i+1]} \leftarrow b_d^{seq[k][i]}$ - $\delta_{seq[k][i]seq[k][i+1]}$
                $loading\_all \leftarrow$ sum([$size_r$ **for** r **in** serving_requests])
                $loading\_parc \leftarrow$ sum([$size_r$ **for** r **in** serving_requests **if** $type_r$ is parcel])
                $loading\_pass \leftarrow$ sum([$size_r$ **for** r **in** serving_requests **if** $type_r$ is passenger])
            end if
        end for
        Routes[k] $\leftarrow$ route
    end for
    return Routes
end function

3.6. Transportation performance indicators

One of the objectives of this thesis is to investigate the efficiency and the service level of waterborne vessels for heterogeneous on-demand service by comparing the mixed purpose vessel system with the conventional fixed purpose vessel system. The investigation is conducted through the analysis of the following key performance indicators (KPIs).

- *Total travel distance (TTD).* The total travel is the distance travelled by all the vessels throughout the day. It is computed by tracking the travelled distance of each vessel between each time step of the rolling horizon. The lower this value is, the higher the efficiency of the system.
- *Total empty travel distance (TETD).* The total empty travel distance is the distance travelled by all the vessels throughout the day without loading any requests. It is computed by tracking the travelled distance of each vessel between each time step and the loading level of the trip. If the loading level is zero, the travelled distance is added to the empty travel distance. The lower the value is, the higher the efficiency of the system.
- *Request met ratio (RMR).* The request met ratio is the ratio between the number of served requests and the total number of requests at the end of the time horizon. It indicates to what extent the requests are served by the service. The higher the value is, the higher the service level is. RMR is defined as,

$$\frac{(number\ of\ served\ requests)}{(total\ number\ of\ requests)} \times 100[\%]$$

4 Model Application

In this chapter, the model proposed in Chapter 3 is performed by taking Fredrikstad in Norway as the case location. The computational experiment in a specific case location is conducted for the following reasons: 1) The proposed model has to be verified, and the performance of the model and solving algorithms must be evaluated. 2) Insights on the efficiency and the service level of the heterogeneous on-demand service by dynamic centralised fleet management of electric waterborne vessels can be obtained through the series of experiments in the case study. The computational experiment starts from the evaluation of the performance of solving the static subproblem proposed in Section 3.3 for each algorithm, the exact method, and the insertion heuristic. Afterwards, the dynamic problem is solved for different demand scenarios which are developed in a stochastic approach, and the performance of the solving method, as well as the results of the transportation performance will be provided.

4.1. Case study

Fredrikstad is selected as the case location for the experiment. Fredrikstad is a city located along the southeast coast of Norway, with a population of approximately 84,000 (Frederikstad Kommune, 2022). Fredrikstad has an urban ferry system connecting the old town and the city centre, and new water metro stops using shared electric ferries are being developed in the new city centre area. Figure 4.1 shows the location of Fredrikstad (left) and its zoomed-in geography (right). The mobility of Fredrikstad has been highly dependent on private cars with a 59% mode share, despite the abundant waterways the city has ("Fredrikstad - SUM", 2024), and the municipality is focused on reducing the dependency on car trips in order to stop the growth of climate emissions. The considered network consists of 8 ferry terminals and is illustrated in Figure 4.2. The terminals have already been used for the conventional ferry service provided by the municipality of Fredrikstad, and these terminals are considered in this case study as well. The terminals are labelled with a unique ID from 0-7.

Hyke ("Hyke", 2021) has been developing a ferry solution which is electrified and also has the margin to be fully autonomous. Hyke is in collaboration with the municipality of Fredrikstad to transform urban mobility and transportation to be more sustainable and accessible to people by utilising the ferry solution they developed and providing mobility and logistic service in the urban waterway of Fredrikstad. The ferry solution of Hyke is shown in Figure 4.3. The modularised interior in the ferry solution enables different allocations of capacity between passengers and goods; thus, it is possible to provide service to passengers and parcels simultaneously. The ferry is designed to be emission-free by full electrification and also to reduce the noise when operating. Not only does the vessel property contribute to more environmentally friendly transportation, but this innovation in the waterway transportation in the city is also expected to stimulate the modal shift of the citizens from private cars to multimodal transportation by providing more comfortable and accessible services.

Based on the mentioned aspects, Fredrikstad is suitable for the case location to apply the proposed model and investigate the potential of electric on-demand waterborne vessels for heterogeneous services in the urban transportation environment.

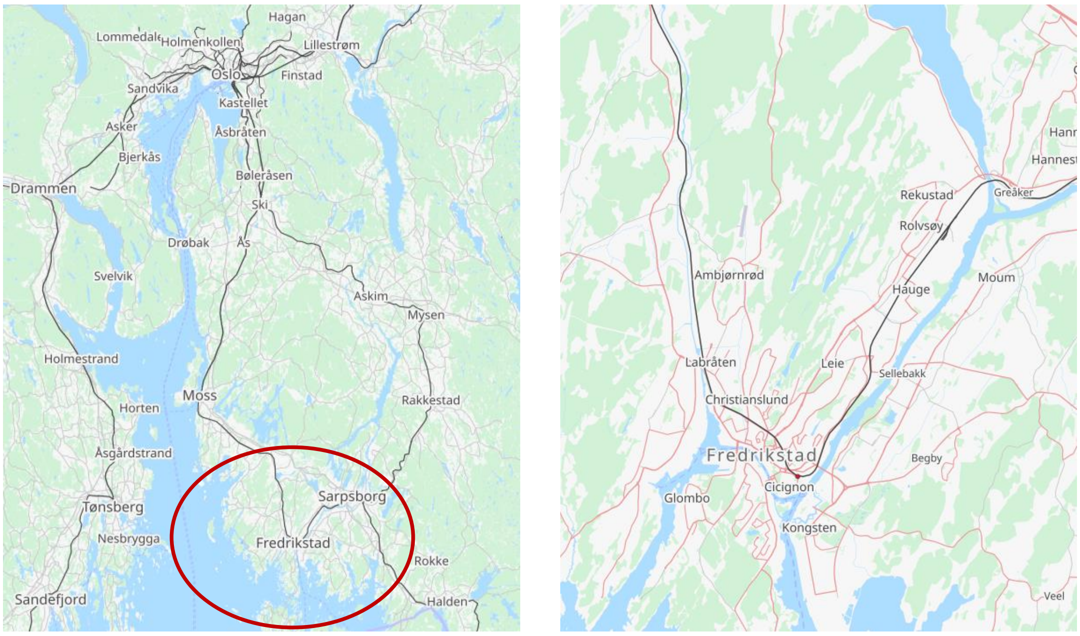


Figure 4.1: Location (left) and geography (right) of Fredrikstad

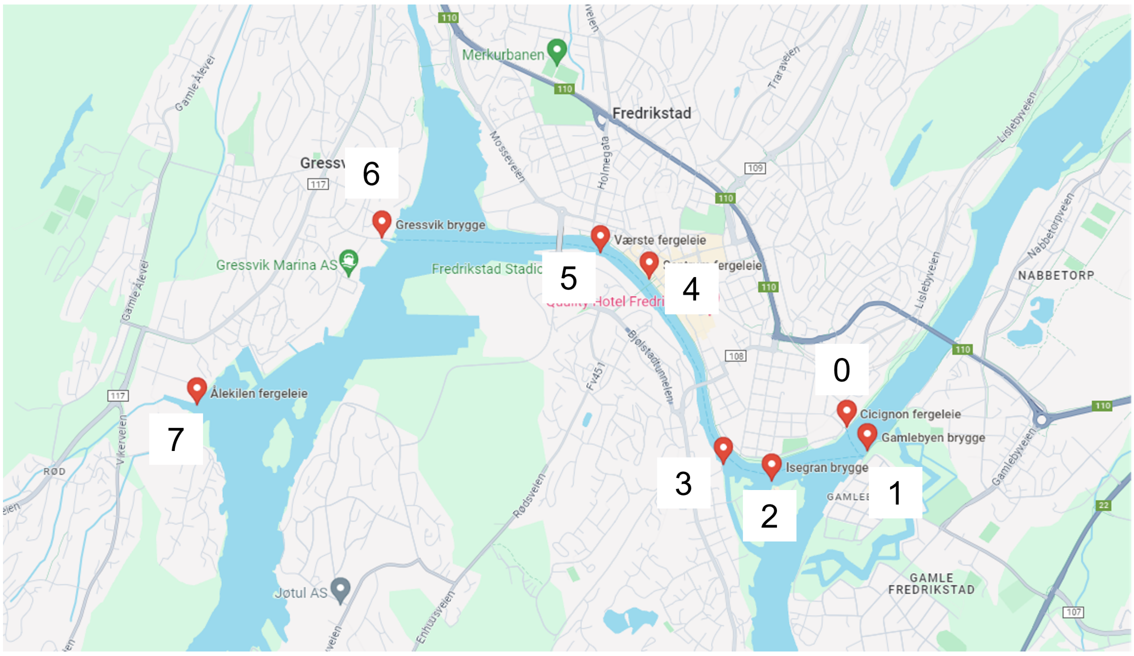


Figure 4.2: Ferry terminals in Fredrikstad (made from GoogleMaps)

Figure 4.3: Hyke's ferry solution

4.2. Static model application

The model application starts from the computational experiments with small instances for the static subproblem. This series of experiments is conducted to evaluate the solving performance of the proposed solving methods. The ferry network of Fredrikstad (Figure 4.2) is utilised for the experiment. 18 instances are generated for the experiment by setting different fleet sizes (number of vessels) and the number of requests. Table 4.1 lists the parameters used for the experiments. The fleet size between two and four and the number of requests between one and six are tested. These two factors influence the complexity of the problem. Therefore, experimenting with different combinations of fleet size and number of requests is expected to provide insights into the solving capability of the proposed methods.

The vessels are assumed to have a homogeneous property and are defined according to the technical description of Hyke's ferry solution ("Hyke", 2021). A vessel is assumed to have a loading capacity of C = 50 passenger units, with the freedom to adjust the capacity allocation in the vessel with a granularity of one passenger unit per request type for mixed purpose vessels. In the static experiments, all vessels are assumed to be mixed purpose. Also, the speed of the vessel is assumed to be constant at 6 knots, which is equivalent to v = 11.112 km/h. The battery consumption per distance is assumed to be r_c = 1.0 kWh/km. Also, the battery capacity of the vessel is set to B = 190 kWh.

Table 4.1: Parameters used for the static experiments

Parameter	Values	Parameter	Values
$\|K\|$	{2, 3, 4}	B	190 kWh
$\|N\|$	8 terminals	r_c	1.0 kWh/km
$\|R\|$	{1, 2, 3, 4, 5, 6}	r	100 kWh/h
C	50 pax	v	11.112 km/h

Each instance is solved by the exact method and the insertion heuristic. The objective value, MIP gap, and computational time are collected for the exact method. The MIP gap represents the difference between the objective value and the lower bound of solutions the solver discovered in the exact method. The objective value and computational time are collected for the insertion heuristic. All experiments are repeated three times to obtain the average values. Each experiment has a time limit of one hour.

4.2.1. Results of static model application

The results of the static experiment are shown in Table 4.2. The objective value, the MIP gap, and the computational time are presented for the exact method for each instance. The objective values and the computational time are presented for each instance for the insertion heuristic. The MIP gap of 0 indicates the solver found the optimal solution, while if the gap is larger than 0, it means the feasible solution is not guaranteed to be the optimum. Instances differ in the fleet size and the number of requests, as shown in Table 4.1. The model is developed in Python 3.12.2, and all experiments are performed on a laptop with an 11th Generation Intel(R) Core(TM) i7-1165G7 @ 2.80GHz and 32GB RAM with a Windows 11 Pro. The routing solution for each instance can be found in Appendix B.

The results indicate the capability of the insertion heuristic in finding optimal/good solutions for all instances in a shorter computational time than the exact method. All instances have the same objective values as the exact method. The solutions in instances 6, 12, and 18 have the same objective values as the exact method, but these solutions are not guaranteed to be the optimal solutions since the time limit of the exact method is reached in these instances.

The insertion heuristic has a significantly shorter computational time than the exact method if the number of requests is less than or equal to 5. It manages to find the solution within a maximum of 15 seconds for these instances. When 6 requests are inserted, both methods require a long computational time to find the solution. Therefore, it can be said that a maximum of 5 requests is the limit of the solving capability of the insertion heuristic under the tested fleet size variation.

Also, the computational time of the insertion heuristic is not linear to the number of requests or the fleet size. This is logical since the heuristic applies a greedy insertion and tries all possibilities of insertion. The increment in the number of requests and fleet size exponentially increases the possible combinations of insertion. Thus, the computational time is expected to drastically increase when the fleet size and the number of requests increase. The results suggest that the number of requests substantially influences the computational time more than the fleet size.

In small instances, such as instances 1, 2, 7, 8, 13, and 14, the exact method found the optimal solution within a minute, which is acceptable for the dynamic model since the time step is set to every minute and providing a solution within a minute will be the benchmark for the dynamic model. However, the computational time of the exact method drastically increases when the number of requests increases. When 6 requests are present, the exact method could not find the optimal solution within the time limit. These results indicate that the solving capability of the exact method is limited to very small instances.

Table 4.2: Comparison between the exact method and the insertion heuristic for the static subproblem

ID	|K|	|R|	Obj Exact [km]	MIP Gap Exact [%]	CPU Exact [s]	Obj Insertion [km]	CPU Insertion [s]
1	2	1	1.301	0.000	1.076	1.301	0.007
2	2	2	3.999	0.000	53.255	3.999	0.004
3	2	3	5.421	0.000	55.852	5.421	0.006
4	2	4	5.725	0.000	140.320	5.725	0.153
5	2	5	5.577	0.000	512.009	5.577	4.007
6	2	6	7.140	12.709	3600.000*	7.140	335.608
7	3	1	1.301	0.000	1.767	1.301	0.001
8	3	2	3.987	0.000	33.208	3.987	0.002
9	3	3	5.410	0.000	66.412	5.410	0.002
10	3	4	5.725	0.000	78.645	5.725	0.175
11	3	5	5.577	0.000	295.413	5.577	14.412
12	3	6	6.976	10.621	3600.000*	6.976	1287.811
13	4	1	1.301	0.000	1.947	1.301	0.001
14	4	2	3.678	0.000	52.450	3.678	0.002
15	4	3	5.100	0.000	146.781	5.100	0.011
16	4	4	5.725	0.000	174.212	5.725	0.569
17	4	5	5.411	0.000	1385.376	5.411	12.560
18	4	6	6.976	17.802	3600.000*	6.976	3415.124

*Time limit of 1 hour reached.

4.3. Dynamic model application

The proposed dynamic model is experimented with to assess the model performance and to obtain insights into the transportation performance of the mixture of capacity in waterborne vessels for heterogeneous on-demand service. By utilising the ferry network of Fredrikstad (Figure 4.2), experiments with different demand scenarios and vessel type combinations are conducted. The fleet size is assumed to be fixed to two. Thus, four vessel type combinations, (mixed, mixed), (mixed, passenger), (mixed, parcel), and (parcel, passenger) are examined. The starting depots of the vessels are assumed to be terminals 4 and 6. These terminals are also where the charging facilities are installed. The parameters regarding the vessel property are the same as Table 4.1. The experiments consider a full usual work day between 6:00-22:00. As mentioned before, the demand for the services is represented as sets of individual requests and the demand scenarios are generated stochastically.

In the dynamic model, the solving time of each static subproblem is limited to one minute. This is because of the necessity of quickly obtaining the next route plan to react to the new requests. As explained in Section 3.5, the new requests will be rejected or postponed, and the previous route plan will be kept until a new feasible route plan is found in the case where the model does not manage to find a feasible solution within a minute. It is also possible that the solver finds a feasible solution, but it is not guaranteed to be the global optimum in the exact method. As long as the route plan is feasible, the route plan will be updated to the new route plan and will be pursued by the vessels until the next time step.

4.3.1. Request generation

A set of travel requests from passengers and delivery requests from parcels are generated to represent the demand for the service. Each request consists of the origin terminal, the destination terminal, the time window, the maximum waiting time, the size, and the type of the request. In this section, the process of generating a set of requests will be presented.

It is ideal to incorporate historical data regarding the demand for the ferry service to create realistic synthetic request data. However, the considered ferry system is still in the conceptual phase and does not have real historical data regarding the demand to apply. Therefore, the set of requests is created by incorporating the typical temporal and spatial patterns of demand for passenger and parcel services in transportation. The temporal distribution of transport demand from passengers is well known to have peak hours and off-peak hours due to common activities such as commuting. Passenger demand has two peak hours during the day: the morning peak and the evening peak. These peak hours derive from commuting trips to work or school. The hours for the rest of the day are off-peak hours, and the passenger demand decreases from the peak hour demand.

As well as the temporal patterns of demand, the spatial patterns of demand will be considered in the request generation. The spatial patterns of demand are influenced by the land use of the areas (Liu et al., 2021). For instance, commuting between a terminal in a residential area to a terminal in an industrial/office area is more likely to happen compared to other terminals, or trips from a residential area to a commercial area are more likely to occur considering the shopping activities. These characteristics will formulate the spatial patterns of demand for the system and will be incorporated into the parameter settings in the case study.

Considering the temporal and spatial distribution of transport demand, an OD and time specific Poisson process is conducted to generate the passenger requests stochastically. The Poisson process is a widely used stochastic process that models the times of arrival to a system at an interval of time. (Robert, 2011) The Poisson process has several characteristics which align with the travel requests made in real life. The first characteristic is the independence of events' occurrence. The occurrence of an event in one time interval does not affect the occurrence of the event in another time interval. This is a realistic assumption to make when it comes to travel requests in real life. Customers do not make requests based on other customers' requests, but they make requests because they want to travel to their destination to fulfil their activity. Another characteristic of the Poisson process is the stationarity. Stationarity means that the probability of an event occurring in a certain length of time is constant and is defined as a constant rate λ.

A Poisson process can be defined as the following. Given a arrival rate of λ and the length of the interval t, the Poisson random variable with mean value of λt is,

$$P(N(t) = k) = \frac{(\lambda t)^k e^{-kt}}{k!}$$

Where $N(t)$ is the number of events in time t. This is the probability of a certain number of events k occurring in the time interval t. Also, the time between two events follows an exponential distribution,

$$f(t) = \lambda e^{-\lambda t}$$

The arrival rate λ is set per OD pair and time period. Given the arrival rate, a set of requests in the time period per OD pair is generated randomly. Each request has an arrival time, and that will be assumed to be the beginning of the pickup time window of the request.

The dynamic model considers a full usual workday between 6:00-22:00, and the time period for passengers is divided into five sections: 6:00-7:00, 7:00-9:30, 9:30-16:00, 16:00-18:30, 18:30-22:00. This division is defined to incorporate the peak hours and the off-peak hours in terms of the passenger demand. 7:00-9:30 and 16:00-18:30 are considered the morning peak and the evening peak, respectively. In these time periods, the parameters are adjusted to have higher arrival rates than other time periods, which indicates that the interval time between two requests is expected to be shorter during these periods than other time periods. In the experiment, the peak hours' arrival rate is set to be 1.5 times larger than the off-peak hours.

The land use around each terminal differs from each other, and these differences are expected to influence the demand from the passengers. The municipality of Fredrikstad published a master plan of urban planning between 2023-2035 ("Kommuneplanens arealdel 2023-2035", 2023). The plan states that the area around terminals 4 and 5 in Figure 4.2 are part of the focused areas of urban development. More residential buildings and business activities are planned for the city. Therefore, it is expected that the demand from/to these terminals will be greater than that of other OD pairs. Based on this assumption, the arrival rate for OD pairs which either the origin or the destination is terminal 4 or 5 is set double of the other OD pairs, and the arrival rate between terminal 4 and 5 is set triple of the other OD pairs. In addition to this demand pattern, it is simply assumed that there is no demand for ferry service between the terminals located on the same island, as well as between the same terminal. Specifically, the pairs (2, 3), (3, 5), (0, 4), and (6, 7) do not have any demand. The details of the parameters can be found in the Appendix A.

Each passenger request has a size which is randomly determined by a discrete uniform distribution between 1-10. Also, it is assumed that the maximum waiting time for each passenger request is 15 minutes, and the latest delivery time is 45 minutes after the request pop up time. The service time of a request is dependent on the size of the request, and for the passenger requests, it is assumed to be 0.25 minutes (15 seconds) per passenger unit. For instance, if the request size is 4, it is assumed that 1 minute is necessary for boarding and leaving the vessel.

The number of parcel requests between an OD pair in a time period is assumed to be known beforehand. This is based on the assumption that the logistics company provides the ferry operator with the information about the parcels to be communicated in advance. Therefore, the ferry operator is able to anticipate how many parcel requests to expect during the day. The stochasticity of the parcel request derives from the uniform distribution of the pop up time of the requests and the required arrival time of the parcel. An uniform distribution with the time period determines the pop up time of a parcel request. Also, parcels have different requirements from passenger requests. In contrast to passenger requests, the waiting time is not important for parcel requests. The important aspect of parcel requests is to arrive at the destination by the required arrival time to ensure the parcels are delivered to the customers as ordered. Therefore, the maximum waiting time is set to large enough for parcel requests and only the delivery time window is applied for them. The time period for parcel requests are divided into 6:00-11:00, 11:00-15:00, and 15:00-21:00. These are set based on the assumption that the parcels can be categorised into three required delivery times: 12:00, 16:00, and 22:00. The delivery by noon considers the case which the parcel has to be delivered to the customer in the afternoon of the same day. Similarly, the delivery time of 16:00 considers the parcels which need to be delivered in the evening of the same day. Finally, the delivery by the end of the day means the parcel will be delivered to the customer the next day. The arrival time of each passenger request is determined by a discrete probability distribution for each time step. Table 4.3 shows the probability of each required delivery time depending on in which time period the request pops up. The required delivery time is randomly determined for each request based on this distribution. It is made sure that the parcel request would have at least an hour between the pop up time and the required delivery time to ensure that the parcel is feasible to be delivered within the time window. The size of a parcel request is also randomly determined by a discrete uniform probability between 1-10. When a parcel request is served by the vessel, it requires some time to board and disembark the parcels to/from the vessel. Similar to the passenger requests, it is assumed that the service time for a parcel request is dependent on its size and is set to require 30 seconds (0.5 minutes) per passenger unit. The request size of parcel requests is defined in the passenger unit for this study.

Table 4.3: Probability of each required delivery time from each pop up time period

		Required delivery time		
		12:00	**16:00**	**22:00**
Pop up time period	**6:00-11:00**	0.6	0.2	0.2
	11:00-15:00	0	0.6	0.4
	15:00-21:00	0	0	1

Two demand scenarios are developed for the experiment, ”high” and ”low”. As the name suggests, the ”high” scenario has higher demand and the ”low” has lower demand. The demand can be summarised as the number of requests throughout the day. The total number of requests for each demand scenario is shown in Table 4.4. The ”low” scenario has 172 requests in total and consists of 36 passenger requests and 136 parcel requests during the whole day. The ”high” scenario has 57 passenger requests and 148 parcel requests and in total 205 requests. Figure 4.4 and Figure 4.5 illustrate the temporal distribution of the cumulative number of requests by taking the time as horizontal axes and the number of requests of each request type occurring at each time as the vertical axes for each demand scenario. It is worth noting that because of the higher arrival rate between the peak hours and the off-peak hours, the increment of passenger requests (red lines) is sharper between the peak hours (7:00-9:30, 16:00-18:30) than the off-peak hours. In contrast, the parcel requests are widely and more or less evenly spread throughout the day.

Table 4.4: Number of requests per demand scenario

Scenario	Total requests	Passenger requests	Parcel requests
low	172	36	136
high	205	57	148

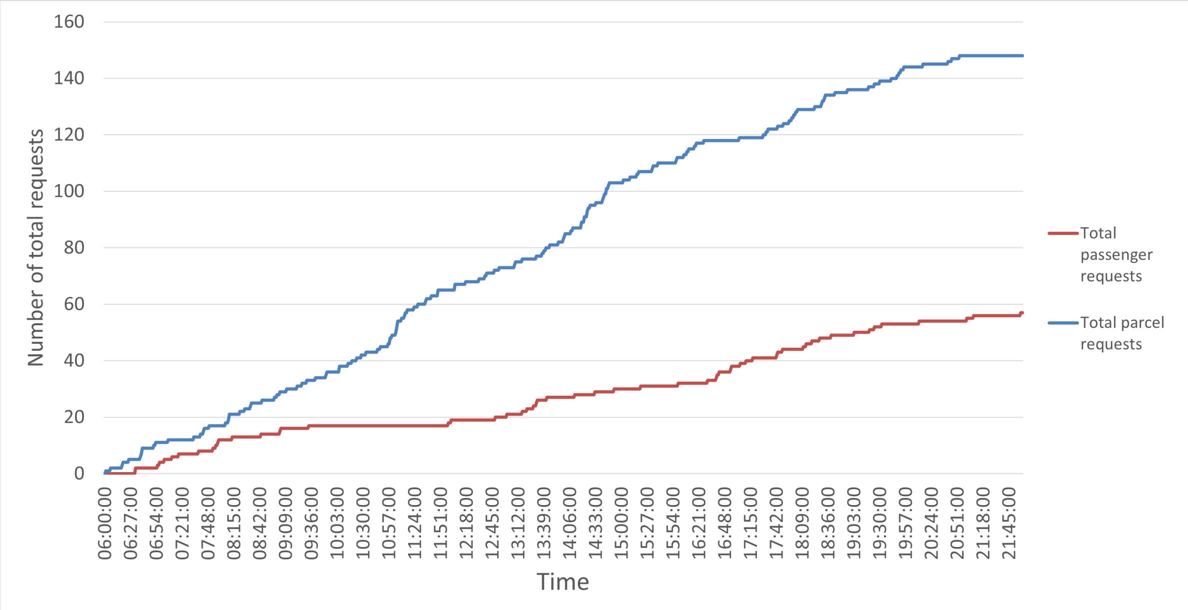


Figure 4.4: Temporal distribution of total number of requests under high demand scenario

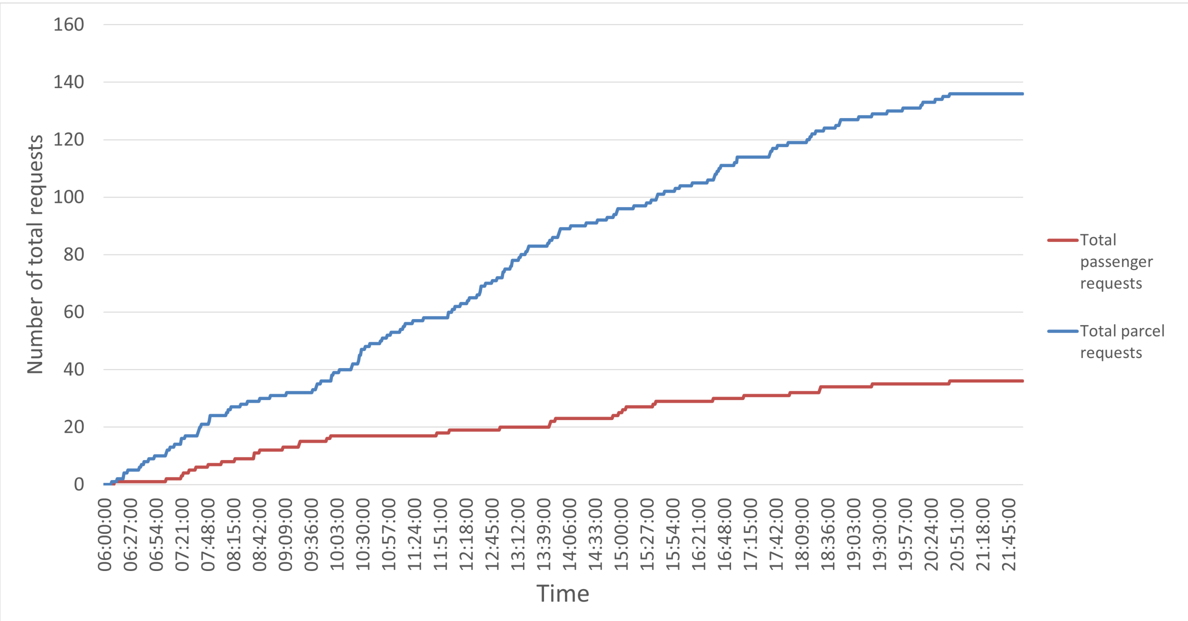


Figure 4.5: Temporal distribution of total number of requests under low demand scenario

4.3.2. Results of solving performance of dynamic model

The results of the dynamic model application are reported in this section. The experiment begins with the performance evaluation of the solving algorithms of the dynamic optimisation for each solving algorithm. Finally, the transportation performance, which mainly focuses on the aforementioned KPIs, is reported. Same as the static model application, the model is developed in Python 3.12.2, and all experiments are performed on a laptop with an 11th Generation Intel(R) Core(TM) i7-1165G7 @ 2.80GHz and 32GB RAM with a Windows 11 Pro.

The dynamic optimisation model is performed for two demand scenarios and four different vessel type combinations. All configurations are performed using both the exact method and the proposed insertion heuristic. The two demand scenarios are defined as it was demonstrated in Section 4.3.1. The fleet size is fixed to two, and the experimented vessel type combinations are (mixed, mixed), (mixed, passenger), (mixed, parcel), and (parcel, passenger). The model performance of the exact method is assessed by the capability of obtaining the optimal solution at each time step, and the computational time. For the insertion heuristic, the computational time is compared to the exact method.

Table 4.5 and 4.6 show the model performance of both methods under each configuration per demand scenario. Every time a new request is inserted into the model, the model executes the static optimisation to determine the new route plan of the vessels to serve the new requests, while ensuring the previous requests' requirements are satisfied. The computational time is the time needed to perform the model for the entire time horizon for each method. The number of time steps the static optimisation was executed are shown in the tables. In the exact method, ”Optimum” indicates the number of time steps in which the model found the optimal route plan, ”sub-optimum” is the number of time steps the model found a feasible route plan, but it is not guaranteed to be the optimal solution, and ”infeasible” is the number of time steps the model could not find a feasible solution within the time limit. The total number of time steps differs between each configuration in the exact method because when the model does not manage to find a feasible solution, the assignment of the new request is postponed to the next time step, and the model tries to find a new route plan again. In contrast, the insertion heuristic has the same number of time steps for each vessel time combination in the same demand scenario since the model is developed to reject new requests if no feasible route is found.

For the high demand scenario, the computational time for the entire time horizon significantly differs between the insertion heuristic and the exact method. As the table shows, the insertion heuristic managed to run the model between 13 - 48 seconds, while the exact method took a minimum of 2876.81 seconds, which is 48.0 minutes. This difference in computational time was expected from the static model application. In the dynamic model, a maximum of three requests were inserted in the same time step. As the results of the static experiment (Table 4.2) suggest, the insertion heuristic finds a solution almost instantly when three requests are inserted, while the exact method took almost a minute to find the optimal solution under three requests with two vessels. The static model was performed between 173 and 176 times in the exact method. The (mixed, parcel) combination managed to find a feasible solution 93.1% of the times, and 84.6% of them were the optimal solution. This was the highest among all vessel type combinations for the high demand scenario. (mixed, mixed) followed by 86.1% of the time finding a feasible solution and 89.3% of them being optimum, (mixed, passenger) with 85.2% and 77.3% of them being optimum, and (parcel, passenger) with 78.9% and 79.0% of them being optimum.

Similar to the high demand scenario, the computational time of the insertion heuristic was significantly shorter than that of the exact method for the low demand scenario. The computational time of the insertion heuristic varies between 24 - 67 seconds, while the exact method took at least 6272 seconds (104 minutes) to perform this scenario. The static model was performed 169-172 times during the time horizon in the exact method. (mixed, parcel) has the highest proportion of obtaining a feasible solution as well as the optimal solution. 95.3% of the time steps reached a feasible solution, and 57.1% of them were the optimal solution. (parcel, passenger) follows with 94.1% of the time obtaining a feasible solution and 47.5% being the optimum. (mixed, mixed) reached a feasible solution for 91.7% of the time with 46.7% being the optimum, and finally (mixed, passenger) had 81.4% of the time finding a feasible solution, which is the lowest among all configurations for this demand scenario. 47.9% of them were the optimal solution.

Table 4.5: Performance of each method under high demand configuration

Vessel types	CPU Exact [s]	Exact (optimum, sub-optimum, infeasible, total) [time steps]	CPU Insertion [s]	Insertion total [time steps]
(mixed, mixed)	2876.81	(133, 16, 24, 173)	13.23	173
(mixed, passenger)	4523.62	(116, 34, 26, 176)	47.41	173
(mixed, parcel)	3035.68	(137, 25, 12, 174)	14.66	173
(parcel, passenger)	4614.02	(109, 29, 37, 175)	15.28	173

Table 4.6: Performance of each method under low demand configuration

Vessel types	CPU Exact [s]	Exact (optimum, sub-optimum, infeasible, total) [time steps]	CPU Insertion [s]	Insertion total [time steps]
(mixed, mixed)	6840.56	(77, 78, 14, 169)	24.15	168
(mixed, passenger)	6909.89	(67, 73, 32, 172)	67.07	168
(mixed, parcel)	6272.47	(92, 69, 8, 169)	37.84	168
(parcel, passenger)	6575.45	(76, 84, 10, 170)	24.19	168

4.3.3. Results of transportation performance in dynamic model

This section reports the results related to the transportation performance for different configurations for the dynamic model. As mentioned before, the dynamic model for all configurations is performed using the exact method and the insertion heuristic. Therefore, two values are presented for each KPI per configuration. In addition to the KPIs, the time series of the loading level of each vessel per configuration is also analysed to provide insights into the operation of vessels throughout the day.

KPIs for high demand scenario

Table 4.7 and 4.8 present the request met ratio (RMR), the total travel distance (TTD), and the total empty travel distance (TETD) of each vessel type combination for high demand scenario solved by the exact method and the insertion heuristic, respectively. (mixed, mixed) recorded high RMR in this scenario, 87.05% for the Exact method and 89.58% for the insertion heuristic. This result is plausible since any request can be assigned to both vessels in (mixed, mixed), which means the closer vessel from the origin of the request was able to pick up the request in most of the cases. On the other hand, (parcel, passenger) recorded the lowest RMR for both solving methods. In contrast to (mixed, mixed), the closer vessel to the pick up terminal was not necessarily able to serve the request because of the fixed purpose of the vessel; thus, more requests were rejected because of the time constraints. (mixed, parcel) and (mixed, passenger) resulted in an intermediate RMR between (mixed, mixed) and (parcel, passenger). It is logical that (mixed, parcel) has a higher RMR than that of (mixed, passenger) since there were more parcel requests than passenger requests generated in this experiment and the dependency on the mixed purpose vessel was relatively lower for (mixed, parcel) than (mixed, passenger).

In the exact method, (mixed, parcel) recorded 207.08 km of TTD and is the lowest TTD among the four combinations. (mixed, passenger) follows with 222.93 km, (mixed, mixed) with 252.98 km, and (parcel, passenger) with 313.91 km. It can clearly be seen that the TTD decreases from the (parcel, passenger) when mixed purpose vessel is available. In fact, it resulted in between 19.4-34.0% reduction of the TTD by introducing the mixed purpose vessel.

In the insertion heuristic, (mixed, passenger) recorded the lowest TTD of 149.34 km. (mixed, parcel) followed by 155.28 km. Unlike the exact method, (parcel, passenger) has slightly lower TTD than (mixed, mixed), with 162.05 km and 177.79 km, respectively. This result derives from the difference in the RMR. As mentioned before, (mixed, mixed) has a significantly higher RMR than that of (parcel, passenger) in the insertion heuristic. It is natural that the more requests the vessels serve, the longer the TTD is. It can be said that a trade-off between the RMR and the TTD is observed from the results of the insertion heuristic.

The TTD is overall lower for the insertion heuristic than those of the exact method. The TTD were 25.0-48.3% lower for the insertion heuristic compared to the exact method. This is possible since there were many cases in which the exact method could only find a feasible solution that is not necessarily the optimal solution for the route plans, and it is assumed that the vessels update their route plans if the solver finds a feasible solution. This led the vessels to pursue route plans that had more travelling distance, and thus, the TTD ended up high. However, the insertion heuristic considers all possible sequences of the order of visiting the terminals. This guarantees that the order of visiting the terminals will be at least found with the minimum total travel distance for the route plan, which is the objective function in the model. Therefore, it is logical that the insertion heuristic obtained lower TTD overall.

In both solving methods, (parcel, passenger) has the largest TETD among the four vessel type combinations. Notably, the availability of mixed purpose vessels decreased the TETD by 63.3-74.9% from (parcel, passenger) in the exact method. The difference in TETD between vessel type combinations is not as large as the exact method in the insertion heuristic. However, (parcel, passenger) still has the largest TETD among the four vessel type combinations, and mixed purpose vessel reduces the TETD by 12.5-34.9% from only having fixed purpose vessels. The reduction of TETD by implementing mixed purpose vessels is expected because the mixed purpose vessels are able to serve any request, and the cases in which a vessel far from the pickup terminal needs to pick up a specific type of request occur less than when fixed purpose vessels are present. The model assigns the nearby mixed purpose vessel to the request rather than the further fixed purpose vessel.

Table 4.7: KPIs for each vessel type combination (high demand, Exact method)

Vessel types	Request met ratio [%]	Total travel distance [km]	Total empty distance [km]
(mixed, mixed)	87.05%	252.98	33.31
(mixed, passenger)	80.31%	222.93	26.90
(mixed, parcel)	88.08%	207.08	22.79
(parcel, passenger)	71.50%	313.91	90.85

Table 4.8: KPIs for each vessel type combination (high demand, Insertion heuristic)

Vessel types	Request met ratio [%]	Total travel distance [km]	Total empty distance [km]
(mixed, mixed)	89.58%	177.79	23.16
(mixed, passenger)	78.13%	149.34	17.22
(mixed, parcel)	82.29%	155.28	18.84
(parcel, passenger)	76.04%	162.05	26.47

KPIs for low demand scenario

Table 4.9 and 4.10 present the KPIs for each vessel type combination for the low demand scenario by each solution method. (mixed, mixed) and (mixed, parcel) recorded similarly high RMR of more than 87% in the exact method. (parcel, passenger) has the third-highest RMR and (mixed, passenger) recorded the lowest RMR. (mixed, mixed) has a significantly high RMR of 86.98% in the insertion heuristic compared to other vessel type combinations. The other three combinations recorded a similar RMR of around 80%. Similar to the high demand scenario, (mixed, mixed) resulted in higher RMR than those of (parcel, passenger). This indicates the improvement of service level by having mixed purpose vessels.

In the insertion heuristic, (mixed, passenger) has 108.59 km of TTD and is significantly lower than other vessel type combinations. (mixed, parcel) followed by 120.20 km and (mixed, mixed) and (parcel, passenger) has a similar TTD between 144 and 149 km. The high TTD in (mixed, mixed) can be because of the higher RMR than other vessel type combinations. As mentioned before in the high demand scenario, it is observed that there is a trade-off between the RMR and the TTD. Naturally, the more requests the vessels serve, the more the vessels need to travel. This trade-off is observed in the low demand scenario as well. (parcel, passenger) has high TTD, and this can be explained by the situations in which a specific vessel needs to pick up a request far away for a fixed purpose.

Similar to the high demand scenario, the TTD in the insertion heuristic resulted in 10.7-42.1% lower than those of the exact method for all vessel type combinations. As mentioned before, this can be because of the assumption that the vessels will update their route plan when a feasible route plan is found, which is not necessarily the optimal route plan that minimises the total travel distance. In the exact method, (mixed, parcel) and (parcel, passenger) have lower TTD than those of (mixed, mixed) and (mixed, passenger). This can be because of the route plan constructed through the optimisation. Also, the TTD highly depends on which request is accepted/rejected. If a request that requires a long travel distance (e.g., terminal 7 to 0) is accepted, the TTD of the vessels increases to serve this request. In contrast, if this request is rejected depending on the vessel types, then the vessels do not need to make long travels. Therefore, it is possible that the limitation of vessel purpose can lead to less TTD, which is highly dependent on the requests occurring in the day.

The results of both methods indicate the reduction of TETD by introducing a mixed purpose vessel. The TETD is reduced by between 13.7-64.5% in the exact method and 39.8-53.1% in the insertion heuristic. This derives from reducing empty trips to pick up a further request with a specific type by introducing mixed purpose vessel. In contrast to the TTD, lower TETD was recorded for the exact method than those of the insertion heuristic. (mixed, passenger) recorded the lowest TETD of 7.33 km. (mixed, parcel) follows with 11.09 km, (mixed, mixed) with 17.83 km, and (parcel, passenger) with 20.65 km. In the insertion heuristic, each vessel type combination has less difference with each other. (mixed, passenger) recorded 14.20 km, (mixed, parcel) with 15.56 km, (mixed, mixed) with 18.23 km, and finally (parcel, passenger) with 30.28 km.

Table 4.9: KPIs for each vessel type combination (low demand, Exact method)

Vessel types	Request met ratio [%]	Total travel distance [km]	Total empty distance [km]
(mixed, mixed)	87.29%	188.57	17.83
(mixed, passenger)	75.98%	187.47	7.33
(mixed, parcel)	87.36%	161.23	11.09
(parcel, passenger)	83.71%	161.65	20.65

Table 4.10: KPIs for each vessel type combination (low demand, Insertion heuristic)

Vessel types	Request met ratio [%]	Total travel distance [km]	Total empty distance [km]
(mixed, mixed)	86.98%	148.63	18.23
(mixed, passenger)	79.77%	108.59	14.20
(mixed, parcel)	81.50%	120.20	15.56
(parcel, passenger)	80.57%	144.38	30.28

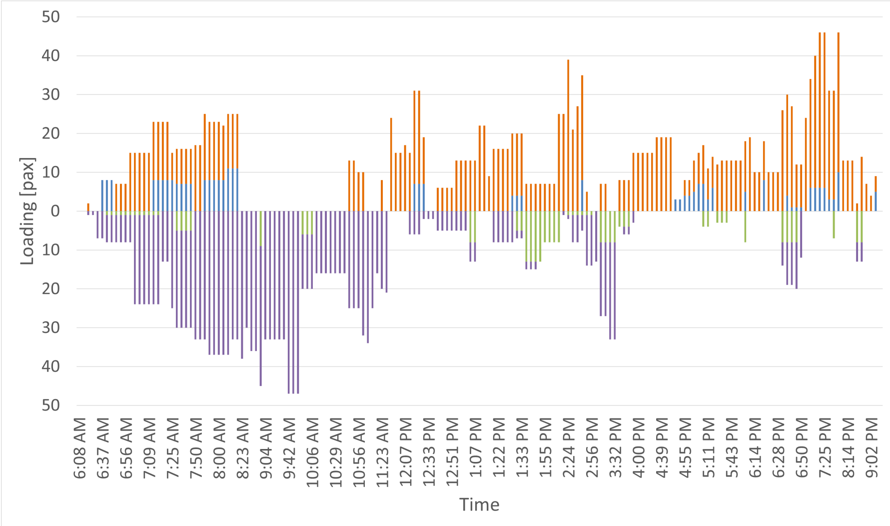
Time series of loading level of each vessel

In addition to the KPIs, the time series of vessels' loading level for each vessel type combination is useful to understand the usage of each vessel during the day. Figure 4.6 and 4.7 illustrate the time series of the loading level of each vessel for all vessel type combinations in high and low demand scenarios solved by the insertion heuristic. For each figure, the horizontal axis is the time and the vertical axis is the loading level represented in passenger unit. Each figure represents both vessels' loading levels at each time step. The loading levels are indicated with vessel 1's load upward from the baseline of 0, and vessel 2's load downward. Orange represents vessel 1's parcel load, blue represents vessel 1's passenger load, purple represents vessel 2's parcel load, and green represents vessel 2's passenger load. The loading capacity of each vessel is 50 passenger units. Since both solving methods resulted in similar patterns, the results from the insertion heuristic will be explained here, and the results of the exact method can be found in Appendix C.

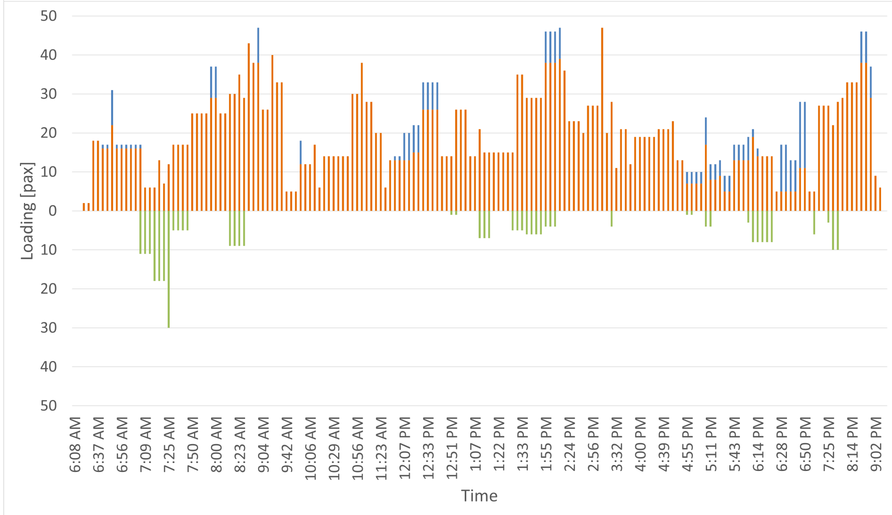
It can be seen in Figure 4.6a that both vessels handled the requests in a good balance in (mixed, mixed) throughout the day compared to other vessel type combinations in high demand scenario. This is supported by the fact that there were fewer time periods in which a vessel had a loading of zero overall, which indicates that the vessel is dwelling at a terminal than other vessel type combinations. For instance, the large difference in the loading level between vessel 1 and vessel 2 in (mixed, passenger) is easy to identify as vessel 1 had a non-zero loading throughout the day, while vessel 2 had zero loading for most of the time, except for several hours including the peak hours (Figure 4.6c). Also, it can be seen in Figure 4.6a that both of the vessels handled the passenger requests in the morning peak and the evening peak in (mixed, mixed). However, vessel 2 served fewer passenger requests in the morning peak and focused on parcel requests, while in the afternoon, the focus switched to vessel 1, where most of the parcel requests were served by vessel 1 and the passenger requests by vessel 2.

Regardless of the vessel type combination, the model preferred to assign the requests to the same vessel during a certain time period regardless of the type of the request rather than trying to distribute the loads equally between the two vessels. This can be because the model's objective function is set to minimise the total travel distance. The simple and effective way to reduce the total travel distance is to reduce the number of trips, and this implicitly means the request service was merged with other requests as much as it could be. Therefore, it is expected that the loading is imbalanced between vessels for a certain period until a new request close to the other vessel occurs.

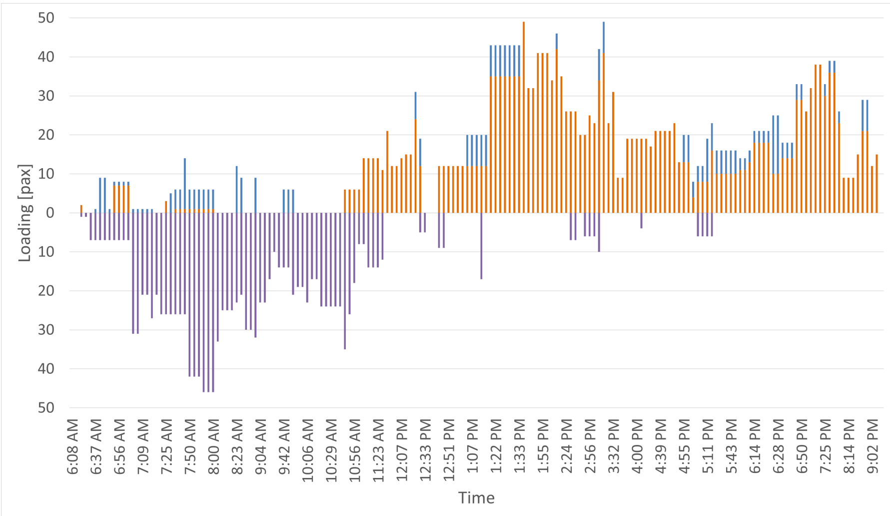
In contrast to the high demand scenario, the loading level of vessels in the low demand scenario was imbalanced in all vessel type combinations, including the (mixed, mixed). As can be seen in Figure 4.7, vessel 1 served most of the requests, and vessel 2 did not serve requests for a long time in all vessel type combinations. As mentioned above, the objective of minimising the total travel distance leads to trying to merge as many requests as possible into one vessel. The imbalance may have occurred because one vessel was enough to serve most of the requests in the low demand scenario, and operating one one vessel led to less total travel distance of the route. Notably, vessel 1 was operating on a loading level close to the capacity when vessel 2 is dedicated to passengers, as shown in Figure 4.7c. Despite vessel 2 being dedicated to passengers, vessel 1 still served the passenger requests in the morning peak in (mixed, passenger) setting.



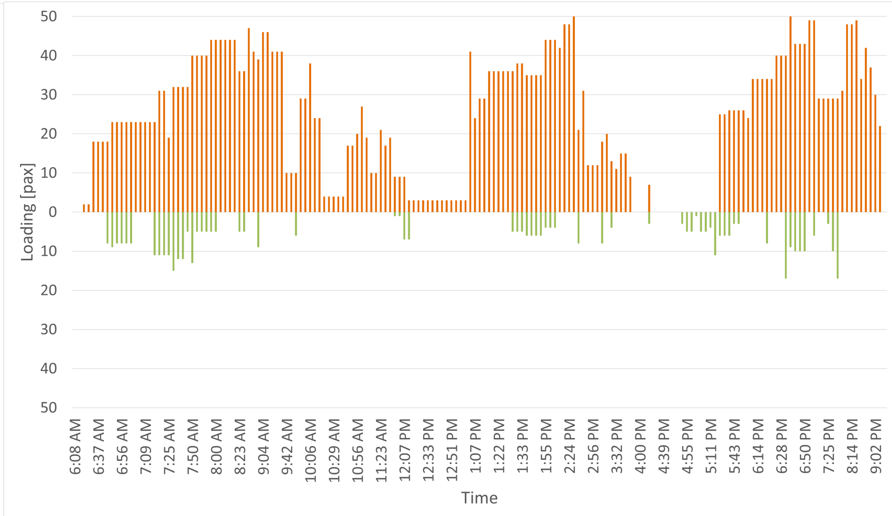
(a) (mixed, mixed)



(b) (mixed, passenger)

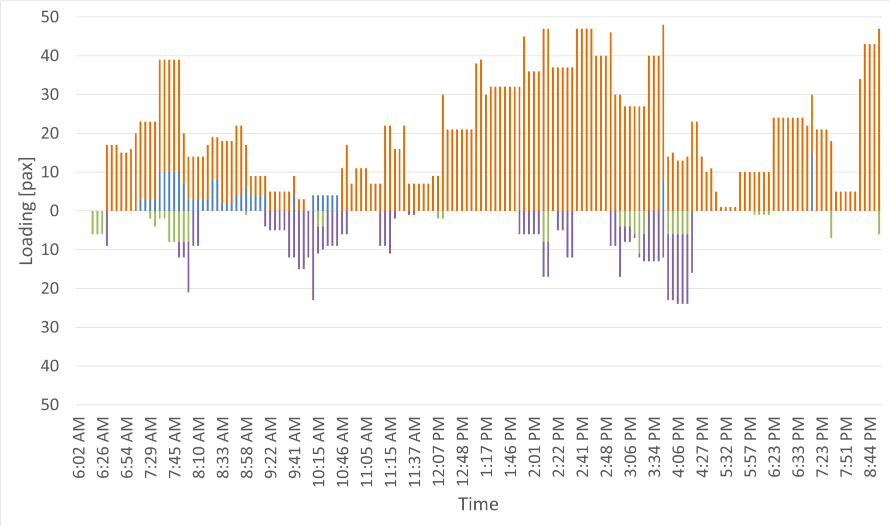


(c) (mixed, parcel)

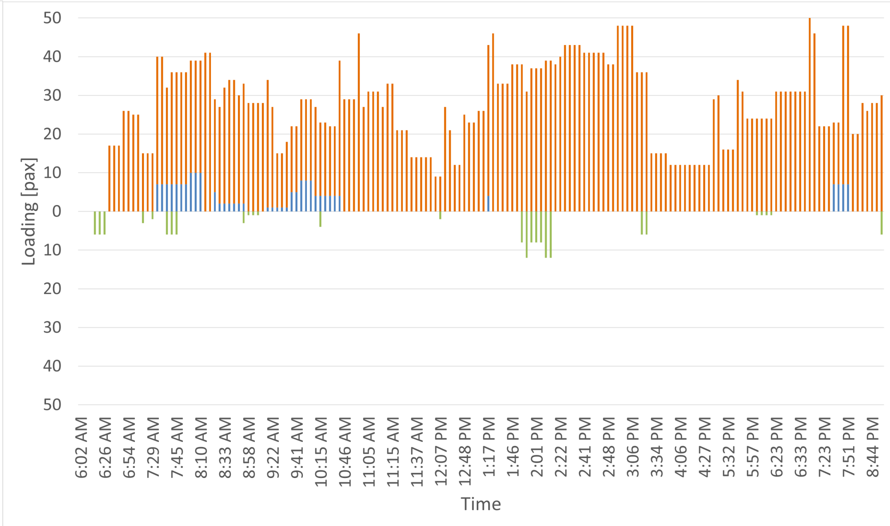


(d) (parcel, passenger)

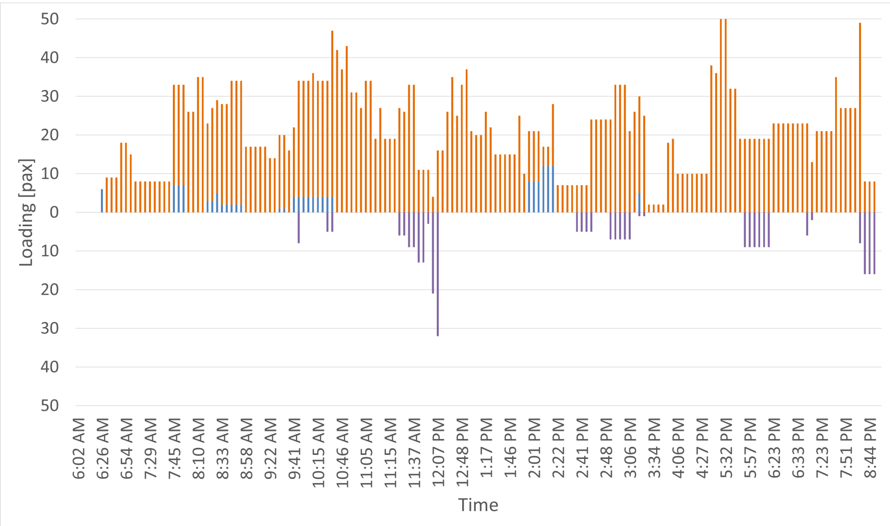
Figure 4.6: Loading level of each vessel in each vessel type combination for high demand scenario by insertion heuristic
orange: parcel vessel 1, blue: passenger vessel 1, purple: parcel vessel 2, green: passenger vessel 2



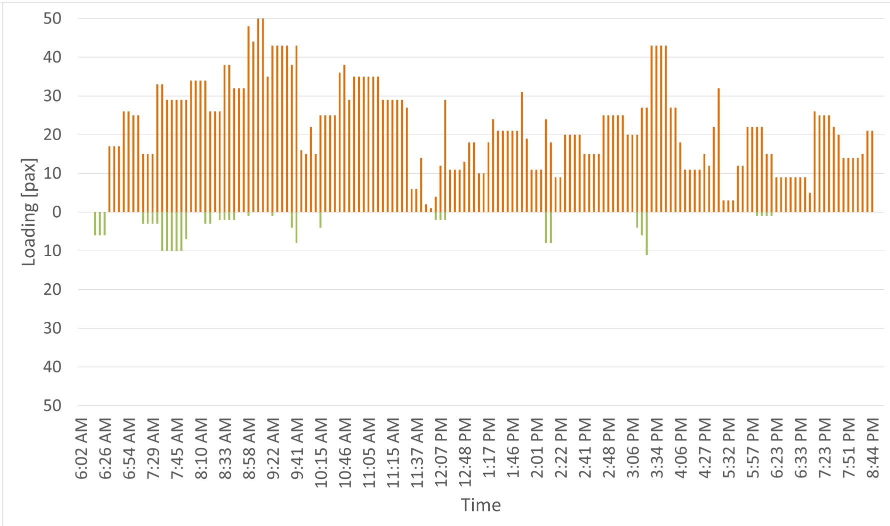
(a) (mixed, mixed)



(b) (mixed, passenger)



(c) (mixed, parcel)



(d) (parcel, passenger)

Figure 4.7: Loading level of each vessel in each vessel type combination for low demand scenario by insertion heuristic
orange: parcel vessel 1, blue: passenger vessel 1, purple: parcel vessel 2, green: passenger vessel 2

5 Discussion

In this chapter, the results obtained from the computational experiment are discussed. Also, the assumptions and limitations of the modelling are discussed.

5.1. Discussion of Results

The experiment of the static optimisation model (Section 4.2.1) indicates the high solving capability of the proposed insertion heuristic. The instance tests demonstrated that the insertion heuristic produces solutions with the same objective values from the exact method in a significantly shorter computational time than the exact method. The results show that the insertion heuristic is capable of solving instances with up to five requests, and when six requests are input, the computational time increases significantly. This is because the algorithm applies greedy insertion. Greedy insertion tries all possible insertion positions for the input requests, and the number of possible combinations increases drastically when the number of requests increases. The shortened computational time derives from the simplification of the route construction in the time domain. The decision variables in the exact method are x, y, and z. x and y are variables related to the travel of the vessels and the requests, respectively. It is necessary to determine if the requests are boarded during each trip, which x is equal to 1. This way of formulation creates another set of decision variables with a size equal to the number of x for each request. Therefore, the number of decision variables significantly grows when the number of requests increases. In addition, the scheduling of the routes, such as the departure time of each trip, is part of the variables in the exact method. The model needs to determine the schedule and the routing simultaneously in the exact method. These aspects increase the solution space to be extremely larger in the exact method than in the insertion heuristic.

In contrast, the proposed insertion heuristic constructs the scheduling of the vessels and the requests after the determination of the routes, based on simple assumptions mentioned in Section 3.5. These assumptions enabled the scheduling and the requests' boarding situations in each trip of the vessels to be deterministic. This is as if it removes the y variables and the time-related variables in the exact method, and it is expected to reduce the solution space significantly. Also, the insertion heuristic only needs to consider the new requests, and the previously assigned or serving requests are only considered in the scheduling construction. In contrast, the exact method takes assigned and serving requests as part of the decision. Although the status of the requests is assured by applying additional constraints to these requests and it contributes to reduce the solution space, the solution space is still large and it requires enormous computational time to find a solution. Finally, the model in the exact method duplicates the network to allow multiple visits to a terminal. The size of the network also affects the resources required to obtain the optimal solution.

In the experiment of the dynamic model, a maximum of three requests occurred in the same time step. As the experiment of static optimisation suggests, the insertion heuristic provides the route plan instantly when less than or equal to three requests are inserted. This led to extremely short computational time for each time step in the dynamic model. This result indicates that the proposed insertion heuristic with greedy insertion is suitable for this dynamic optimisation model with the rolling horizon, and the implementation of more efficient heuristic such as ALNS (Ghilas et al., 2016) would not bring significant improvement in the solving performance for the dynamic model. In contrast, the exact method faced many cases in which the solver did not manage to find a feasible solution within the time limit of one minute. As the performance of the static optimisation model suggests, the model is capable of reaching the optimal solution in cases of up to two or three unassigned requests. With all unassigned, assigned, and serving requests considered, there were mostly more than three requests inputted to the model at each time step, and it is logical that the model had many cases of not finding a feasible solution. Through the evaluation of the methods for solving the dynamic optimisation problem, it can be concluded that the proposed insertion heuristic is effective in providing a good solution for this problem in a short time.

The dynamic model with two solving methods were performed for two different demand scenarios and four different vessel type combinations to obtain insights into the efficiency and the service level of the capacity allocation of waterborne vessels for heterogeneous services. The results show a higher service level and efficiency in the high demand scenario when a mixed purpose vessel is available compared to when only conventional fixed purpose vessels are operating. Allowing the combination of passenger and parcel requests in the same vessel brings more flexibility when assigning requests to a vessel. More cases in which the request was assigned to the nearest vessel were possible thanks to the mixed purpose vessel and this contributed to rejecting fewer requests due to the time constraints because less time was required to pickup the request within the maximum waiting time for passenger requests. Also, vessels were able to pick up and deliver requests with fewer detours, which led to less TTD. TETD decreased by having mixed purpose vessels because they could serve passenger requests while having parcels on board since the parcel requests had looser time window than the passenger requests as long as the parcels were delivered before the required delivery time. Also, it is important to note that the passenger and parcel requests with the same pickup terminal could be picked up at the same time when a mixed purpose vessel is available. This contributed to not only increasing the RMR but also reducing TTD and TETD because the other vessel did not need to visit the same terminal to serve a specific type of request.

Similar to the high demand scenario, it was consistent that implementing mixed purpose vessels contributed to lower TETD in the low demand scenario. Apart from the TETD, there are no notable patterns in the KPIs for this demand scenario using the exact method. This can be because of the solving capability of the model. As shown in Table 4.6, there were many time steps in which the solver could not find a feasible solution or the optimal solution in this demand scenario. The solver especially struggled to find a feasible solution when multiple visits to a terminal were required to construct a feasible route plan. It was often the case that once the solver found a non-optimal feasible solution with multiple visits to a terminal, the solver could not update the route plan with new requests until the multiple visits to a terminal by pursuing the previous route plan were completed. The KPIs are highly dependent on whether the solver manages to find a feasible solution within the time limit. Otherwise, the new requests would be rejected, and route plans would not be updated at each time step. Therefore, it is important to note that the KPIs demonstrated in Table 4.6 are highly influenced by the solving capability of the model, and the influence of the dynamic capacity allocation to the efficiency and the service level of the system cannot be verified. In contrast, the results of the insertion heuristic provide several insights into transport performance. Operating the service with two mixed purpose vessels brought high RMR compared to the other vessel type combinations while maintaining the TTD and the TETD to be low. This result indicates the increase in the service level and the efficiency of the system by introducing dynamic capacity allocation for heterogeneous service. Also, the trade-off between the service level and the efficiency of the system is observed in the result. This is expected since it is natural that the vessels need to travel more to serve more requests.

In the experiment, more parcel requests were generated than passenger requests in both demand scenarios. Only 27.8% (high) and 20.9% (low) of the requests were passenger requests. Therefore, when a passenger-dedicated vessel was present, this vessel spent a long time dwelling. Meanwhile, the other vessel operated and served not only all the parcel requests but also the passenger requests throughout the whole time horizon by merging the service with parcel requests with the same origin/destination. It is important to note that this result is highly dependent on the demand. It is expected that if more passenger requests occur, the passenger-dedicated vessel have less dwelling time and serve the passenger requests more. In the experiment, more parcel requests were generated based on the assumption of the growth of e-commerce and business-related logistics demand compared to the modal shift of citizens from road transport to multi-modal transport, including waterways.

5.2. Discussion of assumptions and limitations

The model for dynamic centralised fleet management is developed with several assumptions. The assumptions, their influence on the results, and the possible improvement methods are discussed in this section.

The dynamic optimisation model was solved in a rolling horizon manner, which solves a static subproblem at the time step when a new request arrives. In the static model, the problem was formulated based on multiple assumptions. The first assumption is that the objective function is defined to minimise the total travel distance. This objective led to the usage of the vessels to be imbalanced throughout the time horizon, as discussed before. In real life, it would not be ideal to operate only one of the vessels out of two heavily, considering the long-term effect of heavy usage (e.g. vessel breakdown likelihood, battery degradation by repetitive charging and exploiting). It would be better to operate all vessels in a good balance to have longer lifetime usage and reduce the risk of vessel breakdown. As mentioned in Chapter 2, the objective function of these PDPs and their variants can vary depending on the purpose of the study. For instance, minimising the discrepancy of loading level between vessels can lead to the assignments of requests to vessels to be distributed as equally as possible between vessels. This may avoid the imbalanced usage of vessels. Another possibility is to maximise the profit of the service company as Su et al., (2022) did. The company is able to determine the operation of vessels to maximise the profit of the company by considering the revenue from logistic companies by serving parcels and also from passenger customers and the operation costs such as travel costs and charging costs.

Another assumption is related to the stochasticity regarding the vessel operation. The vessel speed is assumed to be constant in the model and the associated aspects such as travel time and the battery consumption between two nodes are assumed to be deterministic. In real life, the speed of vessels is not constant. Vessels' speed changes due to many factors, such as the necessity of acceleration/deceleration when approaching or departing a terminal, the weather conditions, and the loading situation. The stochasticity of the speed influences the travel time and the battery consumption between two nodes, and the travel time significantly influences the acceptance/rejection of travel requests since it determines if the vessels manage to serve the requests while ensuring the time windows. Some literature, such as Li et al. (2016) and Wang et al. (2023), incorporated the stochasticity of travel times in a static PDP. However, as far as I know, there are no studies considering the stochasticity in travel times in a dynamic setting for a ferry system. Not only the speed of the vessel but also the distance between two nodes is an important assumption made in this thesis. The distance between two nodes is assumed to be the great-circle distance calculated by the coordinates of the two nodes. This assumption is applied to obtain the travel times. The model will be more realistic if the actual path on the waterways is implemented and the distance/travel times are defined based on the realistic network. Similar to the vessel speed assumption, this affects the capability of reaching terminals within a certain time and would affect the route plan of vessels. In the model, the travel times are underestimated because the distance between two nodes is underestimated. This means the vessels were able to satisfy the time constraint of requests more easily than they would have been if the actual waterways had been considered. Regarding the KPIs, TTD and TETD are underestimated, and the RMR are likely to be overestimated in this model.

The assumptions are made in the request generation as well. The waterborne vessel system for heterogeneous on-demand service is a conceptual system that has not yet been implemented in real life. Therefore, there was no data regarding the demand for the service available. In order to incorporate the stochasticity of the demand in the model, a non-homogeneous Poisson process was applied to generate the request. The parameters in the process were arbitrarily defined by considering little information about the urban demographic and land use of the case location. As the output of the model highly depends on the requests, it is significant to produce a realistic demand dataset to accurately capture the performance of the transport system. Therefore, it is important to collect historical data in order to define realistic parameters for stochastic demand generation.

In the dynamic optimisation model, the requests are inserted into the model gradually and the routing decisions are immediately made every time a new request is inserted to serve the new requests. This approach implicitly prioritises the early requests and the upcoming requests' acceptance/rejection is dependent on the earlier requests. Therefore, despite the developed model optimising the route of vessels at each time step, it does not mean that the service level and the efficiency of the system are optimised for the entire time horizon. For instance, vessels can result in rejecting many requests to serve a single request which occurred earlier than the others, or may need to make extra travels to serve the new requests at the terminal where the vessels departed to serve an earlier request. It is possible to try to incorporate the anticipation of future requests to improve the performance of the transport system. The waiting strategy is one of the common methods used to incorporate the anticipation of future requests. Several waiting strategies are developed for dynamic pick up and delivery problems, such as the wait-first strategy, in which the vehicles wait at their current location for as long as the requests and the route is feasible (Mitrović-Minić and Laporte, 2004).

Related to the anticipation of future requests, the availability of historical data will enable the development of a leaning-based method in the dynamic optimisation model (Cai et al., 2023). Especially, the temporal and spatial pattern in the demand data can be utilised to learn the optimal behaviour of vessels considering the entire time horizon. For instance, the vessels can learn the peak hours and the off-peak hours and wisely decide the acceptance/rejection, the assignment of requests to vessels, and the vessels' route plan so most of the passenger requests during the peak hour can be served, or to postpone the parcel requests to make sure enough capacity is available for the passenger peak hours. Learning-based methods, such as machine learning and reinforcement learning, are capable of deriving the optimal strategies of the operation through training based on historical data. Several studies have implemented learning-based method for vehicle routing problem and its variants (Gao et al., 2024), (Pan and Liu, 2023).

6
Conclusion

This chapter contains the conclusion of this thesis and the recommendation for future application and research. This thesis aimed to develop a model to determine the dispatching of vessels for heterogeneous on-demand service, considering the stochasticity of the demand. In addition, it aimed to investigate the efficiency and the service level of the system by comparing the capacity allocation system by mixed purpose vessels with the conventional fixed purpose vessel system.

A model which dynamically optimises the fleet operation given the generated demand data is developed in this thesis. In demand generation, a non-homogeneous Poisson process was applied to passenger requests to incorporate the temporal and spatial pattern of demand, and a probabilistic approach was also used to reflect the characteristics of parcel deliveries in real life. Given the two generated demand scenarios, the model dynamically solves the dispatching of the waterborne vessels in a rolling horizon manner. Two solving algorithms, the exact method and the insertion heuristic, were proposed to solve the static subproblem at each time step. The output of the experiments was analysed to investigate the solving capability of each algorithm in static and dynamic settings. Also, the output including the KPIs was analysed to obtain insights into the efficiency and the service level of the transport system with mixed purpose fleet compared to the conventional fixed purpose fleet.

In the remaining of this chapter, the answers to the research questions are provided in Section 6.1. Afterwards, recommendations for future application and research are presented in Section 6.2 and Section 6.3.

6.1. Answers to the research questions

1) **How can the dispatching of vessels be determined so that the efficiency of the waterborne vessels system for heterogeneous on-demand service is maximised, considering the stochasticity of the demand?**

   The developed model optimises dispatching the waterborne vessels for heterogeneous on-demand service and the associated route plans so the total travel distance is minimised. Minimising the total travel distance contributes to maximising the efficiency of the system for the fleet. The demand stochasticity was considered by generating individual travel and delivery requests by applying a non-homogeneous Poisson process for passenger requests and a probabilistic approach for parcel requests. The parameters were defined to incorporate the spatial and temporal demand patterns for the services. Taking the generated set of travel and delivery requests as input, the model dynamically optimises the route plan of each vessel to serve requests aiming to minimise the total travel distance of the route plan in a rolling horizon manner. A static subproblem, which contains various constraints related to the capacity, battery level, and the time window, is solved at each time step a new request occurs to determine the new route plan of vessels considering the new requests. Two solving algorithms, the exact method and the insertion heuristic, were proposed to solve the static subproblem. The solving capability and quality of both algorithms were evaluated through the experiment. The insertion heuristic demonstrated its capability of providing good solutions in a significantly shorter computational time than the exact method.

2) **To what extent does the mixture of capacity in vessels improve the efficiency and the service level of the waterborne vessels system?**

The model provides the KPIs related to the efficiency and the service level of the waterborne vessel system. The total travel distance (TTD) and the total empty travel distance (TETD) were defined as the indicators for the efficiency of the service, and the request met ratio (RMR) was defined as the indicator relating to the service level. These KPIs were collected for each configuration, which is a combination of the demand scenario and vessel type combinations, through the case study in Fredrikstad, Norway. The RMR suggests that introducing mixed purpose vessels improves the service level in both demand scenarios as the RMR is higher for (mixed, mixed) than that of (parcel, passenger) by around 6-13 points. Also, the TETD decreases by 12.5-53.1% by introducing mixed purpose vessels compared to the fixed purpose vessels. This is derived from the reduction of empty trips for the vessels to pick up a specific type of request. TTD did not necessarily decrease from fixed purpose vessels when mixed purpose vessels were introduced. This is because of the trade-off observed between RMR and TTD/TETD. The travel distance increases when more requests are served, and the output verifies this trade-off. However, considering the higher RMR for (mixed, mixed) than that of (parcel, passenger) and (mixed, mixed) recording in less TETD and only a slight increment in TTD shows the maintaining the efficiency of the system while providing a high service level. In conclusion, the mixture of capacity in vessels significantly improved both the efficiency and the service level of the waterborne vessel system for heterogeneous on-demand service compared to conventional fixed purpose vessels in the experimented demand scenarios. However, transportation performance is case-specific since it depends highly on demand. Therefore, it is important to accurately capture the demand in practice to obtain insights into transportation performance in real life.

6.2. Recommendations for future application

This section presents some recommendations for the future application of the considered waterborne vessel system in real life. This thesis demonstrated that providing heterogeneous service with a mixture of capacities in vessels improves the efficiency and service level compared to conventional fixed purpose vessels. This result suggests that the current urban ferry systems can utilise their excess capacity to provide logistics services. This will benefit the service-operating company by obtaining profit from delivery companies and the delivery companies by potentially reducing costs by implementing crowd shipping.

It is important to provide a platform for customers so they can easily make requests and the system can quickly determine the vessels' routes accordingly. The proposed insertion heuristic showed its capability of providing good solutions in a very short time. Thus, this algorithm can be a sufficient approach in the system to start the service until more advanced methods can be applied. The platform also enables the service provider to collect the demand data. The historical data of demand would provide the potential for a data-driven approach to determine the operation of vessels for better efficiency and service level for the whole day. However, it is significant to carefully consider the privacy and the security of the data since the collected data will be able to identify individuals.

6.3. Recommendations for future research

The model was developed based on several assumptions as discussed in Section 5.2. The assumptions are mostly related to the network topology and the vessel property which mostly assumed them to be deterministic and homogeneous. These assumptions can be relaxed to set up a more realistic setting in terms of the operation of vessels. The set up of a geographically accurate network and stochasticity in the vessel property and behaviour can possibly be incorporated into the model. In addition, the unavailability of historical data regarding the demand for the service led the model to create a synthetic request dataset arbitrarily. As mentioned before, the results obtained in the experiments are highly dependent on the demand scenario. Therefore, it is significant to provide accurate demand data to the model to accurately capture the impact of dynamic capacity allocation on transportation efficiency and service level. Also, the historical data enables a data-driven approach to optimise the vessel's operation. Learning-based methods are especially expected to be powerful since they can learn from historical data to make decisions about operations by incorporating the anticipation of future demand, as well as the temporal and spatial demand patterns.

Bibliography

Alonso-González, M. J., Liu, T., Cats, O., Van Oort, N., & Hoogendoorn, S. (2018). The potential of demand-responsive transport as a complement to public transport: An assessment framework and an empirical evaluation [Publisher: SAGE Publications Inc]. *Transportation Research Record*, *2672*(8), 879–889. https://doi.org/10.1177/0361198118790842

Arslan, A. M., Agatz, N., Kroon, L., & Zuidwijk, R. (2019). Crowdsourced delivery—a dynamic pickup and delivery problem with ad hoc drivers. *Transportation Science*, *53*(1), 222–235. https://doi.org/10.1287/trsc.2017.0803

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

Behiri, W., Belmokhtar-Berraf, S., & Chu, C. (2018). Urban freight transport using passenger rail network: Scientific issues and quantitative analysis. *Transportation Research Part E: Logistics and Transportation Review*, *115*, 227–245. https://doi.org/10.1016/j.tre.2018.05.002

Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., & Laporte, G. (2007). Static pickup and delivery problems: A classification scheme and survey. *TOP*, *15*(1), 1–31. https://doi.org/10.1007/s11750-007-0009-0

Bertsimas, D., Jaillet, P., & Martin, S. (2019). Online vehicle routing: The edge of optimization in large-scale applications. *Operations Research*, *67*(1), 143–162. https://doi.org/10.1287/opre.2018.1763

Cai, J., Zhu, Q., Lin, Q., Ma, L., Li, J., & Ming, Z. (2023). A survey of dynamic pickup and delivery problems. *Neurocomputing*, *554*, 126631. https://doi.org/10.1016/j.neucom.2023.126631

Chen, H., Hatzenbühler, J., & Jenelius, E. (2022). Pick-up and delivery problem for sequentially consolidated urban transportation with mixed and multi-pupropse vehicle fleet. *Journal of Advanced Transportation*, *2022*, 1–18. https://doi.org/10.1155/2022/2920532

Cleophas, C., Cottrill, C., Ehmke, J. F., & Tierney, K. (2019). Collaborative urban transportation: Recent advances in theory and practice. *European Journal of Operational Research*, *273*(3), 801–816. https://doi.org/10.1016/j.ejor.2018.04.037

Cordeau, J.-F. (2006). A branch-and-cut algorithm for the dial-a-ride problem. *Operations Research*, *54*(3), 573–586. https://doi.org/10.1287/opre.1060.0283

Cordeau, J.-F., & Laporte, G. (2007). The dial-a-ride problem: Models and algorithms. *Annals of Operations Research*, *153*(1), 29–46. https://doi.org/10.1007/s10479-007-0170-8

EuropeanCommision. (2021). *Moving freight on public transit: Best practices, challenges, and opportunities*. Retrieved November 28, 2023, from https://www.tandfonline.com/doi/epdf/10.1080/15568318.2016.1197349?src=getftr

Francesco, P., Luciano, F., Silvia, R. T., & Mario, P. (2013). Horizontal collaboration in logistics: A comprehensive framework. *Research in Logistics Production*. https://www.academia.edu/17369048/Horizontal_Collaboration_in_Logistics_A_Comprehensive_Framework

Frederikstad Kommune. (2022). *Fakta om fredrikstad*. Retrieved December 13, 2023, from https://www.fredrikstad.kommune.no/tjenester/om-fredrikstad/fakta-om-fredrikstad/

Fredrikstad - SUM. (2024). Retrieved June 3, 2024, from https://sum-project.eu/living-labs/fredrikstad/

Gao, Y., Zhang, S., Zhang, Z., & Zhao, Q. (2024). The stochastic share-a-ride problem with electric vehicles and customer priorities. *Computers & Operations Research*, *164*, 106550. https://doi.org/10.1016/j.cor.2024.106550

Geiser, T., Hanne, T., & Dornberger, R. (2020). Best-match in a set of single-vehicle dynamic pickup and delivery problem using ant colony optimization. *Proceedings of the 2020 the 3rd International Conference on Computers in Management and Business*, 126–131. https://doi.org/10.1145/3383845.3383879

Ghilas, V., Demir, E., & Van Woensel, T. (2016). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows and scheduled lines. *Computers & Operations Research*, *72*, 12–30. https://doi.org/10.1016/j.cor.2016.01.018

Hatzenbühler, J., Jenelius, E., Gidófalvi, G., & Cats, O. (2024). Multi-purpose pickup and delivery problem for combined passenger and freight transport. *Transportation*. https://doi.org/10.1007/s11116-024-10482-9

Hyke [Hyke]. (2021, November 24). Retrieved June 4, 2024, from https://hyke.no/

Kamen, P., & Barry, C. D. (2011). *Urban passenger-only ferry systems: Issues, opportunities and technologies*. Retrieved June 7, 2024, from https://trid.trb.org/View/1139746

Karami, F., Vancroonenburg, W., & Vanden Berghe, G. (2020). A periodic optimization approach to dynamic pickup and delivery problems with time windows. *Journal of Scheduling*, *23*(6), 711–731. https://doi.org/10.1007/s10951-020-00650-x

Kommuneplanens arealdel 2023-2035. (2023, June 15). Retrieved June 4, 2024, from https://www.fredrikstad.kommune.no/globalassets/dokumenter/planer/arealplan/planbeskrivelse-med-bestemmelser-og-retningslinjer-kpa2023.pdf

Koyuncu, I., & Yavuz, M. (2019). Duplicating nodes or arcs in green vehicle routing: A computational comparison of two formulations. *Transportation Research Part E: Logistics and Transportation Review*, *122*, 605–623. https://doi.org/10.1016/j.tre.2018.11.003

Le, T. V., Stathopoulos, A., Van Woensel, T., & Ukkusuri, S. V. (2019). Supply, demand, operations, and management of crowd-shipping services: A review and empirical evidence. *Transportation Research Part C: Emerging Technologies*, *103*, 83–103. https://doi.org/10.1016/j.trc.2019.03.023

Li, B., Krushinsky, D., Reijers, H. A., & Van Woensel, T. (2014). The share-a-ride problem: People and parcels sharing taxis. *European Journal of Operational Research*, *238*(1), 31–40. https://doi.org/10.1016/j.ejor.2014.03.003

Li, B., Krushinsky, D., Van Woensel, T., & Reijers, H. A. (2016). The share-a-ride problem with stochastic travel times and stochastic delivery locations. *Transportation Research Part C: Emerging Technologies*, *67*, 95–108. https://doi.org/10.1016/j.trc.2016.01.014

Li, F., Guo, X., Zhou, L., Wu, J., & Li, T. (2022). A capacity matching model in a collaborative urban public transport system: Integrating passenger and freight transportation. *International Journal of Production Research*, *60*(20), 6303–6328. https://doi.org/10.1080/00207543.2021.1991021

Liu, X., Wu, J., Huang, J., Zhang, J., Chen, B. Y., & Chen, A. (2021). Spatial-interaction network analysis of built environmental influence on daily public transport demand. *Journal of Transport Geography*, *92*, 102991. https://doi.org/10.1016/j.jtrangeo.2021.102991

Martin, S., & Tom, V. W. (2016). City logistics: Challenges and opportunities. *Transportation Science*, *50*(2), 579–590. https://doi.org/10.1287/trsc.2016.0675

Masson, R., Trentini, A., Lehuédé, F., Malhéné, N., Péton, O., & Tlahig, H. (2017). Optimization of a city logistics transportation system with mixed passengers and goods. *Journal on Transportation and Logistics*, *6*(1), 81–109. https://doi.org/10.1007/s13676-015-0085-5

Mitrović-Minić, S., & Laporte, G. (2004). Waiting strategies for the dynamic pickup and delivery problem with time windows. *Transportation Research Part B: Methodological*, *38*(7), 635–655. https://doi.org/10.1016/j.trb.2003.09.002

Pan, W., & Liu, S. Q. (2023). Deep reinforcement learning for the dynamic and uncertain vehicle routing problem. *Applied Intelligence*, *53*(1), 405–422. https://doi.org/10.1007/s10489-022-03456-w

Pimentel, C., & Alvelos, F. (2018). Integrated urban freight logistics combining passenger and freight flows ‑mathematical model proposal. *Transportation Research Procedia*, *30*, 80–89. https://doi.org/10.1016/j.trpro.2018.09.010

Robert, G. (2011). *Course notes | discrete stochastic processes | electrical engineering and computer science* [MIT OpenCourseWare]. Retrieved May 20, 2024, from https://ocw.mit.edu/courses/6-262-discrete-stochastic-processes-spring-2011/pages/course-notes/

Savelsbergh, M. W. P., & Sol, M. (1995). The general pickup and delivery problem. *Transportation Science*, *29*(1), 17–29. https://doi.org/10.1287/trsc.29.1.17

Sina Mohri, S., Ghaderi, H., Nassir, N., & Thompson, R. G. (2023). Crowdshipping for sustainable urban logistics: A systematic review of the literature. *Transportation Research Part E: Logistics and Transportation Review*, *178*, 103289. https://doi.org/10.1016/j.tre.2023.103289

Sitek, P., Wikarek, J., Rutczyńska-Wdowiak, K., Bocewicz, G., & Banaszak, Z. (2021). Optimization of capacitated vehicle routing problem with alternative delivery, pick-up and time windows: A modified hybrid approach. *Neurocomputing*, *423*, 670–678. https://doi.org/10.1016/j.neucom.2020.02.126

Stokkink, P., & Geroliminis, N. (2023). A continuum approximation approach to the depot location problem in a crowd-shipping system. *Transportation Research Part E: Logistics and Transportation Review*, *176*, 103207. https://doi.org/10.1016/j.tre.2023.103207

Su, Z., Li, W., Li, J., & Cheng, B. (2022). Heterogeneous fleet vehicle scheduling problems for dynamic pickup and delivery problem with time windows in shared logistics platform: Formulation, instances and algorithms. *International Journal of Systems Science: Operations & Logistics*, *9*(2), 199–223. https://doi.org/10.1080/23302674.2020.1865475

Sun, B., Yang, Y., Shi, J., & Zheng, L. (2019). Dynamic pick-up and delivery optimization with multiple dynamic events in real-world environment. *IEEE Access*, *7*(1), 146209–146220. https://doi.org/10.1109/ACCESS.2019.2944739

Ulmer, M. W., Thomas, B. W., Campbell, A. M., & Woyak, N. (2021). The restaurant meal delivery problem: Dynamic pickup and delivery with deadlines and random ready times. *Transportation Science*, *55*(1), 75–100. https://doi.org/10.1287/trsc.2020.1000

Vonolfen, S., & Affenzeller, M. (2016). Distribution of waiting time for dynamic pickup and delivery problems. *Annals of Operations Research*, *236*(2), 359–382. https://doi.org/10.1007/s10479-014-1683-6

Wang, Z., Dessouky, M., Van Woensel, T., & Ioannou, P. (2023). Pickup and delivery problem with hard time windows considering stochastic and time-dependent travel times. *EURO Journal on Transportation and Logistics*, *12*, 100099. https://doi.org/10.1016/j.ejtl.2022.100099

Yu, V. F., Marye Zegeye, M., Geremew Gebeyehu, S., Indrakarna, P. A. Y., & Jodiawan, P. (2023). The multi-depot general share-a-ride problem. *Expert Systems with Applications*, *213*, 119044. https://doi.org/10.1016/j.eswa.2022.119044

Zhan, X., Szeto, W. Y., & Wang, Y. (2023). The ride-hailing sharing problem with parcel transportation. *Transportation Research Part E: Logistics and Transportation Review*, *172*, 103073. https://doi.org/10.1016/j.tre.2023.103073

Zhang, Y., Negenborn, R. R., & Atasoy, B. (2023). Synchromodal freight transport re-planning under service time uncertainty: An online model-assisted reinforcement learning. *Transportation Research Part C: Emerging Technologies*, *156*, 104355. https://doi.org/10.1016/j.trc.2023.104355

A

Parameters for request generation module

In this appendix, the parameters applied in the request generation module in the computational experiments are presented. Two demand scenarios, ”high” and ”low” were defined. Each demand scenario had different parameters for the non-homogeneous Poisson process of passenger requests. The non-homogeneity is derived from different parameters for each OD pair and time period. In the experiment, 6:00-7:00, 9:30-16:00, and 18:30-22:00 were defined as off-peak hours. In contrast, 6:30-9:00 and 16:00-18:30 were defined as peak hours.

A.1. High demand scenario

Table A.1: Arrival rate for each OD pair in off-peak time periods (high demand)

O/D	0	1	2	3	4	5	6	7
0	0	0.0006	0.0006	0.0006	0	0.0012	0.0006	0.0006
1	0.0006	0	0.0006	0.0006	0.0012	0.0012	0.0006	0.0006
2	0.0006	0.0006	0	0	0.0012	0.0012	0.0006	0.0006
3	0.0006	0.0006	0	0	0.0012	0	0.0006	0.0006
4	0	0.0012	0.0012	0.0012	0	0.0018	0.0012	0.0012
5	0.0012	0.0012	0.0012	0	0.0018	0	0.0012	0.0012
6	0.0006	0.0006	0.0006	0.0006	0.0012	0.0012	0	0
7	0.0006	0.0006	0.0006	0.0006	0.0012	0.0012	0	0

Table A.2: Arrival rate for each OD pair in peak time periods (high demand)

O/D	0	1	2	3	4	5	6	7
0	0	0.0009	0.0009	0.0009	0	0.0018	0.0009	0.0009
1	0.0009	0	0.0009	0.0009	0.0018	0.0018	0.0009	0.0009
2	0.0009	0.0009	0	0	0.0018	0.0018	0.0009	0.0009
3	0.0009	0.0009	0	0	0.0018	0	0.0009	0.0009
4	0	0.0018	0.0018	0.0018	0	0.0027	0.0018	0.0018
5	0.0018	0.0018	0.0018	0	0.0027	0	0.0018	0.0018
6	0.0009	0.0009	0.0009	0.0009	0.0018	0.0018	0	0
7	0.0009	0.0009	0.0009	0.0009	0.0018	0.0018	0	0

A.2. Low demand scenario

Table A.3: Arrival rate for each OD pair in off-peak time periods (low demand)

O/D	0	1	2	3	4	5	6	7
0	0	0.0004	0.0004	0.0004	0	0.0008	0.0004	0.0004
1	0.0004	0	0.0004	0.0004	0.0008	0.0008	0.0004	0.0004
2	0.0004	0.0004	0	0	0.0008	0.0008	0.0004	0.0004
3	0.0004	0.0004	0	0	0.0008	0	0.0004	0.0004
4	0	0.0008	0.0008	0.0008	0	0.0012	0.0008	0.0008
5	0.0008	0.0008	0.0008	0	0.0012	0	0.0008	0.0008
6	0.0004	0.0004	0.0004	0.0004	0.0008	0.0008	0	0
7	0.0004	0.0004	0.0004	0.0004	0.0008	0.0008	0	0

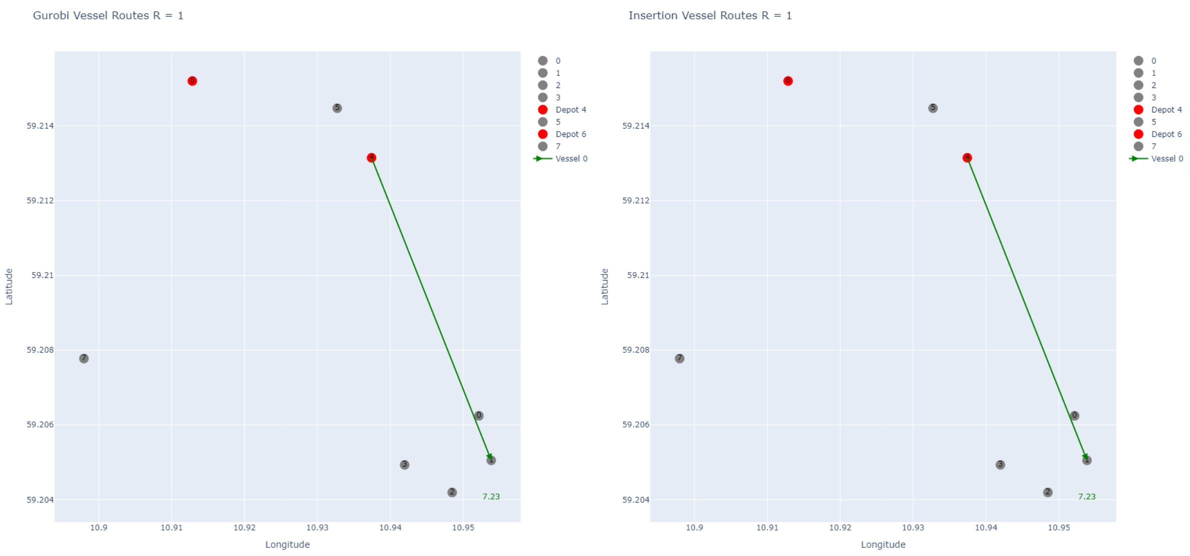
Table A.4: Arrival rate for each OD pair in peak time periods (low demand)

O/D	0	1	2	3	4	5	6	7
0	0	0.0006	0.0006	0.0006	0	0.0012	0.0006	0.0006
1	0.0006	0	0.0006	0.0006	0.0012	0.0012	0.0006	0.0006
2	0.0006	0.0006	0	0	0.0012	0.0012	0.0006	0.0006
3	0.0006	0.0006	0	0	0.0012	0	0.0006	0.0006
4	0	0.0012	0.0012	0.0012	0	0.0018	0.0012	0.0012
5	0.0012	0.0012	0.0012	0	0.0018	0	0.0012	0.0012
6	0.0006	0.0006	0.0006	0.0006	0.0012	0.0012	0	0
7	0.0006	0.0006	0.0006	0.0006	0.0012	0.0012	0	0

B

Obtained solutions from the static subproblem optimisation

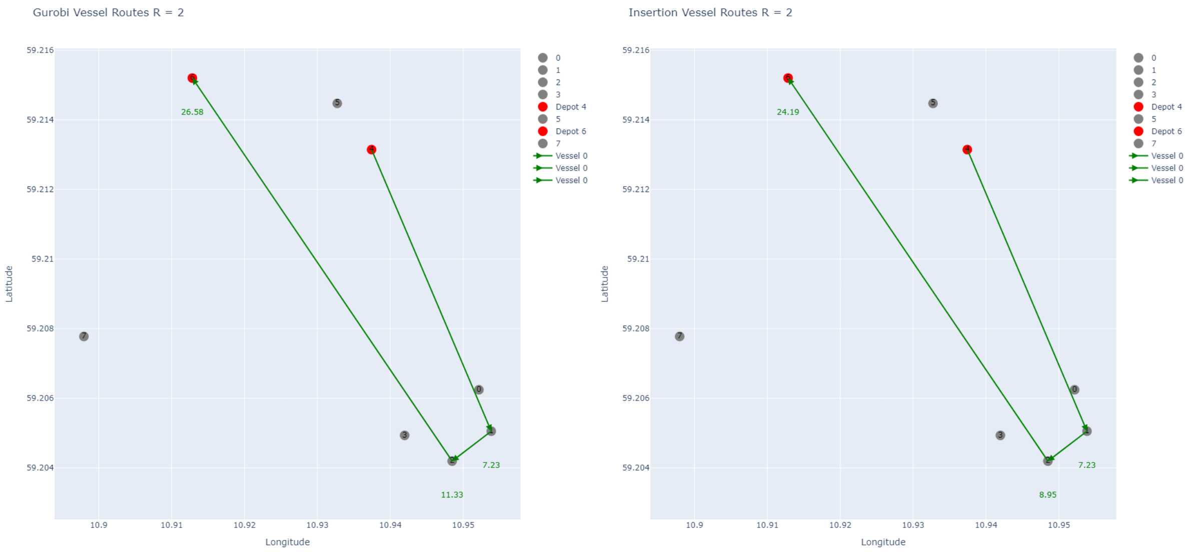
The obtained route plans from the static experiment are presented. The numbers below the visited terminals show the arrival time at each terminal. It can be seen that the visiting orders of terminals are the same between the exact method and the insertion heuristic in all instances. However, the arrival time at each terminal differs in some instances.



(a) Exact method

(b) Insertion heuristic

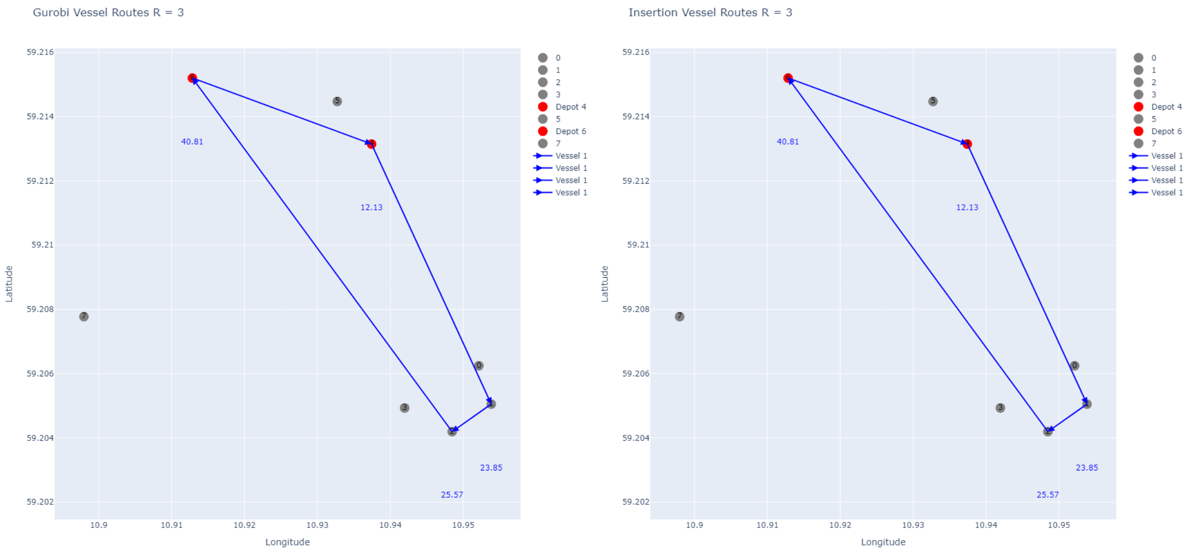
Figure B.1: Obtained route plan from each solving method, $|K| = 2, |R| = 1$



(a) Exact method

(b) Insertion heuristic

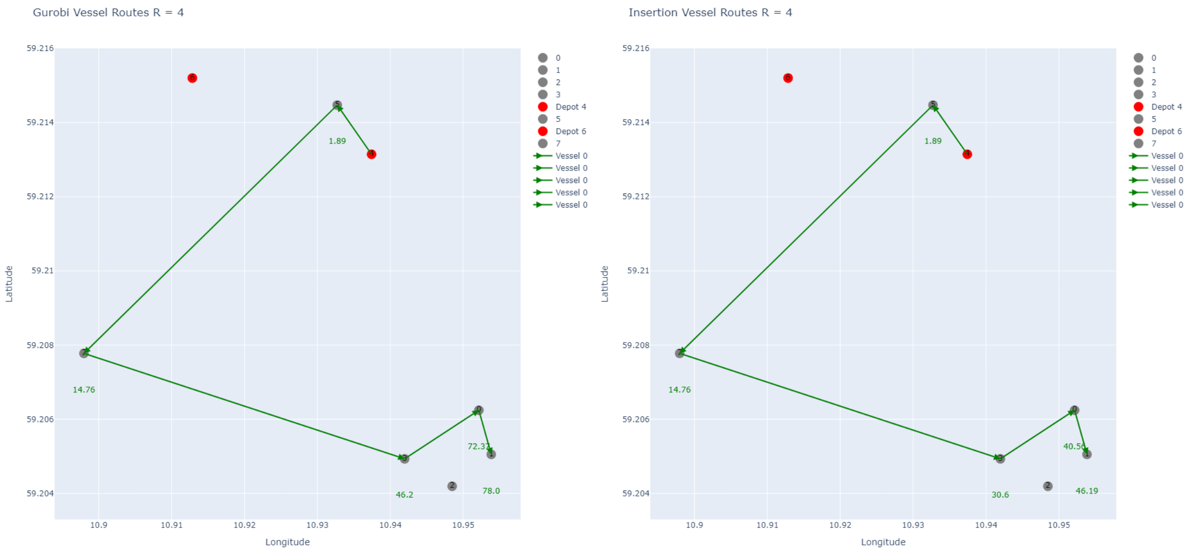
Figure B.2: Obtained route plan from each solving method, $|K| = 2, |R| = 2$



(a) Exact method

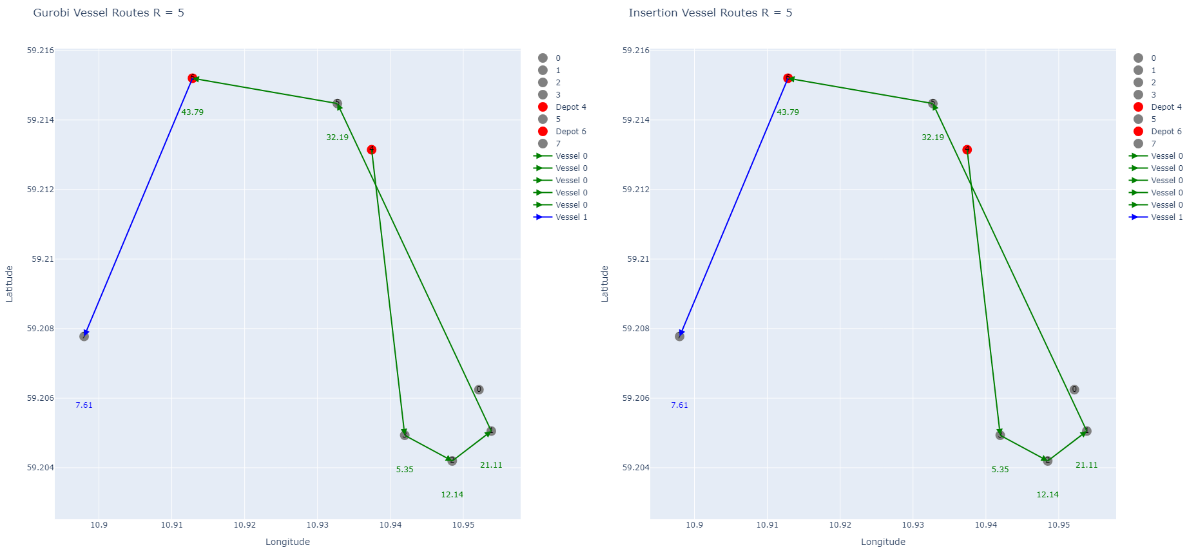
(b) Insertion heuristic

Figure B.3: Obtained route plan from each solving method, $|K| = 2, |R| = 3$



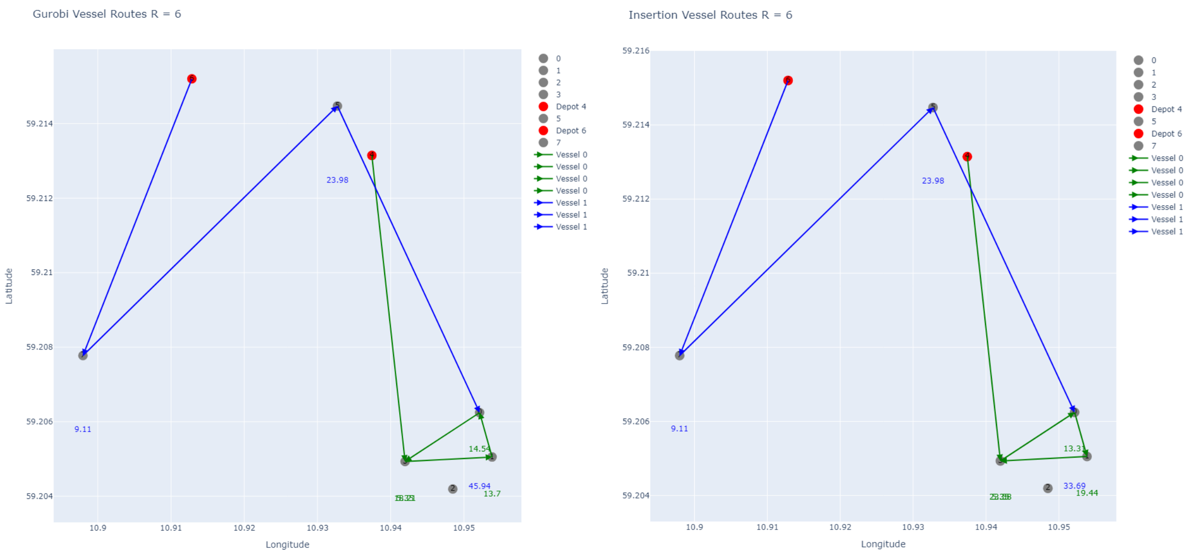
(a) Exact method **(b)** Insertion heuristic

Figure B.4: Obtained route plan from each solving method, $|K| = 2, |R| = 4$



(a) Exact method **(b)** Insertion heuristic

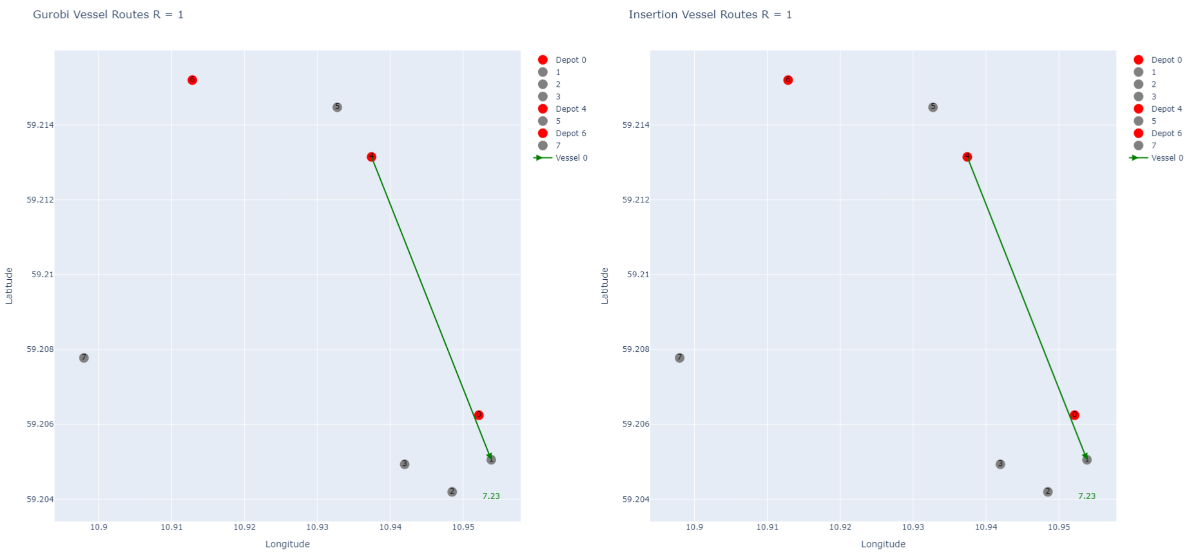
Figure B.5: Obtained route plan from each solving method, $|K| = 2, |R| = 5$



(a) Exact method

(b) Insertion heuristic

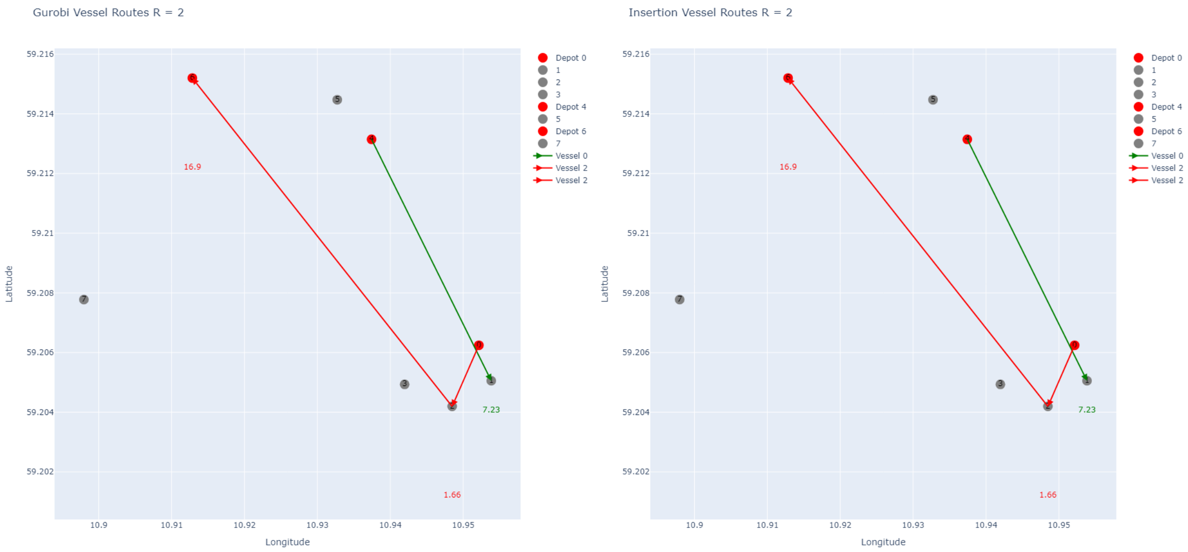
Figure B.6: Obtained route plan from each solving method, $|K| = 2, |R| = 6$



(a) Exact method

(b) Insertion heuristic

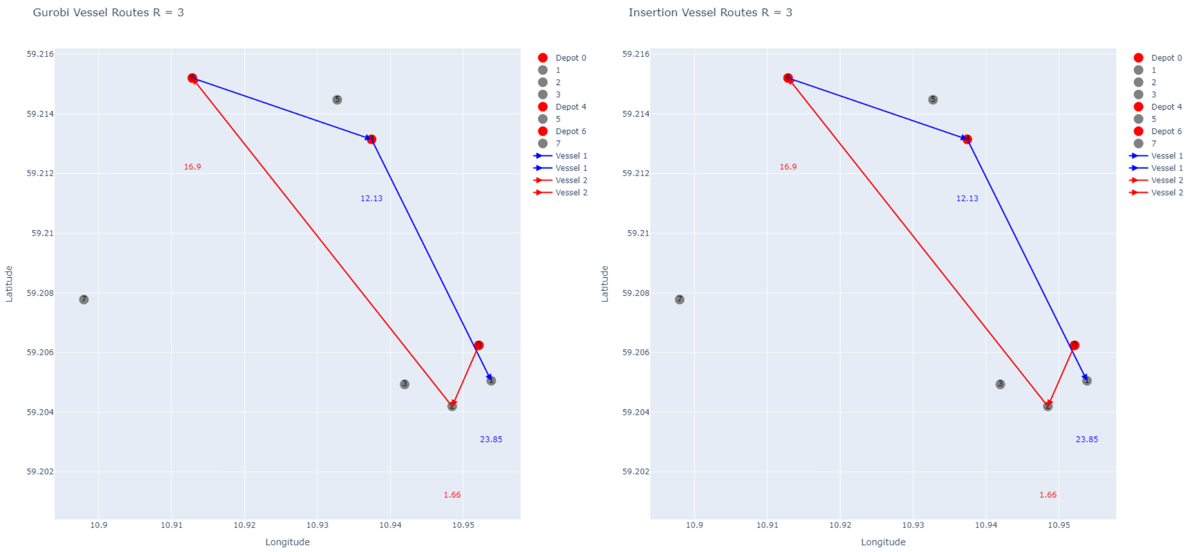
Figure B.7: Obtained route plan from each solving method, $|K| = 3, |R| = 1$



(a) Exact method

(b) Insertion heuristic

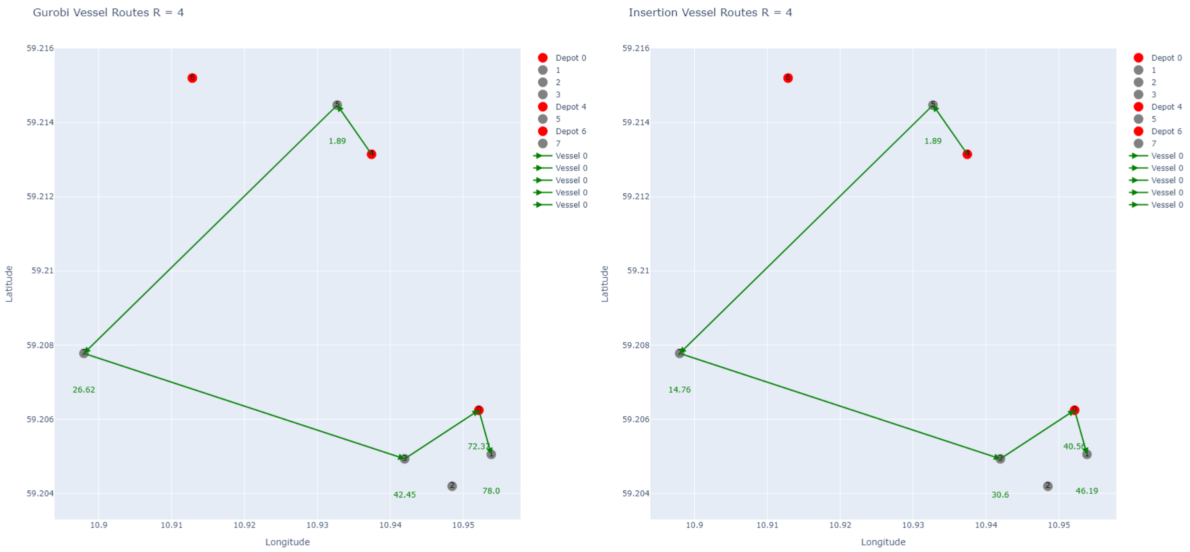
Figure B.8: Obtained route plan from each solving method, $|K| = 3, |R| = 2$



(a) Exact method

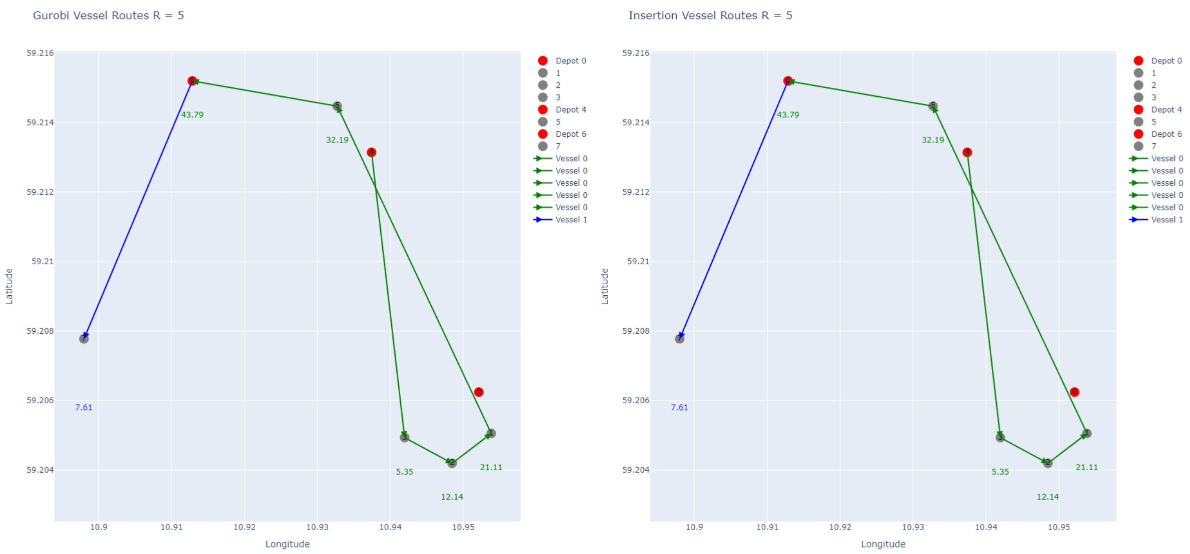
(b) Insertion heuristic

Figure B.9: Obtained route plan from each solving method, $|K| = 3, |R| = 3$



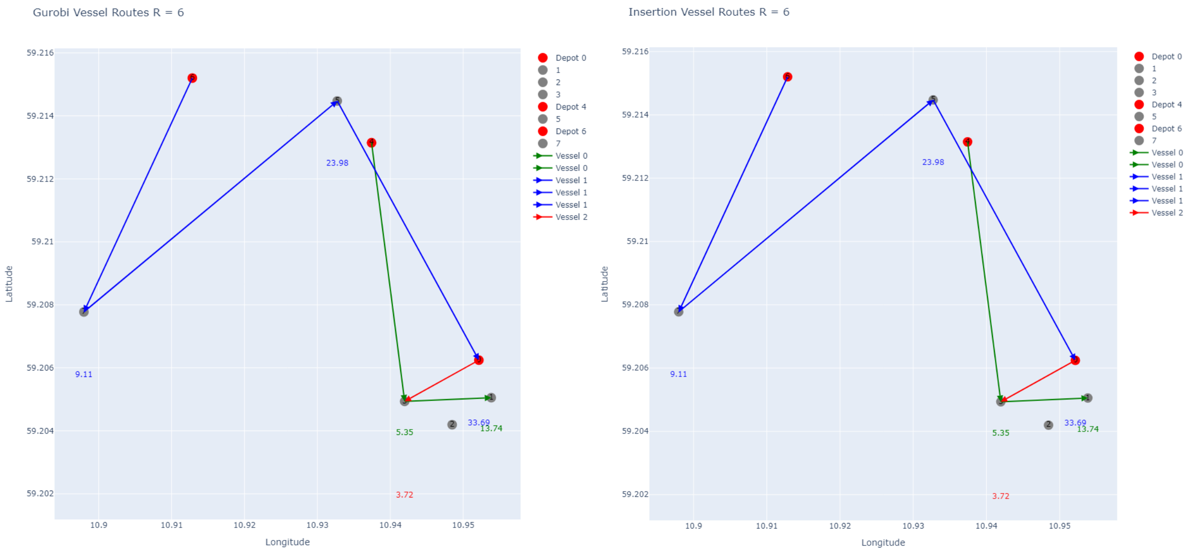
(a) Exact method **(b)** Insertion heuristic

Figure B.10: Obtained route plan from each solving method, $|K| = 3, |R| = 4$



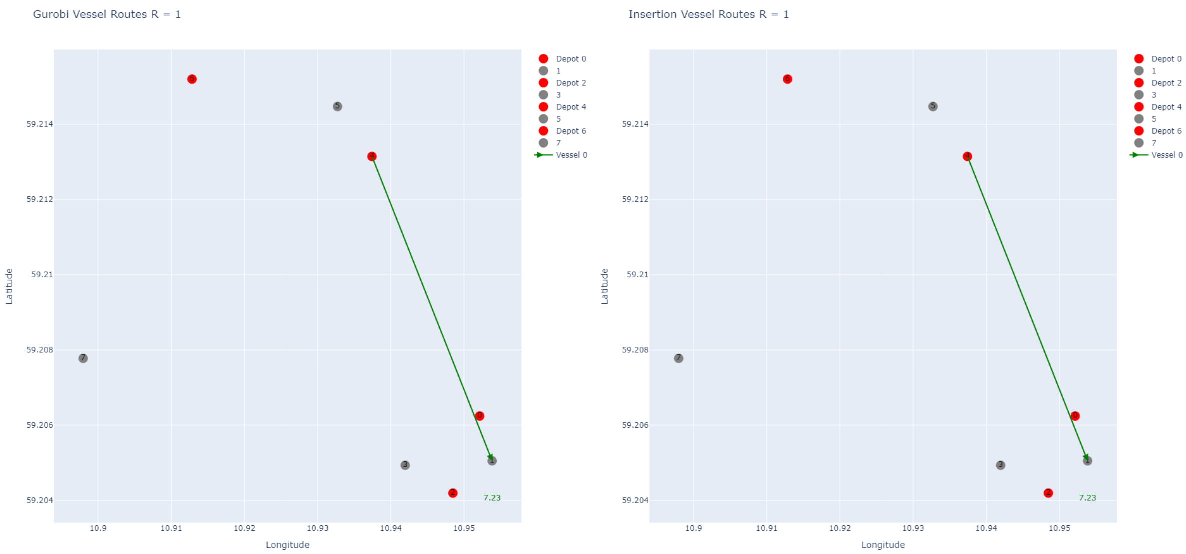
(a) Exact method **(b)** Insertion heuristic

Figure B.11: Obtained route plan from each solving method, $|K| = 3, |R| = 5$



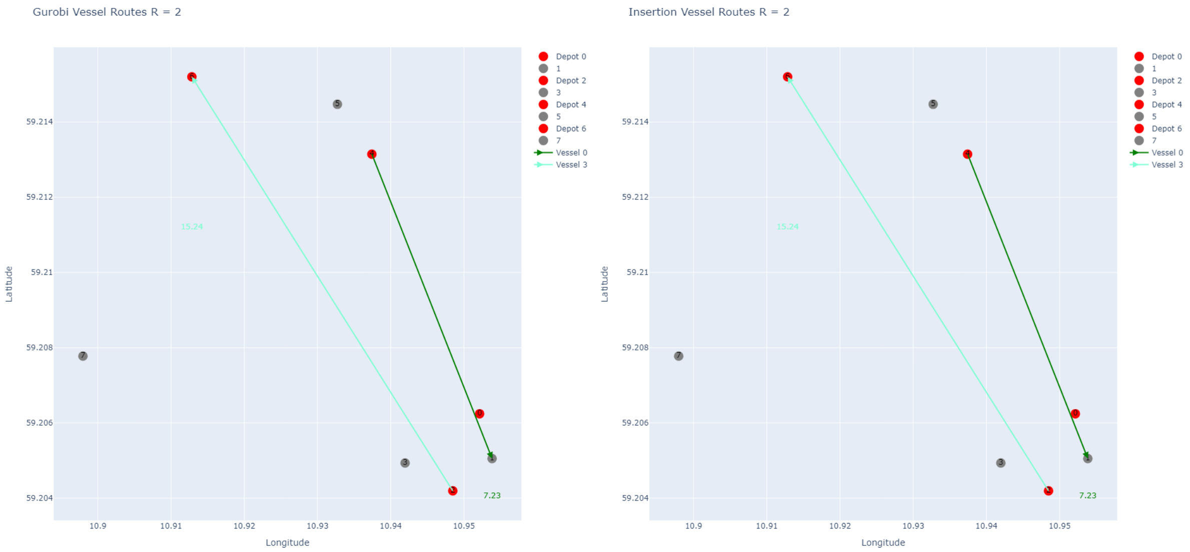
(a) Exact method **(b)** Insertion heuristic

Figure B.12: Obtained route plan from each solving method, $|K| = 3, |R| = 6$



(a) Exact method **(b)** Insertion heuristic

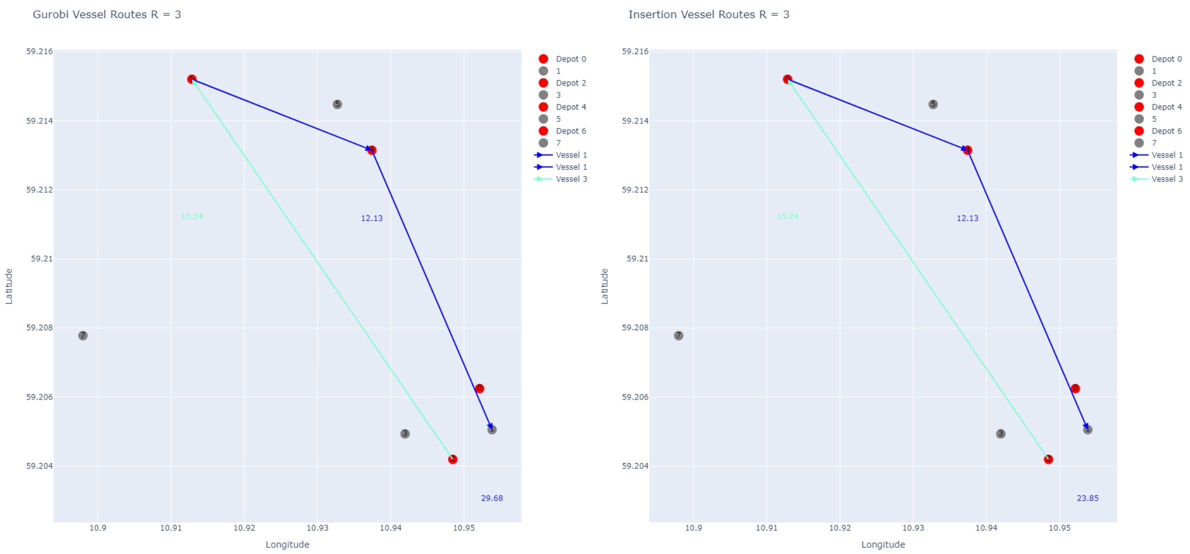
Figure B.13: Obtained route plan from each solving method, $|K| = 4, |R| = 1$



(a) Exact method

(b) Insertion heuristic

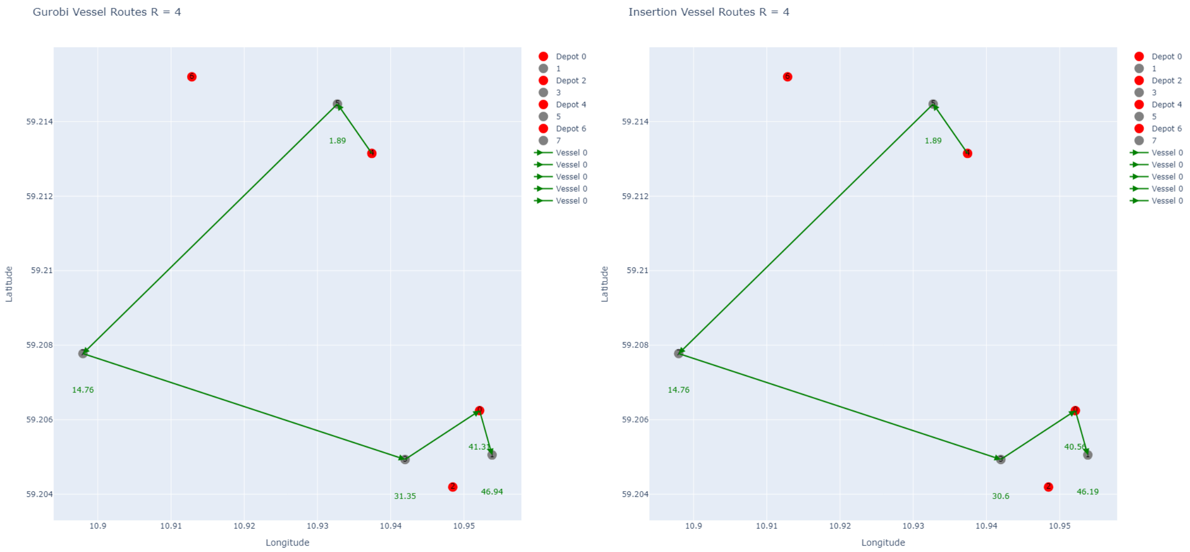
Figure B.14: Obtained route plan from each solving method, $|K| = 4, |R| = 2$



(a) Exact method

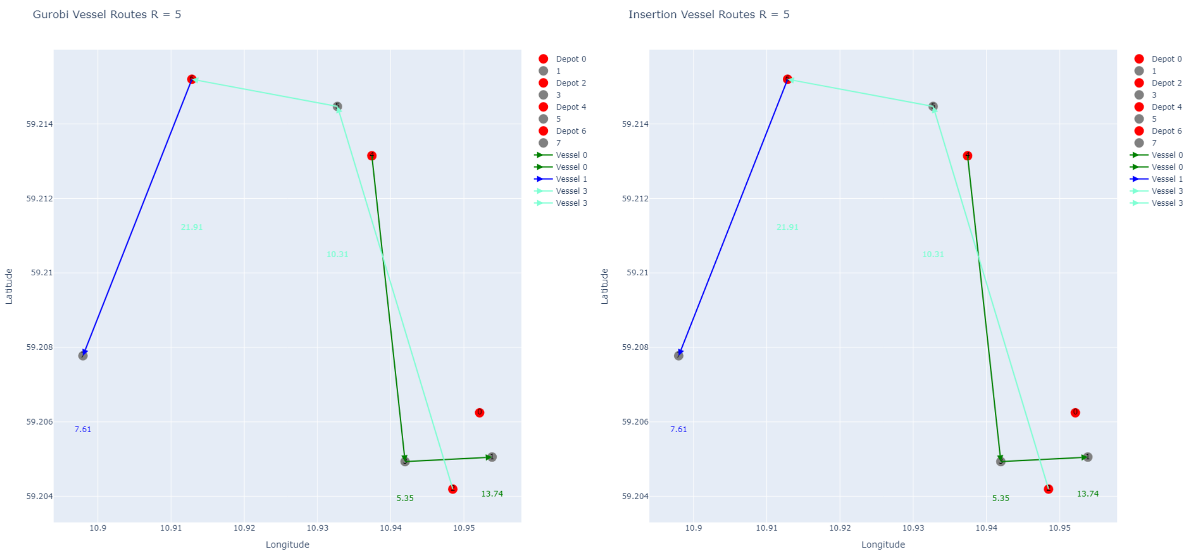
(b) Insertion heuristic

Figure B.15: Obtained route plan from each solving method, $|K| = 4, |R| = 3$



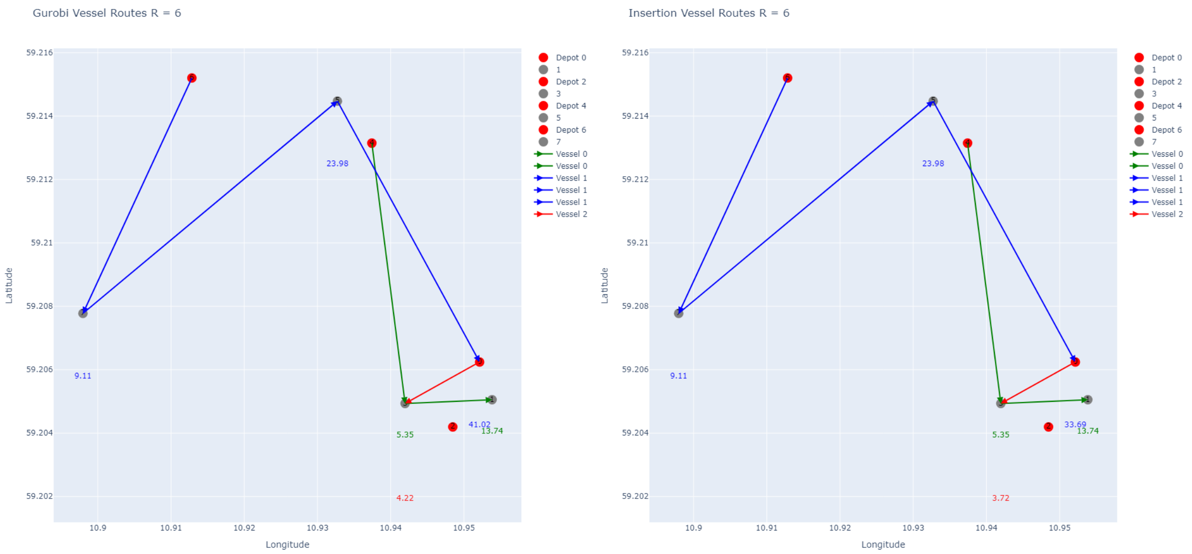
(a) Exact method **(b)** Insertion heuristic

Figure B.16: Obtained route plan from each solving method, $|K| = 4, |R| = 4$



(a) Exact method **(b)** Insertion heuristic

Figure B.17: Obtained route plan from each solving method, $|K| = 4, |R| = 5$



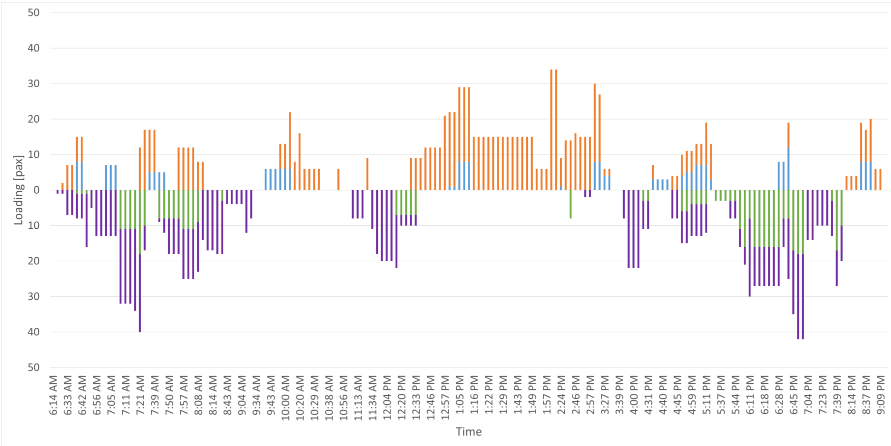
(a) Exact method

(b) Insertion heuristic

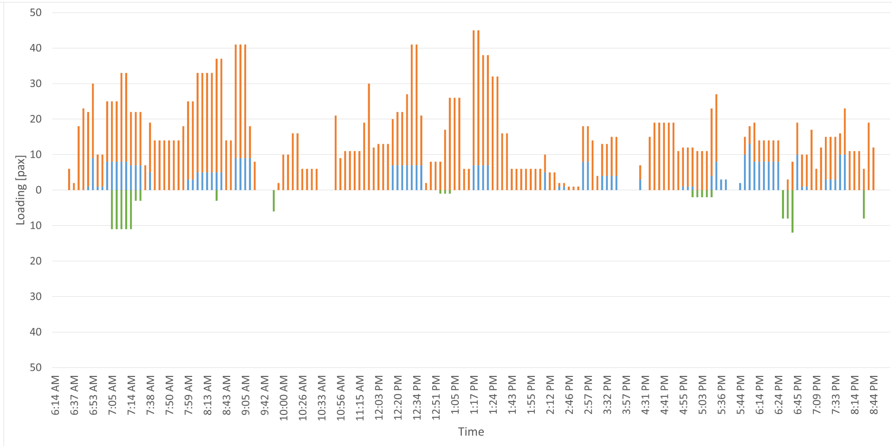
Figure B.18: Obtained route plan from each solving method, $|K| = 4, |R| = 6$

C
Results of the loading level

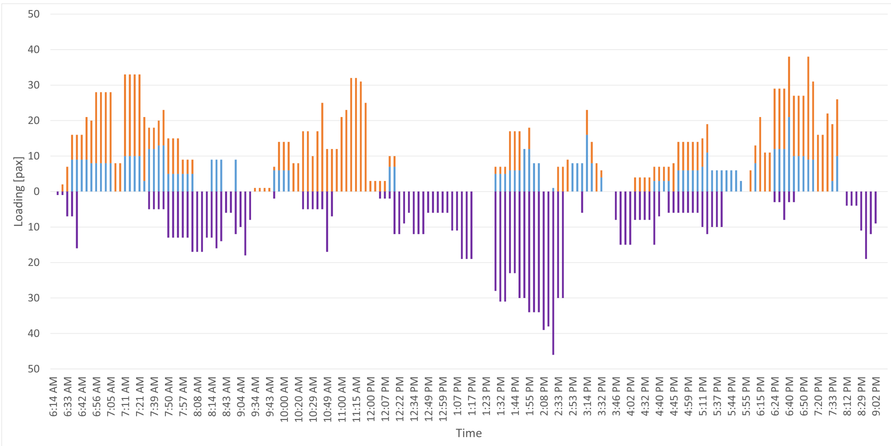
The results of the time series of the loading level for each vessel in each configuration solved by the exact method are presented.



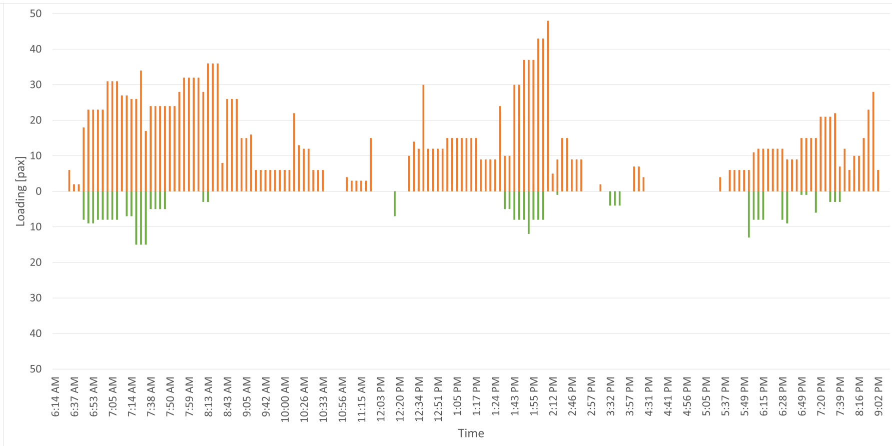
(a) (mixed, mixed)



(b) (mixed, passenger)

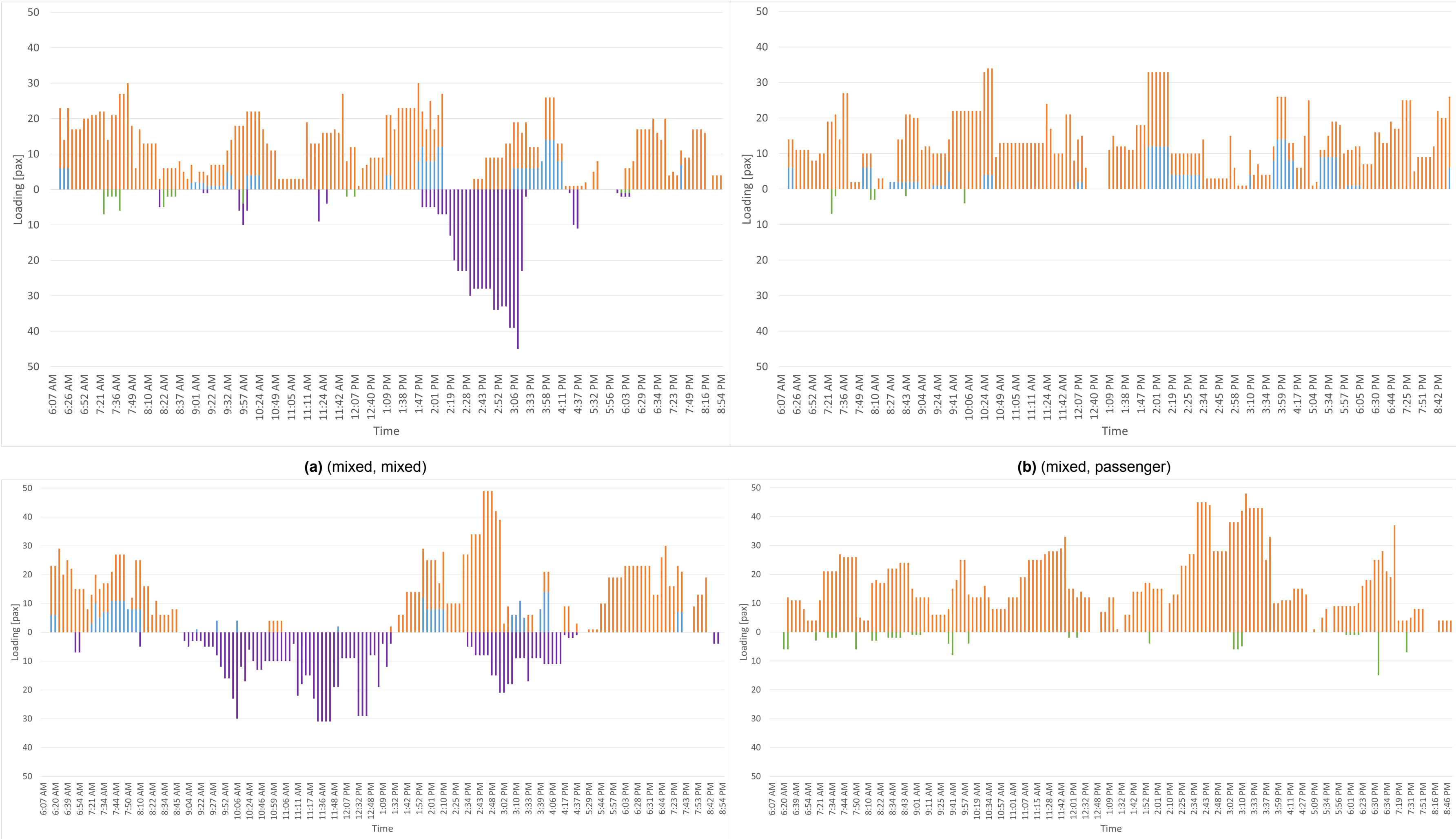


(c) (mixed, parcel)



(d) (parcel, passenger)

Figure C.1: Loading level of each vessel in each vessel type combination for high demand scenario by exact method
orange: parcel vessel 1, blue: passenger vessel 1, purple: parcel vessel 2, green: passenger vessel 2



(a) (mixed, mixed)

(b) (mixed, passenger)

(c) (mixed, parcel)

(d) (parcel, passenger)

Figure C.2: Loading level of each vessel in each vessel type combination for low demand scenario by exact method
orange: parcel vessel 1, blue: passenger vessel 1, purple: parcel vessel 2, green: passenger vessel 2