MODELS AND METHODS FOR CONTAINER PORT CONGESTION MITIGATION

SHUAI JIA

PhD

The Hong Kong Polytechnic University

2020

The Hong Kong Polytechnic University
Department of Logistics and Maritime Studies

Models and Methods for Container Port Congestion Mitigation

Shuai Jia

A thesis submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy

July 2019

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

______________________________(Signed)

___________Shuai Jia___________(Name of Student)

Abstract

Driven by the development of global trade, demand for containerized cargo freight has been growing over the years, leading to frequent vessel calls at major container ports around the world. As a consequence, traffic congestion has increasingly become a burden in the busiest container ports. High levels of congestion impede vessel service, causing long waiting times and severe vessel delays. This thesis studies two optimization problems that aim at mitigating container port congestion caused by the traffic of vessels.

The first problem is a berth allocation problem in which the vessels are classified into deep-sea vessels and feeders. While the arrival times and service times of deep-sea vessels are known to the port operator when berth plans are being devised, the service times of feeders are usually uncertain due to lack of data interchange between the port operator and the feeder operators. The uncertainty of feeders' service times can incur long waiting lines and severe port congestion if the berth plans are poorly devised. To alleviate port congestion and achieve satisfactory vessel service, we allocate berth space to deep-sea vessels and schedule the arrivals of feeders so that the congestion caused by the feeders is under control, while the departure tardiness of deep-sea vessels and the schedule displacements of feeders are minimized. We develop a stochastic optimization model for this problem, and propose a three-phase simulation optimization method, in which the simulation budget is wisely allocated to the solutions explored.

The second problem aims at scheduling the vessel traffic in the port waters by managing the utilization of the navigation channel and the anchorage areas. Navigation channels are fairways for vessels to travel in and out of the terminal basin of a container port. The capacity of a navigation channel is restricted by the number of traffic lanes and safety clearance of vessels, and the availability of a navigation channel is usually affected by tides. The limited capacity and availability of a navigation channel could lead to congestion in the terminal basin. When the navigation channels run out of capacity, the anchorage areas in the terminal basin could serve as a buffer. We develop a mathematical model that simultaneously optimizes the navigation channel traffic and anchorage area utilization. We analyze the complexity of the model and propose a

Lagrangian relaxation heuristic in which the relaxed problem is decomposed into two asymmetric assignment problems.

We evaluate the computational performance of the proposed solution methods using test instances generated based on the operational data of the Yangshan Deep-water Port in Shanghai. Computational results show that the proposed solution methods achieve satisfactory performance within reasonable computation time and outperform benchmark methods in terms of congestion mitigation and vessel service enhancement.

Publications Arising from the Thesis

Jia S, Li C-L, Xu Z (2019) Managing navigation channel traffic and anchorage area utilization of a container port. *Transportation Science* 53(3):728–745.

Jia S, Li C-L, Xu Z (2019) Deep-sea vessel berth planning and feeder arrival scheduling for port congestion mitigation. Submitted for publication.

Acknowledgements

I would like to thank Professor Chung-Lun Li and Professor Zhou Xu for the kind help and fruitful discussions that have made this program successful. I could not have made such achievement without the support of my family.

Contents

Chapter 1

Introduction

Owing to the fast development of global trade, the maritime traffic has seen rapid growth in the last decade. The volume of world seaborne trade increased from 7.7 billion tons in 2006 to 10.3 billion tons in 2016 (UNCTAD 2017). The ever-increasing demand for seaborne cargo freight has given rise to frequent vessel calls at container ports around the world. As a consequence, traffic congestion has increasingly become a burden in the busiest ports. High levels of congestion impede vessel service, causing long waiting times and severe vessel delays. Thus, congestion mitigation has become an important issue faced by the busiest container ports today. Besides infrastructure expansion, which is usually time-consuming and capital-intensive, the following two methods have been implemented in practice for port congestion mitigation: (i) allocating berths to vessels for efficient vessel handling; and (ii) scheduling the vessel traffic in the port waters for efficient vessel movement. This thesis studies a problem of allocating berths to vessels and a problem of scheduling the vessel traffic in the port waters, and proposes optimization models and solution methods for addressing the two problems.

For the first problem, we consider a container port that serves two types of vessels: deep-sea vessels and feeders, where the number of feeders to be served is significantly larger than the number of deep-sea vessels. We allocate berth space to deep-sea vessels and schedule the arrivals of feeders so that the congestion caused by the traffic of feeders is under control, while the departure tardiness of deep-sea vessels and the schedule displacements of feeders are minimized. For the second problem, we consider given berth plans of vessels and schedule the vessel traffic by managing the utilization of the navigation channel and the anchorage areas in the terminal basin, so that vessels can berth and depart as planned. The goal of the thesis is to provide container ports with decision support tools that can be used to improve the current practice of berth utilization and vessel traffic control so that traffic congestion can be well alleviated.

1.1 Berth Allocation and Arrival Scheduling

Vessels served by a container port can usually be classified into two types: deep-sea vessels and feeders (Cordeau et al. 2005; Emde and Boysen 2016; Ursavas and Zhu 2016). Deep-sea vessels are large in size, and they transport containers between ports along long-haul ocean routes. Feeders are small-sized vessels, and they provide transportation services between ports that are relatively close to each other. For container ports located on the estuaries of busy waterways, such as the Port of Shanghai, which is located on the Yangtze River estuary, feeders play an important role in container transshipment. In these container ports, the number of feeders served can be significantly larger than the number of deep-sea vessels.

One important issue faced by a port operator is the need for effective berth planning for both deep-sea vessels and feeders. Deep-sea vessels visit container ports regularly by following their voyage schedules. Thus, the scheduled port arrival and departure times of each deep-sea vessel are known to the port operator. Using electronic data interchange, the port operator can acquire the throughput information of deep-sea vessels. The scheduled arrival and departure times and the throughput information are essential for making detailed service plans for the vessels. However, due to the lack of data interchange between the port operator and the feeder operators, accurate throughput information of feeders may not be available to the port operator in advance. This poses a great challenge for berth planning, as service plans are made several days before vessels arrive. Because of the uncertainties in service times, port operators do not usually reserve berth space for feeders when making berth plans, but allocate berths to feeders dynamically according to certain predetermined service rules (e.g., arbitrarily assigning an available berth to a feeder when it arrives at the port). However, this practice may lead to severe congestion and also lower the port's operational efficiency. On the other hand, developing berth plans for deep-sea vessels and simultaneously taking into account the uncertainties of feeder service times may actually alleviate congestion and thus improve the efficiency of the port.

For a container port such as the Port of Shanghai, which serves a large number of feeders, another important issue faced by the port operator is the need to mitigate port congestion by controlling the waiting lines of feeders. Unlike deep-sea vessels that wait at the anchorage ground upon arrival, feeders usually wait at the terminal basin, which is close to the berths. Long waiting lines of feeders obstruct the traffic in the port, impeding the service of vessels and increasing the high risk of vessel collisions. One method that has been implemented in practice for congestion mitigation is to serve feeders by appointment. That is, the port operator makes adjustments to the voyage schedules of feeders by assigning updated arrival times to the feeders, and the feeders are required to arrive at the port at or close to their assigned arrival times. Assigning arrival times to feeders allows the port operator to reduce the number of arrivals during peak hours. However, the queue length of feeders depends not only on the feeder arrival times, but also on their service times. Hence, to effectively control congestion in the port, the assignment of arrival times to feeders should also take into account the service time uncertainties of feeders.

We study a problem that allocates berths to deep-sea vessels and assigns arrival times to feeders for a container port where the service times of feeders are stochastic with known probability distribution. We develop a stochastic optimization model for the problem. The model aims to minimize the departure delays of deep-sea vessels and schedule displacements of feeders, subject to berth availability and a queue length limit. We develop a simulation optimization method for solving this model. Our model and solution method can be used for tactical berth planning at a container port where information on feeders is limited and mitigating congestion is of great importance.

Berth allocation problems have attracted tremendous research efforts over the past two decades. Various models have been developed by researchers for decision-making at tactical and operational levels; see Steenken, Voß, and Stahlbock (2004), Bierwirth and Meisel (2010, 2015), Kim and Lee (2015), and Gharehgozli, Roy, and de Koster (2016) for comprehensive reviews of the literature. Most of the existing berth allocation models share the common feature that the arrival plans of vessels are predetermined and must be respected by the port operator when devising berth plans. In practice, however, port operators may need to change the vessel arrival plans when making berth plans in order to alleviate congestion in the port. There exist a few works that study the scheduling of vessel arrivals from the perspective of a port operator. Golias et al. (2009) assign arrival times to vessels in a berth allocation model to minimize weighted total waiting time and departure delay of vessels. Alvarez, Longva, and Engebrethsen (2010) and Lang and Veenstra (2010) use simulation to evaluate different arrival scheduling strategies for minimizing the fuel consumption cost of vessels. Du et al. (2015) assign arrival times to vessels in a tidal port to reduce vessel waiting times caused by tidal effect in the navigation channels. Li and Lam (2017) schedule the arrivals of vessels at a container port to resolve vessel conflicts in the fairways nearby the port. De et al. (2018) schedule the berth utilization and the vessel arrivals to minimize the cost incurred by vessel fuel consumption and terminal operations. In our model, the arrival times of deep-sea vessels are input parameters, whereas the arrival times of feeders are decision variables. This model configuration is motivated by the following facts: (i) Deep-sea vessels visit multiple ports by following their voyage schedules; changing the arrival time of a deep-sea vessel at a port may have a significant impact on its schedule for visiting other ports, which is undesirable in practice. (ii) The voyage schedules of feeders are generally quite flexible, as feeders typically transport containers between only two ports. Thus, scheduling the arrivals of feeders helps alleviate port congestion without incurring unacceptable compromises in service quality.

Another important feature of our model is the stochasticity of the service times of feeders. Many researchers have studied berth allocation problems with uncertain vessel information. Some berth allocation models consider uncertain vessel arrival times (see, e.g., Moorthy and Teo 2006), while some models consider uncertain vessel service times (see, e.g., Golias 2011). Some berth allocation models take into account uncertainties in both vessel arrivals and vessel services; see, for example, Han, Lu, and Xi (2010), Zhen, Lee, and Chew (2011), and Liu et al. (2016). Liu, Xiang, and Zheng (2019) provide a summary of the existing works on stochastic berth allocation problems. In most of the existing works, berths are explicitly allocated to vessels in the hope that vessels can start being served as planned at their designated berths even if the vessel information is uncertain. Our model differs from these models in that the berths for serving deep-sea vessels are determined explicitly using complete vessel information, whereas the berths for serving feeders are modeled implicitly due to the uncertainties of feeder service times. For evaluating the performance of the berth plan generated by our model, we adopt an operational service rule that dynamically allocates berths to feeders when the service times of feeders are observed.

In addition to the limited berth availability, which is the major constraint considered in the berth allocation literature, our model considers the queueing behavior of feeders and imposes a restriction on the expected queue length of feeders. This additional constraint is motivated by the need to control vessel congestion in the terminal basin of a busy container port. The queueing behavior of vessels has been studied using queueing and simulation models. These include studies that model the arrival and service processes of vessels as queueing systems and evaluate the port performances under different vessel service patterns (see, e.g., Radmilovich 1992; Zrnić, Dragović, and Radmilović 1999), as well as studies that apply simulation models to evaluate the performance of quay crane and berth allocation policies (see, e.g., Legato and Mazza 2001; Dragović et al. 2005; Dragović, Park, and Radmilović 2006). Existing models on vessel queueing behavior typically focus on the evaluation of predetermined vessel service policies. Our model, however, incorporates vessel queueing behavior into berth allocation and feeder arrival scheduling decisions, so as to optimize berth utilization while at the same time keeping vessel congestion under control.

We develop a simulation optimization method for solving the berth allocation and arrival scheduling problem. Simulation optimization methods are widely used for solving optimization problems where performance of each solution is evaluated by doing simulation experiments. Xu et al. (2015) and Amaran et al. (2016) review various simulation optimization methods and their applications. As noted by Lee, Chew, and Manikam (2006), performing a large number of simulation replications to obtain an accurate estimation of solution performance would consume an unaffordable amount of computation effort, especially when the solution space is large, and thus one should seek to balance the effort spent in running simulations and in sampling solutions. In the literature of berth allocation, there exist a few works that apply simulation optimization for solving problems with uncertainties. However, existing works either allocate a large simulation budget to each solution and incur a heavy computation burden (see, e.g., Han, Lu, and Xi 2010), or attempt to explore the solution space efficiently by ignoring uncertainties and then use simulation to evaluate only a limited number of the visited solutions (see, e.g., Arango et al. 2011, 2013; Legato, Mazza, and Gullì 2014). Unlike these simulation optimization methods, we adapt the solution framework developed by Xu, Nelson, and Hong (2010), which comprises a global phase, a local phase, and a clean-up phase, where different amounts of simulation budget are allocated to different phases. We develop new solution sampling strategies in the global and local phases so as to tackle our berth allocation problem with feeder service time uncertainty.

1.2 Vessel Traffic Scheduling

Vessels that enter or leave a container port need to pass through a navigation channel. The navigation channel is a fairway where vessels receive official pilotage services when traveling between the berths and the open sea. The vessel traffic in a navigation channel is regulated by vessel traffic service (VTS) operators (International Maritime Organization 1997). For tidal ports such as the Port of Shanghai where the availability of the navigation channel is limited due to tidal effect, the management of vessel traffic by VTS operators plays a crucial role in congestion mitigation. In these tidal ports, terminal operators determine the berthing and unberthing times of calling vessels based on their knowledge about the availability of the berths and navigation channels. After receiving the berth plans proposed by the terminal operators, the VTS operator of the port needs to schedule the vessels for traveling through the navigation channels such that the time windows for the vessels to utilize the channels coordinate with the vessels' planned berthing and unberthing times at the terminals. Because of the limited number of traffic lanes and the safety clearance requirement in navigation channels, the number of vessels that can sail in the channels is limited. When the channels run out of capacity, the anchorage area in the terminal basin could serve as a buffer. However, the poor planning of navigation channel and anchorage area utilization often leads to congestion, which lowers terminal operation's efficiency and incurs more vessel emissions. If the VTS operator fails to arrange a schedule for some vessels to utilize the navigation channel, the berth plans will be rejected by the VTS operator, and the terminal operators will have to revise their plans.

Figure 1.1 provides a schematic layout of a container port. During each planning horizon, some calling vessels need to be served at a container port according to the berth plans proposed by the terminal operators. Each vessel arrives at the outer anchorage ground of the port at a given time

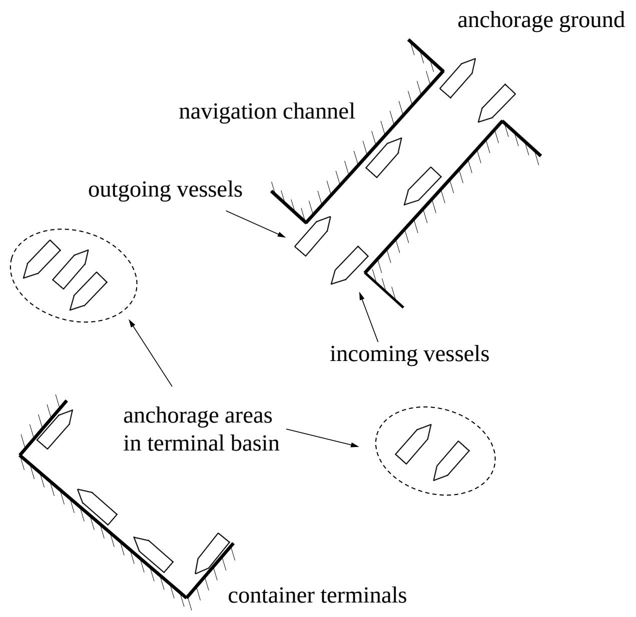

Figure 1.1: Schematic layout of a container port.

with a planned berthing time at which it is expected to arrive at its designated berth. After the vessel completes its service at the berth, it will leave the terminal basin, and is expected to arrive at the outer anchorage ground by a certain expected departure time. Vessels that enter or exit the terminal basin should pass through a navigation channel. The navigation channel consists of two traffic lanes: one for incoming vessels (i.e., vessels that enter the terminal basin for service) and the other for outgoing vessels (i.e., vessels that exit the terminal basin after service). Since each traffic lane bears one-way traffic, incoming or outgoing vessels need to enter a lane one by one and sail in a single line when getting through the channel. When sailing in the navigation channel, vessels in the same traffic lane should keep a safety clearance among them. Because of tidal effect, the water depth in the navigation channel varies over time. As a result, a vessel can only pass through the navigation channel when the water level becomes deep enough (Du et al. 2015, Ding et al. 2016). This tidal constraint, together with the limited capacity of the navigation channel, imposes a serious restriction on the number of vessels that can enter and leave the terminal basin. Vessels in the terminal basin should either moor at berth, or park at some small anchorage areas, which we refer to as *staging anchorages.* The staging anchorages are utilized in the following ways: (i) An incoming vessel that enters the terminal basin earlier than expected should wait at a staging anchorage until it can moor as planned. (ii) After an outgoing vessel finishes service, the vessel should wait at a staging anchorage if it cannot enter the navigation channel due to either inadequate water level or limited channel capacity. To ensure that vessels can berth and depart as planned, the VTS operator should manage the traffic by scheduling the utilization of the navigation channel and the staging anchorages.

We consider a container port with a single navigation channel. We develop and analyze an optimization model that simultaneously allocates space in the anchorage area to vessels and determines the time for each vessel to utilize the navigation channel. The objective of this model is to minimize vessels' berthing and departure tardiness, as well as unsatisfied service requests. The model can be used for constructing weekly "rough-cut" plans that determine the usage of pilots and the number of tugboats needed for guiding the vessels, as well as shorter term plans (e.g., 1- or 2-day plans) that determine the actual schedule of the vessels.

Seaside operations planning problems in container ports have been extensively studied by logistics and operations researchers, and various mathematical models have been developed for different applications. One common feature of the existing models is that problems are formulated from the perspective of a terminal operator. These models include berth allocation models with different spatial, temporal, and handling time attributes, quay crane scheduling models with different task, crane, and interference attributes, as well as various models that integrate berth allocation and quay crane scheduling decisions. Bierwirth and Meisel (2010, 2015) provide comprehensive reviews on various berth allocation and quay crane scheduling problems. For other recent reviews on mathematical models for container terminal operations, see Carlo, Vis, and Roodbergen (2015), Kim and Lee (2015), and Gharehgozli, Roy, and de Koster (2016). In addition to the limited availability of terminal resources, low water level caused by tidal effect in navigation channels and limited space in the anchorage area are other issues faced by many container ports. Yet, most existing seaside operations planning models in the literature have excluded these factors. The tidal condition in navigation channels often leads to tight time windows for vessels with deep drafts to enter and exit the terminal basin. As a result, deep-draft vessels may need to enter and leave their allocated berths at some specific time points, occupy the berth space for a longer time period, or wait at the anchorage area in the terminal basin for berthing and departure.

A number of seaside operations planning models reported in the literature have considered tidal conditions of a port. Barros et al. (2011), Xu, Li, and Leung (2012), Lalla-Ruiz, Melián Batista, and Moreno Vega (2013), Lalla-Ruiz et al. (2016, 2017), Qin, Du, and Sha (2016), and Lalla-Ruiz (2017) develop and analyze different seaside operations planning models with tidal effects at the berths. Ilati, Sheikholeslami, and Hassannayebi (2014), Sheikholeslami, Ilati, and Kobari (2014), Du et al. (2015), Ding et al. (2016), Dadashi et al. (2017), Yu, Wang, and Zhen (2017), and Zhen et al. (2017) develop and analyze different seaside operations planning models by taking the tidal impact on navigation channels into consideration. Existing works that study the tidal impact on a navigation channel typically focus on examining how the tidal windows in the navigation channel affect berth planning decisions. However, the limited traffic capacity of the navigation channel is rarely considered in these works. Since the traffic capacity of the navigation channel is restricted by the number of traffic lanes and the safety clearance of vessels, the number of incoming vessels and the number of outgoing vessel that can enter the navigation channel at a time are limited. This restriction would have a direct impact on the berthing and unberthing times of vessels.

Different from the above mentioned seaside operations planning models which allocate terminal resources to calling vessels, our model takes vessel berthing information as input and focuses on scheduling vessel traffic in a busy container port. Vessel traffic scheduling problems in restricted water areas have been studied by some researchers. Petersen and Taylor (1988), Nauss (2008), Verstichel et al. (2014), and Passchyn et al. (2016) model and solve lock scheduling problems for inland waterways. Ulusçu et al. (2009) and Sluiman (2017) schedule the traffic of transit vessels in straits. There also exist a few works on the scheduling of navigation channels of seaports that are more related to our work. Kelareva et al. (2012) and Kelareva, Tierney, and Kilby (2013) consider the scheduling of incoming and outgoing vessels of a navigation channel with predetermined vessel berthing positions, tidal constraints in the channel, and constraints on tug availability, so as to maximize the cargo throughput of the port. Zhang et al. (2016) also study the scheduling of incoming and outgoing vessels of a navigation channel with predetermined vessel berthing positions and tidal constraints in the channel, where the objective is to minimize the vessels' total waiting time.

Tang et al. (2016) study the master planning of a new container terminal by taking the dimensions of a navigation channel into account to avoid possible bottlenecks for the port's future performance. Lalla-Ruiz, Shi, and Voß (2018) consider a problem that schedules vessels for traveling through a set of navigation channels with tidal constraints. They model the problem as a mixed integer linear program (MILP) and propose a meta-heuristic for minimizing vessels' waiting time at the outer anchorage ground plus the travel time in the navigation channels. Hill et al. (2019) reformulate Lalla-Ruiz, Shi, and Voß's problem as a variant of the multi-mode resource-constrained project scheduling problem and develop a compact MILP that can be solved efficiently by a mathematical programming solver. However, existing works on scheduling vessel traffic in navigation channels of seaports either ignore the utilization of staging anchorages or assume that the capacity of staging anchorages are infinite. The issue of how the utilization of the limited anchorage capacity can mitigate congestion in navigation channels during low-tide periods has yet to be addressed. This study therefore aims to scientifically model, solve, and analyze the anchorage area and navigation channel planning problem of a container port. Similar to the model of Lalla-Ruiz, Shi, and Voß (2018), scheduling the navigation channel traffic of a port is a key component of our model. However, our model considers the assignment of the staging anchorages to vessels, takes into account of the safety clearance of vessels and the pre-determined berth plans when scheduling the vessels, and has an objective function different from that of Lalla-Ruiz, Shi, and Voß's model.

1.3 Contributions

The main contributions of this study is summarized as follows. First, we address two important optimization problems for congestion mitigation in a busy container port. The first problem involves the decisions of allocating berths to deep-sea vessels and scheduling the arrivals of feeders with uncertainties of feeder service times; while the second problem involves the decisions of scheduling the vessel traffic in the navigation channel and managing the utilization of the anchorage areas. We develop optimization models for the two problems. The model of berth allocation and arrival scheduling allocates berth space to deep-sea vessels by respecting the vessel arrival plans, and controls the expected queue length of feeders by adjusting the feeder arrival plans. The model of channel traffic scheduling and anchorage utilization management controls the vessel traffic in the port waters by taking into account the berth plans of vessels. Second, we propose effective solution methods for solving the models. For solving the stochastic optimization model of berth allocation and arrival scheduling, we apply a three-phase simulation optimization method, in which we develop new solution methods for the global and local phases. The simulation optimization method strikes a balance between exploration and exploitation, and allocates the simulation budget to solutions wisely, so that solutions with satisfactory performance can be identified using a reasonable amount of computation effort. For solving the MILP model of channel traffic scheduling and anchorage utilization management, we develop a Lagrangian relaxation heuristic in which the capacity constraint of the anchorage areas is dualized, and show that the remaining problem decomposes into two asymmetric assignment problems, which can be solved in pseudo-polynomial time. Finally, we generate extensive problem instances based on the operational data extracted from the Yangshan Deep-water Port in Shanghai, and compare the computation performances of the proposed solution methods against various benchmark methods via extensive computational experiments. Computational results show that the proposed solution methods outperform the benchmark methods in terms of both efficiency and effectiveness, and thus, the proposed methods can possibly serve as decision support tools for vessel service planning and congestion mitigation in a container port.

Chapter 2

Deep-Sea Vessel Berth Planning and Feeder Arrival Scheduling

2.1 Problem Description and Formulation

We consider a set of vessels that need to be served at a container port during a planning horizon. The vessels are classified into deep-sea vessels and feeders. Each deep-sea vessel has a scheduled port arrival time and a target departure time. Deep-sea vessels strictly follow their voyage schedules and arrive at the port on time. If the port fails to complete service at or before the target departure time of a deep-sea vessel, then departure delay is incurred, resulting in a tardiness penalty cost. Each feeder has an initially scheduled port arrival time. However, in order to mitigate congestion, the port operator may need to control the arrivals of the feeders by altering their arrival plans. The deviation between the assigned port arrival time and the initially scheduled port arrival time of each feeder is referred to as the schedule displacement of the feeder. Schedule displacement of each feeder incurs a penalty cost. Furthermore, to ensure that containers will be successfully transshipped between feeders and deep-sea vessels, the schedule displacement of each feeder cannot exceed a pre-specified upper limit. Consequently, each feeder has a time window during which it is allowed to arrive at the port. In reality, service delays can propagate between feeders and deep-sea vessels due to container transshipment. To account for the dependence between feeders and deep-sea vessels, precedence relations on the services of vessels should be captured in the model. However, such modeling approach has the following drawbacks. First, it significantly increases the complexity of the model. Second, since the service times of feeders are uncertain, a solution to the model may not avoid delay propagations even if the dependence between feeders and deep-sea vessels are taken into account. On the other hand, imposing time windows on the arrivals of feeders helps mitigate delay propagations while eliminating the dependence between feeders and deep-sea vessels, making the model more tractable.

The service times of deep-sea vessels are known to the port operator at the planning stage. However, accurate throughput information of feeders is unavailable when the service plan is developed. Hence, the service times of feeders are uncertain, and berths are usually assigned to feeders dynamically according to chosen operational rules. Uncertainties in the service times can cause stochastic dynamics in the queue length of feeders, which can result in severe congestion when the number of feeders is large but the berths are not effectively utilized for serving the feeders. To mitigate congestion, berth plans of deep-sea vessels and arrival schedules of feeders should be made by taking into account both the uncertainties of feeder service times and the operational service rules of feeders so that the expected queue length of feeders is well controlled.

For the purpose of modeling, we discretize the planning horizon and assume that all time-related parameters are integer-valued. For simplicity, we assume that the port is empty at the beginning of the planning horizon, and the planning horizon is set to be long enough so that all the vessels considered can finish service within the planning horizon. We also assume that all feeders have the same length σ, and that the lengths of deep-sea vessels are multiples of σ. This assumption is justified by the fact that the lengths of deep-sea vessels are much larger than the lengths of feeders, and the lengths of feeders are within a relatively small range compared to the lengths of deep-sea vessels. We discretize the quay into a set of berth segments, with each berth segment having a length equal to the length of a feeder. As a result, each feeder occupies exactly one berth segment while each deep-sea vessel occupies multiple berth segments when being served. In Section 2.4, we discuss how our solution method can be modified to handle problems where the port is not initially empty and the lengths of feeders are non-identical.

The service times of feeders are modeled as random variables with known probability distributions that are independent of the berth plan. The probability distributions of these random variables can be derived by analyzing the historical operational data of the port. We assume that berth segments are assigned to arrived feeders on a first-come first-served basis. When a feeder arrives at the port, its service time will be known, and it will join the waiting line (ties broken randomly if multiple feeders arrive at the same time). Suppose feeder i with service time τ_i is the first feeder in the waiting line. Then, feeder i will start being served, say at time t, once a berth becomes available throughout time interval $[t, t+\tau_i]$ (ties broken randomly if multiple berth segments become available at the same time). When we determine the berth plan, we impose an upper limit on the expected queue length of feeders, so as to control the traffic in the port and the feeder waiting time.

Our problem involves decisions for assigning each deep-sea vessel a berthing time and a berthing position, and for assigning each feeder a port arrival time. The objective is to minimize the weighted total departure tardiness of deep-sea vessels and the weighted total schedule displacement of feeders. We denote this problem as **P**. The mathematical formulation of problem **P** is presented as follows:

Input data:

T: Length of the planning horizon.

N_1: Number of deep-sea vessels to be served.

N_2: Number of feeders to be served.

B: Number of berth segments available.

H_i: Service time of deep-sea vessel i.

R_i: Number of berth segments that deep-sea vessel i needs to occupy.

A_i: Port arrival time of deep-sea vessel i.

D_i: Target port departure time of deep-sea vessel i.

S_i: Initially scheduled port arrival time of feeder i.

$[\underline{E}_i, \bar{E}_i]$: Time interval during which feeder i is allowed to arrive at the port.

$\bar{Q}$: Upper limit on the expected queue length of feeders.

C_{1i}: Unit cost of departure tardiness of deep-sea vessel i.

C_{2i}: Unit cost of schedule displacement of feeder i.

The input data also include the probability distribution of the service time of each feeder, and we let G_i denote the mean service time of feeder i.

Decision variables:

x_{ibt}: $= 1$ if deep-sea vessel i is served by berth segments $b, b+1, \ldots, b+R_i-1$ during the time interval $[t, t+H_i]$; 0 otherwise.

y_{it}: $= 1$ if feeder i is assigned an arrival time t; 0 otherwise.

l_{1i}: Departure tardiness of deep-sea vessel i.

l_{2i}: Schedule displacement of feeder i.

Random variables:

$Q_t(\mathbf{x}, \mathbf{y})$: Number of feeders that are waiting for service during the time interval $[t, t+1]$ if the feeders are served according to a given service plan $(\mathbf{x}, \mathbf{y})$ and the feeders are allocated to the berths on a first-come first-served basis, where $\mathbf{x} = (x_{ibt} \mid i = 1, \ldots, N_1;\ b = 1, \ldots, B;\ t = 0, 1, \ldots, T-1)$ and $\mathbf{y} = (y_{it} \mid i = 1, \ldots, N_2;\ t = 0, 1, \ldots, T-1)$.

Mathematical programming formulation:

$$\mathbf{P}: \text{ minimize } \sum_{i=1}^{N_1} C_{1i} l_{1i} + \sum_{i=1}^{N_2} C_{2i} l_{2i} \tag{2.1}$$

$$\text{subject to } \sum_{b=1}^{B-R_i+1} \sum_{t=A_i}^{T-H_i} x_{ibt} = 1 \quad (i = 1, \ldots, N_1) \tag{2.2}$$

$$\sum_{t=\underline{E}_i}^{\bar{E}_i} y_{it} = 1 \quad (i = 1, \ldots, N_2) \tag{2.3}$$

$$\sum_{i=1}^{N_1} \sum_{b'=\max\{1, b-R_i+1\}}^{\min\{b, B-R_i+1\}} \sum_{t'=\max\{0, t-H_i+1\}}^{\min\{t, T-H_i\}} x_{ib't'} \leq 1 \quad (b = 1, \ldots, B;\ t = 0, 1, \ldots, T-1) \tag{2.4}$$

$$l_{1i} = \max\left\{0, \sum_{b=1}^{B-R_i+1} \sum_{t=A_i}^{T-H_i} (t+H_i) x_{ibt} - D_i\right\} \quad (i = 1, \ldots, N_1) \tag{2.5}$$

$$l_{2i} = \left| \sum_{t=\underline{E}_i}^{\bar{E}_i} t y_{it} - S_i \right| \quad (i = 1, \ldots, N_2) \tag{2.6}$$

$$E[Q_t(\mathbf{x}, \mathbf{y})] \leq \bar{Q} \quad (t = 0, 1, \ldots, T-1) \tag{2.7}$$

$$x_{ibt} \in \{0, 1\} \quad (i = 1, \ldots, N_1;\ b = 1, \ldots, B-R_i+1;\ t = 0, 1, \ldots, T-H_i) \tag{2.8}$$

$$y_{it} \in \{0, 1\} \quad (i = 1, \ldots, N_2;\ t = 0, 1, \ldots, T-1) \tag{2.9}$$

In objective function (2.1), the unit cost C_{1i} represents the weight of the departure tardiness of deep-sea vessel i, and the unit cost C_{2i} represents the weight of the schedule displacement of feeder

i. Thus, objective function (2.1) minimizes the weighted total tardiness and schedule displacement of vessels. Constraint (2.2) ensures that each deep-sea vessel is served during the planning horizon. Constraint (2.3) ensures that each feeder i arrives at the port during its feasible arrival time interval $[\underline{E}_i, \bar{E}_i]$. Constraint (2.4) ensures that each berth segment b is occupied by at most one deep-sea vessel during each time interval $[t, t+1]$. Constraint (2.5) determines the departure tardiness of each deep-sea vessel. Constraints (2.6) determines the schedule displacement of each feeder. Constraint (2.7) imposes an upper limit on the expected queue length of feeders throughout the planning horizon. Constraints (2.8) and (2.9) specify the binary requirements of x_{ibt} and y_{it}.

2.2 Solution Method

Solving problem **P** is highly challenging, because (i) the mathematical formulation contains a large number of constraints and binary decision variables; and (ii) the expected queue length $E[Q_t(\mathbf{x}, \mathbf{y})]$ in constraint (2.7) is dependent on both the service plan $(\mathbf{x}, \mathbf{y})$ and the service time distribution of the feeders. In fact, when $N_2 = 0$ (i.e., there are no feeders), the problem becomes a berth allocation problem with a minimum weighted total tardiness objective. This special case is a generalization of the single machine scheduling problem with a minimum weighted total tardiness objective, which is known to be strongly NP-hard (Garey and Johnson 1979, p. 237). To tackle this difficult problem **P**, we use a simulation optimization method, which is an adaptation of the method proposed by Xu, Nelson, and Hong (2010) for solving fully-constrained discrete simulation optimization problems. Our method runs in three phases: global phase, local phase, and clean-up phase. In the global phase, we attempt to quickly identify a set of solutions with good estimated performance via a genetic algorithm. Since the purpose of the global phase is to explore the solution space efficiently, a relatively small simulation budget is allocated to the evaluation of visited solutions. In the local phase, we construct a set of solution clusters using the solutions generated in the global phase, and for each cluster we identify the most promising area, in which we use local search to obtain a locally optimal solution. In this phase, we allow the solutions to be evaluated more intensively, as the quality of the solutions obtained in this phase will have a higher impact on the overall performance of the simulation optimization method. The clean-up phase selects the best solution

Table 2.1: Parameters used in each phase of the simulation optimization method.

Phase	Parameter	Description
Global phase	s_1	Population size used by the genetic algorithm
	s_2	No. of individual pairs selected for crossover in each iteration
	μ	Mutation rate
	$\bar{\ell}$	Maximum number of iterations
	n_1	No. of simulation replications allocated to each new individual
	n_2	No. of simulation replications allocated to each individual that has been visited before
	λ_0	Initial value of the parameter λ of the relaxed problem $\mathbf{P}'(\lambda)$
	ζ	Step size for updating parameter λ
Local phase	s_3	Maximum number of solutions in each solution cluster
	s_4	Maximum number of solutions evaluated in each iteration
	h	Parameter of the adaptive hyperbox algorithm's stopping condition
	n_3	No. of simulation replications allocated to each visited solution
	$\bar{\lambda}$	Value of the parameter λ used in the local phase
Clean-up phase	δ	Indifference zone
	ϵ	Parameter used to define the confidence level
	n_4	No. of simulation replications allocated to the best solution

among the solutions generated by the local phase, and the performance of the best solution is evaluated with high precision via simulation.

The global phase, local phase, and clean-up phase of our method are described in Sections 2.2.1, 2.2.2, and 2.2.3, respectively. The simulator that we use to evaluate the performance of a solution is presented in Section 2.2.4. The parameters of the algorithms used in different phases are presented in Table 2.1.

2.2.1 Global Phase

The global phase of our simulation optimization method aims to explore the solution space and quickly identify a set of solutions with relatively good estimated performance. This is done via a genetic algorithm. The advantage of using a genetic algorithm is that the algorithm employs a population-based search which considers many good solutions in parallel. Compared to iterative search methods that search the solution space by considering only one solution at a time, a genetic algorithm tends to be more robust to stochastic noise (Xu, Nelson, and Hong 2010) and is widely used as a component of simulation optimization (Fu, Glover, and April 2005; Xu et al. 2015; Amaran et al. 2016).

Our genetic algorithm solves a relaxed version of problem $\mathbf{P}$ iteratively. This relaxed problem, which has a nonnegative parameter λ, is defined as follows:

$$\mathbf{P}'(\lambda): \text{ minimize } \sum_{i=1}^{N_1} C_{1i} l_{1i} + \sum_{i=1}^{N_2} C_{2i} l_{2i} + \lambda\Delta \tag{2.10}$$

$$\text{subject to } \Delta = \max\Big\{0, \max_{0,1,\ldots,T-1}\{E[Q_t(\mathbf{x},\mathbf{y})]\} - \bar{Q}\Big\} \tag{2.11}$$

$$\text{constraints (2.2)–(2.6) and (2.8)–(2.9)}$$

Variable Δ measures the extent to which constraint (2.7) is violated, and λ is the unit penalty on the constraint violation. Clearly, problem $\mathbf{P}'(\lambda)$ is always feasible. We have the following property.

Property 1 *If problem* $\mathbf{P}$ *is feasible, then there exists* $\hat{\lambda} > 0$ *such that any optimal solution of problem* $\mathbf{P}'(\hat{\lambda})$ *is also optimal to problem* $\mathbf{P}$.

Proof: Consider the situation where problem $\mathbf{P}$ is feasible. Note that in each problem instance of $\mathbf{P}$, there are finitely many possible $(\mathbf{x},\mathbf{y})$ values. Let

$$\Delta_{\min} = \min_{(\mathbf{x},\mathbf{y}) \text{ s.t. } \kappa(\mathbf{x},\mathbf{y}) > \bar{Q}} \{\kappa(\mathbf{x},\mathbf{y}) - \bar{Q}\},$$

where

$$\kappa(\mathbf{x},\mathbf{y}) = \max_{t=1,\ldots,T-1}\{E[Q_t(\mathbf{x},\mathbf{y})]\}.$$

Then, $\Delta_{\min} > 0$. Let Z^* denote the optimal solution value of problem $\mathbf{P}$, and let

$$\hat{\lambda} = \frac{Z^*}{\Delta_{\min}} + 1.$$

Because any feasible solution of $\mathbf{P}$ is a feasible solution of $\mathbf{P}'(\lambda)$ with $\Delta = 0$, the optimal solution value of $\mathbf{P}'(\lambda)$ is at most Z^*, for any $\lambda \geq 0$.

Consider a feasible solution of $\mathbf{P}'(\lambda)$ with $\lambda \geq \hat{\lambda}$ and $\Delta > 0$. In this feasible solution, $\Delta \geq \Delta_{\min}$. The objective function value of this feasible solution is at least $\lambda\Delta \geq \hat{\lambda}\Delta_{\min} > Z^*$. Thus, this feasible solution is not optimal. Hence, any optimal solution of $\mathbf{P}'(\lambda)$ with $\lambda \geq \hat{\lambda}$ must satisfy the condition that $\Delta = 0$. Therefore, an optimal solution of $\mathbf{P}'(\lambda)$ with $\lambda \geq \hat{\lambda}$ is also optimal to $\mathbf{P}$. ▮

Property 1 implies that problem $\mathbf{P}$ can be solved optimally by solving $\mathbf{P}'(\lambda)$ with a sufficiently large λ, provided that $\mathbf{P}$ is feasible. Instead of solving problem $\mathbf{P}$ directly, our genetic algorithm solves problem $\mathbf{P}'(\lambda)$ iteratively, where the value of λ is updated periodically. Solving $\mathbf{P}'(\lambda)$ for a variety of λ values enables the genetic algorithm to explore a broader set of possible service plans. When λ is small, the service plans generated tend to have smaller tardiness and displacement costs but be more likely to violate constraint (2.7). When λ is large, the service plans generated tend to be more likely to satisfy constraint (2.7) but have larger tardiness and displacement costs. Both types of service plans, however, may possess certain characteristics of a good solution that can pass onto their offspring in the crossover process.

Let Θ denote the finite set of $(\mathbf{x}, \mathbf{y})$ values that satisfy constraints (2.2)–(2.4) and (2.8)–(2.9). For each $\lambda \geq 0$ and each $(\mathbf{x}, \mathbf{y}) \in \Theta$, define

$$g_\lambda(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N_1} C_{1i} l_{1i} + \sum_{i=1}^{N_2} C_{2i} l_{2i} + \lambda\Delta,$$

where l_{1i}, l_{2i}, and Δ are defined by equations (2.5), (2.6), and (2.11), respectively. We refer to $g_\lambda(\mathbf{x}, \mathbf{y})$ as the performance of the solution $(\mathbf{x}, \mathbf{y})$ for problem $\mathbf{P}'(\lambda)$. The exact value of $g_\lambda(\mathbf{x}, \mathbf{y})$ cannot be obtained easily because $Q_t(\mathbf{x}, \mathbf{y})$ has no closed form over $(\mathbf{x}, \mathbf{y})$. However, since $Q_t(\mathbf{x}, \mathbf{y})$ can be observed through simulation experiments, we can estimate $g_\lambda(\mathbf{x}, \mathbf{y})$ using the sample mean of $Q_t(\mathbf{x}, \mathbf{y})$ observed in multiple independent simulation replications. We let $\bar{g}_\lambda(\mathbf{x}, \mathbf{y})$ denote the estimated performance of $(\mathbf{x}, \mathbf{y})$ obtained via simulation.

In a genetic algorithm, information of a solution is usually encoded into a string of numbers, called an *individual*. In our implementation, each individual is a string $\mathbf{p} = (p_1, \ldots, p_{N_1+N_2})$, which is a permutation of the $(N_1 + N_2)$-tuple $(1, \ldots, N_1 + N_2)$. For each $i = 1, \ldots, N_1 + N_2$, the number p_i represents deep-sea vessel p_i if $p_i \leq N_1$, and represents feeder $p_i - N_1$ if $p_i > N_1$. Furthermore, vessel p_i has a higher service priority than vessel p_j if $i < j$. An individual $\mathbf{p}$ is translated into a solution $(\mathbf{x}, \mathbf{y}) \in \Theta$ via a decoding scheme. This decoding scheme assigns berth segments and service start times to deep-sea vessels. It also assigns arrival times to feeders by reserving some berth space for each feeder. Let $\eta_{bt} = 1$ if berth segment b has been allocated to a vessel during time period $[t, t+1]$, and $\eta_{bt} = 0$ otherwise. The decoding scheme is presented as follows:

Decoding Scheme:

Step 1: Set $\eta_{bt} \leftarrow 0$ for $b = 1, \ldots, B$ and $t = 0, 1, \ldots, T-1$. Set $i \leftarrow 1$.

Step 2: If $p_i \leq N_1$, then (allocate service capacity to deep-sea vessel):

Select the smallest t such that a $b \in \{1, \ldots, B - R_{p_i} + 1\}$ exists with $\sum_{b'=b}^{b+R_{p_i}-1} \sum_{t'=t}^{t+H_{p_i}-1} \eta_{b't'} = 0$. Then, select the smallest b such that $\sum_{b'=b}^{b+R_{p_i}-1} \sum_{t'=t}^{t+H_{p_i}-1} \eta_{b't'} = 0$. Set $x_{p_i bt} \leftarrow 1$. Set $\eta_{b't'} \leftarrow 1$ for $b' = b, b+1, \ldots, b + R_{p_i} - 1$ and $t' = t, t+1, \ldots, t + H_{p_i} - 1$.

Otherwise (allocate service capacity to feeder):

Select the smallest t among $\{\underline{E}_{p_i - N_1}, \ldots, T - G_{p_i - N_1}\}$ such that a $b \in \{1, \ldots, B\}$ exists with $\sum_{t'=t}^{t+G_{p_i-N_1}} \eta_{bt'} = 0$. Then, select the smallest b such that $\sum_{t'=t}^{t+G_{p_i-N_1}} \eta_{bt'} = 0$. Set

$$\bar{t} \leftarrow \begin{cases} t, & \text{if } t \leq \bar{E}_{p_i - N_1}; \\ \bar{E}_{p_i - N_1}, & \text{otherwise.} \end{cases}$$

Set $y_{p_i - N_1, \bar{t}} \leftarrow 1$. Set $\eta_{bt'} \leftarrow 1$ for $t' = t, t+1, \ldots, t + G_{p_i - N_1} - 1$.

Step 3: If $i = N_1 + N_2$, then stop. Otherwise, set $i \leftarrow i + 1$ and go to Step 2.

This decoding scheme allocates service capacity to vessels according to the string of $N_1 + N_2$ numbers in the individual. Specifically, the decoding scheme considers each $i = 1, \ldots, N_1 + N_2$. If $p_i \leq N_1$, then p_i represents deep-sea vessel p_i. In this case, the decoding scheme allocates a feasible service start time and R_{p_i} consecutive berth segments to the vessel so that the departure tardiness of the vessel is minimized. If $p_i > N_1$, then p_i represents feeder $p_i - N_1$. In this case, the decoding scheme assumes that the service duration of feeder i is G_i and allocates a target service start time t and a target berth segment b to feeder i, so that feeder i can complete service as early as possible when being served at berth segment b starting at time t. A feasible arrival time $\bar{t}$ is then assigned to the feeder, so that the waiting time of the feeder is minimized when the feeder's service starts at time t.

Let Ω denote the set of all individuals. It is not difficult to see that by means of the decoding scheme, each individual $\mathbf{p} \in \Omega$ corresponds to one solution $(\mathbf{x}, \mathbf{y}) \in \Theta$. For each $\mathbf{p} \in \Omega$, let $(\mathbf{x}(\mathbf{p}), \mathbf{y}(\mathbf{p}))$ denote the solution obtained by applying the decoding scheme on individual $\mathbf{p}$. The fitness of each individual $\mathbf{p} \in \Omega$ is defined as $1/g_\lambda(\mathbf{x}(\mathbf{p}), \mathbf{y}(\mathbf{p}))$ and is estimated by $1/\bar{g}_\lambda(\mathbf{x}(\mathbf{p}), \mathbf{y}(\mathbf{p}))$. Note that $\bar{g}_\lambda(\mathbf{x}(\mathbf{p}), \mathbf{y}(\mathbf{p}))$ can be obtained by running simulation experiments on the solution $(\mathbf{x}(\mathbf{p}), \mathbf{y}(\mathbf{p}))$. The genetic algorithm, which aims to identify a subset of individuals that result in solutions of $\mathbf{P}$ with small tardiness and displacement costs of vessels, is described as follows:

Genetic Algorithm:

Step 1 (Initialization): Set $\ell \leftarrow 1$ and $\lambda \leftarrow \lambda_0$. Generate an initial population with s_1 individuals. Each individual $\mathbf{p}$ is generated randomly as follows: For each $i = 1, \ldots, N_1 + N_2$, select an available position from $\mathbf{p}$ with all available positions being selected with equal probability, and insert i into the selected position.

Step 2 (Crossover): Randomly select s_2 pairs of distinct individuals from the current population with equal probability. For each pair of individuals $\mathbf{p} = (p_1, \ldots, p_{N_1+N_2})$ and $\bar{\mathbf{p}} = (\bar{p}_1, \ldots, \bar{p}_{N_1+N_2})$, generate two numbers j and j' randomly from $\{1, \ldots, N_1 + N_2\}$ with equal probability, where $j < j'$. Set

$$\mathbf{q} \leftarrow (\underbrace{0, \ldots, 0}_{j-1 \text{ times}}, p_j, p_{j+1}, \ldots, p_{j'}, \underbrace{0, \ldots, 0}_{N_1+N_2-j' \text{ times}}) \text{ and } \bar{\mathbf{q}} \leftarrow (\underbrace{0, \ldots, 0}_{j-1 \text{ times}}, \bar{p}_j, \bar{p}_{j+1}, \ldots, \bar{p}_{j'}, \underbrace{0, \ldots, 0}_{N_1+N_2-j' \text{ times}}).$$

For $i = 1, \ldots, N_1 + N_2$, if $\bar{p}_i$ is not in $\mathbf{q}$ then replace the first 0 in $\mathbf{q}$ with $\bar{p}_i$, and if p_i is not in $\bar{\mathbf{q}}$ then replace the first 0 in $\bar{\mathbf{q}}$ with p_i. Add $\mathbf{q}$ and $\bar{\mathbf{q}}$ to the population.

Step 3 (Mutation): For each individual $\mathbf{p} = (p_1, \ldots, p_{N_1+N_2})$ in the current population, generate a real number $\hat{\mu}$ from a uniform distribution on $[0, 1]$. If $\hat{\mu} \leq \mu$, then randomly select two distinct numbers j and j' from $\{1, \ldots, N_1 + N_2\}$ with equal probability. Swap the positions of p_j and $p_{j'}$ in $\mathbf{p}$.

Step 4 (Evaluation): For each individual $\mathbf{p}$ in the current population, if the individual has been evaluated previously, then perform n_2 additional simulation replications on solution $(\mathbf{x}(\mathbf{p}), \mathbf{y}(\mathbf{p}))$; otherwise perform n_1 simulation replications on solution $(\mathbf{x}(\mathbf{p}), \mathbf{y}(\mathbf{p}))$. Set the fitness of individual $\mathbf{p}$ to be the cumulative sample mean of the solution performance obtained in all replications.

Step 5 (Selection): Sort the individuals of the current population in descending order of fitness. Keep the first s_1 individuals, and discard the other individuals.

Step 6 (Update λ): If $\ell = \bar{\ell}$, then stop. Otherwise, set $\ell \leftarrow \ell + 1$. Consider the first individual $\mathbf{p}$ in the sorted individual list. If the estimated value of $\max_{0,1,\ldots,T-1} \left\{ E[Q_t(\mathbf{x}(\mathbf{p}), \mathbf{y}(\mathbf{p}))] \right\}$ is greater than $\bar{Q}$, then set $\lambda \leftarrow (1 + \zeta)\lambda$. Go to Step 2.

This genetic algorithm starts with a small λ value. When λ is relatively small, the genetic algorithm focuses on searching for solutions that have small vessel tardiness and displacement costs. If the best individual found so far corresponds to a solution of problem $\mathbf{P}'(\lambda)$ that violates constraint (2.7), then the value of λ will be increased in the next iteration so that heavier emphasis will be given to the satisfaction of constraint (2.7). This process is executed repeatedly until the number of iterations reaches the upper limit $\bar{\ell}$.

2.2.2 Local Phase

When the global phase terminates, we obtain a list of $(\mathbf{x}, \mathbf{y})$ values. These $(\mathbf{x}, \mathbf{y})$ values will be used to initialize the local phase. Unlike the global phase, which diversifies the population of individuals by solving the relaxed problem $\mathbf{P}'(\lambda)$ with various λ values, the local phase aims to intensify the search for near-optimal solutions of problem $\mathbf{P}'(\bar{\lambda})$, where $\bar{\lambda}$ is a large input parameter. Note that if $\bar{\lambda}$ is sufficiently large, then an optimal solution of problem $\mathbf{P}'(\bar{\lambda})$ is also optimal to problem $\mathbf{P}$ (see Property 1).

Let $\mathcal{S}$ denote the list of $(\mathbf{x}, \mathbf{y})$ values passed onto the local phase, sorted in ascending order of their estimated performance $\bar{g}_{\bar{\lambda}}(\mathbf{x}, \mathbf{y})$. Let $(\mathbf{x}_j, \mathbf{y}_j)$ denote the jth element in the sorted list $\mathcal{S}$. For each $(\mathbf{x}, \mathbf{y}) \in \Theta$, define a vector of integer values $(\mathbf{e}_1(\mathbf{x}); \mathbf{e}_2(\mathbf{x}); \mathbf{e}_3(\mathbf{y}))$, in which $\mathbf{e}_1(\mathbf{x}) = (e_{11}(\mathbf{x}), \ldots, e_{1N_1}(\mathbf{x}))$, $\mathbf{e}_2(\mathbf{x}) = (e_{21}(\mathbf{x}), \ldots, e_{2N_1}(\mathbf{x}))$, and $\mathbf{e}_3(\mathbf{y}) = (e_{31}(\mathbf{y}), \ldots, e_{3N_2}(\mathbf{y}))$, where

$$e_{1i}(\mathbf{x}) = \sum_{b=1}^{B-R_i+1} \sum_{t=0}^{T-H_i} b x_{ibt} \quad \text{and} \quad e_{2i}(\mathbf{x}) = \sum_{b=1}^{B-R_i+1} \sum_{t=0}^{T-H_i} t x_{ibt}$$

for $i = 1, \ldots, N_1$, and

$$e_{3i}(\mathbf{y}) = \sum_{t=\underline{E}_i}^{\bar{E}_i} t y_{it}$$

for $i = 1, \ldots, N_2$. Note that $e_{1i}(\mathbf{x})$ is the first berth segment allocated to deep-sea vessel i, $e_{2i}(\mathbf{x})$ is the service start time of deep-sea vessel i, and $e_{3i}(\mathbf{y})$ is the arrival time of feeder i.

We define the *distance* between any two solutions $(\mathbf{x}_j, \mathbf{y}_j), (\mathbf{x}_{j'}, \mathbf{y}_{j'}) \in \Theta$ as

$$d_{jj'} = \sqrt{\sum_{i=1}^{N_1} \left[\frac{e_{1i}(\mathbf{x}_j) - e_{1i}(\mathbf{x}_{j'})}{B}\right]^2 + \sum_{i=1}^{N_1} \left[\frac{e_{2i}(\mathbf{x}_j) - e_{2i}(\mathbf{x}_{j'})}{T}\right]^2 + \sum_{i=1}^{N_2} \left[\frac{e_{3i}(\mathbf{y}_j) - e_{3i}(\mathbf{y}_{j'})}{T}\right]^2},$$

which is the Euclidean distance between the normalized vectors $\left(\frac{\mathbf{e}_1(\mathbf{x}_j)}{B}; \frac{\mathbf{e}_2(\mathbf{x}_j)}{T}; \frac{\mathbf{e}_3(\mathbf{y}_j)}{T}\right)$ and $\left(\frac{\mathbf{e}_1(\mathbf{x}_{j'})}{B}; \frac{\mathbf{e}_2(\mathbf{x}_{j'})}{T}; \frac{\mathbf{e}_3(\mathbf{y}_{j'})}{T}\right)$. The local phase is initialized with a set of solution clusters obtained by partitioning solution list $\mathcal{S}$. The solution clusters are generated in such a way that the number of solutions in each cluster is no greater than s_3, and that for each cluster, the distance between the best solution in the cluster and any other solution in the same cluster is relatively small. The procedure for generating solution clusters is given as follows, in which $\mathcal{A}_u$ denotes the uth cluster generated:

Cluster Generation Procedure:

Step 1: Set $u \leftarrow 1$.

Step 2: Set $j \leftarrow \min\{l \mid (\mathbf{x}_l, \mathbf{y}_l) \in \mathcal{S}\}$. Create a new cluster $\mathcal{A}_u \leftarrow \{(\mathbf{x}_j, \mathbf{y}_j)\}$. Set $\mathcal{S} \leftarrow \mathcal{S} \backslash \{(\mathbf{x}_j, \mathbf{y}_j)\}$.

Step 3: If $|\mathcal{A}_u| < s_3$ and $\mathcal{S} \neq \emptyset$, then let k be the element of $\mathcal{S}$ that has the smallest d_{jk} value (ties broken arbitrarily), set $\mathcal{A}_u \leftarrow \mathcal{A}_u \cup \{(\mathbf{x}_k, \mathbf{y}_k)\}$, set $\mathcal{S} \leftarrow \mathcal{S} \setminus \{(\mathbf{x}_k, \mathbf{y}_k)\}$, and repeat Step 3.

Step 4: If $\mathcal{S} = \emptyset$, then stop. Otherwise, set $u \leftarrow u + 1$ and go to Step 2.

For each solution cluster $\mathcal{A}_u$ generated by this procedure, we apply local search to identify a locally optimal solution. We use the adaptive hyperbox algorithm developed by Xu, Nelson, and Hong (2013) as the local search method. The adaptive hyperbox algorithm is a locally convergent random search algorithm that has a special neighborhood structure called the *most promising area*, which was initially proposed by Hong and Nelson (2006). The algorithm randomly samples solutions from the most promising area and then updates the most promising area using the visited solutions. This process is executed repeatedly until the solutions converge or some predetermined stopping conditions are satisfied.

Consider any solution cluster $\mathcal{A}_u$. Let $(\mathbf{x}^*, \mathbf{y}^*)$ be the best sampled solution in $\mathcal{A}_u$; that is, $(\mathbf{x}^*, \mathbf{y}^*) = \arg\min_{(\mathbf{x},\mathbf{y})\in\mathcal{A}_u}\{\bar{g}_{\bar{\lambda}}(\mathbf{x}, \mathbf{y})\}$. For notational convenience, we denote

$$(\mathbf{e}_1; \mathbf{e}_2; \mathbf{e}_3) = ((e_{11}, \ldots, e_{1N_1}); (e_{21}, \ldots, e_{2N_1}); (e_{31}, \ldots, e_{3N_2})) = (\mathbf{e}_1(\mathbf{x}); \mathbf{e}_2(\mathbf{x}); \mathbf{e}_3(\mathbf{y}))$$

and

$$(\mathbf{e}_1^*; \mathbf{e}_2^*; \mathbf{e}_3^*) = ((e_{11}^*, \ldots, e_{1N_1}^*); (e_{21}^*, \ldots, e_{2N_1}^*); (e_{31}^*, \ldots, e_{3N_2}^*)) = (\mathbf{e}_1(\mathbf{x}^*); \mathbf{e}_2(\mathbf{x}^*); \mathbf{e}_3(\mathbf{y}^*)).$$

For any set $\mathcal{V}$ of integer vectors $(\mathbf{e}_1; \mathbf{e}_2; \mathbf{e}_3)$, let

$$L_{1i} = \begin{cases} \max_{(\mathbf{e}_1;\mathbf{e}_2;\mathbf{e}_3)\in\mathcal{V}}\{e_{1i} \mid e_{1i} < e^*_{1i}\}, & \text{if exists;} \\ 1, & \text{otherwise;} \end{cases} \tag{2.12}$$

$$U_{1i} = \begin{cases} \min_{(\mathbf{e}_1;\mathbf{e}_2;\mathbf{e}_3)\in\mathcal{V}}\{e_{1i} \mid e_{1i} > e^*_{1i}\}, & \text{if exists;} \\ B - R_i + 1, & \text{otherwise;} \end{cases} \tag{2.13}$$

$$L_{2i} = \begin{cases} \max_{(\mathbf{e}_1;\mathbf{e}_2;\mathbf{e}_3)\in\mathcal{V}}\{e_{2i} \mid e_{2i} < e^*_{2i}\}, & \text{if exists;} \\ A_i, & \text{otherwise;} \end{cases} \tag{2.14}$$

and

$$U_{2i} = \begin{cases} \min_{(\mathbf{e}_1;\mathbf{e}_2;\mathbf{e}_3)\in\mathcal{V}}\{e_{2i} \mid e_{2i} > e^*_{2i}\}, & \text{if exists;} \\ T - H_i + 1, & \text{otherwise;} \end{cases} \tag{2.15}$$

for $i = 1, \ldots, N_1$. For any $\mathcal{V}$, let

$$L_{3i} = \begin{cases} \max_{(\mathbf{e}_1;\mathbf{e}_2;\mathbf{e}_3)\in\mathcal{V}}\{e_{3i} \mid e_{3i} < e^*_{3i}\}, & \text{if exists;} \\ \underline{E}_i, & \text{otherwise;} \end{cases} \tag{2.16}$$

and

$$U_{3i} = \begin{cases} \min_{(\mathbf{e}_1;\mathbf{e}_2;\mathbf{e}_3)\in\mathcal{V}}\{e_{3i} \mid e_{3i} > e^*_{3i}\}, & \text{if exists;} \\ \bar{E}_i, & \text{otherwise;} \end{cases} \tag{2.17}$$

for $i = 1, \ldots, N_2$. The hyperbox containing $(\mathbf{e}^*_1; \mathbf{e}^*_2; \mathbf{e}^*_3)$ is

$$\mathcal{H} = \big\{(\mathbf{e}_1; \mathbf{e}_2; \mathbf{e}_3) \,\big|\, L_{1i} \le e_{1i} \le U_{1i}; L_{2i} \le e_{2i} \le U_{2i}; L_{3j} \le e_{3j} \le U_{3j}; 1 \le i \le N_1; 1 \le j \le N_2\big\}. \tag{2.18}$$

In other words, $\mathcal{H}$ consists of a set of $(2N_1 + N_2)$-dimensional vectors, in which each dimension has two edges crossing two elements in $\mathcal{V}$ that are the closest to $(\mathbf{e}^*_1; \mathbf{e}^*_2; \mathbf{e}^*_3)$ in that dimension, but not in the same position as $(\mathbf{e}^*_1; \mathbf{e}^*_2; \mathbf{e}^*_3)$.

Let $\mathbb{Z}^{2N_1+N_2}$ denote the $(2N_1 + N_2)$-dimensional integer space. The set $\mathcal{H} \cap \mathbb{Z}^{2N_1+N_2}$ is the most promising area in which the adaptive hyperbox algorithm focuses on sampling solutions. The adaptive hyperbox algorithm iteratively updates $\mathcal{V}$ and $\mathcal{H}$, and selects integer vectors randomly from $\mathcal{H} \cap \mathbb{Z}^{2N_1+N_2}$ in each iteration. Initially, we set $\mathcal{V}$ to

$$\mathcal{V}_0 = \big\{(\mathbf{e}_1; \mathbf{e}_2; \mathbf{e}_3) \bigm| (\mathbf{x}, \mathbf{y}) \in \mathcal{A}_u\big\}.$$

Given any integer vector $(\mathbf{e}_1; \mathbf{e}_2; \mathbf{e}_3) \in \mathcal{H} \cap \mathbb{Z}^{2N_1+N_2}$, the corresponding solution $(\mathbf{x}, \mathbf{y})$ can be obtained as follows: For each $i = 1, \ldots, N_1$, each $b = 1, \ldots, B$, and each $t = 0, 1, \ldots, T-1$, let $x_{ibt} = 1$ if $b = e_{1i}$ and $t = e_{2i}$, and let $x_{ibt} = 0$ otherwise. For each $i = 1, \ldots, N_2$ and each $t = 0, 1, \ldots, T$, let $y_{it} = 1$ if $t = e_{3i}$, and let $y_{it} = 0$ otherwise. Note that the solution $(\mathbf{x}, \mathbf{y})$ generated from $(\mathbf{e}_1; \mathbf{e}_2; \mathbf{e}_3)$ may violate constraint (2.4).

Our implementation of the adaptive hyperbox algorithm is described below:

Adaptive Hyperbox Algorithm:

Step 1: Determine $(\mathbf{x}^*, \mathbf{y}^*)$, $(\mathbf{e}_1^*; \mathbf{e}_2^*; \mathbf{e}_3^*)$, and $\mathcal{V}_0$. Set $\mathcal{V} \leftarrow \mathcal{V}_0$.

Step 2: Determine $\mathcal{H}$ using equations (2.12)–(2.18). Randomly generate s_4 integer vectors from $\mathcal{H} \cap \mathbb{Z}^{2N_1+N_2}$. Each integer vector $(\mathbf{e}_1; \mathbf{e}_2; \mathbf{e}_3)$ is generated as follows: For each $i = 1, \ldots, N_1$, generate e_{1i} and e_{2i} uniformly from $\{L_{1i}, L_{1i}+1, \ldots, U_{1i}\}$ and $\{L_{2i}, L_{2i}+1, \ldots, U_{2i}\}$, respectively. For each $i = 1, \ldots, N_2$, generate e_{3i} uniformly from $\{L_{3i}, L_{3i}+1, \ldots, U_{3i}\}$. Remove duplicates from the generated integer vectors. Let $\mathcal{T}$ be the set of the remaining integer vectors. Set $\mathcal{V} \leftarrow \mathcal{V} \cup \mathcal{T}$.

Step 3: For each $(\mathbf{e}_1; \mathbf{e}_2; \mathbf{e}_3) \in \mathcal{V}$, obtain the corresponding solution $(\mathbf{x}, \mathbf{y})$. If $(\mathbf{x}, \mathbf{y})$ satisfies constraint (2.4), then set $\mathcal{A}_u \leftarrow \mathcal{A}_u \cup \{(\mathbf{x}, \mathbf{y})\}$. Perform n_3 simulation replications on solution $(\mathbf{x}, \mathbf{y})$, and set $\bar{g}_{\bar{\lambda}}(\mathbf{x}, \mathbf{y})$ to be the cumulative sample mean of $g_{\bar{\lambda}}(\mathbf{x}, \mathbf{y})$ among all simulation replications (including the replications performed in previous iterations).

Step 4: Determine $(\mathbf{x}^*, \mathbf{y}^*) = \arg\min_{(\mathbf{x},\mathbf{y}) \in \mathcal{A}_u}\{\bar{g}_{\bar{\lambda}}(\mathbf{x}, \mathbf{y})\}$. Determine $(\mathbf{e}_1^*; \mathbf{e}_2^*; \mathbf{e}_3^*)$. If the stopping condition is satisfied, then stop; otherwise, go to Step 2.

Xu, Nelson, and Hong (2013) have shown that when solutions are uniformly and independently sampled from the most promising area in each iteration, the adaptive hyperbox algorithm is locally convergent; that is, the sequence of solutions generated by the algorithm converges with probability

1 to a locally optimal solution (with regard to a local neighborhood defined by solution distances). In Step 2 of our implementation, the solutions are generated from the integer vectors that are sampled uniformly and independently from $\mathcal{H} \cap \mathbb{Z}^{2N_1+N_2}$. Hence, our implementation of the adaptive hyperbox algorithm is also locally convergent (i.e., the sequence of solutions generated converges with probability 1 to a locally optimal solution of problem $\mathbf{P}'(\bar{\lambda})$) as the number of iterations approaches infinity. However, in practice, stopping conditions are needed to terminate the algorithm at some point. In our implementation, the algorithm is terminated when the current best solution $(\mathbf{x}^*, \mathbf{y}^*)$ is unchanged for h consecutive iterations. The solution $(\mathbf{x}^*, \mathbf{y}^*)$ is then treated as the optimal solution for cluster $\mathcal{A}_u$.

We observe from preliminary computational results that Step 3 of the adaptive hyperbox algorithm frequently generates solutions that violate constraint (2.4). To improve the quality of the solutions generated in Step 3, we develop a solution refinement subroutine. When a solution generated in Step 3 violates constraint (2.4), there are multiple deep-sea vessels occupying a berth segment simultaneously. In such a situation, we select one of these deep-sea vessels and reassign a service start time to the selected vessel, so that no conflicts will be incurred between the selected vessel and any other vessels. This process is executed repeatedly until all conflicts are resolved. Details of the solution refinement subroutine are provided below:

Notations:

$\mathcal{P}$: Set of (b, t) pairs for which constraint (2.4) is violated.

$\mathcal{N}_{bt}$: Set of deep-sea vessels that occupy berth segment b during time period $[t, t+1]$.

ω_{bt}: Number of deep-sea vessels that occupy berth segment b during time period $[t, t+1]$.

$\bar{b}_i$: The first berth segment allocated to deep-sea vessel i.

$\bar{t}_i$: Service start time allocated to deep-sea vessel i.

Solution Refinement Subroutine:

The solution refinement subroutine is presented as follows:

Step 1 (Initialization): For each $i = 1, \ldots, N_1$, set $\bar{b}_i \leftarrow \min\left\{b \mid \sum_{t=0}^{T-H_i} x_{ibt} = 1;\ b = 1, \ldots, B - R_i + 1\right\}$ and $\bar{t}_i \leftarrow \min\left\{t \mid x_{i\bar{b}_i t} = 1;\ t = 0, 1, \ldots, T-1\right\}$. Set $\mathcal{P} \leftarrow \emptyset$. For each $b = 1, \ldots, B$ and $t = 0, 1, \ldots, T-1$, set $\mathcal{N}_{bt} \leftarrow \emptyset$; in addition, if constraint (2.4) is violated for b and t, then set $\mathcal{P} \leftarrow \mathcal{P} \cup \{(b,t)\}$.

Step 2 (Resolving infeasibility): If $\mathcal{P} = \emptyset$, then stop. Otherwise, randomly select an element from $\mathcal{P}$, with all elements having equal probability of being selected. Let (b,t) be the selected element.

Step 2.1: For each $i = 1, \ldots, N_1$ such that $\bar{b}_i \leq b \leq \bar{b}_i + R_i - 1$ and $\bar{t}_i \leq t \leq \bar{t}_i + H_i - 1$, set $\mathcal{N}_{bt} \leftarrow \mathcal{N}_{bt} \cup \{i\}$.

Step 2.2: Randomly select an element from $\mathcal{N}_{bt}$, with all elements of $\mathcal{N}_{bt}$ having equal probability of being selected. Let i be the selected element. For each $b = 1, \ldots, B$ and $t = 0, 1, \ldots, T-1$, set $\omega_{bt} \leftarrow \sum_{i' \in \{1,\ldots,N_1\} \setminus \{i\}} \sum_{b'=\max\{b-R_{i'}+1,1\}}^{b} \sum_{t'=\max\{t-H_{i'}+1,0\}}^{t} x_{i'b't'}$.

Step 2.3: Set $\bar{t}_i \leftarrow \min\left\{t \mid \sum_{b'=\bar{b}_i}^{\bar{b}_i+R_i-1} \sum_{t'=t}^{t+H_i-1} \omega_{b't'} = 0;\ t = A_i, A_i+1, \ldots, T-H_i+1\right\}$. For each $t = 0, 1, \ldots, T-1$, if $t = \bar{t}_i$, then set $x_{i\bar{b}_i t} \leftarrow 1$; otherwise, set $x_{i\bar{b}_i t} \leftarrow 0$. Set $\mathcal{N}_{bt} \leftarrow \mathcal{N}_{bt} \setminus \{i\}$. If $|\mathcal{N}_{bt}| \leq 1$, then set $\mathcal{P} \leftarrow \mathcal{P} \setminus \{(b,t)\}$ and repeat Step 2. Otherwise, go to Step 2.2.

We execute the adaptive hyperbox algorithm with the solution refinement subroutine for each of the solution clusters $\mathcal{A}_u$ generated by the cluster generation procedure. Therefore, at the end of the local phase, we obtain m locally optimal solutions, where m is the number of solution clusters generated by the cluster generation procedure. We then execute a clean-up phase to determine the best solution among the m locally optimal solutions.

2.2.3 Clean-up Phase

At the beginning of the clean-up phase, there are m candidate solutions. If $m = 1$, then we determine accurately whether the solution satisfies constraint (2.7) (and is therefore feasible to problem **P**) with more simulation runs. Otherwise, we need to determine accurately whether the m solutions are feasible to problem **P** and choose the best solution among the feasible solutions. This requires intensive evaluation of the expected queue lengths of the feeders in the solutions.

When the number of candidate solutions is large, evaluating each solution with high precision would require an unaffordable amount of simulation budget. One commonly used method for determining the best solution in a clean-up phase is the ranking-and-selection method proposed by

Boesel, Nelson, and Kim (2003) which aims to minimize the number of simulation replications while ensuring that the selected solution is better than the other alternatives by at least δ with probability at least $1-\epsilon$, where δ is called the indifference zone and $1-\epsilon$ is the desired confidence level. Given the indifference zone, the confidence level, the number of simulation replications previously performed on each solution, and the variation of solution performance obtained in previous simulation replications, the ranking-and-selection method determines the number of additional replications that need to be allocated to each solution, and chooses the solution that has the best cumulative average performance after all additional replications are performed. In our implementation of the clean-up phase, if $m > 1$, then we select the best solution among the m candidate solutions using the ranking-and-selection method. When applying the ranking-and-selection method, the performance of each candidate solution $(\mathbf{x}, \mathbf{y})$ is measured by $g_{\bar{\lambda}}(\mathbf{x}, \mathbf{y})$, which is estimated using $\bar{g}_{\bar{\lambda}}(\mathbf{x}, \mathbf{y})$. When there is only one solution left, we evaluate the solution by running n_4 additional simulation replications, where n_4 is a large input parameter, so that the solution is evaluated with high precision. Let $\hat{Q}_{\max}(\mathbf{x}^*, \mathbf{y}^*)$ be the sample mean of $\max_{t=0,1,\dots,T-1}\{Q_t(\mathbf{x}^*, \mathbf{y}^*)\}$ generated for the best solution $(\mathbf{x}^*, \mathbf{y}^*)$ using n_4 simulation replications. The best solution $(\mathbf{x}^*, \mathbf{y}^*)$ is deemed as feasible to problem $\mathbf{P}$ if $\hat{Q}_{\max}(\mathbf{x}^*, \mathbf{y}^*) \le \bar{Q}$, in which case the solution value obtained for problem $\mathbf{P}$ is $g_0(\mathbf{x}^*, \mathbf{y}^*)$, and is deemed as infeasible to problem $\mathbf{P}$ otherwise.

2.2.4 The Simulator

For any solution $(\mathbf{x}, \mathbf{y}) \in \Theta$, we may use discrete-event simulation to determine the objective function value of $(\mathbf{x}, \mathbf{y})$ and to determine whether $(\mathbf{x}, \mathbf{y})$ satisfies constraint (2.7). In our discrete-event simulation, the events are feeder-arrival and vessel-departure. The feeder-arrival event is triggered whenever a feeder arrives at the port, and the vessel-departure event is triggered whenever a deep-sea vessel or feeder finishes service at the port. Note that the arrival times of feeders are given by the values of the y_{it} variables, and the time intervals during which each berth segment is occupied by deep-sea vessels can be derived using the values of the x_{ibt} variables. When a feeder arrives, the feeder joins the queue, and the service time of the feeder is observed (by sampling). If the feeder is in the first place of the queue and there exist some berth segments that can serve the feeder, then the feeder will be served immediately; otherwise, the feeder will wait in the queue. When vessel-departure occurs, some berth segments are released. If there exist some feeders in the queue at the moment, then the feeders will be assigned to the released berth segments according to the first-come first-served discipline described in Section 2.1. The discrete-event process is terminated when all the vessels are served and the port becomes empty. Each run of the discrete-event simulation results in an observation of $Q_t(\mathbf{x}, \mathbf{y})$ for $t = 0, 1, \ldots, T-1$. The values of $E[Q_t(\mathbf{x}, \mathbf{y})]$ and $g_\lambda(\mathbf{x}, \mathbf{y})$ can be estimated by running the simulation multiple times and taking the sample mean of multiple runs.

2.3 Computational Experiments

The goal of the computational experiments is threefold. First, we would like to evaluate the computational performance of the simulation optimization method for solving problem instances of realistic sizes. For this purpose, we generate problem instances based on the operational data of the Yangshan Deep-water Port in Shanghai, and compare the computational performance of our simulation optimization method with the performances of some benchmark methods. Three genetic algorithms are used as benchmark methods. The first genetic algorithm is the genetic algorithm used in the global phase of our simulation optimization method. The second genetic algorithm is the same as the first genetic algorithm, except that it evaluates the fitness of each individual using the mean service times of feeders rather than the sampled service times. The third genetic algorithm is also the same as the first genetic algorithm, except that it evaluates the fitness of each individual with high precision by assigning a large simulation budget to each visited individual. Since users of our simulation optimization method may want to set different queue length limits for feeders under different situations (e.g., different weather conditions), our second goal is to examine how the computational results are affected when the value of $\bar{Q}$ varies. Our third goal is to evaluate the benefits of controlling the queue length of feeders when making berth plans. For this purpose, we analyze the solutions obtained by the current practice of the port operators that ignore the queue length restriction, and compare the solutions with those obtained by our simulation optimization method.

All algorithms were implemented in C#.Net and run on a computer with a 64-bit Intel i7-6700

3.40GHz CPU and 32GB of RAM.

2.3.1 Problem Instances and Algorithm Parameters

We generate test instances based on the operational data of the Yangshan Deep-water Port in Shanghai. Table 2.2 presents the statistics of the operational data of year 2016 at the port. For each month of the year, the following statistics are presented:

- Vess_Num: Number of deep-sea vessels served.
- Feed_Num: Number of feeders served.
- Vess_Len_Min, Vess_Len_Max, Vess_Len_Avg: Minimum, maximum, and average lengths (in meters) of deep-sea vessels.
- Feed_Len_Min, Feed_Len_Max, Feed_Len_Avg: Minimum, maximum, and average lengths (in meters) of feeders.
- Vess_Time_Min, Vess_Time_Max, Vess_Time_Avg: Minimum, maximum, and average service times (in hours) of deep-sea vessels.
- Feed_Time_Min, Feed_Time_Max, Feed_Time_Avg: Minimum, maximum, and average service times (in hours) of feeders.

Table 2.2: Monthly statistics of the operational data for 2016 at the Yangshan Deep-water Port*

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Average
Vess_Num	381	356	403	382	376	396	411	398	360	363	344	357	377.3
Feed_Num	1158	1061	1058	1122	1241	1162	1211	1250	1161	1104	1104	1131	1146.9
Vess_Len_Min	224.0	228.0	228.0	228.0	228.0	228.0	209.0	228.0	228.0	222.0	222.0	207.0	223.3
Vess_Len_Max	400.0	400.0	400.0	400.0	400.0	400.0	400.0	400.0	400.0	400.0	400.0	400.0	400.0
Vess_Len_Avg	308.1	307.4	312.5	312.8	312.8	316.6	314.5	319.6	318.3	321.0	321.2	320.9	315.5
Feed_Len_Min	75.0	75.0	75.0	75.0	75.0	75.0	75.0	75.0	75.0	75.0	75.0	75.0	75.0
Feed_Len_Max	140.0	140.0	157.0	140.0	140.0	140.0	140.0	140.0	140.0	140.0	140.0	140.0	141.4
Feed_Len_Avg	96.0	96.1	97.3	95.4	95.2	96.6	96.2	96.1	95.1	95.7	96.1	97.0	96.1
Vess_Time_Min	6.2	5.3	5.2	5.5	5.6	6.0	5.5	6.0	5.1	5.2	5.5	5.6	5.6
Vess_Time_Max	37.5	35.8	35.5	33.4	35.7	30.1	28.3	35.7	32.0	35.5	35.5	32.7	34.0
Vess_Time_Avg	14.8	13.2	14.8	14.1	14.3	13.3	14.7	14.9	15.3	16.8	16.8	16.9	15.0
Feed_Time_Min	0.4	0.4	0.7	0.6	0.2	0.7	0.2	0.3	0.3	0.5	0.2	0.5	0.4
Feed_Time_Max	10.0	8.2	9.7	8.1	9.7	9.2	9.1	8.6	9.0	9.1	9.7	8.5	9.1
Feed_Time_Avg	2.8	3.4	4.2	3.2	3.8	4.0	3.2	3.5	4.2	4.5	3.8	4.0	3.7

*These statistics are obtained by ignoring data entries that are either improperly recorded or recorded under abnormal situations (e.g., long vessel service times caused by equipment breakdowns).

It can be observed from Table 2.2 that the number of feeders served in the port is significantly larger than the number of deep-sea vessels. The average numbers of deep-sea vessels and feeders served monthly are 377.3 and 1146.9, respectively, indicating that the daily average numbers of deep-sea vessels and feeders served are 12.6 and 38.2, respectively. In each of our test instances, the number of deep-sea vessels and the number of feeders are set to $N_1 = 25$ and $N_2 = 80$, respectively, which reflect the number of service requests over two days. The reason for using such a parameter setting is that accurate throughput information of deep-sea vessels usually becomes available two days before the vessels arrive at the port. The length of each time period of the planning horizon is set to 1 hour. The length of the planning horizon is set to $T = 96$ (i.e., 4 days), which is long enough to ensure service completion of all vessels.

Since the port has a quay wall of 5600 meters, we set the number of berth segments to $B = 40$, so that each berth segment is 140 meters long, which can accommodate most feeders. The length of each deep-sea vessel ranges between 207 meters and 400 meters. Since the lengths of deep-sea vessels are modeled as multiples of the length of each feeders, we generate the length of each deep-sea vessel i by $R_i = \lceil \frac{\alpha}{140} \rceil$, where α is a real number generated uniformly from $[207, 400]$.

The mean service time of deep-sea vessels over the year is 15.0 hours, and the variance of deep-sea vessel service times is 40.8. Hence, we generate the service time H_i of each deep-sea vessel i using a normal distribution with mean 15.0 and variance 40.8. When generating the value of H_i, we round the sampled value up to the nearest positive integer. In order to evaluate the impact of the feeder service time variance on the solution of problem $\mathbf{P}$, we generate test instances with three different probability distributions of the feeder service times, where the three probability distributions differ from each other only in the variance. The mean service time of feeders over the year is 3.7 hours, and the variance of the feeder service times is 5.2. Hence, our first probability distribution is a normal distribution with mean 3.7 and variance 2.6, our second probability distribution is a normal distribution with mean 3.7 and variance 5.2, and our third probability distribution is a normal distribution with mean 3.7 and variance 7.8. When sampling feeder service times, we truncate the normal distributions so that the sampled service times are non-negative.

We generate the port arrival time A_i of each deep-sea vessel i uniformly from $\{0, 1, \ldots, 48\}$, and set the target departure time to $D_i = A_i + H_i$. We generate the initially scheduled port arrival time S_i of each feeder i uniformly from $\{0, 1, \ldots, 48\}$. A maximum displacement of 12 hours on each feeder arrival time is normally acceptable for port operators. We therefore set the earliest allowed arrival time of each feeder i to $\underline{E}_i = \max\{0, S_i - 12\}$, and set the latest allowed arrival time of each feeder i to $\bar{E}_i = S_i + 12$. Since deep-sea vessels have a higher service priority than feeders, we set the value of C_{1i} to 5 for each $i = 1, \ldots, N_1$, and set the value of C_{2i} to 1 for each $i = 1, \ldots, N_2$.

We consider five different feeder queue length limits by setting $\bar{Q}$ to 5, 10, 15, 20, and 25. As mentioned above, we consider three different feeder service time variances. Thus, there are 15 problem sets. For each problem set, we generate 10 random instances. Hence, there are 150 instances in total. Table 2.3 summaries the configurations of the problem sets used in our computational experiments. The values of the parameters used in our simulation optimization method are provided in Table 2.4.

Table 2.3: Parameter values used in the computational study.

Low service time variance			Medium service time variance			High service time variance		
Problem set	$\bar{Q}$	Service time variance	Problem set	$\bar{Q}$	Service time variance	Problem set	$\bar{Q}$	Service time variance
L1	5	2.6	M1	5	5.2	H1	5	7.8
L2	10	2.6	M2	10	5.2	H2	10	7.8
L3	15	2.6	M3	15	5.2	H3	15	7.8
L4	20	2.6	M4	20	5.2	H4	20	7.8
L5	25	2.6	M5	25	5.2	H5	25	7.8

Table 2.4: Parameter values used in the simulation optimization method.

Phase	Parameter	Value
Global phase	s_1	100
	s_2	100
	μ	0.1
	$\bar{\ell}$	60
	n_1	20
	n_2	5
	λ_0	100
	ζ	0.5
Local Phase	s_3	20
	s_4	85
	h	10
	n_3	50
	$\bar{\lambda}$	10^5
Clean-up phase	δ	1
	ϵ	0.05
	n_4	3000

2.3.2 Benchmark Methods

Genetic algorithms are widely used for solving berth allocation problems with stochastic vessel information. Our method differs from the existing genetic algorithms in that our method combines local search with genetic algorithm and allocates different simulation budgets to solutions in different search phases to strike a balance between exploration and exploitation, whereas existing genetic algorithms lack mechanisms for controlling allocation of the simulation budget. For example, there exist genetic algorithms that simplify the stochasticity of vessel information by replacing the uncertain elements with their expected values, and generate solutions without simulation (see, e.g., Golias 2011; Karafa et al. 2013), as well as genetic algorithms that attempt to find solutions with good performance by assigning a large simulation budget to the visited solutions (see, e.g., Han, Lu, and Xi 2010).

To evaluate the computational performance of our simulation optimization method, we introduce three genetic algorithms as benchmark methods and compare the performance of our method with those of the benchmark methods. Each of the benchmark methods only executes a genetic algorithm followed by a clean-up procedure, without dividing the search process into a global phase and a local phase. The first benchmark method, denoted GA1, is the genetic algorithm used in the global phase of our simulation optimization method (see Section 2.2.1). The second benchmark method, denoted GA2, is the same as the genetic algorithm used in the global phase of our simulation optimization method, except that it evaluates the fitness of each individual using the mean service times of feeders rather than the sampled service times. Hence, the fitness of each individual is deterministic in each iteration of the second benchmark method. The third benchmark method, denoted GA3, is also the same as the genetic algorithm used in the global phase of our simulation optimization method, except that it evaluates the fitness of each individual accurately by assigning a large simulation budget to each visited individual. We use the same simulation budget as in Han, Lu, and Xi (2010), and allocate 750 simulation replications to each solution that needs to be evaluated. When the genetic algorithm of a benchmark method terminates, we execute a clean-up phase to choose the best solution among the solutions generated by the genetic algorithm.

2.3.3 Comparison with Benchmark Methods

In this subsection, we evaluate the computational performance of the simulation optimization method for instances with a medium queue length limit. We solve test instances of problem sets L3, M3, and H3 using the simulation optimization method and the three benchmark methods (i.e., GA1, GA2, and GA3). We run each of the solution methods five times for each test instance, and record the computational results obtained in each run. Table 2.5 summarizes the computational results. For each solution method and each test instance, the "Min," "Max," and "Avg" columns in Table 2.5 report the minimum solution value, the maximum solution value, and the average solution value, respectively, among the five independent runs. The "Time" column reports the average computation time (in seconds) of the five runs.

From Table 2.5, we observe that the average solution values of the simulation optimization method are smaller than those of GA1. This indicates that the local phase of the simulation optimization method improves the solutions generated by the global phase. We also observe from the "Time" columns that the computation time of the simulation optimization method is much longer than that of GA1. This is due to the fact that in the simulation optimization method, the local phase uses a larger simulation budget than the global phase in order to evaluate the performance of solutions more accurately. The running time of GA2 is much shorter than that of the simulation optimization method. However, the average solution values of GA2 are consistently larger than those of the simulation optimization method, indicating that the solutions generated by GA2 are inferior to those generated by the simulation optimization method. This is expected, since the short computation time of GA2 is achieved at the cost of ignoring the uncertainties of feeder service times, which makes the evaluation of solution performance efficient, but which can result in significant errors in the estimation of solution values. On the other hand, GA3 generates better solutions than GA1 and GA2 (see the "Avg" values in Table 2.5). Since the simulation budget used by GA3 for solution evaluation is substantially larger than those used by GA1 and GA2, GA3 estimates solution values more accurately than GA1 and GA2 in each iteration, and is therefore able to generate better solutions. However, because GA3 consumes a large amount of simulation budget, its computation usually requires several hours. Compared to GA3, the

Table 2.5: Computational results for instances of problem sets L3, M3, and H3

	Simulation optimization				GA1				GA2				GA3			
Instance	Min	Max	Avg	Time	Min	Max	Avg	Time	Min	Max	Avg	Time	Min	Max	Avg	Time
L3.1	133	145	140.9	1307.3	180	212	189.7	371.3	179	238	193.8	158.3	135	178	165.2	30410.3
L3.2	238	254	246.5	1538.0	270	323	291.1	432.8	291	355	311.6	100.4	249	295	276.9	28209.2
L3.3	199	216	205.4	1461.7	208	259	236.9	401.3	235	304	250.1	166.4	206	226	210.0	26097.5
L3.4	386	393	389.0	1451.5	406	466	421.2	404.9	455	557	474.8	124.4	382	423	406.7	26558.6
L3.5	349	362	352.7	1402.3	355	438	408.6	400.5	381	531	420.0	104.5	359	416	386.9	27923.2
L3.6	248	261	252.6	1555.7	273	307	287.1	442.8	278	381	321.3	141.1	260	287	283.5	29069.4
L3.7	378	388	383.3	1524.5	394	440	415.0	434.5	393	494	462.4	187.8	380	404	397.5	25396.1
L3.8	69	91	83.7	1410.2	101	180	141.6	411.4	126	192	147.2	186.1	94	168	126.8	27298.0
L3.9	147	161	153.7	1427.5	121	212	176.1	396.1	182	301	261.1	98.0	137	191	162.2	30108.9
L3.10	151	175	159.8	1456.9	171	217	195.2	394.6	167	316	200.6	109.9	165	193	177.1	26708.8
Avg	229.8	244.6	236.8	1453.6	247.9	305.4	276.2	409.0	268.7	366.9	304.3	137.7	236.7	278.1	259.3	27778.0
M3.1	228	271	255.2	1410.3	247	333	306.1	395.6	339	391	354.3	160.7	240	284	270.1	29924.6
M3.2	433	475	438.3	1533.9	461	519	485.2	424.2	476	623	540.3	102.0	433	513	470.3	28206.3
M3.3	420	439	427.5	1512.6	431	484	458.9	402.1	479	556	501.9	163.5	421	458	427.8	25656.7
M3.4	455	473	461.2	1484.7	471	538	484.4	401.3	524	650	566.5	129.7	439	528	462.8	26438.4
M3.5	457	498	469.3	1368.1	487	571	517.0	364.8	492	640	527.2	100.2	455	534	474.0	27448.5
M3.6	504	533	522.6	1559.7	542	589	556.5	423.3	501	589	541.0	137.2	508	561	532.1	28704.7
M3.7	642	675	657.8	1557.7	661	804	689.5	413.5	664	804	745.1	185.1	639	716	671.9	25062.3
M3.8	252	279	270.7	1523.5	285	357	321.9	419.5	310	389	329.4	182.3	284	345	310.7	27299.6
M3.9	242	275	262.7	1488.8	266	335	299.9	383.1	293	384	341.4	97.3	250	312	283.8	30387.4
M3.10	296	322	305.1	1429.8	309	360	333.2	386.4	311	441	347.4	110.4	296	332	323.9	26747.4
Avg	392.8	424.0	407.0	1486.9	415.7	489.0	445.3	401.4	438.9	546.7	479.5	136.9	396.6	458.4	422.7	27587.6
H3.1	327	402	373.4	1661.6	360	471	427.2	398.0	471	512	485.1	162.9	360	397	382.5	30025.1
H3.2	782	850	784.6	1718.8	806	890	832.6	450.6	820	1003	889.8	101.4	777	899	817.5	27829.1
H3.3	664	685	671.9	1688.1	674	739	701.0	418.5	697	813	750.3	169.7	653	723	666.9	25960.5
H3.4	711	741	721.9	1744.5	716	819	727.8	453.5	788	935	837.4	126.1	694	780	711.0	26417.3
H3.5	684	754	705.5	1626.7	695	789	724.7	444.1	731	866	751.5	107.1	694	781	704.6	28377.1
H3.6	729	774	761.6	1750.2	749	830	795.3	465.0	697	723	713.0	139.2	739	802	766.9	28870.1
H3.7	959	1015	984.3	1816.9	985	1135	1016.1	430.1	986	1144	1073.3	184.0	940	1065	998.3	24944.7
H3.8	765	796	787.7	1746.2	802	871	838.6	426.3	823	907	842.2	190.3	777	824	795.6	27286.7
H3.9	558	610	593.1	1649.5	606	660	639.0	439.1	648	733	671.6	102.9	585	650	623.7	30311.9
H3.10	463	491	472.0	1739.4	476	539	499.9	441.5	475	583	518.1	109.6	460	500	497.0	26626.5
Avg	664.2	711.8	685.6	1714.2	686.9	774.3	720.2	436.7	713.6	821.9	753.2	139.3	667.9	742.1	696.4	27664.9

simulation optimization method generates better solutions with a considerably smaller amount of computation effort. Hence, the simulation optimization method outperforms the benchmark methods in minimizing the tardiness and displacement cost of vessels.

Another observation from Table 2.5 is that the solution values obtained by the simulation optimization method in multiple runs do not show high variations, whereas the solution values obtained by the benchmark methods in multiple runs vary considerably. For example, the gap between the “Min” and “Max” values of the simulation optimization method for instance H3.10 is 28, whereas the gaps resulting from GA1, GA2, and GA3 for the same instance are 63, 108, and

40, respectively. This indicates that the simulation optimization method is more robust than the benchmark methods in terms of solution quality. Recall that problem sets L3, M3, and H3 differ from each other only in the variance of feeder service times. Comparing the results obtained for different problem sets, we observe that the solution values tend to increase as the feeder service time variance increases. This is because when the feeders have a higher variation in service time, the solutions tend to reserve longer berthing times for feeders, incurring larger departure tardiness for deep-sea vessels, which increases the overall solution values.

2.3.4 Varying the Queue Length Limit

Next, we investigate how the computational results of the simulation optimization method and the benchmark methods are affected as the upper limit on expected queue length of feeders varies. We consider the test instances with different $\bar{Q}$ values, and we run each of the solution methods five times for each instance. Table 2.6 summarizes the computational results, where each row reports the average computational results of ten instances in a problem set.

Table 2.6: Results for problem sets with different $\bar{Q}$ values.

Problem		Simulation optimization		GA1		GA2		GA3	
set	$\bar{Q}$	Avg	Time	Avg	Time	Avg	Time	Avg	Time
L1	5	315.5	1435.5	385.3	391.2	407.0	125.3	352.2	28380.0
L2	10	275.9	1465.7	332.4	407.9	373.1	101.4	296.4	25675.2
L3	15	236.8	1453.6	276.3	409.0	304.3	137.7	259.3	27778.0
L4	20	199.1	1472.8	229.0	405.1	277.3	150.9	209.1	29331.6
L5	25	174.3	1404.5	190.8	385.3	232.6	157.6	179.8	25211.6
M1	5	643.1	1478.4	701.4	388.4	768.3	131.2	665.9	28531.8
M2	10	501.2	1501.1	546.3	396.2	600.9	124.5	513.5	25999.9
M3	15	407.0	1486.9	445.3	401.4	479.5	136.9	422.7	27587.6
M4	20	328.6	1473.1	347.6	388.7	402.8	134.7	335.1	28402.6
M5	25	282.7	1437.4	310.3	360.5	339.8	156.7	279.0	25134.7
H1	5	893.6	1802.0	936.3	430.1	1003.7	136.1	918.6	30738.3
H2	10	741.4	1760.8	766.7	437.6	817.4	144.3	753.6	29766.1
H3	15	685.6	1714.2	720.2	436.7	753.2	139.3	696.4	27664.9
H4	20	568.9	1687.9	581.6	429.1	606.6	134.2	555.6	28498.1
H5	25	430.3	1467.2	444.2	323.7	474.7	158.6	414.9	29718.9

From Table 2.6, we observe that the average solution values obtained by the simulation optimization method are smaller than those obtained by GA1 for different $\bar{Q}$ values. This shows that the local phase of the simulation optimization method improves the solutions obtained by the global phase for various queue length limits. However, as the value of $\bar{Q}$ increases, the improvement in the solution value generated by the local phase tends to diminish. For example, the improvement in the average solution value obtained by the local phase is 69.8 for problem set L1, where $\bar{Q}$ is set equal to 5. However, the improvement is only 16.5 for problem set L5, where $\bar{Q}$ is set equal to 25. One possible reason for this tendency is that as the queue length limit increases, constraint (2.7) of problem **P** becomes less restrictive, increasing the chance of identifying good solutions in the global phase of the simulation optimization method. The average solution values obtained by GA2 are consistently worse than those obtained by the simulation optimization method, showing that the solutions obtained by ignoring uncertainties are inferior under different queue length limits. On the other hand, the average solution values obtained by GA3 are worse than those obtained by the simulation optimization method when $\bar{Q}$ is small, but are comparable to those obtained by the simulation optimization method when $\bar{Q}$ becomes larger. However, the computation time of GA3 is much longer than that of the simulation optimization method. Overall, the simulation optimization method outperforms the benchmark methods for different queue length limits.

2.3.5 Benefits of Controlling the Queue Length of Feeders

As mentioned in Section 1.1, port operators usually serve deep-sea vessels by ignoring the uncertainties of feeder service times (i.e., they develop detailed berth plans for deep-sea vessels based on complete deep-sea vessel information) and allocate berths to feeders dynamically when feeders arrive at the port. This service strategy can achieve good service levels for deep-sea vessels, as berth space is reserved exclusively for deep-sea vessels. However, this strategy cannot control the waiting times of feeders and is unable to mitigate port congestion incurred by feeder traffic. Unlike the service strategy adopted by port operators, our simulation optimization method generates berth plans for deep-sea vessels by taking into account feeder service times, and controls port congestion and feeder waiting times by restricting the expected queue length of feeders. In this subsection, we compare the performance of the simulation optimization method with the current practice of port operators, and reveal the benefits of controlling the queue length of feeders.

We mimic the current practice of port operators using a sequential decision heuristic. In the current practice, the planners ignore the queue length limit of feeders, i.e., the value of $\bar{Q}$ in constraint (2.7) of problem $\mathbf{P}$ is set to be infinite. Note that after setting $\bar{Q}$ equal to infinity, problem $\mathbf{P}$ decomposes into two subproblems. One subproblem involves only the deep-sea vessels, while the other subproblem involves only the feeders. The sequential decision heuristic first solves a deterministic berth allocation model of the deep-sea vessels, which involves only the x_{ibt} and l_{1i} variables, and then allocates berths to feeders according to the first-come first-served discipline described in Section 2.1. The deterministic berth allocation model of the deep-sea vessels is presented as follows:

$$\text{minimize} \quad \sum_{i=1}^{N_1} C_{1i} l_{1i}$$

$$\text{subject to (2.2), (2.4), (2.5), (2.8)}$$

We solve the deterministic berth allocation model using CPLEX to obtain the total tardiness cost of deep-sea vessels and the values of the x_{ibt} variables. To generate a service plan $(\mathbf{x}, \mathbf{y})$, we need the values of the y_{it} variables. For each $i = 1, \ldots, N_2$ and $t = 0, 1, \ldots, T-1$, we set $y_{it} = 1$ if $t = S_i$, and set $y_{it} = 0$ otherwise. With the value of $(\mathbf{x}, \mathbf{y})$, we run n_4 simulation replications to evaluate the expected queue lengths of feeders and the average waiting times of feeders.

We solve problem sets M1, M2, M3, M4, and M5 using the sequential decision heuristic, and compare the solutions obtained by the sequential decision heuristic with those obtained by the simulation optimization method using the following performance measures: (i) average departure tardiness of deep-sea vessels (AT); (ii) average schedule displacement of feeders (AD); (iii) peak expected queue length of feeders (PQL); and (iv) average waiting time of feeders (AWT). Table 2.7 compares the computational results obtained by the simulation optimization method with those obtained by the sequential decision heuristic for the instances in problem set M3. All time-related values reported in Table 2.7 are in hours.

From Table 2.7, we observe that the average AT and AD values of the sequential decision heuristic are 0.4 and 0.0, respectively, indicating that the sequential decision heuristic performs well in minimizing the departure tardiness of deep-sea vessels and the schedule displacements of feeders. Compared to the sequential decision heuristic, the simulation optimization method has larger AT and AD values. However, the PQL and AWT values of the simulation optimization method are much smaller than those of the sequential decision heuristic. This shows that the simulation

Table 2.7: Results for instances of problem set M3.

	Simulation optimization				Sequential heuristic			
Instance	AT	AD	PQL	AWT	AT	AD	PQL	AWT
M3.1	1.1	1.5	14.9	2.1	0.2	0.0	26.5	4.3
M3.2	2.7	1.2	13.6	4.3	0.5	0.0	37.9	12.4
M3.3	1.5	3.0	14.3	2.6	0.1	0.0	40.3	11.8
M3.4	2.7	1.6	15.0	1.8	0.9	0.0	36.9	8.0
M3.5	2.7	1.7	13.0	2.4	0.8	0.0	41.2	11.0
M3.6	2.5	2.5	14.3	5.0	0.2	0.0	43.3	13.6
M3.7	4.1	1.5	11.7	3.4	1.0	0.0	53.6	18.9
M3.8	1.6	0.8	14.9	3.5	0.0	0.0	42.3	13.3
M3.9	1.2	1.3	14.4	2.9	0.1	0.0	35.7	10.1
M3.10	1.6	1.3	14.7	3.3	0.0	0.0	41.8	11.0
Average	2.2	1.6	14.1	3.1	0.4	0.0	39.9	11.4

optimization method generates solutions that strike a balance between congestion mitigation and deep-sea vessel service quality, and performs much better than the sequential decision method in reducing feeder waiting times.

Figure 2.1 depicts the average performance measures obtained by the simulation optimization method for different $\bar{Q}$ values, as well as performance measures obtained by the sequential decision heuristic. From Figure 2.1, we observe that the average departure tardiness of deep-sea vessels and the average schedule displacement of feeders generated by the simulation optimization method tend to decrease as $\bar{Q}$ increases, whereas the peak expected queue length of feeders and the average waiting time of feeders tend to increase as $\bar{Q}$ increases. On the other hand, since the sequential decision heuristic ignores the queue length restriction of feeders, its computational results are independent of the $\bar{Q}$ values. The comparisons show that the simulation optimization method is more flexible than the sequential decision heuristic in controlling the expected queue length of feeders, enabling terminal operators to quantify the trade-off between alleviating port congestion and enhancing vessel service quality. For container ports that serve a large number of feeders (e.g., the Port of Shanghai), long waiting lines of feeders can result in severe congestion in the port basin, impeding the service of vessels and creating a high risk of vessel collisions. Hence, the simulation optimization method would be more preferable than the current practice of port operators for berth allocation and congestion mitigation.

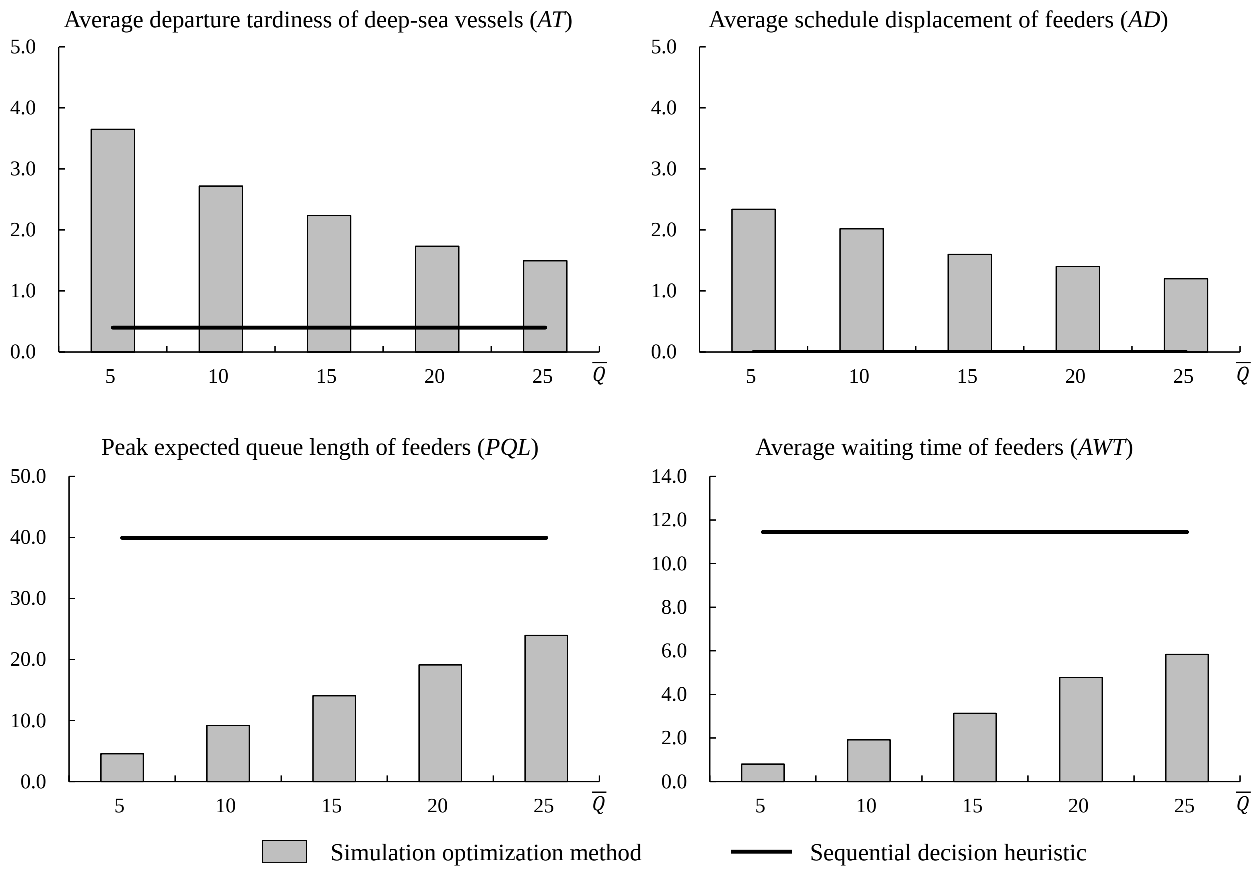

Figure 2.1: Performance measures of solutions obtained for different $\bar{Q}$ values.

2.4 Extensions

Our solution method can be modified easily to handle different extensions of the problem. First, consider the situation where some vessels are being served and some feeders are waiting for service at the beginning of the planning horizon. Having some vessels being served at time 0 implies that some berth segments are unavailable for some initial time periods. We can modify our simulation optimization method to handle this situation. This is done by modifying the decoding scheme in such a way that allocating berth segments to vessels during the berth segments' unavailable period is disallowed. Having some feeders waiting for service at time 0 implies that those arrived feeders have known service times, and thus the simulator must be modified accordingly. Next, consider the situation where the feeder lengths are non-identical. In this situation, the first-come first-served service rule described in Section 2.1 needs to be modified to ensure that a feeder can only be served when there are sufficient consecutive berth segments available to accommodate the feeder. Our simulation optimization method can be applied to this situation if we modify the decoding scheme by allocating berth segments to a feeder only when there are sufficient consecutive berth segments available. Finally, we remark that by modifying the implementation of the simulator, our simulation optimization method is also applicable to the situation where feeders are served according to particular service rules other than the first-come first-served rule.

Chapter 3

Managing Navigation Channel Traffic and Anchorage Area Utilization

3.1 Problem Description and Formulation

Consider a planning horizon of length T, during which n_1 incoming vessels and n_2 outgoing vessels are scheduled for service at the container terminals of a port. Note that the number of incoming vessels may be different from the number of outgoing vessels, as a vessel that enters the terminal basin during a planning cycle may leave in another cycle. Each incoming vessel i has an arrival time A_i. Thus, vessel i may enter the navigation channel at or after time A_i. Incoming vessel i also has a planned berthing time B_i as specified by the berth plans. Each outgoing vessel i has a planned service completion time E_i at the berth, and a target departure time D_i (i.e., time for the vessel to arrive at the outer anchorage ground after service).

The water level in the navigation channel is tide-dependent, and a vessel can only sail in the channel when the water becomes deep enough to hold the draft. Because the water level rises and drops with the fluctuation of tide, each vessel i is subject to a number of tidal windows, during which it can sail through the channel with a satisfactory water level. The vessels' tidal windows are "nested," i.e., the tidal windows for vessels with a deeper draft are subsets of the tidal windows for vessels with a shallower draft. Figure 3.1 illustrates the variation of the water level over time, and the nested tidal windows for vessels with different drafts. We let u_i denote the number of tidal windows that vessel i has during the planning horizon, and $[\underline{w}_{il}, \bar{w}_{il}]$ denote the lth tidal window for vessel i.

Each vessel i has been assigned to a berthing position $b(i)$ where the vessel should moor and

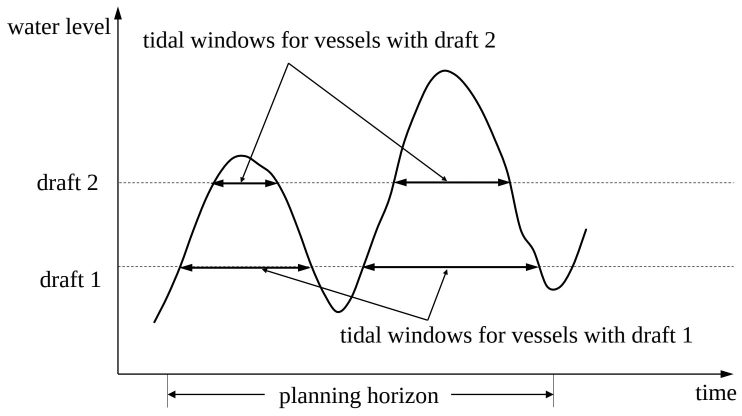
water level
tidal windows for vessels with draft 2
draft 2
draft 1
tidal windows for vessels with draft 1
planning horizon
time

Figure 3.1: Nested tidal windows of vessels with different drafts.

be handled. Here, $b(i)$ represents a berth if the cargo terminal has a discrete berth layout, and a specific position along the quay if the cargo terminal has a continuous berth layout. As mentioned in Section 1.2, incoming vessels may utilize the staging anchorages if they arrive at the terminal basin too early, and outgoing vessels may utilize the staging anchorages if they need to wait for the availability of the navigation channel. Denote $\bar{\tau}$ as the navigation time in the navigation channel. When traveling through the navigation channel, vessels in the same traffic lane must keep a safety clearance σ, in terms of the length of travel time between two successive vessels. There are m separate staging anchorages in the terminal basin, which may be located at different locations. We denote $\tau_{0,b(i)}$, τ'_{0k}, and $\tau''_{k,b(i)}$ as the travel time between the end of the navigation channel and berthing position of vessel i, the travel time between the end of the navigation channel and staging anchorage k, and the travel time between berthing position of vessel i and staging anchorage k, respectively (see Figure 3.2).

To comply with the predetermined berth plans, an incoming vessel should not berth earlier than its planned berthing time B_i in order to avoid interference on the berthing activities of other vessels. In practice, some flexibility is allowed for the berthing of each incoming vessel, but a deadline $\bar{B}_i$ ($\bar{B}_i \geq B_i$) on the actual berthing time of each incoming vessel i is imposed by the cargo terminal. Therefore, vessel i should berth within the feasible time interval $[B_i, \bar{B}_i]$ so as to complete service on time. If the actual berthing time is later than B_i, berthing tardiness is incurred, resulting in a

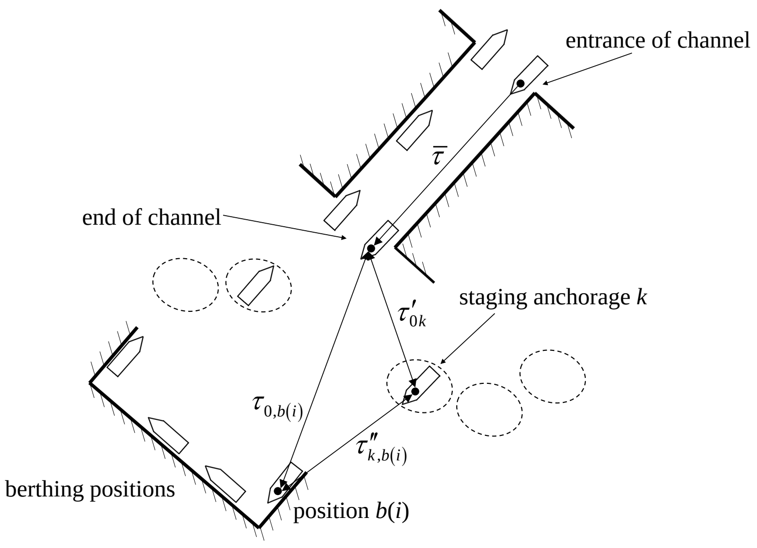

Figure 3.2: Possible movements of vessels at the seaport and their corresponding travel times.

penalty cost. The tardiness penalty for late berthing is C_{1i} per time unit for each incoming vessel i. If the actual berthing time is later than $\bar{B}_i$, service completion will be delayed, which may affect the actual berthing time of subsequent vessels, causing uncontrolled execution of the predetermined berth plans. Each outgoing vessel i must unberth at time E_i. After unberthing, the outgoing vessel will either sail directly through the navigation channel to the outer anchorage ground for departure, or wait at a staging anchorage before sailing through the channel. On-time or early departure is preferred, but late departure, i.e., departure later than D_i, is allowed. The tardiness penalty for late departure is C_{2i} per unit time for each outgoing vessel i. If a vessel cannot be served at its minimum requirement (i.e., an incoming vessel i cannot berth before its berthing deadline $\bar{B}_i$ or an outgoing vessel i cannot unberth at time E_i), the berth plan for the vessel will need to be revised by the terminal operator. We let C_{0i} be a large lump-sum penalty on such incidence for each vessel i.

We assume that all time-related parameters in our model are multiples of the safety clearance σ. Because σ is relatively small compared to other travel time parameters, the inaccuracy of the solution caused by this assumption is relatively insignificant. Thus, by setting $\sigma = 1$, all time-related parameters in our model become integer-valued. The problem involves decisions for determining the actual berthing time of incoming vessels, the actual departure time of outgoing vessels, and the time points for the incoming and outgoing vessels to enter the channel. In addition, the utilization of the staging anchorages (i.e., whether a vessel should occupy a staging anchorage or not, which staging anchorage is assigned to a vessel, and the time points for a vessel to enter and leave the staging anchorage) must also be determined. The objective of the problem is to minimize the total cost incurred by berthing tardiness of incoming vessels and departure tardiness of outgoing vessels, as well as the cost incurred by the unsatisfied service requests. We denote this problem as **P**. Additional assumptions of problem **P** are as follows:

(i) The outer anchorage ground is beyond the navigation channel and has infinite capacity. Therefore, the utilization of the outer anchorage ground is not considered in our model.

(ii) Each staging anchorage can accommodate exactly one vessel. In addition, a vessel cannot stay partially at one staging anchorage and partially at another staging anchorage.

(iii) All the staging anchorages are available for the vessels throughout the planning horizon.

(iv) The travel speed is constant and identical for all vessels in the navigation channel; the travel speed is also constant and identical for all vessels in the terminal basin. Thus, the time for traveling through the navigation channel is identical for all vessels, and the time for traveling between any two locations in the terminal basin is symmetric and identical for all vessels.

For notational convenience, we number the incoming vessels from 1 to n_1, and the outgoing vessels from $n_1 + 1$ to $n_1 + n_2$. Let M denote a large number. Problem **P** can be formulated as an MILP as follows.

Input parameters:

T: Length of planning horizon

n_1: Number of incoming vessels

n_2: Number of outgoing vessels

m: Number of staging anchorages

$b(i)$: Designated berthing position for vessel i

u_i: Number of tidal windows for vessel i

$[\underline{w}_{il}, \bar{w}_{il}]$: lth tidal window for vessel i

A_i: Arrival time of incoming vessel i (i.e., the earliest possible time to enter the navigation channel)

B_i: Planned berthing time of incoming vessel i

$\bar{B}_i$: Latest allowed berthing time of incoming vessel i

E_i: The time that outgoing vessel i must leave berthing position $b(i)$

D_i: Expected departure time of outgoing vessel i

$\bar{\tau}$: Amount of time for a vessel to travel through the navigation channel

$\tau_{0,b(i)}$: Amount of time for vessel i to travel directly between the end of the navigation channel and berthing position $b(i)$

τ'_{0k}: Amount of time for a vessel to travel between the end of the navigation channel and staging anchorage k ($\tau'_{00} = 0$)

$\tau''_{k,b(i)}$: Amount of time for vessel i to travel between staging anchorage k and berthing position $b(i)$ ($\tau''_{0,b(i)} = \tau_{0,b(i)}$)

C_{1i}: Unit cost of berthing tardiness for incoming vessel i

C_{2i}: Unit cost of departure tardiness for outgoing vessel i

C_{0i}: A (large) lump-sum cost if the service request of vessel i cannot be satisfied.

Decision variables:

x_{it}: =1 if vessel i enters the navigation channel at time t; 0 otherwise

y_{ik}: =1 if vessel i occupies staging anchorage k; 0 otherwise

y_{i0}: =1 if vessel i travels directly between the end of the channel and berthing position $b(i)$ without occupying any staging anchorage; 0 otherwise

z_{ikt}: =1 if vessel i occupies staging anchorage k at time t; 0 otherwise

e_i: Time point for vessel i to enter a staging anchorage

f_i: Time point for vessel i to leave a staging anchorage

g_i: Berthing time of incoming vessel i

L_{1i}: Berthing tardiness of incoming vessel i

L_{2i}: Departure tardiness of outgoing vessel i

U_i: =1 if the service request of vessel i cannot be satisfied; 0 otherwise.

MILP formulation:

$$\mathbf{P}: \text{ minimize } \sum_{i=1}^{n_1} C_{1i}L_{1i} + \sum_{i=n_1+1}^{n_1+n_2} C_{2i}L_{2i} + \sum_{i=1}^{n_1+n_2} C_{0i}U_i \tag{3.1}$$

$$\text{subject to } \sum_{i=1}^{n_1} x_{it} \leq 1 \quad (t = 0, 1, \ldots, T) \tag{3.2}$$

$$\sum_{i=n_1+1}^{n_1+n_2} x_{it} \leq 1 \quad (t = 0, 1, \ldots, T) \tag{3.3}$$

$$\sum_{t=0}^{T} x_{it} = 1 - U_i \quad (i = 1, \ldots, n_1 + n_2) \tag{3.4}$$

$$x_{it} = 0 \quad (i = 1, \ldots, n_1;\ t = 0, 1, \ldots, A_i - 1) \tag{3.5}$$

$$x_{it} = 0 \quad (i = 1, \ldots, n_1 + n_2;\ t \in \{0, 1, \ldots, T\} \setminus \bigcup_{l=1}^{u_i} [\underline{w}_{il}, \bar{w}_{il} - \bar{\tau}]) \tag{3.6}$$

$$\sum_{k=0}^{m} y_{ik} = 1 - U_i \quad (i = 1, \ldots, n_1 + n_2) \tag{3.7}$$

$$e_i = \sum_{t=0}^{T} (t + \bar{\tau}) x_{it} + \sum_{k=0}^{m} \tau'_{0k} y_{ik} \quad (i = 1, \ldots, n_1) \tag{3.8}$$

$$f_i = g_i - \sum_{k=0}^{m} \tau''_{k,b(i)} y_{ik} \quad (i = 1, \ldots, n_1) \tag{3.9}$$

$$e_i = \sum_{k=0}^{m} (E_i + \tau''_{k,b(i)}) y_{ik} \quad (i = n_1 + 1, \ldots, n_1 + n_2) \tag{3.10}$$

$$f_i = \sum_{t=0}^{T} t x_{it} - \sum_{k=0}^{m} \tau'_{0k} y_{ik} \quad (i = n_1 + 1, \ldots, n_1 + n_2) \tag{3.11}$$

$$f_i - M(1 - y_{i0}) \leq e_i \leq f_i \quad (i = 1, \ldots, n_1 + n_2) \tag{3.12}$$

$$\sum_{i=1}^{n_1+n_2} z_{ikt} \leq 1 \quad (k = 1, \ldots, m;\ t = 0, 1, \ldots, T) \tag{3.13}$$

$$e_i - M(1 - z_{ikt}) \leq t \leq f_i + M(1 - z_{ikt})$$
$$(i = 1, \ldots, n_1 + n_2;\ k = 1, \ldots, m;\ t = 0, 1, \ldots, T) \tag{3.14}$$

$$f_i - e_i + 1 - M(1 - y_{ik}) \leq \sum_{t=0}^{T} z_{ikt} \leq M y_{ik}$$
$$(i = 1, \ldots, n_1 + n_2;\ k = 1, \ldots, m) \tag{3.15}$$

$$B_i(1 - U_i) \leq g_i \leq \bar{B}_i(1 - U_i) \quad (i = 1, \ldots, n_1) \tag{3.16}$$

$$L_{1i} \geq g_i - B_i \quad (i = 1, \ldots, n_1) \tag{3.17}$$

$$L_{2i} \geq \sum_{t=0}^{T} (t + \bar{\tau}) x_{it} - D_i(1 - U_i) \quad (i = n_1 + 1, \ldots, n_1 + n_2) \tag{3.18}$$

$$x_{it} \in \{0, 1\} \quad (i = 1, \ldots, n_1 + n_2;\ t = 0, 1, \ldots, T) \tag{3.19}$$

$$y_{ik} \in \{0, 1\} \quad (i = 1, \ldots, n_1 + n_2;\ k = 0, 1, \ldots, m) \tag{3.20}$$

$$z_{ikt} \in \{0,1\} \quad (i,j = 1,\ldots,n_1+n_2;\ k=1,\ldots,m;\ t=0,1,\ldots,T) \tag{3.21}$$

$$U_i \in \{0,1\} \quad (i=1,\ldots,n_1+n_2) \tag{3.22}$$

$$e_i, f_i \geq 0 \quad (i=1,\ldots,n_1+n_2) \tag{3.23}$$

$$g_i, L_{1i} \geq 0 \quad (i=1,\ldots,n_1) \tag{3.24}$$

$$L_{2i} \geq 0 \quad (i=n_1+1,\ldots,n_1+n_2) \tag{3.25}$$

Objective function (3.1) minimizes the total penalty, which includes the total berthing tardiness penalty of the incoming vessels, the total departure tardiness penalty of the outgoing vessels, and the penalty for unsatisfied service requests of vessels. Constraint (3.2) ensures that at most one incoming vessel can enter the navigation channel from the outer anchorage ground at each time point. Constraint (3.3) ensures that at most one outgoing vessel can enter the channel from the terminal basin at each time point. Constraint (3.4) requires each accepted vessel to enter the channel once. Constraint (3.5) specifies that an incoming vessel i cannot enter the channel before A_i. Constraint (3.6) requires each accepted vessel i to enter the channel during time interval $[\underline{w}_{il}, \bar{w}_{il} - \bar{\tau}]$ for some $l = 1,\ldots,u_i$, so that the vessel can pass through the channel within tidal window $[\underline{w}_{il}, \bar{w}_{il}]$. Constraint (3.7) ensures that each accepted vessel either travels directly between the end of the navigation channel and its designated berthing position, or makes use of exactly one staging anchorage. Constraints (3.8) and (3.9) define the time points when incoming vessel i enters and leaves a staging anchorage (if $y_{i0} = 1$, then e_i and f_i are both set equal to the time point when incoming vessel i arrives at the end of the navigation channel). Similarly, constraints (3.10) and (3.11) define the time points when outgoing vessel i enters and leaves a staging anchorage. Constraint (3.12) specifies a relationship between e_i and f_i. It also implies that $f_i = e_i$ if vessel i does not make use of any staging anchorage. Constraint (3.13) ensures that at most one vessel dwells at staging anchorage k at each time point. We refer to this constraint as the capacity constraint of staging anchorage k. Constraint (3.14) ensures that z_{ikt} can be set equal to 1 only when $t \in [e_i, f_i]$, i.e., vessel i dwells at a staging anchorage only within time window $[e_i, f_i]$. Constraint (3.15) states that if vessel i does not make use of staging anchorage k, then it cannot dwell at staging anchorage k at any time point; otherwise it must occupy staging anchorage k for $f_i - e_i + 1$ time points (i.e., time points $e_i, e_i + 1, \ldots, f_i$). Constraint (3.16) ensures that each incoming vessel i berths within the feasible time interval $[B_i, \bar{B}_i]$ to satisfy its service request. Constraint (3.17) determines the berthing tardiness L_{1i} for each incoming vessel i. Constraint (3.18) determines the departure tardiness L_{2i} for each outgoing vessel i. Finally, constraints (3.19)–(3.25) specify the binary and nonnegativity requirements of the decision variables.

A simple example that illustrates problem **P** is given as follows:

- Length of planning horizon: $T = 12$
- Number of incoming vessels: $n_1 = 2$
- Number of outgoing vessels: $n_2 = 2$
- Number of staging anchorages: $m = 1$
- Number of tidal window for vessel i: $u_i = 1$ for $i = 1, 2, 3, 4$
- Tidal windows for incoming vessels: $[\underline{w}_{11}, \bar{w}_{11}] = [3, 8]$; $[\underline{w}_{21}, \bar{w}_{21}] = [0, 12]$
- Tidal windows for outgoing vessels: $[\underline{w}_{31}, \bar{w}_{31}] = [3, 8]$; $[\underline{w}_{41}, \bar{w}_{41}] = [0, 12]$
- Arrival times of incoming vessels: $A_1 = 2$; $A_2 = 3$
- Berthing time windows of incoming vessels: $[B_1, \bar{B}_1] = [11, 12]$; $[B_2, \bar{B}_2] = [9, 10]$
- Time points that outgoing vessels must leave their berthing positions: $E_3 = 0$; $E_4 = 2$
- Expected departure times of outgoing vessels: $D_3 = 7$; $D_4 = 10$
- Amount of time for a vessel to travel through the navigation channel: $\bar{\tau} = 5$
- Amount of time for vessel i to travel directly between the end of the navigation channel and berthing position $b(i)$: $\tau_{0,b(i)} = 1$ for $i = 1, 2, 3, 4$
- Amount of time for a vessel to travel between the end of the navigation channel and the staging anchorage: $\tau'_{01} = 1$
- Amount of time for vessel i to travel between the staging anchorage and berthing position $b(i)$: $\tau''_{1,b(i)} = 1$ for $i = 1, 2, 3, 4$
- Unit costs of berthing tardiness for incoming vessels: $C_{11} = 2$; $C_{12} = 3$
- Unit costs of departure tardiness for outgoing vessels: $C_{23} = 2$; $C_{24} = 3$
- Lump-sum cost if the service request of vessel i cannot be satisfied: $C_{0i} = 100$ for $i = 1, 2, 3, 4$

In this example, there is only one staging anchorage, and each vessel has only one tidal window.

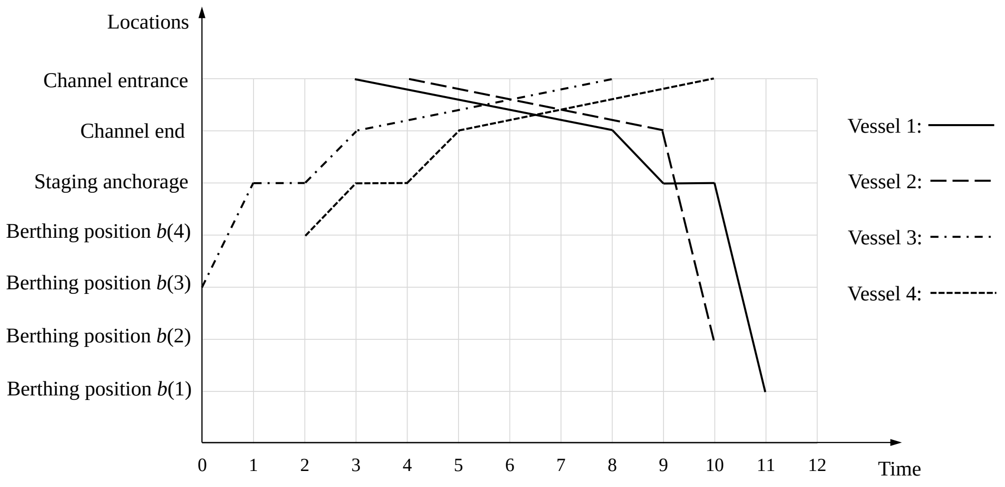

Figure 3.3: Routes of vessels in a feasible solution of the example.

A solution of this example is depicted in Figure 3.3. In this solution, incoming vessel 1 travels through the navigation channel during the time interval $[3, 8]$, travels to the staging anchorage, and then travels to berthing position $b(1)$. Vessel 1 arrives at the staging anchorage at $e_1 = 8 + \tau'_{01} = 9$ and leaves the staging anchorage at $f_1 = 10$. Thus, the berthing tardiness of vessel 1 is $L_{11} = f_1 + \tau''_{1,b(1)} - B_1 = 0$, and the tardiness cost of the vessel is $C_{11}L_{11} = 0$. Incoming vessel 2 travels through the navigation channel during the time interval $[4, 9]$ and then travels directly to berthing position $b(2)$ without making use of the staging anchorage. The berthing tardiness of vessel 2 is $L_{12} = 9 + \tau_{0,b(2)} - B_2 = 1$, and the tardiness cost of the vessel is $C_{12}L_{12} = 3$. Outgoing vessel 3 leaves berthing position $b(3)$ at time $E_3 = 0$, travels to the staging anchorage, and then travels to the navigation channel. Vessel 3 arrives at the staging anchorage at $e_3 = 0 + \tau''_{1,b(3)} = 1$, and leaves the staging anchorage at $f_3 = 2$. Thus, the departure tardiness of vessel 3 is $L_{23} = \max\{0,\ f_3 + \tau'_{01} + \bar{\tau} - D_3\} = 1$, and the tardiness cost of the vessel is $C_{23}L_{23} = 2$. Outgoing vessel 4 leaves berthing position $b(4)$ at time $E_4 = 2$, travels to the staging anchorage, and then travels to the navigation channel. Vessel 4 arrives at the staging anchorage at $e_4 = 2 + \tau''_{1,b(4)} = 3$, and leaves the staging anchorage at $f_4 = 4$. Thus, the departure tardiness of vessel 4 is $L_{24} = \max\{0,\ f_4 + \tau'_{01} + \bar{\tau} - D_4\} = 0$, and the tardiness cost of the vessel is $C_{24}L_{24} = 0$. Since the service requests of all vessels are satisfied, the total cost of the solution is $C_{11}L_{11} + C_{22}L_{22} + C_{23}L_{23} + C_{24}L_{24} = 5$. Note that in this solution, (i) each outgoing vessel i unberths at E_i; (ii) each incoming vessel i berths within its berthing time window $[B_i, \bar{B}_i]$; (iii) each incoming vessel enters the navigation channel no earlier than its arrival time; (iv) each vessel passes through the channel during its tidal window; (v) at any time point, there is at most one incoming vessel entering the channel and at most one outgoing vessel entering the channel; and (vi) at any time point, there is at most one vessel occupying the staging anchorage. Hence, this solution is feasible.

The following theorem states the computational complexity of problem **P**.

Theorem 1 *Problem* **P** *is NP-hard in the strong sense.*

Proof: We transform 3-Dimensional Matching (3DM) to the decision version of problem **P**. Given disjoint sets $X_\alpha = \{\alpha_1, \dots, \alpha_r\}$, $X_\beta = \{\beta_1, \dots, \beta_r\}$, and $X_\gamma = \{\gamma_1, \dots, \gamma_r\}$, and a set $S \subseteq X_\alpha \times X_\beta \times X_\gamma$, the 3DM problem asks whether there exists a subset $S' \subseteq S$ such that $|S'| = r$ and no two elements of S' agree in any coordinate. 3DM is known to be strongly NP-hard (Garey and Johnson 1979). For notational convenience, we denote $S = \{\phi_1, \dots, \phi_s\}$ and $\phi_k = (\alpha_{\mu(k,1)}, \beta_{\mu(k,2)}, \gamma_{\mu(k,3)})$ for $k = 1, \dots, s$. We assume that $s \geq r$ (as the case where $s < r$ is trivial).

Given an arbitrary instance of 3DM, we construct a corresponding instance of problem **P** as follows. There are $s + 2r$ incoming vessels, 0 outgoing vessels, and s staging anchorages. All vessels have the same tidal windows. Among the incoming vessels, vessels $1, \dots, r$ correspond to the elements in X_α; vessels $r + 1, \dots, 2r$ correspond to the elements in X_β; vessels $2r + 1, \dots, 3r$ correspond to the elements in X_γ; and vessels $3r + 1, \dots, s + 2r$ do not correspond to any element of $X_\alpha \cup X_\beta \cup X_\gamma$. Specifically, we let

$$T = s + 8r;$$

$$n_1 = s + 2r;$$

$$n_2 = 0;$$

$$m = s;$$

$$u_i = 4, \text{ for } i = 1, \dots, s + 2r;$$

$$[\underline{w}_{i1}, \bar{w}_{i1}] = [0, s - r], \text{ for } i = 1, \dots, s + 2r;$$

$$[\underline{w}_{i2}, \bar{w}_{i2}] = [s, s + r], \text{ for } i = 1, \dots, s + 2r;$$

$$[\underline{w}_{i3}, \bar{w}_{i3}] = [s + 2r, s + 3r], \text{ for } i = 1, \dots, s + 2r;$$

$$[\underline{w}_{i4}, \bar{w}_{i4}] = [s+4r, s+5r], \text{ for } i = 1, \dots, s+2r;$$

$$A_i = \begin{cases} s, & \text{for } i = 1, \dots, r; \\ s+2r, & \text{for } i = r+1, \dots, 2r; \\ s+4r, & \text{for } i = 2r+1, \dots, 3r; \\ 0, & \text{for } i = 3r+1, \dots, s+2r; \end{cases}$$

$$B_i = \bar{B}_i = \begin{cases} s+3r, & \text{for } i = 1, \dots, r; \\ s+5r, & \text{for } i = r+1, \dots, 2r; \\ s+7r, & \text{for } i = 2r+1, \dots, 3r; \\ s+8r, & \text{for } i = 3r+1, \dots, s+2r; \end{cases}$$

$$C_{1i} = C_{0i} = 1, \text{ for } i = 1, \dots, s+2r;$$

$$\bar{\tau} = 1;$$

$$\tau_{0,b(i)} = \tau'_{0k} = r, \text{ for } i = 1, \dots, s+2r \text{ and } k = 1, \dots, s;$$

$$\tau''_{k,b(i)} = \begin{cases} r, & \text{if } (1 \le i \le r \text{ and } i = \mu(k,1)) \\ & \quad \text{or } (r+1 \le i \le 2r \text{ and } i - r = \mu(k,2)) \\ & \quad \text{or } (2r+1 \le i \le 3r \text{ and } i - 2r = \mu(k,3)); \\ 2r, & \text{otherwise,} \end{cases}$$

for $i = 1, \dots, s+2r$ and $k = 1, \dots, s$.

Clearly, this transformation is pseudo-polynomial. We will show that there exists a feasible solution to the constructed instance of problem **P** with a zero total cost if and only if the answer to the 3DM problem is "yes." Suppose the answer to the 3DM problem is "yes." Denote $S' = \{\phi_{\pi(1)}, \dots, \phi_{\pi(r)}\}$, where $\pi(1), \dots, \pi(r) \in \{1, \dots, s\}$. Then, consider the following solution to the constructed instance of problem **P**: For $i = 1, \dots, r$, vessel i enters the navigation channel at time $s + i - 1$; vessel $r + i$ enters the navigation channel at time $s + 2r + i - 1$; and vessel $2r + i$ enters the navigation channel at time $s + 4r + i - 1$. For $i = 3r + 1, \dots, s + 2r$, vessel i enters the navigation channel at time $i - 3r - 1$. In other words, vessels $1, \dots, r$ pass through the channel one by one during the tidal window $[s, s + r]$; vessels $r + 1, \dots, 2r$ pass through the channel one by one during the tidal window $[s + 2r, s + 3r]$; vessels $2r + 1, \dots, 3r$ pass through the channel one by one during the tidal window $[s + 4r, s + 5r]$; and vessels $3r + 1, \dots, s + 2r$ pass through the channel one by one during the tidal window $[0, s-r]$. Each of these vessels occupies a staging anchorage before traveling to its designated berthing position. For $i = 1, \ldots, r$, vessels $\mu(\pi(i), 1)$, $r + \mu(\pi(i), 2)$, and $2r + \mu(\pi(i), 3)$ occupy staging anchorage $\pi(i)$ until r time units before their planned berthing time and then travel to their designated berthing positions. Vessels $3r+1, \ldots, s+2r$ are arbitrarily assigned to the other $s-r$ staging anchorages. These $s-r$ vessels will stay at their staging anchorages until time $s+6r$ and then travel to their designated berthing positions. Note that for all $i = 1, \ldots, s+2r$, vessel i arrives at $b(i)$ at exactly its planned berthing time B_i.

Since no two elements of S' agree in any coordinate, $\mu(\pi(1), 1), \ldots, \mu(\pi(r), 1)$ are distinct elements of $\{1, \ldots, r\}$. Similarly, $\mu(\pi(1), 2), \ldots, \mu(\pi(r), 2)$ are distinct elements of $\{1, \ldots, r\}$, and $\mu(\pi(1), 3), \ldots, \mu(\pi(r), 3)$ are distinct elements of $\{1, \ldots, r\}$. Hence, vessels $1, \ldots, r$ occupy different staging anchorages; vessels $r+1, \ldots, 2r$ occupy different staging anchorages; and vessels $2r+1, \ldots, 3r$ occupy different staging anchorages. Vessels $1, \ldots, r$ occupy the staging anchorages no later than time $s+2r$. Vessels $r+1, \ldots, 2r$ occupy the staging anchorages no earlier than time $s+3r+1$ and no later than time $s+4r$. Vessels $2r+1, \ldots, 3r$ occupy the staging anchorages no earlier than time $s+5r+1$. Thus, at any point in time, each staging anchorage is occupied by no more than one vessel. Hence, this solution is feasible. Since all service requests of vessels are satisfied and all vessels arrive on time at their designated berthing positions, this solution has a zero total cost.

Conversely, suppose that there exists a feasible solution to the constructed instance of problem **P** with a zero total cost. In this solution, for all $i = 1, \ldots, s+2r$, vessel i must berth at time B_i. Since $A_{2r+1} = \cdots = A_{3r} = s+4r$, vessels $2r+1, \ldots, 3r$ must pass through the navigation channel during the tidal window $[s+4r, s+5r]$ in order to arrive at their berthing positions on time. Since at most r incoming vessels can pass through the tidal window $[s+4r, s+5r]$, this tidal window is fully utilized by vessels $2r+1, \ldots, 3r$. Since $A_{r+1} = \cdots = A_{2r} = s+2r$, vessels $r+1, \ldots, 2r$ must pass through the navigation channel during the tidal window $[s+2r, s+3r]$ in order to arrive at their berthing positions on time. Since at most r incoming vessels can pass through the tidal window $[s+2r, s+3r]$, this tidal window is fully utilized by vessels $r+1, \ldots, 2r$. Since $A_1 = \cdots = A_r = s$, vessels $1, \ldots, r$ must pass through the navigation channel during the tidal window $[s, s+r]$ in order to arrive at their berthing positions on time. Since at most r incoming vessels can pass through the tidal window $[s, s+r]$, this tidal window is fully utilized by vessels $1, \ldots, r$. Now, the only tidal window left for vessels $3r+1, \ldots, 2r+s$ is $[0, s-r]$. Thus, vessels $3r+1, \ldots, s+2r$ must pass through the channel during this tidal window.

For $i = 1, \ldots, r$, vessel i reaches the end of the navigation channel at a time point within the time interval $[s+1, s+r]$; vessel $r+i$ reaches the end of the navigation channel at a time point within the time interval $[s+2r+1, s+3r]$; and vessel $2r+i$ reaches the end of the navigation channel at a time point within the time interval $[s+4r+1, s+5r]$. Thus, these $3r$ vessels reach the end of the channel at least $2r$ time units before their planned berthing time. Hence, each of them must visit a staging anchorage before traveling to its designated berthing position. Clearly, each of vessels $3r+1, \ldots, 2r+s$ must also visit a staging anchorage before traveling to its designated berthing position. For $i = 1, \ldots, 2r+s$, we let k_i denote the staging anchorage which vessel i occupies.

For $i = 1, \ldots, r$, vessel i reaches a staging anchorage at a time point within the time interval $[s+r+1, s+2r]$, which is less than $2r$ time units before B_i. Vessel $r+i$ reaches a staging anchorage at a time point within the time interval $[s+3r+1, s+4r]$, which is less than $2r$ time units before B_i. Vessel $2r+i$ reaches a staging anchorage at a time point within the time interval $[s+5r+1, s+6r]$, which is less than $2r$ time units before B_i. Note that $\tau''_{k,b(i)}$ equals either r or $2r$ for any i and k. Therefore, vessels $1, \ldots, 3r$ must leave their staging anchorages exactly r time units before their planned berthing time in order to avoid berthing tardiness penalty. Thus, the solution of the constructed instance has the following properties: (i) Vessels $1, \ldots, r$ and $3r+1, \ldots, 2r+s$ are occupying s different staging anchorages at time $s+2r$; vessels $r+1, \ldots, 2r$ and $3r+1, \ldots, 2r+s$ are occupying s different staging anchorages at time $s+4r$; and vessels $2r+1, \ldots, 3r$ and $3r+1, \ldots, 2r+s$ are occupying s different staging anchorages at time $s+6r$. (ii) $\mu(k_i, 1) = i$ for $i = 1, \ldots, r$; $\mu(k_i, 2) = i - r$ for $i = r+1, \ldots, 2r$; and $\mu(k_i, 3) = i - 2r$ for $i = 2r+1, \ldots, 3r$ (i.e., for $1, \ldots, 3r$, vessel i occupies staging anchorage k_i such that $\tau''_{k_i,b(i)} = r$).

Property (i) implies that vessels $1, \ldots, r$ occupy different staging anchorages, i.e., $k_1, \ldots, k_r$ are distinct values; vessels $r+1, \ldots, 2r$ occupy different staging anchorages, i.e., $k_{r+1}, \ldots, k_{2r}$ are distinct values; and vessels $2r+1, \ldots, 3r$ occupy different staging anchorages, i.e., $k_{2r+1}, \ldots, k_{3r}$ are distinct values. This, together with property (ii), implies that $\mu(k_1, 1), \ldots, \mu(k_r, 1)$ are distinct values, $\mu(k_{r+1}, 2), \ldots, \mu(k_{2r}, 2)$ are distinct values, and $\mu(k_{2r+1}, 3), \ldots, \mu(k_{3r}, 3)$ are distinct values. Property (i) also implies that

$$\{k_1, \ldots, k_r\} = \{k_{r+1}, \ldots, k_{2r}\} = \{k_{2r+1}, \ldots, k_{3r}\};$$

i.e., the r staging anchorages occupied by vessels $1, \ldots, r$ are the same staging anchorages occupied by vessels $r+1, \ldots, 2r$, which are the same staging anchorages occupied by vessels $2r+1, \ldots, 3r$ (while the other $s-r$ staging anchorages are occupied by vessels $3r+1, \ldots, 2r+s$). Hence, $\mu(k_1, 1), \ldots, \mu(k_r, 1)$ are distinct values; $\mu(k_1, 2), \ldots, \mu(k_r, 2)$ are distinct values; and $\mu(k_1, 3), \ldots,$ $\mu(k_r, 3)$ are distinct values.

Let $S' = \{\phi_{k_1}, \ldots, \phi_{k_r}\}$. Then, $|S'| = r$, and no two elements of S' agree in any coordinate. This completes the proof of the theorem. ▮

Remark 1 *In the proof of Theorem 1, we have shown that the answer to the 3-Dimensional Matching problem is "yes" if and only if the constructed instance of problem* **P** *has a zero total cost. Thus, unless $P = NP$, there exists no approximation algorithm for problem* **P** *with a constant bound on the relative error.*

Remark 2 *In the proof of Theorem 1, the constructed instance of problem* **P** *has no outgoing vessel, and all travel times satisfy the triangle inequality. Thus, problem* **P** *is NP-hard in the strong sense even when there are only incoming vessels and all travel times satisfy the triangle inequality.*

3.2 Solution Method

In this section, we present a Lagrangian relaxation heuristic for problem **P**. We first present the Lagrangian dual problem, and show that the subproblems obtained by relaxing the capacity constraints of the staging anchorages can be solved in pseudo-polynomial time. We then present a heuristic for constructing a feasible primal solution from the solution to the Lagrangian dual problem, as well as the subgradient optimization procedure for searching the Lagrangian multipliers.

3.2.1 The Lagrangian Relaxation Problem and Its Subproblems

Relaxing the staging anchorage capacity constraint (3.13) and placing it in the objective function of problem $\mathbf{P}$ with Lagrangian multipliers $\lambda_{kt} \geq 0$ ($k = 1, \ldots, m$; $t = 0, 1, \ldots, T$), we obtain the following relaxed problem:

$$\begin{array}{lll} \mathbf{P}(\lambda): & \text{minimize} & \sum_{i=1}^{n_1} C_{1i}L_{1i} + \sum_{i=n_1+1}^{n_1+n_2} C_{2i}L_{2i} + \sum_{i=1}^{n_1+n_2} C_{0i}U_i + \sum_{k=1}^{m} \sum_{t=0}^{T} \lambda_{kt}(\sum_{i=1}^{n_1+n_2} z_{ikt} - 1) \\ & \text{subject to} & (3.2)\text{–}(3.12) \text{ and } (3.14)\text{–}(3.25) \end{array}$$

where λ denotes the vector of the λ_{kt} values. Let $L(\lambda)$ denote the optimal objective value of problem $\mathbf{P}(\lambda)$. The goal of the Lagrangian relaxation heuristic is to solve the following Lagrangian dual problem:

$$\mathbf{P}_{\text{LD}}: \quad \max_{\lambda} L(\lambda).$$

We search the optimal Lagrangian multipliers by applying the subgradient optimization method, which will be described in the next section. In the following, we will show that given the values of the Lagrangian multipliers, problem $\mathbf{P}(\lambda)$ can be solved in pseudo-polynomial time.

After dropping the constant term $-\sum_{k=1}^{m} \sum_{t=0}^{T} \lambda_{kt}$ from the objective function, problem $\mathbf{P}(\lambda)$ can be decomposed into the following two independent subproblems:

$$\begin{array}{lll} \mathbf{P}_{\text{in}}(\lambda): & \text{minimize} & \sum_{i=1}^{n_1} \left\{C_{1i}L_{1i} + C_{0i}U_i + \sum_{k=1}^{m} \sum_{t=0}^{T} \lambda_{kt}z_{ikt}\right\} \\ & \text{subject to} & (3.2), (3.4)\text{–}(3.9), (3.12), (3.14)\text{–}(3.17), \text{ and } (3.19)\text{–}(3.24) \\ & & (\text{for } i = 1, \ldots, n_1 \text{ only}) \end{array}$$

and

$$\begin{array}{lll} \mathbf{P}_{\text{out}}(\lambda): & \text{minimize} & \sum_{i=n_1+1}^{n_1+n_2} \left\{C_{2i}L_{2i} + C_{0i}U_i + \sum_{k=1}^{m} \sum_{t=0}^{T} \lambda_{kt}z_{ikt}\right\} \\ & \text{subject to} & (3.3)\text{–}(3.4), (3.6)\text{–}(3.7), (3.10)\text{–}(3.12), (3.14)\text{–}(3.15), (3.18)\text{–}(3.23), \text{ and } (3.25) \\ & & (\text{for } i = n_1+1, \ldots, n_1+n_2 \text{ only}) \end{array}$$

Subproblem $\mathbf{P}_{\text{in}}(\lambda)$ involves only decision variables of the incoming vessels, while subproblem $\mathbf{P}_{\text{out}}(\lambda)$ involves only decision variables of the outgoing vessels. We will show that these subproblems can both be transformed into asymmetric assignment problems.

We first consider subproblem $\mathbf{P}_{\text{in}}(\lambda)$. For ease of presentation, we introduce a dummy time point $T+i$ and denote $x_{i,T+i} = U_i$ for each incoming vessel $i = 1, \ldots, n_1$. For $i = 1, \ldots, n_1$ and $t = 0, 1, \ldots, T + n_1$, define

$$F_{it}(\lambda) = \begin{cases} \min_{k=0,1,\ldots,m} \left\{F_{it}^{(k)}(\lambda)\right\}, & \text{if } A_i \le t \le T \text{ and } t \in \bigcup_{l=1}^{u_i} [\underline{w}_{il}, \bar{w}_{il} - \bar{\tau}]; \\ C_{0i}, & \text{if } t = T + i; \\ +\infty, & \text{otherwise;} \end{cases}$$

where

$$F_{it}^{(k)}(\lambda) = \begin{cases} C_{1i}(t + \bar{\tau} + \tau_{0,b(i)} - B_i), & \text{if } k = 0 \text{ and } B_i \le t + \bar{\tau} + \tau_{0,b(i)} \le \bar{B}_i; \\ C_{1i}(t + \bar{\tau} + \tau'_{0k} + \tau''_{k,b(i)} - B_i) + \lambda_{k,t+\bar{\tau}+\tau'_{0k}}, & \text{if } k > 0 \text{ and } B_i \le t + \bar{\tau} + \tau'_{0k} + \tau''_{k,b(i)} \le \bar{B}_i; \\ \sum_{t'=t+\bar{\tau}+\tau'_{0k}}^{B_i - \tau''_{k,b(i)}} \lambda_{kt'}, & \text{if } k > 0 \text{ and } t + \bar{\tau} + \tau'_{0k} + \tau''_{k,b(i)} < B_i; \\ +\infty, & \text{otherwise.} \end{cases}$$

Here, $F_{it}(\lambda)$ represents the cost of letting incoming vessel i enter the navigation channel at time t in the Lagrangian subproblem, $F_{it}^{(k)}(\lambda)$ represents this cost when the vessel is assigned staging anchorage k (if $k > 0$), and $F_{it}^{(0)}(\lambda)$ represents this cost when the vessel is not assigned any staging anchorage. Define

$$\begin{aligned} \mathbf{P}'_{\text{in}}(\lambda): \text{ minimize } \quad & \textstyle\sum_{i=1}^{n_1} \sum_{t=0}^{T+n_1} F_{it}(\lambda) x_{it} \\ \text{subject to } \quad & \textstyle\sum_{i=1}^{n_1} x_{it} \le 1 \quad (t = 0, 1, \ldots, T + n_1) \\ & \textstyle\sum_{t=0}^{T+n_1} x_{it} = 1 \quad (i = 1, \ldots, n_1) \\ & x_{it} \in \{0, 1\} \quad (i = 1, \ldots, n_1;\ t = 0, 1, \ldots, T + n_1) \end{aligned}$$

Lemma 1 *Solving problem $\mathbf{P}'_{\text{in}}(\lambda)$ yields the optimal objective value of problem $\mathbf{P}_{\text{in}}(\lambda)$.*

Proof: We divide the proof into two parts. In the first part, we show that the optimal objective value of $\mathbf{P}'_{\text{in}}(\lambda)$ is less than or equal to the optimal objective value of $\mathbf{P}_{\text{in}}(\lambda)$. Let $\{x^*_{it}, y^*_{ik}, z^*_{ikt}, e^*_i, f^*_i, g^*_i, L^*_{1i}, U^*_i\}$ be an optimal solution of $\mathbf{P}_{\text{in}}(\lambda)$. We construct a solution $\{x^{**}_{it}\}$ of problem $\mathbf{P}'_{\text{in}}(\lambda)$ by setting

$$x^{**}_{it} = \begin{cases} x^*_{it}, & \text{if } t \le T; \\ U^*_i, & \text{if } t = T + i; \\ 0, & \text{otherwise;} \end{cases}$$

for $i = 1, \ldots, n_1$ and $t = 0, 1, \ldots, T + n_1$. Constraint (3.2) implies that $\sum_{i=1}^{n_1} x_{it}^* \leq 1$ for $t = 0, 1, \ldots, T$. Thus, for $t = 0, 1, \ldots, T$,

$$\sum_{i=1}^{n_1} x_{it}^{**} = \sum_{i=1}^{n_1} x_{it}^* \leq 1.$$

For $t = T + 1, \ldots, T + n_1$,

$$\sum_{i=1}^{n_1} x_{it}^{**} = x_{t-T,t}^{**} = U_{t-T}^* \leq 1.$$

Constraint (3.4) implies that $\sum_{t=0}^{T} x_{it}^* = 1 - U_i^*$ for $i = 1, \ldots, n_1$. Thus, for $i = 1, \ldots, n_1$,

$$\sum_{t=0}^{T+n_1} x_{it}^{**} = \sum_{t=0}^{T} x_{it}^{**} + \sum_{t=T+1}^{T+n_1} x_{it}^{**} = \sum_{t=0}^{T} x_{it}^* + U_i^* = 1.$$

Hence, $\{x_{it}^{**}\}$ is a feasible solution of $\mathbf{P}'_{\text{in}}(\lambda)$.

We now show that the objective value of this feasible solution is less than or equal to $\sum_{i=1}^{n_1} \{C_{1i} L_{1i}^* + C_{0i} U_i^* + \sum_{k=1}^{m} \sum_{t=0}^{T} \lambda_{kt} z_{ikt}^*\}$, which is the optimal objective value of $\mathbf{P}_{\text{in}}(\lambda)$. To do so, we show that for $i = 1, \ldots, n_1$, the inequality

$$\sum_{t=0}^{T+n_1} F_{it}(\lambda) x_{it}^{**} \leq C_{1i} L_{1i}^* + C_{0i} U_i^* + \sum_{k=1}^{m} \sum_{t=0}^{T} \lambda_{kt} z_{ikt}^* \tag{3.26}$$

holds. We divide the analysis into two cases.

Case 1: $U_i^* = 1$. In this case, from (3.4), we have $x_{i0}^* = x_{i1}^* = \cdots = x_{iT}^* = 0$. Thus, $x_{i,T+i}^{**} = 1$ and $x_{it}^{**} = 0$ if $t \neq T + i$. We have

$$\sum_{t=0}^{T+n_1} F_{it}(\lambda) x_{it}^{**} = F_{i,T+i}(\lambda) = C_{0i} = C_{0i} U_i^*,$$

and hence inequality (3.26) holds.

Case 2: $U_i^* = 0$. In this case, by (3.4) and (3.19), exactly one of $x_{i0}^*, x_{i1}^*, \ldots, x_{iT}^*$ is equal to 1. Let t_i^* be the value of t such that $x_{it}^* = 1$. By (3.7) and (3.20), exactly one of $y_{i0}^*, y_{i1}^*, \ldots, y_{im}^*$ is equal to 1. Let k_i^* be the value of k such that $y_{ik}^* = 1$. Then, by (3.8),

$$e_i^* = t_i^* + \bar{\tau} + \tau'_{0k_i^*}. \tag{3.27}$$

By (3.9),

$$g_i^* = f_i^* + \tau''_{k_i^*, b(i)}. \tag{3.28}$$

By (3.14) and (3.15),

$$z_{ikt}^* = \begin{cases} 1, & \text{if } k = k_i^* \text{ and } e_i^* \le t \le f_i^*; \\ 0, & \text{otherwise;} \end{cases}$$

which implies that

$$\sum_{k=1}^{m}\sum_{t=0}^{T} \lambda_{kt} z_{ikt}^* = \sum_{t=0}^{T} \lambda_{k_i^* t} z_{ik_i^* t}^* = \sum_{t=e_i^*}^{f_i^*} \lambda_{k_i^* t}. \tag{3.29}$$

Note that in this case, $x_{i,T+1}^{**} = \cdots = x_{i,T+n_1}^{**} = 0$. Furthermore, by (3.5) and (3.6), $t_i^* \in \{A_i, \ldots, T\} \cap \bigcup_{l=1}^{u_i} [\underline{w}_{il}, \bar{w}_{il} - \bar{\tau}]$. Hence,

$$\sum_{t=0}^{T+n_1} F_{it}(\lambda) x_{it}^{**} = \sum_{t=0}^{T} F_{it}(\lambda) x_{it}^{**} = \sum_{t=0}^{T} F_{it}(\lambda) x_{it}^{*} = F_{it_i^*}(\lambda) = \min_{k=0,1,\ldots,m} \{F_{it_i^*}^{(k)}(\lambda)\} \le F_{it_i^*}^{(k_i^*)}(\lambda). \tag{3.30}$$

From (3.12), (3.16), and (3.17), we have

$$e_i^* \le f_i^*, \tag{3.31}$$

$$B_i \le g_i^* \le \bar{B}_i, \tag{3.32}$$

and

$$g_i^* - B_i \le L_{1i}^*. \tag{3.33}$$

From (3.27), (3.28), (3.31), and (3.32), we have

$$t_i^* + \bar{\tau} + \tau'_{0k_i^*} + \tau''_{k_i^*, b(i)} \le \bar{B}_i. \tag{3.34}$$

We further divide this case into three subcases.

Case 2.1: $k_i^* = 0$. In this case, by (3.12), $e_i^* = f_i^*$. This, together with (3.27) and (3.28), implies that $g_i^* = t_i^* + \bar{\tau} + \tau'_{00} + \tau''_{0,b(i)}$, or equivalently,

$$g_i^* = t_i^* + \bar{\tau} + \tau_{0,b(i)}. \tag{3.35}$$

From (3.32) and (3.35), we have $B_i \le t_i^* + \bar{\tau} + \tau_{0,b(i)} \le \bar{B}_i$. Thus, by (3.30), (3.33)–(3.35), and the definition of $F_{it}^{(k)}(\lambda)$,

$$\sum_{t=0}^{T+n_1} F_{it}(\lambda) x_{it}^{**} \le F_{it_i^*}^{(0)}(\lambda) = C_{1i}(t_i^* + \bar{\tau} + \tau_{0,b(i)} - B_i) = C_{1i}(g_i^* - B_i) \le C_{1i} L_{1i}^*,$$

and hence inequality (3.26) holds.

Case 2.2: $k_i^* > 0$ and $B_i \le t_i^* + \bar{\tau} + \tau'_{0k_i^*} + \tau''_{k_i^*,b(i)}$. From (3.34), we have $B_i \le t_i^* + \bar{\tau} + \tau'_{0k_i^*} + \tau''_{k_i^*,b(i)} \le \bar{B}_i$. Thus, by (3.27)–(3.31), (3.33), and the definition of $F_{it}^{(k)}(\lambda)$,

$$\begin{aligned}
\sum_{t=0}^{T+n_1} F_{it}(\lambda) x_{it}^{**} &\le F_{it_i^*}^{(k_i^*)}(\lambda) = C_{1i}(t_i^* + \bar{\tau} + \tau'_{0k_i^*} + \tau''_{k_i^*,b(i)} - B_i) + \lambda_{k_i^*, t_i^* + \bar{\tau} + \tau'_{0k_i^*}} \\
&= C_{1i}(e_i^* + \tau''_{k_i^*,b(i)} - B_i) + \lambda_{k_i^* e_i^*} \le C_{1i}(f_i^* + \tau''_{k_i^*,b(i)} - B_i) + \lambda_{k_i^* e_i^*} \\
&= C_{1i}(g_i^* - B_i) + \lambda_{k_i^* e_i^*} \le C_{1i} L_{1i}^* + \lambda_{k_i^* e_i^*} \le C_{1i} L_{1i}^* + \sum_{t=e_i^*}^{f_i^*} \lambda_{k_i^* t} = C_{1i} L_{1i}^* + \sum_{k=1}^{m} \sum_{t=0}^{T} \lambda_{kt} z_{ikt}^*,
\end{aligned}$$

and hence inequality (3.26) holds.

Case 2.3: $k_i^* > 0$ and $t_i^* + \bar{\tau} + \tau'_{0k_i^*} + \tau''_{k_i^*,b(i)} < B_i$. In this case, by (3.27)–(3.30), (3.32), and the definition of $F_{it}^{(k)}(\lambda)$,

$$\sum_{t=0}^{T+n_1} F_{it}(\lambda) x_{it}^{**} \le F_{it_i^*}^{(k_i^*)}(\lambda) = \sum_{t=t_i^* + \bar{\tau} + \tau'_{0k_i^*}}^{B_i - \tau''_{k_i^*,b(i)}} \lambda_{k_i^* t} = \sum_{t=e_i^*}^{B_i - g_i^* + f_i^*} \lambda_{k_i^* t} \le \sum_{t=e_i^*}^{f_i^*} \lambda_{k_i^* t} = \sum_{k=1}^{m} \sum_{t=0}^{T} \lambda_{kt} z_{ikt}^*,$$

and hence inequality (3.26) holds.

Summarizing the above cases, we conclude that the optimal objective value of $\mathbf{P}'_{\text{in}}(\lambda)$ is no greater than the optimal objective value of $\mathbf{P}_{\text{in}}(\lambda)$. This completes the first part of the proof.

In the second part, we show that the optimal objective value of $\mathbf{P}_{\text{in}}(\lambda)$ is less than or equal to the optimal objective value of $\mathbf{P}'_{\text{in}}(\lambda)$. Let $\{x_{it}^{**}\}$ be an optimal solution of problem $\mathbf{P}'_{\text{in}}(\lambda)$ with a finite objective value. Note that for $i = 1, \ldots, n_1$, either $x_{i,T+i}^{**} = 1$ or exactly one of $x_{i0}^{**}, x_{i1}^{**}, \ldots, x_{iT}^{**}$ is equal to 1. For each $i = 1, \ldots, n_1$, we let t_i^* be the value of t such that $x_{it}^{**} = 1$, and let $k_i^* = \arg\min_{k=0,1,\ldots,m} \{F_{it_i^*}^{(k)}(\lambda)\}$, with ties broken arbitrarily. We construct a feasible solution $\{x_{it}^*, y_{ik}^*, z_{ikt}^*, e_i^*, f_i^*, g_i^*, L_{1i}^*, U_i^*\}$ of problem $\mathbf{P}_{\text{in}}(\lambda)$ as follows:

$$x_{it}^* = x_{it}^{**} \quad (i = 1, \ldots, n_1;\ t = 0, 1, \ldots, T);$$

$$U_i^* = x_{i,T+i}^{**} \quad (i = 1, \ldots, n_1);$$

$$y_{ik}^* = \begin{cases} 1, & \text{if } k = k_i^* \text{ and } U_i^* = 0, \\ 0, & \text{otherwise,} \end{cases} \qquad (i = 1, \ldots, n_1;\ k = 0, 1, \ldots, m);$$

$$e_i^* = \begin{cases} t_i^* + \bar{\tau} + \tau'_{0k_i^*}, & \text{if } U_i^* = 0, \\ 0, & \text{otherwise,} \end{cases} \qquad (i = 1, \ldots, n_1);$$

$$g_i^* = \begin{cases} \max\{t_i^* + \bar{\tau} + \tau'_{0k_i^*} + \tau''_{k_i^*,b(i)}, B_i\}, & \text{if } U_i^* = 0 \text{ and } k_i^* > 0, \\ t_i^* + \bar{\tau} + \tau_{0,b(i)}, & \text{if } U_i^* = 0 \text{ and } k_i^* = 0, \\ 0, & \text{otherwise,} \end{cases} \quad (i = 1, \ldots, n_1);$$

$$f_i^* = \begin{cases} g_i^* - \tau''_{k_i^*,b(i)}, & \text{if } U_i^* = 0, \\ g_i^*, & \text{otherwise,} \end{cases} \quad (i = 1, \ldots, n_1);$$

$$z_{ikt}^* = \begin{cases} 1, & \text{if } k = k_i^*,\ U_i^* = 0, \text{ and } e_i^* \le t \le f_i^*, \\ 0, & \text{otherwise,} \end{cases} \quad (i = 1, \ldots, n_1;\ k = 1, \ldots, m;\ t = 0, 1, \ldots, T);$$

$$L_{1i}^* = \max\{g_i^* - B_i,\ 0\} \quad (i = 1, \ldots, n_1).$$

It is easy to see that this solution satisfies constraints (3.2), (3.4), (3.7)–(3.9), (3.14), (3.15), (3.17), and (3.19)–(3.24) (for $i = 1, \ldots, n_1$). Since the objective value of solution $\{x_{it}^{**}\}$ is finite, $F_{it_i^*}(\lambda)$ is finite for all $i = 1, \ldots, n_1$. Thus, for $i = 1, \ldots, n_1$, either $t_i^* = T + i$ or $t_i^* \in \{A_i, \ldots, T\} \cap \bigcup_{l=1}^{u_i} [\underline{w}_{il}, \bar{w}_{il} - \bar{\tau}]$, which implies that $x_{it}^* = 0$ if $t \notin \{A_i, \ldots, T\} \cap \bigcup_{l=1}^{u_i} [\underline{w}_{il}, \bar{w}_{il} - \bar{\tau}]$. Hence, x_{it}^* satisfies constraints (3.5) and (3.6) (for $i = 1, \ldots, n_1$). For $i = 1, \ldots, n_1$, if $U_i^* = 1$, then $e_i^* = f_i^* = y_{i0}^* = 0$, and thus e_i^*, f_i^*, and y_{i0}^* satisfy (3.12). If $U_i^* = 0$, then $e_i^* = t_i^* + \bar{\tau} + \tau'_{0k_i^*} \le g_i^* - \tau''_{k_i^*,b(i)} = f_i^*$, where the inequality becomes an equality when $k_i^* = 0$ (i.e., when $y_{i0}^* = 1$), and thus e_i^*, f_i^*, and y_{i0}^* satisfy (3.12). Hence, this solution satisfies constraint (3.12). For $i = 1, \ldots, n_1$, if $U_i^* = 0$, then $t_i^* \ne T + i$, which implies that $F_{it_i^*}(\lambda) = \min_{k=0,1,\ldots,m} \{F_{it_i^*}^{(k)}(\lambda)\} = F_{it_i^*}^{(k_i^*)}(\lambda)$ and that $F_{it_i^*}^{(k_i^*)}(\lambda)$ is finite. Thus, either $k_i^* = 0$ and $B_i \le t_i^* + \bar{\tau} + \tau_{0,b(i)} \le \bar{B}_i$, or $k_i^* > 0$ and $t_i^* + \bar{\tau} + \tau'_{0k_i^*} + \tau''_{k_i^*,b(i)} \le \bar{B}_i$. In both cases, $B_i \le g_i^* \le \bar{B}_i$. Hence, this solution also satisfies constraint (3.16). Therefore, $\{x_{it}^*, y_{ik}^*, z_{ikt}^*, e_i^*, f_i^*, g_i^*, L_{1i}^*, U_i^*\}$ is a feasible solution of $\mathbf{P}_{\text{in}}(\lambda)$.

We now show that the objective value of this feasible solution is less than or equal to $\sum_{i=1}^{n_1} \sum_{t=0}^{T+n_1} F_{it}(\lambda) x_{it}^{**}$, which is the optimal objective value of $\mathbf{P}'_{\text{in}}(\lambda)$. To do so, we show that for $i = 1, \ldots, n_1$, the inequality

$$C_{1i} L_{1i}^* + C_{0i} U_i^* + \sum_{k=1}^{m} \sum_{t=0}^{T} \lambda_{kt} z_{ikt}^* \le \sum_{t=0}^{T+n_1} F_{it}(\lambda) x_{it}^{**} \tag{3.36}$$

holds. We divide the analysis into two cases.

Case 1: $U_i^* = 1$. In this case, $x_{i,T+i}^{**} = 1$, $g_i^* = 0$, and $z_{ikt}^* = 0$ for $k = 1, \ldots, m$ and $t =$

$0, 1, \ldots, T$. Thus, $L^*_{1i} = 0$ and $\sum_{k=1}^m \sum_{t=0}^T \lambda_{kt} z^*_{ikt} = 0$. We have

$$C_{1i}L^*_{1i} + C_{0i}U^*_i + \sum_{k=1}^m \sum_{t=0}^T \lambda_{kt} z^*_{ikt} = C_{0i} = F_{i,T+i}(\lambda) \le \sum_{t=0}^{T+n_1} F_{it}(\lambda) x^{**}_{it},$$

and hence inequality (3.36) holds.

Case 2: $U^*_i = 0$. In this case, $t^*_i \neq T + i$. Thus, $t^*_i \in \{A_i, \ldots, T\} \cap \bigcup_{l=1}^{u_i} [\underline{w}_{il}, \bar{w}_{il} - \bar{\tau}]$. Hence,

$$\sum_{t=0}^{T+n_1} F_{it}(\lambda) x^{**}_{it} = F_{it^*_i}(\lambda) = \min_{k=0,1,\ldots,m} \{F^{(k)}_{it^*_i}(\lambda)\} = F^{(k^*_i)}_{it^*_i}(\lambda). \tag{3.37}$$

We further divide this case into three subcases.

Case 2.1: $k^*_i = 0$. In this case, if $t^*_i + \bar{\tau} + \tau_{0,b(i)} \notin [B_i, \bar{B}_i]$, then $F^{(k^*_i)}_{it^*_i}(\lambda) = +\infty$, and by (3.37), $\sum_{t=0}^{T+n_1} F_{it}(\lambda) x^{**}_{it} = +\infty$, which implies that inequality (3.36) holds. If $B_i \le t^*_i + \bar{\tau} + \tau_{0,b(i)} \le \bar{B}_i$, then

$$F^{(k^*_i)}_{it^*_i}(\lambda) = C_{1i}(t^*_i + \bar{\tau} + \tau'_{0,b(i)} - B_i). \tag{3.38}$$

Note that in this case, $\sum_{k=1}^m \sum_{t=0}^T \lambda_{kt} z^*_{ikt} = 0$ and $L^*_{1i} = g^*_i - B_i = t^*_i + \bar{\tau} + \tau_{0,b(i)} - B_i$. By (3.37) and (3.38),

$$C_{1i}L^*_{1i} + C_{0i}U^*_i + \sum_{k=1}^m \sum_{t=0}^T \lambda_{kt} z^*_{ikt} = C_{1i}(t^*_i + \bar{\tau} + \tau_{0,b(i)} - B_i) = F^{(k^*_i)}_{it^*_i}(\lambda) = \sum_{t=0}^{T+n_1} F_{it}(\lambda) x^{**}_{it},$$

and hence inequality (3.36) holds.

Case 2.2: $k^*_i > 0$ and $B_i \le t^*_i + \bar{\tau} + \tau'_{0k^*_i} + \tau''_{k^*_i,b(i)}$. In this case, if $t^*_i + \bar{\tau} + \tau'_{0k^*_i} + \tau''_{k^*_i,b(i)} > \bar{B}_i$, then $F^{(k^*_i)}_{it^*_i}(\lambda) = +\infty$, and by (3.37), $\sum_{t=0}^{T+n_1} F_{it}(\lambda) x^{**}_{it} = +\infty$, which implies that inequality (3.36) holds. If $t^*_i + \bar{\tau} + \tau'_{0k^*_i} + \tau''_{k^*_i,b(i)} \le \bar{B}_i$, then

$$F^{(k^*_i)}_{it^*_i}(\lambda) = C_{1i}(t^*_i + \bar{\tau} + \tau'_{0k^*_i} + \tau''_{k^*_i,b(i)} - B_i) + \lambda_{k^*_i, t^*_i + \bar{\tau} + \tau'_{0k^*_i}}. \tag{3.39}$$

Note that in this case, $g^*_i = t^*_i + \bar{\tau} + \tau'_{0k^*_i} + \tau''_{k^*_i,b(i)}$, which implies that

$$L^*_{1i} = t^*_i + \bar{\tau} + \tau'_{0k^*_i} + \tau''_{k^*_i,b(i)} - B_i. \tag{3.40}$$

In addition, $e^*_i = f^*_i = t^*_i + \bar{\tau} + \tau'_{0k^*_i}$, which implies that

$$\sum_{k=1}^m \sum_{t=0}^T \lambda_{kt} z^*_{ikt} = \sum_{t=e^*_i}^{f^*_i} \lambda_{k^*_i t} = \lambda_{k^*_i, t^*_i + \bar{\tau} + \tau'_{0k^*_i}}. \tag{3.41}$$

From (3.37) and (3.39)–(3.41), we have

$$C_{1i}L_{1i}^* + C_{0i}U_i^* + \sum_{k=1}^{m}\sum_{t=0}^{T}\lambda_{kt}z_{ikt}^* = C_{1i}(t_i^* + \bar{\tau} + \tau'_{0k_i^*} + \tau''_{k_i^*,b(i)} - B_i) + \lambda_{k_i^*, t_i^*+\bar{\tau}+\tau'_{0k_i^*}}$$

$$= F_{it_i^*}^{(k_i^*)}(\lambda) = \sum_{t=0}^{T+n_1} F_{it}(\lambda)x_{it}^{**},$$

and hence inequality (3.36) holds.

Case 2.3: $k_i^* > 0$ and $t_i^* + \bar{\tau} + \tau'_{0k_i^*} + \tau''_{k_i^*,b(i)} < B_i$. In this case, $g_i = B_i$, which implies that $f_i^* = B_i - \tau''_{k_i^*,b(i)}$ and $L_{1i}^* = 0$. Note that

$$\sum_{k=1}^{m}\sum_{t=0}^{T}\lambda_{kt}z_{ikt}^* = \sum_{t=e_i^*}^{f_i^*}\lambda_{k_i^*t}. \tag{3.42}$$

By (3.37), (3.42), and the definition of $F_{ik}^{(k)}(\lambda)$,

$$C_{1i}L_{1i}^* + C_{0i}U_i^* + \sum_{k=1}^{m}\sum_{t=0}^{T}\lambda_{kt}z_{ikt}^* = \sum_{t=e_i^*}^{f_i^*}\lambda_{k_i^*t} = \sum_{t=t_i^*+\bar{\tau}+\tau'_{0k_i^*}}^{B_i-\tau''_{k_i^*,b(i)}}\lambda_{k_i^*t} = F_{it_i^*}^{(k_i^*)}(\lambda) = \sum_{t=0}^{T+n_1}F_{it}(\lambda)x_{it}^{**},$$

and hence inequality (3.36) holds.

Summarizing the above cases, we conclude that the optimal objective value of $\mathbf{P}_{\text{in}}(\lambda)$ is no greater than the optimal objective value of $\mathbf{P}'_{\text{in}}(\lambda)$. Combining the results of the two parts, we conclude that problems $\mathbf{P}_{\text{in}}(\lambda)$ and $\mathbf{P}'_{\text{in}}(\lambda)$ have the same optimal objective value. Hence, solving $\mathbf{P}'_{\text{in}}(\lambda)$ yields the optimal objective value of $\mathbf{P}_{\text{in}}(\lambda)$. ▮

Next, we consider subproblem $\mathbf{P}_{\text{out}}(\lambda)$. Again, we introduce a dummy time point $T+i$ and denote $x_{i,T+i} = U_i$ for each outgoing vessel $i = n_1+1, \ldots, n_1+n_2$. Similar to subproblem $\mathbf{P}'_{\text{in}}(\lambda)$, for $i = n_1+1, \ldots, n_1+n_2$ and $t = 0, 1, \ldots, T+n_2$, we define

$$G_{it}(\lambda) = \begin{cases} \min_{k=0,1,\ldots,m}\left\{G_{it}^{(k)}(\lambda)\right\}, & \text{if } t \in \bigcup_{l=1}^{u_i}[\underline{w}_{il}, \bar{w}_{il} - \bar{\tau}]; \\ C_{0i}, & \text{if } t = T + i - n_1; \\ +\infty, & \text{otherwise;} \end{cases}$$

where

$$G_{it}^{(k)}(\lambda) = \begin{cases} C_{2i}\max\{t + \bar{\tau} - D_i,\, 0\}, & \text{if } k = 0 \text{ and } t = E_i + \tau_{0,b(i)}; \\ C_{2i}\max\{t + \bar{\tau} - D_i,\, 0\} + \sum_{t'=E_i+\tau''_{k,b(i)}}^{t-\tau'_{0k}}\lambda_{kt'}, & \text{if } k > 0 \text{ and } t \geq E_i + \tau''_{k,b(i)} + \tau'_{0k}; \\ +\infty, & \text{otherwise.} \end{cases}$$

Define

$$\begin{aligned}
\mathbf{P}'_{\text{out}}(\lambda):\ \ \text{minimize}\quad & \textstyle\sum_{i=n_1+1}^{n_1+n_2}\sum_{t=0}^{T+n_2} G_{it}(\lambda)x_{it}\\
\text{subject to}\quad & \textstyle\sum_{i=n_1+1}^{n_1+n_2} x_{it}\le 1\ \ (t=0,1,\ldots,T+n_2)\\
& \textstyle\sum_{t=0}^{T+n_2} x_{it}=1\ \ (i=n_1+1,\ldots,n_1+n_2)\\
& x_{it}\in\{0,1\}\ \ (i=n_1+1,\ldots,n_1+n_2;\ t=0,1,\ldots,T+n_2)
\end{aligned}$$

Lemma 2 *Solving problem $\mathbf{P}'_{\text{out}}(\lambda)$ yields the optimal objective value for problem $\mathbf{P}_{\text{out}}(\lambda)$.*

Proof: We divide the proof into two parts. In the first part, we show that the optimal objective value of $\mathbf{P}'_{\text{out}}(\lambda)$ is less than or equal to the optimal objective value of $\mathbf{P}_{\text{out}}(\lambda)$. Let $\{x^*_{it}, y^*_{ik}, z^*_{ikt}, e^*_i, f^*_i, L^*_{2i}, U^*_i\}$ be an optimal solution of $\mathbf{P}_{\text{out}}(\lambda)$. We construct a solution $\{x^{**}_{it}\}$ of problem $\mathbf{P}'_{\text{out}}(\lambda)$ by setting

$$x^{**}_{it}=\begin{cases} x^*_{it}, & \text{if } t\le T;\\ U^*_i, & \text{if } t=T+i-n_1;\\ 0, & \text{otherwise;}\end{cases}$$

for $i=n_1+1,\ldots,n_1+n_2$ and $t=0,1,\ldots,T+n_2$. Constraint (3.3) implies that $\sum_{i=n_1+1}^{n_1+n_2} x^*_{it}\le 1$ for $t=0,1,\ldots,T$. Thus, for $t=0,1,\ldots,T$,

$$\sum_{i=n_1+1}^{n_1+n_2} x^{**}_{it}=\sum_{i=n_1+1}^{n_1+n_2} x^*_{it}\le 1.$$

For $t=T+1,\ldots,T+n_2$,

$$\sum_{i=n_1+1}^{n_1+n_2} x^{**}_{it}=x^{**}_{t-T+n_1,t}=U^*_{t-T+n_1}\le 1.$$

Constraint (3.4) implies that $\sum_{t=0}^{T} x^*_{it}=1-U^*_i$ for $i=n_1+1,\ldots,n_1+n_2$. Thus, for $i=n_1+1,\ldots,n_1+n_2$,

$$\sum_{t=0}^{T+n_2} x^{**}_{it}=\sum_{t=0}^{T} x^{**}_{it}+\sum_{t=T+1}^{T+n_2} x^{**}_{it}=\sum_{t=0}^{T} x^*_{it}+U^*_i=1.$$

Hence, $\{x^{**}_{it}\}$ is a feasible solution of $\mathbf{P}'_{\text{out}}(\lambda)$.

We now show that the objective value of this feasible solution is less than or equal to $\sum_{i=n_1+1}^{n_1+n_2}\{C_{2i}L^*_{2i}+C_{0i}U^*_i+\sum_{k=1}^{m}\sum_{t=0}^{T}\lambda_{kt}z^*_{ikt}\}$, which is the optimal objective value of $\mathbf{P}_{\text{out}}(\lambda)$.

To do so, we show that for $i = n_1 + 1, \ldots, n_1 + n_2$, the inequality

$$\sum_{t=0}^{T+n_2} G_{it}(\lambda) x_{it}^{**} \le C_{2i} L_{2i}^* + C_{0i} U_i^* + \sum_{k=1}^{m} \sum_{t=0}^{T} \lambda_{kt} z_{ikt}^* \tag{3.43}$$

holds. We divide the analysis into two cases.

Case 1: $U_i^* = 1$. In this case, from (3.4), we have $x_{i0}^* = x_{i1}^* = \cdots = x_{iT}^* = 0$. Thus, $x_{i,T+i-n_1}^{**} = 1$ and $x_{it}^{**} = 0$ if $t \neq T + i - n_1$. We have

$$\sum_{t=0}^{T+n_2} G_{it}(\lambda) x_{it}^{**} = G_{i,T+i-n_1}(\lambda) = C_{0i} = C_{0i} U_i^*,$$

and hence inequality (3.43) holds.

Case 2: $U_i^* = 0$. In this case, by (3.4) and (3.19), exactly one of $x_{i0}^*, x_{i1}^*, \ldots, x_{iT}^*$ is equal to 1. Let t_i^* be the value of t such that $x_{it}^* = 1$. By (3.7) and (3.20), exactly one of $y_{i0}^*, y_{i1}^*, \ldots, y_{im}^*$ is equal to 1. Let k_i^* be the value of k such that $y_{ik}^* = 1$. Then, by (3.10),

$$e_i^* = E_i + \tau''_{k_i^*, b(i)}. \tag{3.44}$$

By (3.11),

$$f_i^* = t_i^* - \tau'_{0,k_i^*}. \tag{3.45}$$

By (3.14) and (3.15),

$$z_{ikt}^* = \begin{cases} 1, & \text{if } k = k_i^* \text{ and } e_i^* \le t \le f_i^*; \\ 0, & \text{otherwise;} \end{cases}$$

which implies that

$$\sum_{k=1}^{m} \sum_{t=0}^{T} \lambda_{kt} z_{ikt}^* = \sum_{t=0}^{T} \lambda_{k_i^* t} z_{ik_i^* t}^* = \sum_{t=e_i^*}^{f_i^*} \lambda_{k_i^* t}. \tag{3.46}$$

Note that in this case, $x_{i,T+1}^{**} = \cdots = x_{i,T+n_2}^{**} = 0$. Furthermore, by (3.6), $t_i^* \in \bigcup_{l=1}^{u_i} [\underline{w}_{il}, \bar{w}_{il} - \bar{\tau}]$. Hence,

$$\sum_{t=0}^{T+n_2} G_{it}(\lambda) x_{it}^{**} = \sum_{t=0}^{T} G_{it}(\lambda) x_{it}^{**} = \sum_{t=0}^{T} G_{it}(\lambda) x_{it}^* = G_{it_i^*}(\lambda) = \min_{k=0,1,\ldots,m} \left\{ G_{it_i^*}^{(k)}(\lambda) \right\} \le G_{it_i^*}^{(k_i^*)}(\lambda). \tag{3.47}$$

From (3.12), we have

$$e_i^* \le f_i^*. \tag{3.48}$$

From (3.18) and (3.25), we have

$$\max\{t_i^* + \bar{\tau} - D_i,\, 0\} \leq L_{2i}^*. \tag{3.49}$$

From (3.44), (3.45), and (3.48), we have

$$t_i^* \geq E_i + \tau''_{k_i^*,b(i)} + \tau'_{0k_i^*}. \tag{3.50}$$

We further divide this case into two subcases.

Case 2.1: $k_i^* = 0$. In this case, by (3.12), $e_i^* = f_i^*$. This, together with (3.44) and (3.45), implies that $t_i^* = E_i + \tau'_{00} + \tau''_{0,b(i)}$, or equivalently,

$$t_i^* = E_i + \tau_{0,b(i)}. \tag{3.51}$$

Thus, by (3.47), (3.49), (3.51), and the definition of $G_{it}^{(k)}(\lambda)$,

$$\sum_{t=0}^{T+n_2} G_{it}(\lambda)x_{it}^{**} \leq G_{it_i^*}^{(0)}(\lambda) = C_{2i}\max\{t_i^* + \bar{\tau} - D_i,\, 0\} \leq C_{2i}L_{2i}^*,$$

and hence inequality (3.43) holds.

Case 2.2: $k_i^* > 0$. In this case, by (3.44)–(3.47), (3.49), (3.50), and the definition of $G_{it}^{(k)}(\lambda)$,

$$\begin{aligned}\sum_{t=0}^{T+n_2} G_{it}(\lambda)x_{it}^{**} &\leq G_{it_i^*}^{(k_i^*)}(\lambda) = C_{2i}\max\{t_i^* + \bar{\tau} - D_i, 0\} + \sum_{t=E_i+\tau''_{k_i^*,b(i)}}^{t_i^*-\tau'_{0k_i^*}} \lambda_{k_i^*t} \\ &\leq C_{2i}L_{2i}^* + \sum_{t=e_i^*}^{f_i^*} \lambda_{k_i^*t} = C_{2i}L_{2i}^* + \sum_{k=1}^{m}\sum_{t=0}^{T} \lambda_{kt}z_{ikt}^*,\end{aligned}$$

and hence inequality (3.43) holds.

Summarizing the above cases, we conclude that the optimal objective value of $\mathbf{P}'_{\text{out}}(\lambda)$ is no greater than the optimal objective value of $\mathbf{P}_{\text{out}}(\lambda)$. This completes the first part of the proof.

In the second part, we show that the optimal objective value of $\mathbf{P}_{\text{out}}(\lambda)$ is less than or equal to the optimal objective value of $\mathbf{P}'_{\text{out}}(\lambda)$. Let $\{x_{it}^{**}\}$ be an optimal solution of problem $\mathbf{P}'_{\text{out}}(\lambda)$ with a finite objective value. Note that for $i = n_1 + 1, \ldots, n_1 + n_2$, either $x_{i,T+i-n_1}^{**} = 1$ or exactly one of $x_{i0}^{**}, x_{i1}^{**}, \ldots, x_{iT}^{**}$ is equal to 1. For each $i = n_1 + 1, \ldots, n_1 + n_2$, we let t_i^* be the value of t such that $x_{it}^{**} = 1$, and let $k_i^* = \arg\min_{k=0,1,\ldots,m}\{G_{it_i^*}^{(k)}(\lambda)\}$, with ties broken arbitrarily. We construct a feasible solution $\{x^*_{it}, y^*_{ik}, z^*_{ikt}, e^*_i, f^*_i, L^*_{2i}, U^*_i\}$ of problem $\mathbf{P}_{\text{out}}(\lambda)$ as follows:

$$x^*_{it} = x^{**}_{it} \quad (i = n_1+1, \ldots, n_1+n_2;\ t = 0, 1, \ldots, T);$$

$$U^*_i = x^{**}_{i,T+i-n_1} \quad (i = n_1+1, \ldots, n_1+n_2);$$

$$y^*_{ik} = \begin{cases} 1, & \text{if } k = k^*_i \text{ and } U^*_i = 0, \\ 0, & \text{otherwise,} \end{cases} \quad (i = n_1+1, \ldots, n_1+n_2;\ k = 0, 1, \ldots, m);$$

$$e^*_i = \begin{cases} E_i + \tau''_{k^*_i, b(i)}, & \text{if } U^*_i = 0, \\ 0, & \text{otherwise,} \end{cases} \quad (i = n_1+1, \ldots, n_1+n_2);$$

$$f^*_i = \begin{cases} t^*_i - \tau'_{0,k^*_i}, & \text{if } U^*_i = 0, \\ 0, & \text{otherwise,} \end{cases} \quad (i = n_1+1, \ldots, n_1+n_2);$$

$$z^*_{ikt} = \begin{cases} 1, & \text{if } k = k^*_i,\ U^*_i = 0, \text{ and } e^*_i \le t \le f^*_i, \\ 0, & \text{otherwise,} \end{cases} \quad (i = n_1+1, \ldots, n_1+n_2;\ k = 1, \ldots, m;\ t = 0, 1, \ldots, T);$$

$$L^*_{2i} = \begin{cases} \max\{t^*_i + \bar{\tau} - D_i,\ 0\}, & \text{if } U^*_i = 0, \\ 0, & \text{otherwise,} \end{cases} \quad (i = n_1+1, \ldots, n_1+n_2).$$

It is easy to see that this solution satisfies constraints (3.3), (3.4), (3.7), (3.10), (3.11), (3.14), (3.15), (3.18)–(3.23), and (3.25) (for $i = n_1+1, \ldots, n_1+n_2$). Since the objective value of solution $\{x^{**}_{it}\}$ is finite, $G_{it^*_i}(\lambda)$ is finite for all $i = n_1+1, \ldots, n_1+n_2$. Thus, for $i = n_1+1, \ldots, n_1+n_2$, either $t^*_i = T+i-n_1$ or $t^*_i \in \bigcup_{l=1}^{u_i} [\underline{w}_{il}, \bar{w}_{il} - \bar{\tau}]$, which implies that $x^*_{it} = 0$ if $t \notin \bigcup_{l=1}^{u_i} [\underline{w}_{il}, \bar{w}_{il} - \bar{\tau}]$. Hence, x^*_{it} satisfies constraint (3.6) (for $i = n_1+1, \ldots, n_1+n_2$). For $i = n_1+1, \ldots, n_1+n_2$, if $U^*_i = 1$, then $e^*_i = f^*_i = y^*_{i0} = 0$, and thus e^*_i, f^*_i, and y^*_{i0} satisfy (3.12). If $U^*_i = 0$, then $t^*_i \ne T+i-n_1$, which implies that $G_{it^*_i}(\lambda) = \min_{k=0,1,\ldots,m} \left\{G^{(k)}_{it^*_i}(\lambda)\right\} = G^{(k^*_i)}_{it^*_i}(\lambda)$ and that $G^{(k^*_i)}_{it^*_i}(\lambda)$ is finite. Thus, either $k^*_i = 0$ and $t^*_i = E_i + \tau_{0,b(i)}$, or $k^*_i > 0$ and $t^*_i \ge E_i + \tau''_{k^*_i,b(i)} + \tau'_{0k^*_i}$. In the first case, $y^*_{i0} = 1$ and $e^*_i = E_i + \tau''_{k^*_i,b(i)} = E_i + \tau_{0,b(i)} = t^*_i = f^*_i$, and thus e^*_i, f^*_i, and y^*_{i0} satisfy (3.12). In the second case, $y^*_{i0} = 0$ and $e^*_i = E_i + \tau''_{k^*_i,b(i)} \le t^*_i - \tau'_{0k^*_i} = f^*_i$, and thus e^*_i, f^*_i, and y^*_{i0} also satisfy (3.12). Hence, this solution satisfies constraint (3.12). Therefore, $\{x^*_{it}, y^*_{ik}, z^*_{ikt}, e^*_i, f^*_i, L^*_{2i}, U^*_i\}$ is a feasible solution of $\mathbf{P}_{\text{out}}(\lambda)$.

We now show that the objective value of this feasible solution is less than or equal to $\sum_{i=n_1+1}^{n_1+n_2} \sum_{t=0}^{T+n_2} G_{it}(\lambda) x^{**}_{it}$, which is the optimal objective value of $\mathbf{P}'_{\text{out}}(\lambda)$. To do so, we show that for $i = n_1 + 1, \ldots, n_1 + n_2$, the inequality

$$C_{2i}L^*_{2i} + C_{0i}U^*_i + \sum_{k=1}^{m}\sum_{t=0}^{T} \lambda_{kt} z^*_{ikt} \le \sum_{t=0}^{T+n_2} G_{it}(\lambda)x^{**}_{it} \tag{3.52}$$

holds. We divide the analysis into two cases.

Case 1: $U^*_i = 1$. In this case, $x^{**}_{i,T+i-n_1} = 1$, $L^*_{2i} = 0$, and $z^*_{ikt} = 0$ for $k = 1, \ldots, m$ and $t = 0, 1, \ldots, T$. Thus, $\sum_{k=1}^{m}\sum_{t=0}^{T} \lambda_{kt} z^*_{ikt} = 0$. We have

$$C_{2i}L^*_{2i} + C_{0i}U^*_i + \sum_{k=1}^{m}\sum_{t=0}^{T} \lambda_{kt} z^*_{ikt} = C_{0i} = G_{i,T+i-n_1}(\lambda) \le \sum_{t=0}^{T+n_2} G_{it}(\lambda)x^{**}_{it},$$

and hence inequality (3.52) holds.

Case 2: $U^*_i = 0$. In this case, $t^*_i \ne T + i - n_1$. Thus, $t^*_i \in \bigcup_{l=1}^{u_i}[\underline{w}_{il}, \bar{w}_{il} - \bar{\tau}]$. Hence,

$$\sum_{t=0}^{T+n_2} G_{it}(\lambda)x^{**}_{it} = G_{it^*_i}(\lambda) = \min_{k=0,1,\ldots,m} \left\{G^{(k)}_{it^*_i}(\lambda)\right\} = G^{(k^*_i)}_{it^*_i}(\lambda). \tag{3.53}$$

We further divide this case into two subcases.

Case 2.1: $k^*_i = 0$. In this case, if $t^*_i \ne E_i + \tau_{0,b(i)}$, then $G^{(k^*_i)}_{it^*_i}(\lambda) = +\infty$, and by (3.53), $\sum_{t=0}^{T+n_2} G_{it}(\lambda)x^{**}_{it} = +\infty$, which implies that inequality (3.52) holds. If $t^*_i = E_i + \tau_{0,b(i)}$, then

$$G^{(k^*_i)}_{it^*_i}(\lambda) = C_{2i}\max\{t^*_i + \bar{\tau} - D_i,\, 0\}. \tag{3.54}$$

Note that in this case, $\sum_{k=1}^{m}\sum_{t=0}^{T} \lambda_{kt} z^*_{ikt} = 0$ and $L^*_{2i} = \max\{t^*_i + \bar{\tau} - D_i,\, 0\}$. By (3.53) and (3.54),

$$C_{2i}L^*_{2i} + C_{0i}U^*_i + \sum_{k=1}^{m}\sum_{t=0}^{T} \lambda_{kt} z^*_{ikt} = C_{2i}\max\{t^*_i + \bar{\tau} - D_i,\, 0\} = G^{(k^*_i)}_{it^*_i}(\lambda) = \sum_{t=0}^{T+n_2} G_{it}(\lambda)x^{**}_{it},$$

and hence inequality (3.52) holds.

Case 2.2: $k^*_i > 0$. In this case, if $t^*_i < E_i + \tau''_{k^*_i,b(i)} + \tau'_{0k^*_i}$, then $G^{(k^*_i)}_{it^*_i}(\lambda) = +\infty$, and by (3.53), $\sum_{t=0}^{T+n_2} G_{it}(\lambda)x^{**}_{it} = +\infty$, which implies that inequality (3.52) holds. If $t^*_i \ge E_i + \tau''_{k^*_i,b(i)} + \tau'_{0k^*_i}$, then by the definition of $G^{(k)}_{it}(\lambda)$,

$$G^{(k^*_i)}_{it^*_i}(\lambda) = C_{2i}\max\{t^*_i + \bar{\tau} - D_i,\, 0\} + \sum_{t=E_i+\tau''_{k^*_i,b(i)}}^{t^*_i - \tau'_{0k^*_i}} \lambda_{k^*_i t} = C_{2i}\max\{t^*_i + \bar{\tau} - D_i,\, 0\} + \sum_{t=e^*_i}^{f^*_i} \lambda_{k^*_i t}. \tag{3.55}$$

Note that in this case,

$$L^*_{2i} = \max\{t^*_i + \bar{\tau} - D_i,\, 0\} \tag{3.56}$$

and

$$\sum_{k=1}^{m}\sum_{t=0}^{T}\lambda_{kt}z^{*}_{ikt}=\sum_{t=e_i^*}^{f_i^*}\lambda_{k_i^*t}. \tag{3.57}$$

By (3.53) and (3.55)–(3.57),

$$C_{2i}L^*_{2i}+C_{0i}U^*_i+\sum_{k=1}^{m}\sum_{t=0}^{T}\lambda_{kt}z^*_{ikt}=C_{2i}\max\{t^*_i+\bar{\tau}-D_i,\,0\}+\sum_{t=e_i^*}^{f_i^*}\lambda_{k_i^*t}=G^{(k_i^*)}_{it_i^*}(\lambda)=\sum_{t=0}^{T+n_2}G_{it}(\lambda)x^{**}_{it},$$

and hence inequality (3.52) holds.

Summarizing the above cases, we conclude that the optimal objective value of $\mathbf{P}_{\text{out}}(\lambda)$ is no greater than the optimal objective value of $\mathbf{P}'_{\text{out}}(\lambda)$. Combining the results of the two parts, we conclude that problems $\mathbf{P}_{\text{out}}(\lambda)$ and $\mathbf{P}'_{\text{out}}(\lambda)$ have the same optimal objective value. Hence, solving $\mathbf{P}'_{\text{out}}(\lambda)$ yields the optimal objective value of $\mathbf{P}_{\text{out}}(\lambda)$. ▮

Problems $\mathbf{P}'_{\text{in}}(\lambda)$ and $\mathbf{P}'_{\text{out}}(\lambda)$ are $n_1\times\big(T+n_1+1\big)$ and $n_2\times\big(T+n_2+1\big)$ asymmetric assignment problems, respectively, and can be solved by the Hungarian method whose running time is polynomial in the problem size. Since the sizes of these asymmetric assignment problems are linear functions of T, these two problems can be solved in pseudo-polynomial time.

3.2.2 Upper Bound Heuristic and the Subgradient Method

Given any vector λ of λ_{kt} values, the optimal objective value of problem $\mathbf{P}(\lambda)$, i.e., $L(\lambda)$, is a lower bound on the optimal objective value of problem $\mathbf{P}$. We now develop a heuristic method for constructing a feasible solution of problem $\mathbf{P}$ based on the optimal solution of problem $\mathbf{P}(\lambda)$. The objective value of this feasible solution serves as an upper bound on the optimal objective value of problem $\mathbf{P}$, and this upper bound is used for updating the Lagrangian multipliers in a subgradient optimization framework, which will be described later. In the optimal solution of problem $\mathbf{P}(\lambda)$, each vessel i either cannot be served successfully (i.e., $\sum_{t=0}^{T}x_{it}=0$) or is assigned a time point for entering the navigation channel (i.e., $\sum_{t=0}^{T}x_{it}=1$). A feasible solution of problem $\mathbf{P}$ can be constructed by fixing these x_{it} values and then solving the MILP of problem $\mathbf{P}$. After fixing the x_{it} values, the MILP of problem $\mathbf{P}$ can be simplified as follows. Denote $\Omega=\{i\mid\sum_{t=0}^{T}x_{it}=1;\ i=1,\ldots,n_1+n_2\}$. For $i,j\in\Omega$ such that $i\neq j$, define new binary variables $\delta_{ij}=1$ if vessel i arrives at its staging anchorage earlier than vessel j, and $\delta_{ij}=0$ otherwise. Then, a feasible solution to problem $\mathbf{P}$ can be obtained by solving the following problem:

$$\begin{aligned}
\mathbf{P}_{\text{ub}}: \text{ minimize } \quad & \textstyle\sum_{i=1}^{n_1} C_{1i}L_{1i} + \sum_{i=n_1+1}^{n_1+n_2} C_{2i}L_{2i} + \sum_{i=1}^{n_1+n_2} C_{0i}U_i \\
\text{subject to } \quad & (3.7)\text{–}(3.12),\ (3.16)\text{–}(3.18),\ (3.20),\ (3.22)\text{–}(3.25) \\
& e_j \geq f_i + 1 - M(1-\delta_{ij}) \quad (i,j \in \Omega;\ i \neq j) \qquad (3.58) \\
& \delta_{ij} + \delta_{ji} \geq y_{ik} + y_{jk} - 1 \quad (i,j \in \Omega;\ i \neq j;\ k = 1,\ldots,m) \qquad (3.59) \\
& \delta_{ij} \in \{0,1\} \quad (i,j \in \Omega;\ i \neq j) \qquad (3.60)
\end{aligned}$$

Problem $\mathbf{P}_{\text{ub}}$ differs from problem $\mathbf{P}$ in that (i) x_{it} has become an input parameter, (ii) variable z_{ikt} and constraints (3.2)–(3.6) and (3.13)–(3.15) have been removed, and (iii) variable δ_{ij} and constraints (3.58)–(3.59) have been introduced to ensure that $[e_i, f_i]$ and $[e_j, f_j]$ do not overlap if vessels i and j make use of the same staging anchorage. After solving problem $\mathbf{P}_{\text{ub}}$, we let $z_{ikt} = 0$ if $y_{ik} = 0$ or $t \notin [e_i, f_i]$, and let $z_{ikt} = 1$ if $y_{ik} = 1$ and $t \in [e_i, f_i]$. It is easy to see that these z_{ikt} values satisfy constraints (3.13)–(3.15). Hence, solving problem $\mathbf{P}_{\text{ub}}$ yields a feasible solution of problem $\mathbf{P}$. Note that the number of variables in problem $\mathbf{P}_{\text{ub}}$ is independent of T; therefore, the number of decision variables and the number of constraints of problem $\mathbf{P}_{\text{ub}}$ are polynomial in the input size.

We apply the standard subgradient optimization method (see Held, Wolfe, and Crowder 1974) to update the Lagrangian multipliers. The subgradient for constraint (3.13) at the ℓ-th iteration of the subgradient algorithm is the vector Δ^ℓ with components

$$\Delta_{kt}^\ell = \textstyle\sum_{i=1}^{n_1+n_2} z_{ikt}^\ell - 1,$$

for $k = 1,\ldots,m$ and $t = 0,1,\ldots,T$. Denote λ_{kt}^ℓ as the value of λ_{kt} at the ℓ-th iteration of the subgradient algorithm. The value of λ_{kt} is updated as follows:

$$\lambda_{kt}^\ell = \begin{cases} 0, & \text{if } \ell = 1; \\ \max\{0, \lambda_{kt}^{\ell-1} + \zeta^{\ell-1}\Delta_{kt}^{\ell-1}\}, & \text{if } \ell \geq 2; \end{cases} \qquad (3.61)$$

where ζ^ℓ is the step size at the ℓ-th iteration.

Let Z_{LD} be the optimal objective value of the Lagrangian dual problem $\mathbf{P}_{\text{LD}}$, and $\underline{Z}(\lambda^\ell)$ be the optimal objective value of problem $\mathbf{P}(\lambda^\ell)$. According to Fisher (2004), the most commonly used step size rule is in the following form:

$$\zeta^\ell = \varepsilon \frac{Z^* - \underline{Z}(\lambda^\ell)}{\|\Delta^\ell\|^2}, \tag{3.62}$$

where Z^* is an estimate of Z_{LD}, and ε is a step size control parameter. Held, Wolfe, and Crowder (1974) showed that if Z^* is a lower bound of Z_{LD} and $0 < \varepsilon \leq 2$, then $\underline{Z}(\lambda^\ell)$ converges to either Z^* or a value between Z^* and Z_{LD}. Since a good lower bound of Z_{LD} is typically not known, Z^* is usually set to be an upper bound of the optimal objective value of the primal problem $\mathbf{P}$. Meanwhile, to ensure convergence of $\underline{Z}(\lambda^\ell)$, ε is often decreased iteratively so that ζ^ℓ converges to 0.

In our implementation, we determine the step size according to (3.62). We can make use of an upper bound $\bar{Z}(\lambda^\ell)$ obtained by solving problem $\mathbf{P}_{\text{ub}}$ to estimate Z_{LD}. However, as mentioned earlier, the cost coefficients of unsatisfied service requests (i.e., the C_{0i} values) are often significantly larger than the cost coefficients of berthing and departure tardiness (i.e., the C_{1i} and C_{2i} values). Hence, $\bar{Z}(\lambda^\ell)$ can be much larger than Z_{LD} if $U_i = 1$ for some i in the upper bound solution but $U_i = 0$ for all i in an optimal solution of problem $\mathbf{P}$. To overcome this pitfall, we use $Z^* = \min\{\bar{Z}(\lambda^\ell), 2\underline{Z}(\lambda^\ell)\}$ as a better estimate of Z_{LD}.

We start the subgradient algorithm with ε initially set to 1. Let $\bar{Z}$ and $\underline{Z}$ denote the best upper bound and the best lower bound found so far, respectively. When the value of $\underline{Z}$ is not improved for 5 consecutive iterations, we reduce ε by 20%. The subgradient algorithm is terminated when (i) we reach 100 iterations, or (ii) the optimality gap, measured by $\big(\bar{Z} - \underline{Z}\big)/\underline{Z} \times 100\%$, is below 1%. A pseudo-code of our solution method is provided below:

Notations:

ℓ: Iteration number

θ^ℓ: Optimality gap at the ℓth iteration, which is measured by $\big(\bar{Z} - \underline{Z}\big)/\underline{Z} \times 100\%$

κ: An integer that records the number of consecutive iterations during which $\underline{Z}$ is not improved

The pseudo-code is given as follows:

Lagrangian Relaxation Heuristic:

```
    Initialization:
1:  $\ell \leftarrow 1$; $\kappa \leftarrow 0$; $\varepsilon \leftarrow 1$; $\underline{Z} \leftarrow 0$; $\bar{Z} \leftarrow +\infty$; $\theta^\ell \leftarrow +\infty$; $\lambda^\ell_{kt} \leftarrow 0$ $(k = 1, \ldots, m;\ t = 0, \ldots, T)$.
```

```
    *Subgradient optimization procedure:*
2:  **While** $\ell \leq 100$ and $\theta^\ell \geq 1\%$
3:      Solve problems $\mathbf{P}_{\text{in}}(\lambda^\ell)$ and $\mathbf{P}_{\text{out}}(\lambda^\ell)$ to obtain optimal $x_{it}$ values for problem $\mathbf{P}(\lambda^\ell)$ and the lower bound $\underline{Z}(\lambda^\ell)$;
4:      **If** $\underline{Z}(\lambda^\ell) > \underline{Z}$ **Then**
5:          $\underline{Z} \leftarrow \underline{Z}(\lambda^\ell)$; $\kappa \leftarrow 0$
6:      **Else**
7:          $\kappa \leftarrow \kappa + 1$.
8:      **End If**
9:      Solve problem $\mathbf{P}_{\text{ub}}$ with the $x_{it}$ values obtained in Step 3 to obtain the $z_{ikt}^\ell$ values and the upper bound $\bar{Z}(\lambda^\ell)$;
10:     $\bar{Z} \leftarrow \min\{\bar{Z}, \bar{Z}(\lambda^\ell)\}$; $\theta^\ell \leftarrow \big(\bar{Z} - \underline{Z}\big)/\underline{Z}$;
11:     **If** $\kappa \geq 5$ **Then**
12:         $\varepsilon \leftarrow 0.8 \times \varepsilon$; $\kappa \leftarrow 0$.
13:     **End If**
14:     $\Delta_{kt}^\ell \leftarrow \sum_{i=1}^{n_1+n_2} z_{ikt}^\ell - 1$ $(k = 1, \ldots, m;\ t = 0, \ldots, T)$;
15:     $Z^* \leftarrow \min\{\bar{Z}(\lambda^\ell), 2\underline{Z}(\lambda^\ell)\}$; $\zeta^\ell \leftarrow \varepsilon(Z^* - \underline{Z}(\lambda^\ell))/(\|\Delta^\ell\|^2)$;
16:     $\ell \leftarrow \ell + 1$; $\lambda_{kt}^\ell \leftarrow \max\{0, \lambda_{kt}^{\ell-1} + \zeta^{\ell-1}\Delta_{kt}^{\ell-1}\}$ $(k = 1, \ldots, m;\ t = 0, \ldots, T)$.
17: **End While**
```

Steps 2–17 form the main body of the Lagrangian relaxation heuristic. Step 3 solves the Lagrangian relaxation problem. Steps 4–8 update the best lower bound $\underline{Z}$. Step 9 executes the upper bound heuristic to obtain a feasible solution of problem **P**. Step 10 updates the best upper bound $\bar{Z}$ and the optimality gap. Steps 11–13 determine if ε should be reduced by 20%. Steps 14–17 update the step size, the iteration number, and the Lagrangian multipliers. The output of the Lagrangian relaxation heuristic is the solution corresponding to the best upper bound $\bar{Z}$.

3.3 Computational Experiments

The goal of the computational experiments is threefold. First, we would like to evaluate the computational performance of the Lagrangian relaxation heuristic for problems of different sizes.

For this purpose, we generate problem instances with planning horizons of different lengths and compute the optimality gaps of the solutions obtained by the heuristic. We also compare the computational performance of Lagrangian relaxation heuristic with those of benchmark solutions. Two benchmark solution methods are used for comparison. One method is to solve the MILP of problem **P** directly using CPLEX, a well-known mathematical programming solver. Another method is to adopt a rule-based approach which mimics the current practice of the VTS operator at the Yangshan Deep-water Port in Shanghai. Since in practice some parameter values may be different under different situations, our second goal is to investigate how the performance of the Lagrangian relaxation heuristic is affected as these parameters vary. In particular, since weather conditions may affect the availability of staging anchorages in the terminal basin, we examine how the results are affected as the number of staging anchorages varies. We also analyze the sensitivity of the Lagrangian heuristic's performance to different settings of tardiness penalties (i.e., the C_{1i} and C_{2i} values). Because it is not easy to quantify the tardiness penalties, a solution with lower sensitivity to the values of the tardiness penalties is considered more robust, and thus of higher practicality. Our third goal is to evaluate the benefits of taking the anchorage area's capacity into consideration when planning navigation channel traffic. To achieve this, we analyze the solutions obtained by several vessel sequencing policies when the anchorage area's capacity is ignored.

All algorithms were implemented in C#.Net and ran on a computer with a 64-bit Intel i7-6700 3.40GHz CPU and 32GB RAM. In the Lagrangian relaxation heuristic, the upper bound was obtained by solving the MILP of problem $\mathbf{P}_{\text{ub}}$. All MILPs were solved by CPLEX 12.5 with default configurations.

3.3.1 Generation of Problem Instances

In this subsection, we describe the test instances used in our computational experiments. These test instances are randomly generated with the parameter setting selected based on the physical layout and the characteristics of the operational data of the Yangshan Deep-water Port of Shanghai. Figure 3.4 depicts the physical layout of the Yangshan Deep-water Port. The port has one navigation channel with two traffic lanes. One traffic lane is for incoming vessels, and the other is for outgoing vessels. We let $U\{\alpha, .., \beta\}$ denote the random number generator which returns a

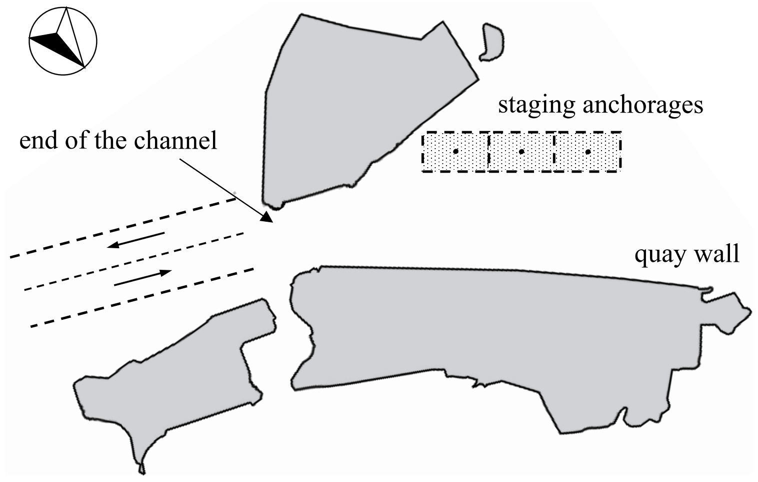

Figure 3.4: Layout of the Yangshan Deep-water Port (not to scale).

uniformly distributed random integer from $\{\alpha, \alpha+1, \ldots, \beta\}$, and we say that $R \sim U\{\alpha, .., \beta\}$ if R is a random number generated by $U\{\alpha, .., \beta\}$. We let $U[\alpha, \beta]$ denote the random number generator which returns a uniformly distributed random real number lying within the interval $[\alpha, \beta]$, and we say that $R \sim U[\alpha, \beta]$ if R is a random number generated by $U[\alpha, \beta]$.

The safety clearance between vessels traveling through the navigation channel is set equal to 10 minutes. Thus, each time unit represents 10 minutes. The amount of time for a vessel to travel through the navigation channel $\bar{\tau}$ is set equal to 12 time units. We generate test instances with planning horizons varying from 1 day to 7 days (i.e., T is set equal to $144, 288, \ldots, 1008$ time units) in three categories, namely the low-traffic case, the medium-traffic case, and the heavy-traffic case, and develop 21 problem sets. For each problem set, we generate 5 random test instances, and thus, there are 105 test instances in total. For a test instance with a planning horizon of d days, the number of incoming vessels is obtained by setting $n_1 \sim U\{10d, .., 12d\}$ for the low-traffic case, $n_1 \sim U\{12d, .., 14d\}$ for the medium-traffic case, and $n_1 \sim U\{14d, .., 16d\}$ for the heavy-traffic case. The number of outgoing vessels n_2 is set equal to n_1. See Table 3.1 for a summary of the problem sets.

The Yangshan Deep-water Port possesses a continuous quay wall divided into 16 berth segments. Hence, in each test instance, there are 16 berthing positions, numbered from 1 to 16. For example, if vessel i is designated to berth at the 10th position, then $b(i) = 10$. The number of staging

Table 3.1: Problem sets used in the computational study.

	Low-traffic		Medium-traffic		Heavy-traffic	
T	Problem set	n_1 and n_2	Problem set	n_1 and n_2	Problem set	n_1 and n_2
144	L-1	$U\{10,..,12\}$	M-1	$U\{12,..,14\}$	H-1	$U\{14,..,16\}$
288	L-2	$U\{20,..,24\}$	M-2	$U\{24,..,28\}$	H-2	$U\{28,..,32\}$
432	L-3	$U\{30,..,36\}$	M-3	$U\{36,..,42\}$	H-3	$U\{42,..,48\}$
576	L-4	$U\{40,..,48\}$	M-4	$U\{48,..,56\}$	H-4	$U\{56,..,64\}$
720	L-5	$U\{50,..,60\}$	M-5	$U\{60,..,70\}$	H-5	$U\{70,..,80\}$
864	L-6	$U\{60,..,72\}$	M-6	$U\{72,..,84\}$	H-6	$U\{84,..,96\}$
1008	L-7	$U\{70,..,84\}$	M-7	$U\{84,..,98\}$	H-7	$U\{98,..,102\}$

anchorages is set to $m = 3$. The travel speed of all vessels in the terminal basin is set equal to 1000 meters per time unit. The coordinates of the berthing positions, staging anchorages, and end of navigation channel, measured in meters, are provided in Table 3.2. The travel times $\tau_{0,b(i)}$, τ'_{0k}, and $\tau''_{k,b(i)}$ are obtained by dividing the Euclidean distances between the two locations concerned by the travel speed of the vessel and rounding the results to the nearest integers. For example, suppose vessel i is assigned to staging anchorage 1 and $b(i) = 10$. Since the Euclidean distance between staging anchorage 1 and berthing position 10 is $\sqrt{(3500-1800)^2+(0-2000)^2} \approx 2625$ meters, we have $\tau''_{1,b(i)} = 2625/1000 \approx 3$.

Table 3.3 presents the statistics of the operational data of year 2016 at the Yangshan Deep-water Port. For incoming vessels, the following statistics are presented:

- VESS_IN: Average number of scheduled incoming vessels per day;

Table 3.2: Coordinates of different locations.

Berthing positions and their coordinates		Staging anchorages and their coordinates		Coordinates of the end of navigation channel
1	(350, 0)	1	(1800, 2000)	(0, 600)
2	(700, 0)	2	(2800, 2000)	
3	(1050, 0)	3	(3800, 2000)	
4	(1400, 0)			
5	(1750, 0)			
6	(2100, 0)			
7	(2450, 0)			
8	(2800, 0)			
9	(3150, 0)			
10	(3500, 0)			
11	(3850, 0)			
12	(4200, 0)			
13	(4550, 0)			
14	(4900, 0)			
15	(5250, 0)			
16	(5600, 0)			

- BTH_{min}, BTH_{max}, BTH_{avg}: Earliest, latest, and average planned berthing time of the scheduled incoming vessels;
- BTH_LEAD: Average length (in minutes) of the interval between arrival time and planned berthing time of the scheduled incoming vessels;
- BTH_DELAY: Average length (in minutes) of the interval between the planned berthing time and the actual berthing time of the scheduled incoming vessels;
- DP_DFT_IN: Proportion of deep-draft vessels (i.e., vessels with drafts not less than 12.5 meters) among all the scheduled incoming vessels;
- $\text{DFT\_IN}_{\text{min}}$, $\text{DFT\_IN}_{\text{max}}$, $\text{DFT\_IN}_{\text{avg}}$: Minimum, maximum, and average draft (in meters) of the deep-draft incoming vessels.

For outgoing vessels, the following statistics are presented:

- VESS_OUT: Average number of scheduled outgoing vessels per day;
- UBT_{min}, UBT_{max}, UBT_{avg}: Earliest, latest, and average planned unberthing time of the scheduled outgoing vessels;
- DPT_LEAD: Average length (in minutes) of the interval between unberthing time and expected departure time of the scheduled outgoing vessels.

For each month, the average number of scheduled incoming vessels and the average number of scheduled outgoing vessels are almost the same. The value ranges between 11.5 and 13.4, with a mean of 12.3 (see the VESS_IN and VESS_OUT values in Table 3.3). Therefore, in each test instance of our computational study, we set $n_1 = n_2$. To capture the variation in the number of scheduled vessels, we generate three categories of test instances, namely the low-traffic case, the medium-traffic case, and the heavy-traffic case. For instances of the low-traffic case, 10 incoming vessels and 10 outgoing vessels are scheduled for service per day; for instances of the medium-traffic case, 12 incoming vessels and 12 outgoing vessels are scheduled for service per day; and for instances of the heavy-traffic case, 14 incoming vessels and 14 outgoing vessels are scheduled for service per day. Note that the berth plans that we extracted are executed berth plans, i.e., they have been verified by the VTS operator. In the berth plans initially devised and proposed by the terminal operators, the number of scheduled incoming and outgoing vessels may be larger than

Table 3.3: Monthly statistics of the operational data of 2016 at the Yangshan Deep-water Port*

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Average
Incoming vessels													
VESS_IN	12.5	12.2	13.0	12.8	12.2	13.3	13.2	12.9	11.9	11.7	11.5	11.5	12.2
BTH_{min}	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00
BTH_{max}	23:59	23:59	23:59	23:59	23:30	23:59	23:59	23:59	23:30	23:30	23:30	23:59	23:49
BTH_{avg}	11:58	11:48	12:05	12:45	12:14	11:24	12:04	12:25	12:06	12:00	11:40	12:21	12:04
BTH_LEAD	1521	1871	2531	1923	1078	1480	1533	1267	1614	1869	2117	2181	1749
BTH_DELAY	1530	1470	1530	1790	1560	1560	1531	1531	1650	1560	1510	1650	1656
DP_DFT_IN	0.28	0.26	0.22	0.26	0.27	0.28	0.25	0.25	0.23	0.21	0.20	0.22	0.24
$\text{DFT\_IN}_{\text{min}}$	12.5	12.6	12.5	12.7	12.5	12.5	12.6	12.5	12.6	12.5	12.5	12.5	12.5
$\text{DFT\_IN}_{\text{max}}$	14.3	14.3	15	14	14.4	14.2	15.2	14.5	14.3	14.5	14.7	14.6	14.4
$\text{DFT\_IN}_{\text{avg}}$	13.1	13.3	13.5	13.3	13.6	13.6	13.4	13	13.1	13.6	13.2	13.6	13.3
Outgoing vessels													
VESS_OUT	12.5	12.2	13.0	12.8	12.3	13.3	13.4	12.9	11.9	11.7	11.5	11.6	12.3
UBT_{min}	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00
UBT_{max}	23:30	23:30	23:30	23:30	23:30	23:30	23:30	23:59	23:30	23:30	23:30	23:30	23:32
UBT_{avg}	11:39	10:44	11:43	12:24	12:03	11:23	11:54	11:44	12:07	12:11	11:34	12:00	11:47
DPT_LEAD	354	−83	498	425	−424	−288	40	14	43	464	719	802	214

*These statistics are for large vessels only. This is because small vessels and barges need not enter the navigation channel when arriving at or departing from the port. In addition, small vessels and large vessels do not occupy the same anchorage area. Therefore, we only consider large vessels in our test instances.

that in the executed berth plans, as service requests of some vessels may have been rejected by the VTS operator due to insufficient traffic capacity. To capture the difference in the number of scheduled vessels between the executed berth plans and the initial berth plans, we enlarge both the number of scheduled incoming vessels and the number of scheduled outgoing vessels by ϵ for each instance, where $\epsilon \sim U\{0, .., 2d\}$, and d is the number of days in the planning horizon. Consequently, for each instance with a planning horizon of d days, the number of incoming vessels (or outgoing vessels) is generated by $U\{10d, .., 12d\}$, $U\{12d, .., 14d\}$, and $U\{14d, .., 16d\}$ for the low-traffic case, medium-traffic case, and heavy-traffic case, respectively.

Both the planned berthing times of incoming vessels and the planned unberthing times of outgoing vessels shown in Table 3.3 range between 00:00 and 23:59, with a mean value around 12:00 (see the BTH and UBT values in the table), indicating that the B_i and E_i values are evenly distributed throughout the planning horizon. Since the travel time of a vessel is no more than 20 time units (12 time units in the navigation channel and, according to the locations shown in Table 3.2, no more than 8 time units in the terminal basin), we generate the values of B_i by $U\{20, .., T\}$, and generate the values of E_i by $U\{0, .., T-20\}$. For the incoming vessels, the

BTH_LEAD values range between 1078 and 2531 minutes (i.e., between 107.8 and 253.1 time units). Thus, for each incoming vessel i, we set $A_i = \max\{0, B_i - R\}$ with $R \sim U\{100, .., 250\}$. The BTH_DELAY values range between 1470 and 1790 minutes (i.e., between 147.0 and 179.0 time units). Thus, for each incoming vessel i, we set $\bar{B}_i = \min\{B_i + R, T\}$ with $R \sim U\{150, .., 180\}$. For the outgoing vessels, the DPT_LEAD values range between -424 and 802 minutes (i.e., between -42.4 and 80.2 time units). Thus, for each outgoing vessel i, we set $D_i = \max\{0, E_i + R\}$ with $R \sim U\{-40, .., 80\}$.

Vessels with draft greater than 12.5 meters are classified as "deep-draft vessels." We refer to those vessels with draft no greater than 12.5 meters as "non-deep-draft vessels." Non-deep-draft vessels are unaffected by the tide and have a single tidal window $[0, T]$. Among the incoming vessels, about 24% of them are deep-draft vessels (see the DP_DFT_IN value in the "Average" column of Table 3.3). To reflect this characteristic, in our computational study we randomly select 24% of the incoming vessels to be deep-draft vessels. The drafts of deep-draft vessels range between 12.5 meters and 15.2 meters, with a mean of 13.3 meters (see the DFT_IN values in Table 3.3). Thus, we generate the drafts of deep-draft incoming vessels by $U[12.5, 15.2]$ meters. The drafts of outgoing vessels are not recorded at the Yangshan Deep-water Port. However, the proportion of deep-draft vessels among the outgoing vessels is about the same as that of the incoming vessels. We therefore generate deep-draft vessels and their drafts for the outgoing vessels the same way as we do for the incoming vessels. We derive the tidal windows of the deep-draft vessels based on their drafts and the information of the tide. The average water level of the port is 16 meters, the average highest and lowest tide of the port is ± 1.5 meters, and the average duration of each tidal cycle of the port is 12 hours (i.e., 72 time units). Following the method of Du et al. (2015), we simulate the tidal pattern using a sine curve. Hence, we let the water level at time t be $16 + 1.5 \sin \frac{\pi t}{36}$. A vessel is allowed to navigate through the navigation channel when the water level is greater than or equal to the draft of the vessel plus a seabed clearance of 2 meters. Therefore, the tidal windows of a deep-water level with draft ξ is given as $\left\{t \;\middle|\; 16 + 1.5 \sin \frac{\pi t}{36} \geq \xi + 2;\ 0 \leq t \leq T\right\}$.

Deep-draft vessels have a higher service priority than non-deep-draft vessels. We set the tardiness penalties C_{1i} and C_{2i} to 2 if vessel i is a deep-draft vessel, and set C_{1i} and C_{2i} to 1 if vessel

i is a non-deep-draft vessel. The objective of our model includes two types of penalties, namely the penalties on berthing and departure tardiness of vessels, and the penalties on unsatisfied vessel service requests. In practice, satisfying vessel service requests is of the highest priority. To reflect this priority, we set C_{0i} to a significantly larger value than C_{1i} and C_{2i}. For each vessel i, we set C_{0i} equal to 10000. Thus, if the objective value of a heuristic solution exceeds 10000, it indicates that either the given berth plans are impossible to satisfy all service requests due to limited traffic capacity of the port, or the heuristic fails to identify a solution that satisfies all service requests.

3.3.2 Benchmark Methods

To evaluate the computational performance of the Lagrangian relaxation heuristic, we introduce two benchmark methods for solving problem **P** and compare the performance of the Lagrangian relaxation heuristic with the performances of these benchmark methods. The first benchmark method is to solve the MILP formulation of problem **P** presented in Section 3.1 directly by CPLEX. The second benchmark method is a rule-based heuristic, which simulates the decision of a VTS operator. In the following, we provide a description on the manual decision process.

The VTS officers at the Yangshan Deep-water Port currently adopt a rule-based approach to manage the traffic in the navigation channel and the utilization of the staging anchorages. This rule-based heuristic gives priority to outgoing vessels to ensure on-time unberthing, so as to provide berth vacancy for subsequent vessels. It first considers the outgoing vessels one by one according to their unberthing times. In each step, the heuristic allocates a time slot of the navigation channel to the outgoing vessel being considered. It also allocates some time slots of a staging anchorage to this vessel if the vessel cannot travel to the channel and exit the port directly. If two outgoing vessels have the same unberthing time, then the one with the higher service priority (i.e., larger C_{2i} value) is considered first. Next, the heuristic considers the incoming vessels one by one according to their planned berthing times. In each step, it allocates the next available time slot of the navigation channel to the incoming vessel being considered. It also allocates some time slots of a staging anchorage to this vessel if the vessel is too early to travel to its berth directly. If two incoming vessels have the same planned berthing time, then the one with the higher service priority (i.e., larger C_{1i} value) is considered first. The pseudo-code of the rule-based heuristic used to simulate the manual decision process is presented below:

Notations:

TC: The objective value

$\mathcal{V}_1$, $\mathcal{V}_2$: Sets of the incoming and outgoing vessels, respectively, that have not been considered

$\mathcal{T}_1$, $\mathcal{T}_2$: Sets of available time points for the incoming and outgoing vessels, respectively, to enter the navigation channel

$\mathcal{S}_k$: Set of available time points for staging anchorage k

$\mathcal{W}$: A set for storing the indices of vessels

e_i^k, f_i^k: The values of e_i and f_i, respectively, if vessel i makes use of staging anchorage k

The rule-based heuristic is presented as follows:

Rule-based Heuristic:

```
    *Initialization:*
1:  $TC \leftarrow 0$; $\mathcal{V}_1 \leftarrow \{1,\dots,n_1\}$; $\mathcal{V}_2 \leftarrow \{n_1+1,\dots,n_1+n_2\}$; $\mathcal{T}_1 \leftarrow \{0,1,\dots,T\}$; $\mathcal{T}_2 \leftarrow \{0,1,\dots,T\}$;
    $\mathcal{S}_k \leftarrow \{0,1,\dots,T\}$ $(k=1,\dots,m)$; $t \leftarrow 0$.
    *Managing traffic of outgoing vessels:*
2:  **While** $t \le T$
3:    $\mathcal{W} \leftarrow \{i \in \mathcal{V}_2 \mid E_i = t\}$;
4:    **While** $\mathcal{W} \neq \emptyset$
5:      $i \leftarrow \arg\max_{j\in\mathcal{W}}\{C_{2j}\}$ (ties broken arbitrarily);
6:      **If** $t+\tau_{0,b(i)} \in \mathcal{T}_2 \cap \bigcup_{l=1}^{u_i}[\underline{w}_{il}, \bar{w}_{il}-\bar{\tau}]$ **Then**
7:        $\mathcal{T}_2 \leftarrow \mathcal{T}_2 \setminus \{t+\tau_{0,b(i)}\}$; $U_i \leftarrow 0$; $\mathcal{V}_2 \leftarrow \mathcal{V}_2 \setminus \{i\}$; $TC \leftarrow TC + C_{2i}\max\{0, t+\tau_{0,b(i)}+\bar{\tau}-D_i\}$
8:      **Else**
9:        $e_i^k \leftarrow E_i + \tau''_{k,b(i)}$; $f_i^k \leftarrow \min\{t' \in \mathcal{T}_2 \cap \bigcup_{l=1}^{u_i}[\underline{w}_{il}, \bar{w}_{il}-\bar{\tau}] \mid t' \ge e_i^k + \tau'_{0k}\} - \tau'_{0k}$ $(k=1,\dots,m)$;
10:       **If** $\exists k = 1,\dots,m$ such that $[e_i^k, f_i^k] \subseteq \mathcal{S}_k$ **Then**
11:         $e_i \leftarrow e_i^k$; $f_i \leftarrow f_i^k$; $\mathcal{T}_2 \leftarrow \mathcal{T}_2 \setminus \{f_i+\tau'_{0k}\}$; $\mathcal{S}_k \leftarrow \mathcal{S}_k \setminus [e_i, f_i]$; $U_i \leftarrow 0$; $\mathcal{V}_2 \leftarrow \mathcal{V}_2 \setminus \{i\}$;
            $TC \leftarrow TC + C_{2i}\max\{0, f_i+\tau'_{0k}+\bar{\tau}-D_i\}$.
12:       **End If**
13:     **End If**
14:     $\mathcal{W} \leftarrow \mathcal{W} \setminus \{i\}$.
15:   **End While**
```

```
16:     $t \leftarrow t + 1$.
17: **End While**
    *Managing traffic of incoming vessels:*
18: $t \leftarrow 0$.
19: **While** $t \leq T$
20:     $\mathcal{W} \leftarrow \big\{i \in \mathcal{V}_1 \mid B_i = \min_{j \in \mathcal{V}_1}\{B_j\};\ t \geq A_i;\ t \in \bigcup_{l=1}^{u_i}[\underline{w}_{il}, \bar{w}_{il} - \bar{\tau}]\big\}$;
21:     **While** $\mathcal{W} \neq \emptyset$
22:         $i \leftarrow \arg\max_{j \in \mathcal{W}}\{C_{1j}\}$ (ties broken arbitrarily);
23:         **If** $B_i \leq t + \bar{\tau} + \tau_{0,b(i)} \leq \bar{B}_i$ **Then**
24:             $\mathcal{T}_1 \leftarrow \mathcal{T}_1 \setminus \{t\}$; $U_i \leftarrow 0$; $\mathcal{V}_1 \leftarrow \mathcal{V}_1 \setminus \{i\}$; $TC \leftarrow TC + C_{1i}\big(t + \bar{\tau} + \tau_{0,b(i)} - B_i\big)$; $\mathcal{W} \leftarrow \emptyset$
25:         **Else**
26:             $e_i^k \leftarrow t + \bar{\tau} + \tau'_{0k}$; $f_i^k \leftarrow e_i^k$ if $e_i^k + \tau''_{k,b(i)} \geq B_i$; $f_i^k \leftarrow B_i - \tau''_{k,b(i)}$ if $e_i^k + \tau''_{k,b(i)} < B_i$
            $(k = 1, \ldots, m)$;
27:             **If** $\exists k = 1, \ldots, m$ such that $[e_i^k, f_i^k] \subseteq \mathcal{S}_k$ and $f_i^k + \tau''_{k,b(i)} \leq \bar{B}_i$ **Then**
28:                 $e_i \leftarrow e_i^k$; $f_i \leftarrow f_i^k$; $\mathcal{T}_1 \leftarrow \mathcal{T}_1 \setminus \{t\}$; $\mathcal{S}_k \leftarrow \mathcal{S}_k \setminus [e_i, f_i]$; $U_i \leftarrow 0$; $\mathcal{V}_1 \leftarrow \mathcal{V}_1 \setminus \{i\}$;
                $TC \leftarrow TC + C_{1i}\max\{0, f_i + \tau''_{k,b(i)} - B_i\}$; $\mathcal{W} \leftarrow \emptyset$
29:             **Else**
30:                 $\mathcal{W} \leftarrow \mathcal{W} \setminus \{i\}$.
31:             **End If**
32:         **End If**
33:     **End While**
34:     $t \leftarrow t + 1$.
35: **End While**
    *Including the costs of unsatisfied service requests:*
36: **For** $i \in \mathcal{V}_1 \cup \mathcal{V}_2$
37:     $U_i \leftarrow 1$ and $TC \leftarrow TC + C_{0i}$.
```

Steps 2–35 form the main body of the heuristic for scheduling vessel traffic. The main body consists of two parts. The first part (i.e., steps 2–17) schedules the traffic of the outgoing vessels, whereas the second part (i.e., steps 18–35) schedules the traffic of the incoming vessels. In each iteration of the first part, a time point t is considered. Given the time point t, step 3 identifies the outgoing vessels that should unberth at time t. If such outgoing vessels are identified (whose indices are stored in the set $\mathcal{W}$), the one with the highest service priority is then selected in step 5 (with ties broken arbitrarily). Then, steps 6–13 determine whether the selected outgoing vessel can enter the navigation channel or not. If the outgoing vessel cannot enter the channel directly, then the vessel is assigned a staging anchorage, provided that at least one staging anchorage is available. If the outgoing vessel is assigned a time point for entering the navigation channel, the vessel is then eliminated from the outgoing vessel list $\mathcal{V}_2$, and its departure tardiness cost is then added to the total cost TC. Each of the outgoing vessels in the set $\mathcal{W}$ is eliminated from the set in step 14 after being treated in steps 6–13. In each iteration of the second part, a time point t is considered. Given the time point t, step 20 identifies the incoming vessels that can enter the navigation channel at time t, among which the ones with the earliest planned berthing time are selected with their indices stored in the set $\mathcal{W}$. Among the incoming vessels in $\mathcal{W}$, the one with the highest service priority is selected in step 22 (with ties broken arbitrarily). Then, steps 23–32 determine whether the selected incoming vessel can berth within its berthing time window or not. If the incoming vessel cannot travel to its berthing position directly and berth within its berthing time window, then the vessel is assigned a staging anchorage, provided that at least one staging anchorage is available. If the incoming vessel is assigned a time point for entering the navigation channel, the vessel is then eliminated from the incoming vessel list $\mathcal{V}_1$, and its berthing tardiness cost is then added to the total cost TC. Each of the incoming vessels in the set $\mathcal{W}$ is eliminated from the set after being treated in steps 23–32. When the iterations of the main body are exhausted, vessels that remain in the sets $\mathcal{V}_1$ and $\mathcal{V}_2$ will not be served. Thus, in steps 36–37, the cost for unsatisfied service requests is added to the total cost TC for each of the vessels that remain in the sets $\mathcal{V}_1$ and $\mathcal{V}_2$.

3.3.3 Comparison with Benchmark Methods

We solve the problem sets presented in Table 3.1 in Section 3.3.1 using the Lagrangian relaxation heuristic and the two benchmark methods, namely solving the MILP directly via CPLEX (which we refer to as the "MILP/CPLEX method") and the rule-based heuristic. When solving each problem, the running time of the MILP/CPLEX method is restricted to no more than 3600 seconds.

Table 3.4: Computational results for the tested problem sets.

Problem	Lagrangian Relaxation					MILP/CPLEX					Rule-based Heuristic				
set	#I	#R	G1 (%)	G2 (%)	Time	#I	#R	G1 (%)	G2 (%)	Time	#I	#R	G1 (%)	G2 (%)	Time
Low-traffic															
L-1	0	0	0.0	0.0	0.1	0	0	0.0	0.0	11.5	1	0.4	≥ 100	0.5	0.0
L-2	0	0	2.7	2.7	3.1	0	0	5.5	5.5	2414.1	1	0.2	≥ 100	12.1	0.0
L-3	0	0	0.7	0.7	3.1	1	0.4	≥ 100	11.4	3168.4	0	0	14.3	14.3	0.0
L-4	0	0	0.0	0.0	0.8	3	1.0	≥ 100	6.6	3580.6	1	0.2	≥ 100	16.4	0.0
L-5	1	0.2	≥ 100	1.6	21.1	5	7.2	≥ 100	–	3600	2	0.6	≥ 100	9.5	0.0
L-6	0	0	1.2	1.2	36.8	5	26.8	≥ 100	–	3600	2	0.4	≥ 100	21.0	0.0
L-7	2	0.4	≥ 100	1.4	75.1	5	82.8	≥ 100	–	3600	1	0.6	≥ 100	18.3	0.0
Medium-traffic															
M-1	0	0	0.0	0.0	0.1	0	0	0.0	0.0	16.1	0	0	4.4	4.4	0.0
M-2	0	0	1.9	1.9	1.7	0	0	8.4	8.4	2932.5	2	0.4	≥ 100	21.2	0.0
M-3	0	0	3.1	3.1	13.6	2	0.4	≥ 100	27.3	3600	0	0	14.5	14.5	0.0
M-4	0	0	3.2	3.2	34.4	5	4.8	≥ 100	–	3600	0	0	26.5	26.5	0.0
M-5	0	0	4.4	4.4	60.7	5	7.6	≥ 100	–	3600	1	0.2	≥ 100	21.8	0.0
M-6	0	0	1.5	1.5	98.7	5	28.4	≥ 100	–	3600	1	0.2	≥ 100	23.2	0.0
M-7	1	0.4	≥ 100	4.8	176.2	5	87.8	≥ 100	–	3600	1	0.4	≥ 100	32.2	0.0
Heavy-traffic															
H-1	0	0	0.1	0.1	0.2	0	0	0.0	0.0	54.3	1	0.4	≥ 100	3.2	0.0
H-2	0	0	0.5	0.5	1.6	1	0.4	≥ 100	18.4	2978.1	0	0	23.4	23.4	0.0
H-3	0	0	1.8	1.8	20.8	4	2.2	≥ 100	29.1	3600	2	0.4	≥ 100	23.7	0.0
H-4	1	0.2	≥ 100	4.7	39.5	5	6.0	≥ 100	–	3600	2	1.2	≥ 100	30.7	0.0
H-5	0	0	7.7	7.7	106.8	5	22.8	≥ 100	–	3600	1	0.2	≥ 100	35.6	0.0
H-6	1	0.4	≥ 100	4.4	133.5	5	91.4	≥ 100	–	3600	3	1.8	≥ 100	40.9	0.0
H-7	0	0	4.2	4.2	273.1	5	134.0	≥ 100	–	3600	3	0.6	≥ 100	33.6	0.0

Table 3.4 reports the computational results obtained by the Lagrangian relaxation heuristic, the MILP/CPLEX method, and the rule-based heuristic, where each row summarizes the results of 5 random test instances. For each problem set and each solution method, the "#I" column reports the number of instances (out of 5 instances) in which some service requests cannot be satisfied, and the "#R" column reports the average number of unsatisfied service requests. Detailed computational results, including the upper bound and lower bound values of each test instance, are provided in Appendix B.

The "G1" columns report the average optimality gaps of the objective values of the test instances in each problem set. The optimality gap of a test instance is given by $(UB - LB)/LB \times 100\%$, where UB is the objective value of the solution generated by the solution method concerned, and LB is the lower bound generated by the Lagrangian relaxation heuristic. Recall that the penalty on each unsatisfied service request, C_{0i}, is significantly larger than the tardiness penalties C_{1i} and C_{2i}. Hence, if there exists an instance for which all vessel service requests are satisfied in the lower bound solution but some are not satisfied in the upper bound solution, then the optimality gap of that instance, and thus the G1 value of the problem set, would become very large. As a supplementary to the G1 columns, we report in the "G2" columns the average optimality gaps of those test instances for which all vessel service requests are satisfied in the upper bound solutions. Hence, the G2 values represent the accuracy of the solution methods on determining the berthing and departure tardiness costs. For each problem set, if service requests are not satisfied in each of the 5 instances, then the G2 value for this problem set is unavailable. The running time (in seconds) of the solution methods is reported in the "Time" columns.

From Table 3.4, we observe that the values in the "#I" column of Lagrangian relaxation heuristic are small, showing that the Lagrangian relaxation heuristic could identify complete vessel schedules for most of the test instances. In particular, for those instances with planning horizons no more than 3 days (i.e., instances in L-1, L-2, L-3, M-1, M-2, M-3, H-1, H-2, and H-3), the Lagrangian relaxation heuristic successfully finds solutions with no unsatisfied service request. In contrast, both the MILP/CPLEX method and rule-based heuristic fail to identify complete vessel schedules frequently. Values in the "#R" columns show that the Lagrangian relaxation heuristic results in very few unsatisfied service requests. However, the MILP/CPLEX method results in far more unsatisfied service requests. For instances with planning horizons longer than 3 days, the number of unsatisfied service requests generated by the MILP/CPLEX method increases dramatically as the length of the planning horizon increases. The rule-based heuristic also results in more unsatisfied service requests than the Lagrangian relaxation heuristic. Thus, the Lagrangian relaxation heuristic outperforms the benchmark methods by far in minimizing the number of unsatisfied service requests.

To evaluate the performance of the solution methods on minimizing the berthing and departure tardiness, we compare the optimality gaps on the tardiness cost, i.e., the values in the G2 columns. The performance of the Lagrangian relaxation algorithm is as good as that of CPLEX for small-sized instances in L-1 and M-1. For larger problem instances, the G2 values of MILP/CPLEX are either unavailable or greater than those of the Lagrangian relaxation heuristic. The G2 values of rule-based heuristic are significantly worse than those of the Lagrangian relaxation heuristic. Hence, the Lagrangian relaxation heuristic is superior over the benchmark methods in minimizing the berthing and departure tardiness of vessels. Overall, the average optimality gap of the Lagrangian relaxation heuristic on the tardiness cost is within 8% for each of the tested problem sets. The Lagrangian relaxation heuristic reaches the stopping criteria within a reasonable amount of computer time. On the other hand, the MILP/CPLEX method finds optimal solutions within the running time limit only for small instances.

From Table 3.4, we observe that the #I, #R, G1, and G2 values of all three solution methods tend to increase as the length of the planning horizon increases, indicating that it is more difficult to schedule the vessel traffic for longer term planning. The performance of the three solution methods is also affected by the traffic density. As the traffic density increases, it is more difficult to ensure on-time berthing and departure, and thus the objective values of the solutions generated by the three solution methods, as well as the Lagrangian relaxation lower bound, tend to increase as the traffic density increases (see Tables A1–A3 in Appendix B). In addition, we observe that the G1 and G2 values of all three solution methods also tend to increase as the traffic becomes heavier. This shows the increase in difficulty in finding good solutions as the traffic density increases. However, the #I and #R values of Lagrangian relaxation heuristic are hardly affected by traffic density. This shows that the Lagrangian relaxation heuristic is quite capable in minimizing unsatisfied service requests under heavy traffic conditions.

As mentioned earlier, the rule-based heuristic mimics the current practice of the VTS operator at the Yangshan Deep-water Port in Shanghai. Hence, a comparison between the performance of the Lagrangian relaxation heuristic and that of the rule-based heuristic reflects the benefit of using the Lagrangian relaxation heuristic. Table 3.5 summarizes the overall improvement generated by the Lagrangian relaxation heuristic as opposed to the rule-based heuristic. The comparison shows that the Lagrangian relaxation heuristic results in a 17.9% reduction in vessel tardiness. The most significant benefit brought by the Lagrangian relaxation heuristic is that the Lagrangian relaxation solutions have substantially fewer unsatisfied service requests. Because the terminal operators will have to adjust the berth plans when vessels' service requests cannot be satisfied, the Lagrangian relaxation heuristic can help reduce the frequency of berth plan adjustment.

Table 3.5: Performance improvement of Lagrangian relaxation over rule-based heuristic.

	Number of instances with unsatisfied service requests	Average number of unsatisfied service requests per instance	Average tardiness cost per instance	Average total cost per instance
Lagrangian Relaxation	6	0.1	1183.4	1945.3
Rule-based Heuristic	25	0.3	1440.9	5345.7
Percentage Improvement	76.0%	82.9%	17.9%	63.6%

3.3.4 Varying Anchorage Capacity and Tardiness Penalties

In the computational study presented in Section 3.3.3, we have set the number of staging anchorages to 3. However, in practice, the number of available staging anchorages in the terminal basin is different under different situations. Thus, we investigate how the computational performance of the Lagrangian relaxation heuristic is affected as the anchorage capacity varies. For this purpose, we solve the problem sets with the longest planning horizon (i.e., L-7, M-7, and H-7) using the Lagrangian relaxation heuristic with different values of m.

We analyze the computational results with m set equal to 1, 2, 3, 4, and 5. When changing the value of m, we regenerate the locations of the staging anchorages by evenly dividing the anchorage area of the terminal basin (see Figure 3.4) into m parts. Table 3.6 summarizes the computational results.

Table 3.6: Computational results under different values of m.

Problem set	m	#I	#R	G1 (%)	G2 (%)	Time
L-7	1	5	4.4	≥ 100	–	49.0
	2	2	2.8	≥ 100	8.2	78.2
	3	2	0.4	≥ 100	1.4	75.1
	4	0	0	0.5	0.4	25.9
	5	0	0	0.1	0.1	11.6
M-7	1	5	5	≥ 100	–	96.9
	2	3	1	≥ 100	11.4	112.7
	3	1	0.4	≥ 100	4.8	176.2
	4	1	0.2	≥ 100	1.7	89.6
	5	0	0	0.2	0.2	20.1
H-7	1	5	3.2	≥ 100	–	150.2
	2	2	0.4	≥ 100	17.8	201.2
	3	0	0	4.2	4.2	273.1
	4	0	0	1.2	1.2	172.5
	5	0	0	0.2	0.2	52.8

From these results, we observe that the Lagrangian relaxation heuristic is successful in identifying complete vessel schedules when m equals 5, but often fails to do so when m equals 1 and 2. This is because when the number of staging anchorages decreases, the chance that the optimal solution contains no unsatisfied service request drops. We also observe that the optimality gap G2 improves as m increases. This is because the Lagrangian relaxation problem $\mathbf{P}(\lambda)$ is developed by dualizing constraint (3.13). Since the purpose of constraint (3.13) is to ensure that at most one vessel dwells at each staging anchorage at each time point, increasing the number of staging anchorages makes this constraint less restrictive, and, as a result, tightens the lower bound.

In the computational study presented in Section 3.3.3, we have set the tardiness penalties C_{1i} and C_{2i} to 1 or 2 depending on whether vessel i is a deep-draft vessel or not. In practice, there may be other factors affecting a vessel's service priority. Thus, we investigate the sensitivity of the Lagrangian relaxation heuristic performance to a change in the tardiness penalties' distribution. For this purpose, we generate new instances from the problem sets L-7, M-7, and H-7 by randomly perturbing the tardiness penalties of vessels. Let $\nu \in [0, 1)$ be a parameter, and let C'_{1i} and C'_{2i} be the tardiness penalties of vessels after perturbation. For each instance in the original data set, we generate new instances by setting $C'_{1i} = \rho C_{1i}$ with $\rho \sim U[1-\nu, 1+\nu]$ for each $i = 1, \ldots, n_1$, and setting $C'_{2i} = \rho C_{2i}$ with $\rho \sim U[1-\nu, 1+\nu]$ for each $i = n_1 + 1, \ldots, n_1 + n_2$. For each instance in problem sets L-10, M-10, and H-10, we generate four new instances with ν set equal to 0.1, 0.2, 0.3, 0.4. We solve these new instances using the Lagrangian relaxation heuristic.

Table 3.7: Computational results under different values of ν.

Problem set	ν	#I	#R	G1 (%)	G2 (%)	Time
L-7	0	2	0.4	≥ 100	1.4	75.1
	0.1	2	0.4	≥ 100	1.3	76.1
	0.2	2	0.4	≥ 100	1.4	93.4
	0.3	2	0.4	≥ 100	1.2	86.2
	0.4	2	0.4	≥ 100	1.2	80.2
M-7	0	1	0.4	≥ 100	4.8	176.2
	0.1	1	0.4	≥ 100	5.4	170.7
	0.2	1	0.4	≥ 100	5.8	172.3
	0.3	1	0.4	≥ 100	5.3	185.1
	0.4	1	0.4	≥ 100	6.0	182.2
H-7	0	0	0	4.2	4.2	273.1
	0.1	0	0	4.8	4.8	291.4
	0.2	0	0	4.9	4.9	282.6
	0.3	0	0	5.2	5.2	292.5
	0.4	0	0	4.7	4.7	269.1

Table 3.7 summarizes the computational results under different values of ν. From these results,

we observe that the #I and #R values are insensitive to the change in ν value, showing the number of unsatisfied service requests is insensitive to the change of tardiness penalties. We also observe that the G1 and G2 values are also insensitive to the change in ν value. This implies that the performance of the Lagrangian relaxation heuristic is insensitive to the C_{1i} and C_{2i} values.

3.3.5 Sequencing Vessels by Ignoring Anchorage Capacity

One key feature of our optimization model is that the capacity of the anchorage area and the capacity of the navigation channel are considered simultaneously. If the capacity of the anchorage area is ignored, then the problem becomes a pure vessel-sequencing problem that can be solved efficiently via some heuristic rules. However, the effectiveness of the solution is affected if the capacity of the anchorage area is not considered when the traffic of the navigation channel is scheduled. In this subsection, we computationally assess the benefits of taking the anchorage area's capacity into consideration when planning navigation channel traffic.

Suppose the capacity of the anchorage area is ignored. Then, the main problem is to determine the sequence of incoming vessels that enters the navigation channel from the outer anchorage ground, as well as the sequence of outgoing vessels that enters the navigation channel from the port basin. We consider several vessel sequencing policies that other researchers have considered for port management, including the first-come first-served (FCFS) policy, the shortest time windows length (STW) policy, and the random sequencing (RS) policy; see Cordeau et al. (2005) and Lalla-Ruiz, Shi, and Voß (2018).

We computationally evaluate the performances of the FCFS, STW, and RS policies for sequencing incoming and outgoing vessels when these policies are applied to our model. When applying the FCFS policy to our problem, we let the incoming vessels enter the navigation channel in ascending order of their port arrival times, and let the outgoing vessels enter the navigation channel in ascending order of their unberthing time, with ties broken arbitrarily. When applying the STW policy to our problem, we let the incoming vessels enter the navigation channel in ascending order of their tidal window lengths, and let the outgoing vessels enter the navigation channel in ascending order of their tidal window lengths, with ties broken arbitrarily. When applying the RS policy to our problem, a random sequence of incoming vessels is generated by selecting vessels one by one with equal probability, and a random sequence of outgoing vessels is generated by selecting vessels one by one with equal probability.

Because we have ignored the anchorage capacity when scheduling the navigation channel traffic, the anchorage area may run out of capacity during execution. Thus, we need a mechanism to allocate the anchorage space. In our computational experiments, after obtaining the sequence of incoming vessels and the sequence of outgoing vessels, we allocate the staging anchorages to the vessels in a greedy manner as follows:

```
Step 1: Set $t \leftarrow 0$.
Step 2: Consider the next vessel $i$ in the given outgoing vessel sequence.
    Step 2.1: Determine the earliest time point $t' \geq t+1$ at which vessel $i$ can enter the navigation channel.
    Step 2.2: If vessel $i$ cannot travel directly to the navigation channel and reach the end of the channel at time $t'$, then search for an available staging anchorage for vessel $i$ (with ties broken arbitrarily), and let $i$ be an unsatisfied service request if no staging anchorage is available.
    Step 2.3: Set $t \leftarrow t'$. If vessel $i$ is not the last vessel in the outgoing vessel sequence, then repeat Step 2.
Step 3: Set $t \leftarrow 0$.
Step 4: Consider the next vessel $i$ in the given incoming vessel sequence.
    Step 4.1: Determine the earliest time point $t' \geq t+1$ at which vessel $i$ can enter the navigation channel.
    Step 4.2: If vessel $i$ cannot travel directly to its berthing position after entering the channel at time point $t'$, then search for an available staging anchorage for vessel $i$ so that vessel $i$ can reach $b(i)$ during the time window $[B_i, \bar{B}_i]$ (with ties broken arbitrarily), and let $i$ be an unsatisfied service request if no such staging anchorage is available.
    Step 4.3: Set $t \leftarrow t'$. If vessel $i$ is not the last vessel in the incoming vessel sequence, then repeat Step 4.
```

Note that in the above steps, priority is given to outgoing vessels in order to ensure on-time unberthing and to provide berth vacancy for subsequent vessels. As mentioned by Lalla-Ruiz, Shi, and Voß (2018), the RS policy has the advantage that different vessel sequences can be obtained by repeating the policy multiple times and that each run requires very little time effort. Hence, in our computational study, if the RS policy fails to generate a solution with no unsatisfied vessel service, then we repeat the RS policy with different vessel sequences until either (i) a solution with no unsatisfied vessel service is generated, or (ii) the RS policy has been repeated 100 times.

We test the FCFS, STW, and RS policies using the same test instances as in Section 3.3.3. Table 3.8 reports the results, where each row summarizes the results of 5 random test instances. Since the execution of these policies requires very little computational time, the running times are not included in this table. Detailed computational results for each test instance are provided in Appendix B.

Table 3.8: Comparison with different vessel sequencing policies.

Problem	Lagrangian Relaxation				FCFS				STW				RS			
set	#I	#R	G1 (%)	G2 (%)	#I	#R	G1 (%)	G2 (%)	#I	#R	G1 (%)	G2 (%)	#I	#R	G1 (%)	G2 (%)
Low-traffic																
L-1	0	0	0.0	0.0	0	0	≥ 100	≥ 100	0	0	≥ 100	≥ 100	0	0	97.1	97.1
L-2	0	0	2.7	2.7	0	0	39.4	39.4	5	9.2	≥ 100	–	5	16.2	≥ 100	–
L-3	0	0	0.7	0.7	0	0	44.7	44.7	5	20.2	≥ 100	–	5	25.2	≥ 100	–
L-4	0	0	0.0	0.0	0	0	43.3	43.3	5	35.4	≥ 100	–	5	37.8	≥ 100	–
L-5	1	0.2	≥ 100	1.6	0	0	52.1	52.1	5	46.2	≥ 100	–	5	51	≥ 100	–
L-6	0	0	1.2	1.2	0	0	65.8	65.8	5	59	≥ 100	–	5	63.6	≥ 100	–
L-7	2	0.4	≥ 100	1.4	2	0.6	≥ 100	50.7	5	72.8	≥ 100	–	5	74.8	≥ 100	–
Medium-traffic																
M-1	0	0	0.0	0.0	0	0	33.1	33.1	0	0	41.8	41.8	0	0	≥ 100	≥ 100
M-2	0	0	1.9	1.9	0	0	69.7	69.7	4	12	≥ 100	≥ 100	4	16.8	≥ 100	≥ 100
M-3	0	0	3.1	3.1	0	0	54.7	54.7	5	26.2	≥ 100	–	5	34.6	≥ 100	–
M-4	0	0	3.2	3.2	0	0	80.7	80.7	5	46.6	≥ 100	–	5	50.6	≥ 100	–
M-5	0	0	4.4	4.4	0	0	70.3	70.3	5	60.4	≥ 100	–	5	62.6	≥ 100	–
M-6	0	0	1.5	1.5	1	0.2	≥ 100	87.9	5	74.2	≥ 100	–	5	76	≥ 100	–
M-7	1	0.4	≥ 100	4.8	1	0.4	≥ 100	58.0	5	88	≥ 100	–	5	91	≥ 100	–
Heavy-traffic																
H-1	0	0	0.1	0.1	0	0	≥ 100	≥ 100	0	0	≥ 100	≥ 100	0	0	≥ 100	≥ 100
H-2	0	0	0.5	0.5	0	0	≥ 100	≥ 100	5	15.2	≥ 100	–	5	23.4	≥ 100	–
H-3	0	0	1.8	1.8	1	0.2	≥ 100	78.0	5	37	≥ 100	–	5	35.6	≥ 100	–
H-4	1	0.2	≥ 100	4.7	1	0.2	≥ 100	95.1	5	55.2	≥ 100	–	5	56.4	≥ 100	–
H-5	0	0	7.7	7.7	0	0	≥ 100	≥ 100	5	70	≥ 100	–	5	71.8	≥ 100	–
H-6	1	0.4	≥ 100	4.4	1	0.4	≥ 100	87.3	5	84	≥ 100	–	5	88.6	≥ 100	–
H-7	0	0	4.2	4.2	1	0.2	≥ 100	88.1	5	98.4	≥ 100	–	5	≥ 100	≥ 100	–

From Table 3.8, we observe that the #I and #R values of the FCFS policy are comparable to those of the Lagrangian relaxation heuristic, and are significantly smaller than those of the STW and RS policies. This is because berth planners typically plan the incoming vessels' berthing times based on their arrival times, and plan the outgoing vessels' unberthing times based on their expected departure times. Thus, in the given data, the latest allowed berthing times of the incoming vessels are positively correlated with their arrival times, and the unberthing times of the outgoing vessels are positively correlated with their expected departure times. Since the FCFS policy sequences the incoming and outgoing vessels based on the their arrival and unberthing times, respectively, it requires less anchorage capacity for the vessels than the other policies. Hence, it is relatively easy for the FCFS policy to satisfy all service requests. However, since these vessel sequencing policies ignore the utilization of the staging anchorages when they sequence the vessels, the staging anchorages tend to be utilized ineffectively in the solutions generated by these policies. Therefore, as indicated by the G2 values in Table 3.8, the performances of the FCFS, STW, and RS policies in minimizing the berthing and departure tardiness of vessels are worse than that of the Lagrangian relaxation heuristic. Overall, the computational results indicate that integrating the management of the anchorage area utilization into the decision for channel traffic control can significantly improve the vessel service.

3.4 Extensions

In this section, we discuss extensions of problem **P** that may arise in practical operations, and propose generalizations of our model and solution approaches for handling generalized problems.

3.4.1 Multiple Navigation Channels

Some seaports have more than one navigation channel, such as the Waigaoqiao Port in Shanghai (Lalla-Ruiz, Shi, and Voß 2016). In these seaports, incoming and outgoing vessels may utilize any of the navigation channels. In this scenario, an additional decision that needs to be made is to determine which navigation channel should be assigned to each vessel. Denote p as the number of navigation channels, where each channel has a one-way traffic lane for each direction. For each $i = 1, \ldots, n_1 + n_2$ and $q = 1, \ldots, p$, denote u_{iq} and $[\underline{w}_{iql}, \bar{w}_{iql}]$ as the number of tidal windows of vessel i and the lth tidal window of vessel i, respectively, in the qth navigation channel. For each $q = 1, \ldots, p$, denote $\bar{\tau}_q$ as the amount of time for a vessel to travel through the qth navigation channel. Let x_{iqt} be the decision variable that takes value 1 if vessel i enters navigation channel q at time t, and takes value 0 otherwise. In the MILP of problem **P**, constraints (3.4), (3.6), (3.8), (3.11), and (3.18) should be replaced by the following constraints:

$$\sum_{q=1}^{p} \sum_{t=0}^{T} x_{iqt} = 1 - U_i \quad (i = 1, \ldots, n_1 + n_2);$$

$$x_{iqt} = 0 \quad (i = 1, \ldots, n_1 + n_2;\ q = 1, \ldots, p;\ t \in \{0, 1, \ldots, T\} \setminus \bigcup_{l=1}^{u_{iq}} [\underline{w}_{iql}, \bar{w}_{iql} - \bar{\tau}_q]);$$

$$e_i = \sum_{q=1}^{p} \sum_{t=0}^{T} (t + \bar{\tau}_q) x_{iqt} + \sum_{k=0}^{m} \tau'_{0k} y_{ik} \quad (i = 1, \ldots, n_1);$$

$$f_i = \sum_{q=1}^{p} \sum_{t=0}^{T} t x_{iqt} - \sum_{k=0}^{m} \tau'_{0k} y_{ik} \quad (i = n_1 + 1, \ldots, n_1 + n_2);$$

$$L_{2i} \geq \sum_{q=1}^{p} \sum_{t=0}^{T} (t + \bar{\tau}_q) x_{iqt} - D_i (1 - U_i) \quad (i = n_1 + 1, \ldots, n_1 + n_2).$$

In addition, in constraints (3.2)–(3.3), (3.5), and (3.19), variable x_{it} should be replaced by x_{iqt}, and parameter $\bar{\tau}$ should be replaced by $\bar{\tau}_q$. These constraints should also be imposed for each $q = 1, \ldots, p$. In the Lagrangian relaxation heuristic, problems $\mathbf{P}'_{\text{in}}$ and $\mathbf{P}'_{\text{out}}$ should be modified as follows:

$$\begin{aligned}
\mathbf{P}'_{\text{in}}(\lambda):\ \text{minimize} \quad & \sum_{i=1}^{n_1} \sum_{q=1}^{p} \sum_{t=0}^{T+n_1} F_{iqt}(\lambda) x_{iqt} \\
\text{subject to} \quad & \sum_{i=1}^{n_1} x_{iqt} \leq 1 \quad (q = 1, \ldots, p;\ t = 0, 1, \ldots, T + n_1) \\
& \sum_{q=1}^{p} \sum_{t=0}^{T+n_1} x_{iqt} = 1 \quad (i = 1, \ldots, n_1) \\
& x_{iqt} \in \{0, 1\} \quad (i = 1, \ldots, n_1;\ q = 1, \ldots, p;\ t = 0, 1, \ldots, T + n_1)
\end{aligned}$$

and

$$\begin{aligned}
\mathbf{P}'_{\text{out}}(\lambda):\ \text{minimize} \quad & \sum_{i=n_1+1}^{n_1+n_2} \sum_{q=1}^{p} \sum_{t=0}^{T+n_2} G_{iqt}(\lambda) x_{iqt} \\
\text{subject to} \quad & \sum_{i=n_1+1}^{n_1+n_2} x_{iqt} \leq 1 \quad (q = 1, \ldots, p;\ t = 0, 1, \ldots, T + n_2) \\
& \sum_{q=1}^{p} \sum_{t=0}^{T+n_2} x_{iqt} = 1 \quad (i = n_1 + 1, \ldots, n_1 + n_2) \\
& x_{iqt} \in \{0, 1\} \quad (i = n_1 + 1, \ldots, n_1 + n_2;\ q = 1, \ldots, p;\ t = 0, 1, \ldots, T + n_2)
\end{aligned}$$

where $F_{iqt}(\lambda)$ and $G_{iqt}(\lambda)$ are defined by replacing u_i with u_{iq}, replacing $\bar{\tau}$ with $\bar{\tau}_q$, and replacing $[\underline{w}_{il}, \bar{w}_{il} - \bar{\tau}]$ with $[\underline{w}_{iql}, \bar{w}_{iql} - \bar{\tau}_q]$ in functions $F_{it}(\lambda)$ and $G_{it}(\lambda)$, respectively. Problems $\mathbf{P}'_{\text{in}}(\lambda)$ and $\mathbf{P}'_{\text{out}}(\lambda)$ are now $n_1 \times (T + n_1 + 1)p$ and $n_2 \times (T + n_2 + 1)p$ asymmetric assignment problems, respectively, and can be solved in pseudo-polynomial time.

3.4.2 Heterogeneous Staging Anchorages

In problem $\mathbf{P}$, each staging anchorage can accommodate exactly one vessel. In reality, however, the capacities of the staging anchorages may be different. In such case, the number of vessels that a staging anchorage can accommodate depends on its capacity. For each $k = 1\ldots, m$, denote s_k as the capacity of staging anchorage k. For each $i = 1, \ldots, n_1 + n_2$, denote u_i as the size of vessel i. The problem with heterogeneous staging anchorages can be modeled by adding the constraint

$$y_{ik} = 0 \quad (i = 1, \ldots, n_1 + n_2;\ k = 1, \ldots, m;\ u_i > s_k),$$

and replacing constraint (3.13) with the constraint

$$\textstyle\sum_{i=1}^{n_1+n_2} u_i z_{ikt} \le s_k \quad (k = 1, \ldots, m;\ t = 0, 1, \ldots, T). \tag{3.63}$$

In the Lagrangian relaxation heuristic, after relaxing constraint (3.63), the objective function of problem $\mathbf{P}(\lambda)$ becomes

$$\text{minimize}\ \ \textstyle\sum_{i=1}^{n_1} C_{1i}L_{1i} + \sum_{i=n_1+1}^{n_1+n_2} C_{2i}L_{2i} + \sum_{i=1}^{n_1+n_2} C_{0i}U_i + \sum_{k=1}^{m}\sum_{t=0}^{T} \lambda_{kt}(\sum_{i=1}^{n_1+n_2} u_i z_{ikt} - s_k).$$

When solving subproblems $\mathbf{P}'_{\text{in}}(\lambda)$ and $\mathbf{P}'_{\text{out}}(\lambda)$, the cost functions $F_{it}^{(k)}(\lambda)$ and $G_{it}^{(k)}(\lambda)$ are modified as follows:

$$F_{it}^{(k)}(\lambda) = \begin{cases} C_{1i}(t + \bar{\tau} + \tau_{0,b(i)} - B_i), & \text{if } k = 0, \text{ and } B_i \le t + \bar{\tau} + \tau_{0,b(i)} \le \bar{B}_i; \\ C_{1i}(t+\bar{\tau}+\tau'_{0k}+\tau''_{k,b(i)}-B_i)+u_i\lambda_{k,t+\bar{\tau}+\tau'_{0k}}, & \text{if } k>0,\ u_i \le s_k, \text{ and } B_i \le t+\bar{\tau}+\tau'_{0k}+\tau''_{k,b(i)} \le \bar{B}_i; \\ u_i \sum_{t'=t+\bar{\tau}+\tau'_{0k}}^{B_i-\tau''_{k,b(i)}} \lambda_{kt'}, & \text{if } k > 0,\ u_i \le s_k, \text{ and } t + \bar{\tau} + \tau'_{0k} + \tau''_{k,b(i)} < B_i; \\ +\infty, & \text{otherwise} \end{cases}$$

$(i = 1, \ldots, n_1;\ k = 0, 1, \ldots, m;\ t = 0, 1, \ldots, T + n_1)$;

$$G_{it}^{(k)}(\lambda) = \begin{cases} C_{2i} \max\{t + \bar{\tau} - D_i,\ 0\}, & \text{if } k = 0, \text{ and } t = E_i + \tau_{0,b(i)}; \\ C_{2i} \max\{t+\bar{\tau}-D_i,\ 0\}+u_i \sum_{t'=E_i+\tau''_{k,b(i)}}^{t-\tau'_{0k}} \lambda_{kt'}, & \text{if } k>0,\ u_i \le s_k, \text{ and } t \ge E_i+\tau''_{k,b(i)}+\tau'_{0k}; \\ +\infty, & \text{otherwise} \end{cases}$$

$(i = n_1 + 1, \ldots, n_1 + n_2;\ k = 0, 1, \ldots, m;\ t = 0, 1, \ldots, T + n_2)$.

Note that problem $\mathbf{P}_{\text{ub}}$ is no longer valid for modeling the upper bound problem as the heterogeneity of the staging anchorages' capacities are not taken into account in the MILP of problem

$\mathbf{P}_{\text{ub}}$. Hence, a new upper bound method should be devised for generating feasible solutions of problem $\mathbf{P}$. Given the values of the x_{it} variables in a lower bound solution, one possible way for constructing a feasible solution is to plug all the x_{it} values into the updated MILP of problem $\mathbf{P}$ and solve the MILP.

3.4.3 Flexible Unberthing Times of Outgoing Vessels

If the solution of problem $\mathbf{P}$ contains some unsatisfied service requests of outgoing vessels, the terminal operators will need to adjust their berth plans. One frequently occurred adjustment is to delay the outgoing vessels' unberthing times. Such delay in unberthing is often possible but undesirable as it could reduce the berth productivity (a berth is not productive when an outgoing vessel is waiting for unberthing at the berth). To resolve this issue, an alternative model would be to allow an outgoing vessel i to stay at berth $b(i)$ beyond time E_i at a cost. The alternative model that allows such flexibility can be described as follows. Each outgoing vessel i is allowed to unberth within a time window $[E_i, \bar{E}_i]$, where $\bar{E}_i$ is the deadline for unberthing. An unberthing tardiness cost of $C_{3i} > 0$ per unit time is incurred when vessel i is unberthed after time E_i. Let h_i be the actual unberthing time and L_{3i} be the unberthing tardiness of vessel i. Then, the MILP of problem $\mathbf{P}$ can be modified by changing the objective function to

$$\textstyle\sum_{i=1}^{n_1} C_{1i}L_{1i} + \sum_{i=n_1+1}^{n_1+n_2} (C_{2i}L_{2i} + C_{3i}L_{3i}) + \sum_{i=1}^{n_1+n_2} C_{0i}U_i,$$

and replacing constraint (3.10) by the following constraints:

$$\begin{aligned}
&e_i = \textstyle\sum_{k=0}^{m}(h_i + \tau''_{k,b(i)})y_{ik} \quad (i = n_1+1, \ldots, n_1+n_2);\\
&E_i \le h_i \le \bar{E}_i \quad (i = n_1+1, \ldots, n_1+n_2);\\
&L_{3i} \ge h_i - E_i \quad (i = n_1+1, \ldots, n_1+n_2);\\
&L_{3i} \ge 0 \quad (i = n_1+1, \ldots, n_1+n_2).
\end{aligned}$$

These changes should also be made to problem $\mathbf{P}_{\text{ub}}$ when deriving the upper bound solution. When solving subproblem $\mathbf{P}'_{\text{out}}(\lambda)$, the cost function $G_{it}^{(k)}(\lambda)$ should be modified as follows:

$$G_{it}^{(k)}(\lambda) = \begin{cases} C_{2i}\max\{t+\bar{\tau}-D_i,\, 0\} + C_{3i}(t-E_i-\tau_{0,b(i)}), & \text{if } k=0 \text{ and } E_i+\tau_{0,b(i)} \le t \le \bar{E}_i+\tau_{0,b(i)}; \\ C_{2i}\max\{t+\bar{\tau}-D_i,\, 0\} & \\ \quad +\min_{\theta=E_i,\ldots,\min\{\bar{E}_i,t-\tau'_{0k}-\tau''_{k,b(i)}\}}\{\Gamma_{ikt}(\theta)\}, & \text{if } k>0 \text{ and } t \ge E_i+\tau''_{k,b(i)}+\tau'_{0k}; \\ +\infty, & \text{otherwise;} \end{cases}$$

where

$$\Gamma_{ikt}(\theta) = C_{3i}(\theta - E_i) + \sum_{t'=\theta+\tau''_{k,b(i)}}^{t-\tau'_{0k}} \lambda_{kt'}.$$

In this equation, θ is the unberthing time of outgoing vessel i. Note that solving the modified Lagrangian relaxation subproblem $\mathbf{P}'_{\text{out}}(\lambda)$ requires more computational time than solving the original subproblem, because in the modified subproblem a search of the θ value among $E_i, E_i+1, \ldots, \min\{\bar{E}_i, t-\tau'_{0k}-\tau''_{k,b(i)}\}$ is required when determining each $G_{it}^{(k)}(\lambda)$ value.

Chapter 4

Conclusions and Suggestions for Future Research

This thesis studies two optimization problems for container port congestion mitigation. The first problem aims to optimize the berth allocation of deep-sea vessels and schedule arrivals of feeders by taking into account a queue length restriction of feeders, and the second problem aims to scheduling the vessel traffic in the port waters by optimizing the utilization of the navigation channel and the anchorage areas.

For the problem of allocating berths to deep-sea vessels and scheduling the arrivals of feeders, we develop a stochastic optimization model to account for the uncertainties of feeder service times. We solve the stochastic optimization model using a simulation optimization method that searches the solution space via a global phase, a local phase, and a clean-up phase, and allocates different amounts of simulation budget to different search phases in order to balance the effort spent on solution evaluation and solution sampling. We generate problem instances based on the operational data of the Yangshan Deep-water Port in Shanghai, and compare the computational performance of the simulation optimization method with those of three benchmark methods that share the same solution sampling strategy but are assigned different amounts of simulation budget for solution evaluation. The computational results indicate that for the solution sampling strategy employed by the benchmark methods, allocating a larger simulation budget to solution evaluation improves the capability of generating good solutions but leads to considerably stronger demand for computation effort, and vice versa. On the other hand, our simulation optimization method finds good solutions with a reasonable amount of computation effort and thus outperforms the benchmark methods. We also compare the solutions obtained by the simulation optimization method with those obtained by a sequential decision heuristic that mimics the current practice of port operators. The comparisons show that the simulation optimization method provides the flexibility of allocating berth space to deep-sea vessels and scheduling the arrivals of feeders under different queue length limits, whereas the sequential decision heuristic is incapable of controlling the queue length of feeders. Hence, the simulation optimization method would be a promising decision support tool for berth allocation and congestion mitigation in a port that serves a large number of feeders.

For the problem of scheduling channel traffic and anchorage area utilization, we develop a MILP model for the problem and show that the problem is strongly NP-hard. We also show that after relaxing the anchorage capacity constraint, the resulting subproblems can be transformed into two asymmetric assignment problems that are solvable in pseudo-polynomial time. Based on this observation, we develop a Lagrangian relaxation heuristic for the problem. We generate test instances based on the layout and operational data of the Yangshan Deep-Water Port in Shanghai, and compare the computational performance of the Lagrangian relaxation heuristic with those of CPLEX and a rule-based heuristic that mimics the current practice of a VTS operator. The computational results indicate that the manual decision process for scheduling vessel traffic often results in large tardiness penalties and a large number of unsatisfied service requests, while CPLEX finds optimal solutions within reasonable computation times only for the smallest instances. On the other hand, the Lagrangian relaxation heuristic provides high-quality solutions in relatively short computation times for instances with planning horizons varying from one day to one week. Therefore, the Lagrangian relaxation heuristic achieves a good balance between effectiveness and efficiency, and thus can serve as a decision support tool for congestion mitigation and improving vessel service levels at container ports. Furthermore, we computationally test the performances of various vessel sequencing policies by ignoring the anchorage capacity, and find that the solutions obtained by ignoring the anchorage capacity incur significantly larger penalty costs than the solutions obtained by the Lagrangian relaxation heuristic. This indicates the importance of considering the anchorage capacity when scheduling the navigation channel traffic.

Future research could focus on improving the performance of the proposed solution methods and extending the current models and solution methods for solving a wide range of problems.

In the decoding scheme described in Section 2.2.1, the vessels are considered sequentially when assigning berths, and each deep-sea vessel is assigned an earliest feasible service start time. However, it may be possible to delay the berthing of a deep-sea vessel without increasing its tardiness, so that the tardiness of the subsequent vessel can be reduced. To achieve better assignment of service start times for the deep-sea vessels, a more sophisticated algorithm should be embedded into the decoding scheme. The development of such algorithm is left for future research. Our simulation-optimization method can be adapted for solving a variety of port operation management problems with uncertainties. For example, in a yard crane scheduling problem, the yard cranes need to serve container trucks that pick up containers from or deliver containers to the yard. Since the arrival times of the container trucks are uncertain to the port operator, it would be important to take into account the stochastic arrivals of container trucks when scheduling the yard cranes. To solve different port operation management problems with uncertainties, the solution sampling strategy used in the global phase and the local phase can be tailored to the particular problem in order to explore the solution space efficiently. The simulation budget allocation strategy (i.e., allocating different amounts of simulation budget to solutions in different search phases) will therefore still be useful for enhancing the computational efficiency. A variety of extensions on the channel traffic and anchorage utilization management problem can be investigated in future research. For instance, vessels are usually guided by tugboats when sailing in or out of a seaport (see, e.g., Ilati, Sheikholeslami, and Hassannayebi 2014). When the availability of tugboats is limited, decisions for scheduling the tugboats should be integrated into the channel traffic and anchorage utilization management problem. Some seaports have navigation channels with single traffic lane. Those seaports are often highly congested as incoming and outgoing vessels share the same traffic lane when traveling through a navigation channel. Modeling and solving a problem with incoming and outgoing vessels sharing a single traffic lane could be a future research topic. Another interesting research direction could be to develop and analyze a model that integrates the decisions for berth allocation, navigation channel traffic, and staging anchorage utilization, since such integration could significantly improve port performance in terms of throughput enhancement and congestion mitigation. However, as mentioned in Section 1.2, berth allocation decisions are made by terminal operators whereas vessel traffic is regulated by VTS operators. Thus, a model that integrates berth allocation and vessel traffic management is applicable only if there is an authority that could centralize the decision making for both terminal operations and vessel traffic regulations.

References

Alvarez JF, Longva T, Engebrethsen ES (2010) A methodology to assess vessel berthing and speed optimization policies. *Maritime Economics & Logistics* 12(4):327–346.

Amaran S, Sahinidis NV, Sharda B, Bury SJ (2016) Simulation optimization: A review of algorithms and applications. *Annals of Operations Research* 240(1):351–380.

Arango C, Cortés P, Muñuzuri J, Onieva L (2011) Berth allocation planning in Seville inland port by simulation and optimisation. *Advanced Engineering Informatics* 25(3):452–461.

Arango C, Cortés P, Onieva L, Escudero A (2013) Simulation-optimization models for the dynamic berth allocation problem. *Computer-Aided Civil and Infrastructure Engineering* 28(10):769–779.

Barros VH, Costa TS, Oliveira ACM, Lorena LAN (2011) Model and heuristic for berth allocation in tidal bulk ports with stock level constraints. *Computers & Industrial Engineering* 60(4):606–613.

Bierwirth C, Meisel F (2010) A survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research* 202(3):615–627.

Bierwirth C, Meisel F (2015) A follow-up survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research* 244(3):675–689.

Boesel J, Nelson BL, Kim S-H (2003) Using ranking and selection to "clean up" after simulation optimization. *Operations Research* 51(5):814–825.

Cordeau J-F, Laporte G, Legato P, Moccia L (2005) Models and tabu search heuristics for the berth-allocation problem. *Transportation Science* 39(4):526–538.

Dadashi A, Dulebenets MA, Golias MM, Sheikholeslami A (2017) A novel continuous berth scheduling model at multiple marine container terminals with tidal considerations. *Maritime Business Review* 2(2):142–157.

De A, Pratap S, Kumar A, Tiwari MK (2018) A hybrid dynamic berth allocation planning problem with fuel costs considerations for container terminal port using chemical reaction optimization approach. *Annals of Operations Research*, forthcoming, https://doi.org/10.1007/s10479-018-3070-1.

Ding Y, Jia S, Gu T, Li C-L (2016) SGICT builds an optimization-based system for daily berth planning. *Interfaces* 46(4):281–296.

Dragović B, Park NK, Radmilović Z (2006) Ship-berth link performance evaluation: Simulation and analytical approaches. *Maritime Policy & Management* 33(3):281–299.

Dragović B, Park NK, Radmilović Z, Maraš V (2005) Simulation modelling of ship-berth link with priority service. *Maritime Economics & Logistics* 7(4):316–335.

Du Y, Chen Q, Lam JSL, Xu Y, Cao JX (2015) Modeling the impacts of tides and the virtual arrival policy in berth allocation. *Transportation Science* 49(4):939–956.

Emde S, Boysen N (2016) Berth allocation in container terminals that service feeder ships and deep-sea vessels. *Journal of the Operational Research Society* 67(4):551–563.

Fisher ML (2004) The Lagrangian relaxation method for solving integer programming problems. *Management Science* 50(12):1861–1871.

Fu MC, Glover FW, April J (2005) Simulation optimization: A review, new developments, and applications. Kuhl ME, Steiger NM, Armstrong FB, Joines JA, eds., *Proceedings of the 2005 Winter simulation conference, Orlando, FL,* 83–95.

Garey MR, Johnson DS (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Freeman, New York).

Gharehgozli AH, Roy D, de Koster R (2016) Sea container terminals: New technologies and OR models. *Maritime Economics & Logistics* 18(2):103–140.

Golias MM (2011) A bi-objective berth allocation formulation to account for vessel handling time uncertainty. *Maritime Economics & Logistics* 13(4):419–441.

Golias MM, Saharidis GK, Boile M, Theofanis S, Ierapetritou MG (2009) The berth allocation problem: Optimizing vessel arrival time. *Maritime Economics & Logistics* 11(4):358–377.

Han X-L, Lu Z-Q, Xi L-F (2010) A proactive approach for simultaneous berth and quay crane scheduling problem with stochastic arrival and handling time. *European Journal of Operational Research* 207(3):1327–1340.

Held M, Wolfe P, Crowder HP (1974) Validation of subgradient optimization. *Mathematical Programming* 6:62–88.

Hill A, Lalla-Ruiz E, Voß S, Goycoolea M (2019) A multi-mode resource-constrained project scheduling reformulation for the waterway ship scheduling problem. *Journal of Scheduling* 22(2):173–182

Hong LJ, Nelson BL (2006) Discrete optimization via simulation using COMPASS. *Operations Research* 54(1):115–129.

Ilati G, Sheikholeslami A, Hassannayebi E (2014) A simulation-based optimization approach for integrated port resource allocation problem. *Promet – Traffic & Transportation* 26(3):243–255.

International Maritime Organization (1997) Resolution A.857(20): Guidelines for vessel traffic services. http://www.maritime-vts.co.uk/A857.pdf.

Karafa J, Golias MM, Ivey S, Saharidis GKD, Leonardos N (2013) The berth allocation problem with stochastic vessel handling times. *The International Journal of Advanced Manufacturing Technology* 65(1–4):473–484.

Kelareva E, Brand S, Kilby P, Thiébaux S, Wallace M (2012) CP and MIP methods for ship scheduling with time-varying draft. *Proceedings of the Twenty-Second International Conference on Automated Planning and Scheduling* (Sao Paulo, Brazil), 110–118.

Kelareva E, Tierney K, Kilby P (2013) CP methods for scheduling and routing with time-dependent task costs. *Lecture Notes in Computer Science* 7874:111–127.

Kim KH, Lee H (2015) Container terminal operation: Current trends and future challenges. Lee C-Y, Meng Q, eds. *Handbook of Ocean Container Transport Logistics* (Springer, Switzerland), 43–73.

Lalla Ruiz E, Melián Batista B, Moreno Vega JM (2013) Adaptive variable neighbourhood search for berth planning in maritime container terminals. *Proceedings of the Workshop on Constraint Satisfaction Techniques for Planning and Scheduling Problems* (Beijing, China), 35–43.

Lalla-Ruiz E (2017) Intelligent management of seaside logistic operations at maritime container terminals. *4OR* 15(2):217–218.

Lalla-Ruiz E, Expósito-Izquierdo C, Melián-Batista B, Moreno-Vega JM (2016) A set-partitioning-based model for the berth allocation problem under time-dependent limitations. *European Journal of Operational Research* 250(3):1001–1012.

Lalla-Ruiz E, Shi X, Voß S (2018) The waterway ship scheduling problem. *Transportation Research Part D* 60:191–209.

Lalla-Ruiz E, Voß S, Expósito-Izquierdo C, Melián-Batista B, Moreno-Vega JM (2017) A POPMUSIC-based approach for the berth allocation problem under time-dependent limitations. *Annals of Operations Research* 253:871–897.

Lang N, Veenstra A (2010) A quantitative analysis of container vessel arrival planning strategies. *OR Spectrum* 32(3):477–499.

Lee LH, Chew EP, Manikam P (2006) A general framework on the simulation-based optimization under fixed computing budget. *European Journal of Operational Research* 174(3):1828–1841.

Legato P, Mazza RM (2001) Berth planning and resources optimisation at a container terminal via discrete event simulation. *European Journal of Operational Research* 133(3):537–547.

Legato P, Mazza RM, Gullì D (2014) Integrating tactical and operational berth allocation decisions via Simulation-Optimization. *Computers & Industrial Engineering* 78:84–94

Li Q, Lam JSL (2017) Conflict resolution for enhancing shipping safety and improving navigational traffic within a seaport: Vessel arrival scheduling. *Transportmetrica A: Transport Science* 13(8):727–741.

Liu C, Xiang X, Zhang C, Zheng L (2016) A decision model for berth allocation under uncertainty considering service level using an adaptive differential evolution algorithm. *Asia-Pacific Journal of Operational Research* 33(6):1650049 (28 pages).

Liu C, Xiang X, Zheng L (2019) A two-stage robust optimization approach for the berth allocation problem under uncertainty. *Flexible Services and Manufacturing Journal*, forthcoming, https://doi.org/10.1007/s10696-019-09343-w

Moorthy R, Teo CP (2006) Berth management in container terminal: The template design problem. *OR Spectrum* 28(4):495-518.

Nauss RM (2008) Optimal sequencing in the presence of setup times for tow/barge traffic through a river lock. *European Journal of Operational Research* 187(3):1268–1281.

Passchyn W, Coene S, Briskorn D, Hurink JL, Spieksma FCR, Vanden Berghe G (2016) The lockmaster's problem. *European Journal of Operational Research* 251(2):432–441.

Petersen ER, Taylor AJ (1988) An optimal scheduling system for the Welland Canal. *Transportation Science* 22(3):173–185.

Qin T, Du Y, Sha M (2016) Evaluating the solution performance of IP and CP for berth allocation with time-varying water depth. *Transportation Research Part E* 87:167–185.

Radmilovich ZR (1992) Ship-berth link as bulk queueing system in ports. *Journal of Waterway, Port, Coastal, and Ocean Engineering* 118(5):474–495.

Sheikholeslami A, Ilati Gh, Kobari M (2014) The continuous dynamic berth allocation problem at a marine container terminal with tidal constraints in the access channel. *International Journal of Civil Engineering* 12(3):344–353.

Sluiman FJ (2017) Transit vessel scheduling. *Naval Research Logistics* 64(3):225–248.

Steenken D, Voß S, Stahlbock R (2004) Container terminal operation and operations research—a classification and literature review. *OR Spectrum* 26(1):3–49.

Tang G, Wang W, Song X, Guo Z, Yu X, Qiao F (2016) Effect of entrance channel dimensions on berth occupancy of container terminals. *Ocean Engineering* 117:174–187.

Ulusçu ÖS, Özbaş B, Altıok T, Or İ, Yılmaz T (2009) Transit vessel scheduling in the Strait of Istanbul. *The Journal of Navigation* 62(1):59–77.

UNCTAD (2017) Review of maritime transport 2017. *United Nations Conference on Trade and Development* (Geneva, Switzerland).

Ursavas E, Zhu SX (2016) Optimal policies for the berth allocation problem under stochastic nature. *European Journal of Operational Research* 255(2):380–387.

Verstichel J, De Causmaecker P, Spieksma F, Vanden Berghe G (2014) The generalized lock scheduling problem: An exact approach. *Transportation Research Part E* 65:16–34.

Xu D, Li C-L, Leung JY-T (2012) Berth allocation with time-dependent physical limitations on vessels. *European Journal of Operational Research* 216(1):47–56.

Xu J, Huang E, Chen C-H, Lee LH (2015) Simulation optimization: A review and exploration in the new era of cloud computing and big data. *Asia-Pacific Journal of Operational Research*

32(03):1550019 (34 pages).

Xu J, Nelson BL, Hong LJ (2010) Industrial strength COMPASS: A comprehensive algorithm and software for optimization via simulation. *ACM Transactions on Modeling and Computer Simulation* 20(1):3:1–3:29.

Xu J, Nelson BL, Hong LJ (2013) An adaptive hyperbox algorithm for high-dimensional discrete optimization via simulation problems. *INFORMS Journal on Computing* 25(1):133–146.

Yu S, Wang S, Zhen L (2017) Quay crane scheduling problem with considering tidal impact and fuel consumption. *Flexible Services and Manufacturing Journal* 29(3–4):345–368.

Zhang X, Lin J, Guo Z, Liu T (2016) Vessel transportation scheduling optimization based on channel-berth coordination. *Ocean Engineering* 112:145–152.

Zhen L, Lee LH, Chew EP (2011) A decision model for berth allocation under uncertainty. *European Journal of Operational Research* 212(1):54–68.

Zhen L, Liang Z, Zhuge D, Lee LH, Chow EP (2017) Daily berth planning in a tidal port with channel flow control. *Transportation Research Part B* 106:193–217.

Zrnić DN, Dragović BM, Radmilović ZR (1999) Anchorage-ship-berth link as multiple server queuing system. *Journal of Waterway, Port, Coastal, and Ocean Engineering* 125(5):232–240.

Appendix

Tables A1–A3 provide the detailed computational results discussed in Section 3.3.3, the MILP/CPLEX method, and the rule-based heuristic, where each row reports the results of each test instance. The #R and UB values obtained by these three methods, as well as the LB values obtained by the Lagrangian relaxation heuristic, are reported. The "G" columns report the optimality gaps of the heuristic solution values of each test instance, and the "Time" columns report the computational time (in seconds).

Tables A4–A6 provide the detailed computational results discussed in Section 3.3.5, where each row reports the results of each test instance. The #R and UB values obtained by these four vessel sequencing policies, as well as the LB values obtained by the Lagrangian relaxation heuristic, are reported. The "G" columns report the optimality gaps of the objective function values generated by the four policies, and the "Time" columns report the computational time (in seconds).

Table A1: Details of the computational results reported in Table 3.4 (low-traffic instances).

Problem set	Instance	Lagrangian Relaxation					MILP/CPLEX				Rule-based Heuristic			
		#R	UB	LB	G (%)	Time	#R	UB	G (%)	Time	#R	UB	G (%)	Time
L-1	1	0	138	138.0	0.0	0.1	0	138	0.0	14.8	0	140	1.4	0.0
	2	0	92	92.0	0.0	0.1	0	92	0.0	8.4	0	92	0.0	0.0
	3	0	102	102.0	0.0	0.1	0	102	0.0	11.6	0	102	0.0	0.0
	4	0	356	356.0	0.0	0.1	0	356	0.0	9.8	0	358	0.6	0.0
	5	0	91	91.0	0.0	0.1	0	91	0.0	12.6	2	20091	$\geq$ 100	0.0
L-2	1	0	672	668.0	0.6	0.2	0	668	0.0	287.6	0	673	0.7	0.0
	2	0	532	482.9	10.2	5.3	0	532	10.2	3600	1	10586	$\geq$ 100	0.0
	3	0	804	796.0	1.0	3.3	0	922	15.8	3600	0	893	12.2	0.0
	4	0	480	480.0	0.0	0.2	0	480	0.0	982.6	0	506	5.4	0.0
	5	0	544	535.2	1.7	6.6	0	544	1.7	3600	0	697	30.2	0.0
L-3	1	0	575	575.0	0.0	3.1	0	713	24.0	3600	0	722	25.6	0.0
	2	0	662	662.0	0.0	0.4	0	796	20.2	3600	0	730	10.3	0.0
	3	0	595	595.0	0.0	0.5	0	603	1.3	3600	0	657	10.4	0.0
	4	0	523	523.0	0.0	0.4	0	523	0.0	1442.1	0	523	0.0	0.0
	5	0	614	593.1	3.5	10.9	2	20638	$\geq$ 100	3600	0	744	25.4	0.0
L-4	1	0	804	804.0	0.0	0.8	1	10816	$\geq$ 100	3600	0	904	12.4	0.0
	2	0	861	861.0	0.0	1.0	0	861	0.0	3502.8	0	862	0.1	0.0
	3	0	816	816.0	0.0	0.8	1	11058	$\geq$ 100	3600	0	1004	23.0	0.0
	4	0	579	579.0	0.0	0.8	0	655	13.1	3600	0	753	30.1	0.0
	5	0	886	886.0	0.0	0.8	3	30840	$\geq$ 100	3600	1	10970	$\geq$ 100	0.0
L-5	1	0	1331	1321.5	0.7	19.3	7	71497	$\geq$ 100	3600	1	11518	$\geq$ 100	0.0
	2	0	906	868.8	4.3	40.5	12	121366	$\geq$ 100	3600	0	1102	26.8	0.0
	3	0	895	891.0	0.4	1.4	7	71922	$\geq$ 100	3600	0	897	0.7	0.0
	4	0	1357	1345.0	0.9	1.7	3	31606	$\geq$ 100	3600	0	1359	1.0	0.0
	5	1	10985	968.2	$\geq$ 100	42.7	7	71766	$\geq$ 100	3600	2	21175	$\geq$ 100	0.0
L-6	1	0	903	903.0	0.0	20.2	83	832151	$\geq$ 100	3600	0	1035	14.6	0.0
	2	0	1260	1259.0	0.1	2.7	12	121986	$\geq$ 100	3600	1	11363	$\geq$ 100	0.0
	3	0	1454	1408.8	3.2	71.9	9	92426	$\geq$ 100	3600	0	1813	28.7	0.0
	4	0	1973	1930.3	2.2	86.3	18	183727	$\geq$ 100	3600	1	12308	$\geq$ 100	0.0
	5	0	1516	1508.0	0.5	2.9	12	123402	$\geq$ 100	3600	0	1806	19.8	0.0
L-7	1	1	12059	2063.3	$\geq$ 100	115.2	98	981470	$\geq$ 100	3600	0	2668	29.3	0.0
	2	1	12134	2070.4	$\geq$ 100	103.8	95	952787	$\geq$ 100	3600	3	32475	$\geq$ 100	0.0
	3	0	1699	1651.7	2.9	95.8	79	795146	$\geq$ 100	3600	0	1940	17.5	0.0
	4	0	1538	1524.5	0.9	34.8	93	932379	$\geq$ 100	3600	0	1744	14.4	0.0
	5	0	2457	2448.6	0.3	26.0	49	495957	$\geq$ 100	3600	0	2741	11.9	0.0

Table A2: Details of the computational results reported in Table 3.4 (medium-traffic instances).

Problem set	Instance	Lagrangian Relaxation					MILP/CPLEX				Rule-based Heuristic			
		#R	*UB*	*LB*	G (%)	Time	#R	*UB*	G (%)	Time	#R	*UB*	G (%)	Time
M-1	1	0	300	300.0	0.0	0.1	0	300	0.0	11.1	0	362	20.7	0.0
	2	0	297	297.0	0.0	0.1	0	297	0.0	10.1	0	297	0.0	0.0
	3	0	280	280.0	0.0	0.1	0	280	0.0	20.5	0	280	0.0	0.0
	4	0	159	159.0	0.0	0.1	0	159	0.0	13.4	0	161	1.3	0.0
	5	0	272	272.0	0.0	0.1	0	272	0.0	25.5	0	272	0.0	0.0
M-2	1	0	270	270.0	0.0	0.3	0	340	25.9	3600	1	10298	$\geq$ 100	0.0
	2	0	522	522.0	0.0	0.3	0	522	0.0	2530.3	0	694	33.0	0.0
	3	0	595	595.0	0.0	0.4	0	595	0.0	3562.6	1	10790	$\geq$ 100	0.0
	4	0	597	597.0	0.0	0.3	0	597	0.0	1369.7	0	598	0.2	0.0
	5	0	687	627.1	9.6	7.5	0	729	16.3	3600	0	818	30.4	0.0
M-3	1	0	1188	1162.2	2.2	20.0	1	11642	$\geq$ 100	3600	0	1449	24.7	0.0
	2	0	718	663.9	8.2	14.9	0	880	32.6	3600	0	773	16.4	0.0
	3	0	983	958.3	2.6	16.2	0	993	3.4	3600	0	1009	5.1	0.0
	4	0	630	626.0	0.6	0.8	0	914	46.0	3600	0	758	21.1	0.0
	5	0	628	615.9	2.0	16.2	1	10756	$\geq$ 100	3600	0	648	5.2	0.0
M-4	1	0	1680	1539.9	9.1	45.1	5	52325	$\geq$ 100	3600	0	1893	22.9	0.0
	2	0	952	922.8	3.2	32.3	4	41735	$\geq$ 100	3600	0	1104	19.6	0.0
	3	0	1297	1269.7	2.2	48.2	7	71693	$\geq$ 100	3600	0	1694	33.4	0.0
	4	0	1504	1484.5	1.3	41.2	4	42162	$\geq$ 100	3600	0	1795	20.9	0.0
	5	0	1360	1359.0	0.1	5.3	4	41849	$\geq$ 100	3600	0	1842	35.5	0.0
M-5	1	0	1970	1884.1	4.6	86.6	9	92438	$\geq$ 100	3600	0	2611	38.6	0.0
	2	0	2067	2030.1	1.8	74.1	10	103114	$\geq$ 100	3600	0	2227	9.7	0.0
	3	0	1348	1174.7	14.8	64.4	9	92104	$\geq$ 100	3600	1	11598	$\geq$ 100	0.0
	4	0	1795	1779.0	0.9	4.6	5	52252	$\geq$ 100	3600	0	1922	8.0	0.0
	5	0	1097	1097.0	0.0	74.0	5	51596	$\geq$ 100	3600	0	1434	30.7	0.0
M-6	1	0	2046	2002.4	2.2	128.5	41	415137	$\geq$ 100	3600	0	2572	28.4	0.0
	2	0	1390	1390.0	0.0	125.7	26	264468	$\geq$ 100	3600	0	1745	25.5	0.0
	3	0	2255	2179.8	3.4	122.6	27	272941	$\geq$ 100	3600	1	12840	$\geq$ 100	0.0
	4	0	2125	2092.6	1.5	112.1	33	335243	$\geq$ 100	3600	0	2493	19.1	0.0
	5	0	1438	1436.0	0.1	4.5	15	152901	$\geq$ 100	3600	0	1718	19.6	0.0
M-7	1	0	2511	2262.0	11.0	175.6	101	1013989	$\geq$ 100	3600	0	3273	44.7	0.0
	2	0	2601	2535.1	2.6	189.2	65	655988	$\geq$ 100	3600	0	3349	32.1	0.0
	3	0	1352	1339.2	1.0	154.3	49	495803	$\geq$ 100	3600	0	1655	23.6	0.0
	4	0	2297	2191.9	4.8	189.5	110	1103285	$\geq$ 100	3600	0	2816	28.5	0.0
	5	2	22379	2819.2	$\geq$ 100	172.4	114	1142223	$\geq$ 100	3600	2	23206	$\geq$ 100	0.0

Table A3: Details of the computational results reported in Table 3.4 (heavy-traffic instances).

Problem set	Instance	Lagrangian Relaxation					MILP/CPLEX				Rule-based Heuristic			
		#R	UB	LB	G (%)	Time	#R	UB	G (%)	Time	#R	UB	G (%)	Time
H-1	1	0	232	232.0	0.0	0.1	0	232	0.0	16.1	2	20232	≥ 100	0.0
	2	0	86	86.0	0.0	0.1	0	86	0.0	34.3	0	90	4.7	0.0
	3	0	192	192.0	0.0	0.2	0	192	0.0	21.1	0	192	0.0	0.0
	4	0	149	148.0	0.7	0.2	0	148	0.0	153.7	0	160	8.1	0.0
	5	0	250	250.0	0.0	0.1	0	250	0.0	46.4	0	250	0.0	0.0
H-2	1	0	529	523.9	1.0	4.8	0	612	16.8	3600	0	614	17.2	0.0
	2	0	465	465.0	0.0	0.3	0	601	29.2	3600	0	554	19.1	0.0
	3	0	414	411.3	0.7	1.8	0	525	27.7	3600	0	597	45.2	0.0
	4	0	396	393.0	0.8	0.4	2	20615	≥ 100	3600	0	456	16.0	0.0
	5	0	341	341.0	0.0	0.4	0	341	0.0	490.5	0	407	19.4	0.0
H-3	1	0	1318	1314.0	0.3	29.5	2	21339	≥ 100	3600	0	1478	12.5	0.0
	2	0	1069	1063.6	0.5	25.3	0	1376	29.1	3600	1	11483	≥ 100	0.0
	3	0	1238	1232.0	0.5	0.9	5	52099	≥ 100	3600	1	11265	≥ 100	0.0
	4	0	1010	1010.0	0.0	26.2	3	31132	≥ 100	3600	0	1245	23.3	0.0
	5	0	1010	936.3	7.9	22.3	1	11240	≥ 100	3600	0	1266	35.2	0.0
H-4	1	0	1253	1251.0	0.2	1.7	3	31339	≥ 100	3600	0	1502	20.1	0.0
	2	0	2111	2046.2	3.2	56.9	7	71841	≥ 100	3600	0	2283	11.6	0.0
	3	1	11445	1448.8	≥ 100	72.0	10	101673	≥ 100	3600	1	12523	≥ 100	0.0
	4	0	873	869.0	0.5	2.1	1	11288	≥ 100	3600	0	1394	60.4	0.0
	5	0	1361	1181.9	15.2	64.8	9	91836	≥ 100	3600	5	51844	≥ 100	0.0
H-5	1	0	1277	1273.0	0.3	3.7	20	202927	≥ 100	3600	0	1396	9.7	0.0
	2	0	1635	1527.4	7.0	111.7	59	594185	≥ 100	3600	0	2312	51.4	0.0
	3	0	1791	1615.6	10.9	121.0	11	113108	≥ 100	3600	1	12312	≥ 100	0.0
	4	0	2325	2102.0	10.6	177.8	12	123674	≥ 100	3600	0	2876	36.8	0.0
	5	0	1823	1661.0	9.8	119.9	12	124542	≥ 100	3600	0	2404	44.7	0.0
H-6	1	0	2345	2337.4	0.3	56.1	30	305530	≥ 100	3600	3	32701	≥ 100	0.0
	2	0	2034	2014.0	1.0	26.1	87	874460	≥ 100	3600	0	2695	33.8	0.0
	3	0	1520	1500.1	1.3	185.4	160	1600637	≥ 100	3600	4	42021	≥ 100	0.0
	4	2	22610	2630.6	≥ 100	202.3	89	896332	≥ 100	3600	2	23521	≥ 100	0.0
	5	0	2723	2366.6	15.1	197.4	91	914753	≥ 100	3600	0	3500	47.9	0.0
H-7	1	0	2247	2167.7	3.7	269.0	128	1282685	≥ 100	3600	1	12578	≥ 100	0.0
	2	0	2816	2635.9	6.8	275.2	182	1820975	≥ 100	3600	1	13455	≥ 100	0.0
	3	0	2725	2638.8	3.3	334.4	143	1432143	≥ 100	3600	0	3788	43.6	0.0
	4	0	1878	1789.4	4.9	267.6	106	1065635	≥ 100	3600	1	12555	≥ 100	0.0
	5	0	1985	1942.3	2.2	219.2	111	1113555	≥ 100	3600	0	2402	23.7	0.0

Table A4: Details of the computational results reported in Table 3.8 (low-traffic instances).

Problem set	Instance	LB	Lagrangian Relaxation				FCFS				STW				RS			
			#R	UB	G (%)	Time	#R	UB	G (%)	Time	#R	UB	G (%)	Time	#R	UB	G (%)	Time
L-1	1	138.0	0	138	0.0	0.1	0	270	95.7	0.0	0	322	≥ 100	0.0	0	156	13.0	0.0
	2	92.0	0	92	0.0	0.1	0	354	≥ 100	0.0	0	288	≥ 100	0.0	0	368	≥ 100	0.0
	3	102.0	0	102	0.0	0.1	0	188	84.3	0.0	0	236	≥ 100	0.0	0	157	53.9	0.0
	4	356.0	0	356	0.0	0.1	0	518	45.5	0.0	0	534	50.0	0.0	0	457	28.4	0.0
	5	91.0	0	91	0.0	0.1	0	102	12.1	0.0	0	159	74.7	0.0	0	173	90.1	0.0
L-2	1	668.0	0	672	0.6	0.2	0	1272	90.4	0.0	12	121221	≥ 100	0.0	17	170669	≥ 100	0.0
	2	482.9	0	532	10.2	5.3	0	613	26.9	0.0	12	120476	≥ 100	0.0	18	180476	≥ 100	0.0
	3	796.0	0	804	1.0	3.3	0	1111	39.6	0.0	12	121088	≥ 100	0.0	16	161064	≥ 100	0.0
	4	480.0	0	480	0.0	0.2	0	564	17.5	0.0	10	100822	≥ 100	0.0	19	190480	≥ 100	0.0
	5	535.2	0	544	1.7	6.6	0	657	22.8	0.0	0	1986	≥ 100	0.0	11	111093	≥ 100	0.0
L-3	1	575.0	0	575	0.0	3.1	0	897	56.0	0.0	18	181217	≥ 100	0.0	27	270705	≥ 100	0.0
	2	662.0	0	662	0.0	0.4	0	885	33.7	0.0	15	151692	≥ 100	0.0	27	270702	≥ 100	0.0
	3	595.0	0	595	0.0	0.5	0	881	48.1	0.0	24	240921	≥ 100	0.0	22	220879	≥ 100	0.1
	4	523.0	0	523	0.0	0.4	0	722	38.0	0.0	22	220603	≥ 100	0.0	23	231051	≥ 100	0.0
	5	593.1	0	614	3.5	10.9	0	875	47.5	0.0	22	220760	≥ 100	0.0	27	270724	≥ 100	0.0
L-4	1	804.0	0	804	0.0	0.8	0	1421	76.7	0.0	35	350892	≥ 100	0.0	37	371264	≥ 100	0.1
	2	861.0	0	861	0.0	1.0	0	1134	31.7	0.0	37	370966	≥ 100	0.0	36	360996	≥ 100	0.1
	3	816.0	0	816	0.0	0.8	0	1015	24.4	0.0	32	321124	≥ 100	0.0	37	371142	≥ 100	0.0
	4	579.0	0	579	0.0	0.8	0	790	36.4	0.0	37	370581	≥ 100	0.0	39	390581	≥ 100	0.1
	5	886.0	0	886	0.0	0.8	0	1304	47.2	0.0	36	361002	≥ 100	0.0	40	400925	≥ 100	0.3
L-5	1	1321.5	0	1331	0.7	19.3	0	1612	22.0	0.0	49	491326	≥ 100	0.0	53	531326	≥ 100	0.2
	2	868.8	0	906	4.3	40.5	0	1525	75.5	0.0	47	470926	≥ 100	0.0	48	481032	≥ 100	0.1
	3	891.0	0	895	0.4	1.4	0	1536	72.4	0.0	43	431403	≥ 100	0.0	48	481022	≥ 100	0.1
	4	1345.0	0	1357	0.9	1.7	0	1770	31.6	0.0	52	521347	≥ 100	0.0	53	531585	≥ 100	0.2
	5	968.2	1	10985	≥ 100	42.7	0	1537	58.7	0.0	40	401819	≥ 100	0.0	53	530961	≥ 100	0.1
L-6	1	903.0	0	903	0.0	20.2	0	1768	95.8	0.0	54	541325	≥ 100	0.0	62	620903	≥ 100	0.2
	2	1259.0	0	1260	0.1	2.7	0	1942	54.2	0.0	56	561445	≥ 100	0.0	64	641263	≥ 100	0.2
	3	1408.8	0	1454	3.2	71.9	0	2603	84.8	0.0	63	631375	≥ 100	0.0	63	631375	≥ 100	0.1
	4	1930.3	0	1973	2.2	86.3	0	3055	58.3	0.0	64	642270	≥ 100	0.0	66	661928	≥ 100	0.2
	5	1508.0	0	1516	0.5	2.9	0	2049	35.9	0.0	58	581510	≥ 100	0.0	63	631514	≥ 100	0.4
L-7	1	2063.3	1	12059	≥ 100	115.2	0	3293	59.6	0.0	72	722063	≥ 100	0.0	74	742063	≥ 100	0.5
	2	2070.4	1	12134	≥ 100	103.8	1	13092	≥ 100	0.0	75	752043	≥ 100	0.0	76	762043	≥ 100	0.5
	3	1651.7	0	1699	2.9	95.8	2	22913	≥ 100	0.0	73	731663	≥ 100	0.0	73	731663	≥ 100	1.5
	4	1524.5	0	1538	0.9	34.8	0	2345	53.8	0.0	72	721524	≥ 100	0.0	75	751524	≥ 100	0.2
	5	2448.6	0	2457	0.3	26.0	0	3393	38.6	0.0	72	722450	≥ 100	0.0	76	762450	≥ 100	0.4

Table A5: Details of the computational results reported in Table 3.8 (medium-traffic instances).

Problem	Instance	LB	Lagrangian Relaxation				FCFS				STW				RS			
set			#R	UB	G (%)	Time	#R	UB	G (%)	Time	#R	UB	G (%)	Time	#R	UB	G (%)	Time
M-1	1	300.0	0	300	0.0	0.1	0	441	47.0	0.0	0	536	78.7	0.0	0	651	≥ 100	0.0
	2	297.0	0	297	0.0	0.1	0	413	39.1	0.0	0	407	37.0	0.0	0	624	≥ 100	0.0
	3	280.0	0	280	0.0	0.1	0	306	9.3	0.0	0	347	23.9	0.0	0	336	20.0	0.0
	4	159.0	0	159	0.0	0.1	0	232	45.9	0.0	0	228	43.4	0.0	0	692	≥ 100	0.0
	5	272.0	0	272	0.0	0.1	0	338	24.3	0.0	0	343	26.1	0.0	0	400	47.1	0.0
M-2	1	270.0	0	270	0.0	0.3	0	471	74.4	0.0	17	170484	≥ 100	0.0	20	200671	≥ 100	0.0
	2	522.0	0	522	0.0	0.3	0	715	37.0	0.0	0	1654	≥ 100	0.0	0	2403	≥ 100	0.0
	3	595.0	0	595	0.0	0.4	0	899	51.1	0.0	15	150596	≥ 100	0.0	22	220596	≥ 100	0.0
	4	597.0	0	597	0.0	0.3	0	1388	≥ 100	0.0	10	101433	≥ 100	0.0	21	210677	≥ 100	0.0
	5	627.1	0	687	9.6	7.5	0	964	53.7	0.0	18	181049	≥ 100	0.0	21	210678	≥ 100	0.0
M-3	1	1162.2	0	1188	2.2	20.0	0	1722	48.2	0.0	20	202210	≥ 100	0.0	35	351158	≥ 100	0.0
	2	663.9	0	718	8.2	14.9	0	884	33.2	0.0	32	320664	≥ 100	0.0	34	340664	≥ 100	0.1
	3	960.3	0	983	2.4	16.2	0	1741	81.3	0.0	27	271245	≥ 100	0.0	34	341109	≥ 100	0.0
	4	626.0	0	630	0.6	0.8	0	968	54.6	0.0	24	242026	≥ 100	0.0	35	350626	≥ 100	0.0
	5	615.9	0	628	2.0	16.2	0	961	56.0	0.0	28	280836	≥ 100	0.0	35	350612	≥ 100	0.0
M-4	1	1539.9	0	1680	9.1	45.1	0	2643	71.6	0.0	52	521519	≥ 100	0.0	51	511754	≥ 100	0.1
	2	922.8	0	952	3.2	32.3	0	1434	55.4	0.0	41	411372	≥ 100	0.0	47	470896	≥ 100	0.1
	3	1269.7	0	1297	2.2	48.2	0	3019	≥ 100	0.0	50	501460	≥ 100	0.0	53	531272	≥ 100	0.1
	4	1484.5	0	1504	1.3	41.2	0	2258	52.1	0.0	45	451691	≥ 100	0.0	51	511455	≥ 100	0.0
	5	1359.0	0	1360	0.1	5.3	0	2536	86.6	0.0	45	451642	≥ 100	0.0	51	511366	≥ 100	0.1
M-5	1	1884.1	0	1970	4.6	86.6	0	2794	48.3	0.0	61	611854	≥ 100	0.0	66	661820	≥ 100	0.1
	2	2030.1	0	2067	1.8	74.1	0	3624	78.5	0.0	63	632026	≥ 100	0.0	65	652300	≥ 100	0.1
	3	1174.7	0	1348	14.8	64.4	0	2043	73.9	0.0	57	571205	≥ 100	0.0	61	611159	≥ 100	0.1
	4	1779.0	0	1795	0.9	4.6	0	2698	51.7	0.0	61	611801	≥ 100	0.0	59	591816	≥ 100	0.1
	5	1097.0	0	1097	0.0	74.0	0	2183	99.0	0.0	60	601433	≥ 100	0.0	62	621249	≥ 100	0.2
M-6	1	2002.4	0	2046	2.2	128.5	0	2835	41.6	0.0	74	742003	≥ 100	0.0	76	762003	≥ 100	0.1
	2	1390.0	0	1390	0.0	125.7	0	2602	87.2	0.0	73	731393	≥ 100	0.0	74	741400	≥ 100	0.2
	3	2179.8	0	2255	3.4	122.6	1	14286	≥ 100	0.0	74	742167	≥ 100	0.0	76	762167	≥ 100	0.4
	4	2092.6	0	2125	1.5	112.1	0	4392	≥ 100	0.0	73	732221	≥ 100	0.0	75	752221	≥ 100	0.6
	5	1436.0	0	1438	0.1	4.5	0	3055	≥ 100	0.0	77	771565	≥ 100	0.0	79	791565	≥ 100	0.4
M-7	1	2262.0	0	2511	11.0	175.6	0	3036	34.2	0.0	87	872197	≥ 100	0.0	90	902217	≥ 100	0.2
	2	2535.1	0	2601	2.6	189.2	0	4393	73.3	0.0	89	892662	≥ 100	0.0	93	932662	≥ 100	0.4
	3	1339.2	0	1352	1.0	154.3	0	2391	78.5	0.0	82	821330	≥ 100	0.0	86	861337	≥ 100	0.1
	4	2191.9	0	2297	4.8	189.5	0	3203	46.1	0.0	90	902195	≥ 100	0.0	93	932195	≥ 100	0.4
	5	2819.2	2	22379	≥ 100	172.4	2	24529	≥ 100	0.0	92	922905	≥ 100	0.0	93	932729	≥ 100	1.0

Table A6: Details of the computational results reported in Table 3.8 (heavy-traffic instances).

Problem set	Instance	LB	Lagrangian Relaxation				FCFS				STW				RS			
			#R	UB	G (%)	Time	#R	UB	G (%)	Time	#R	UB	G (%)	Time	#R	UB	G (%)	Time
H-1	1	232.0	0	232	0.0	0.1	0	430	85.3	0.0	0	595	≥ 100	0.0	0	466	≥ 100	0.0
	2	86.0	0	86	0.0	0.1	0	280	≥ 100	0.0	0	347	≥ 100	0.0	0	292	≥ 100	0.0
	3	192.0	0	192	0.0	0.2	0	426	≥ 100	0.0	0	567	≥ 100	0.0	0	539	≥ 100	0.0
	4	148.0	0	149	0.7	0.2	0	315	≥ 100	0.0	0	296	≥ 100	0.0	0	616	≥ 100	0.0
	5	250.0	0	250	0.0	0.1	0	449	79.6	0.0	0	409	63.6	0.0	0	702	≥ 100	0.0
H-2	1	523.9	0	529	1.0	4.8	0	845	61.3	0.0	16	161368	≥ 100	0.0	19	191222	≥ 100	0.0
	2	465.0	0	465	0.0	0.3	0	928	99.6	0.0	0	2400	≥ 100	0.0	20	200982	≥ 100	0.0
	3	411.3	0	414	0.7	1.8	0	795	93.3	0.0	19	190505	≥ 100	0.0	26	260424	≥ 100	0.0
	4	393.0	0	396	0.8	0.4	0	1143	≥ 100	0.0	19	190466	≥ 100	0.0	26	260559	≥ 100	0.0
	5	341.0	0	341	0.0	0.4	0	625	83.3	0.0	22	221393	≥ 100	0.0	26	260551	≥ 100	0.0
H-3	1	1314.0	0	1318	0.3	29.5	0	2477	88.5	0.0	38	381733	≥ 100	0.0	31	311921	≥ 100	0.1
	2	1063.6	0	1069	0.5	25.3	0	1926	81.1	0.0	37	371425	≥ 100	0.0	38	381275	≥ 100	0.1
	3	1232.0	0	1238	0.5	0.9	1	12052	≥ 100	0.0	40	401343	≥ 100	0.0	34	341755	≥ 100	0.1
	4	1010.0	0	1010	0.0	26.2	0	1859	84.1	0.0	35	351180	≥ 100	0.0	36	361293	≥ 100	0.1
	5	936.3	0	1010	7.9	22.3	0	1480	58.1	0.0	35	351387	≥ 100	0.0	39	390903	≥ 100	0.0
H-4	1	1251.0	0	1253	0.2	1.7	0	2355	88.2	0.0	52	521642	≥ 100	0.0	55	551344	≥ 100	0.1
	2	2046.2	0	2111	3.2	56.9	0	3517	71.9	0.0	55	552013	≥ 100	0.0	53	532187	≥ 100	0.1
	3	1448.8	1	11445	≥ 100	72.0	1	13291	≥ 100	0.0	59	591275	≥ 100	0.0	58	581449	≥ 100	0.1
	4	869.0	0	873	0.5	2.1	0	2270	≥ 100	0.0	54	541114	≥ 100	0.0	59	590877	≥ 100	0.2
	5	1181.9	0	1361	15.2	64.8	0	2303	94.8	0.0	56	561425	≥ 100	0.0	57	571387	≥ 100	0.1
H-5	1	1273.0	0	1277	0.3	3.7	0	2882	≥ 100	0.0	70	701273	≥ 100	0.0	70	701293	≥ 100	0.1
	2	1527.4	0	1635	7.0	111.7	0	2812	84.1	0.0	71	711540	≥ 100	0.0	72	721606	≥ 100	0.1
	3	1615.6	0	1791	10.9	121.0	0	3903	≥ 100	0.0	67	671815	≥ 100	0.0	72	721595	≥ 100	0.3
	4	2102.0	0	2325	10.6	177.8	0	3722	77.1	0.0	72	722079	≥ 100	0.0	73	732037	≥ 100	0.4
	5	1661.0	0	1823	9.8	119.9	0	2895	74.3	0.0	70	701624	≥ 100	0.0	72	721624	≥ 100	0.1
H-6	1	2337.4	0	2345	0.3	56.1	0	4122	76.4	0.0	82	822337	≥ 100	0.0	86	862337	≥ 100	0.3
	2	2014.0	0	2034	1.0	26.1	0	3955	96.4	0.0	87	872026	≥ 100	0.0	89	892099	≥ 100	0.4
	3	1500.1	0	1520	1.3	185.4	0	3069	≥ 100	0.0	86	861504	≥ 100	0.0	89	891504	≥ 100	0.6
	4	2630.6	2	22610	≥ 100	202.3	2	24603	≥ 100	0.0	88	882461	≥ 100	0.0	92	922461	≥ 100	0.4
	5	2366.6	0	2723	15.1	197.4	0	3565	50.6	0.0	77	772503	≥ 100	0.0	87	872245	≥ 100	0.1
H-7	1	2167.7	0	2247	3.7	269.0	1	14067	≥ 100	0.0	100	1002145	≥ 100	0.0	100	1002145	≥ 100	0.2
	2	2635.9	0	2816	6.8	275.2	0	4283	62.5	0.0	91	912842	≥ 100	0.0	100	1002638	≥ 100	0.6
	3	2638.8	0	2725	3.3	334.4	0	5042	91.1	0.0	104	1042585	≥ 100	0.0	108	1082597	≥ 100	0.5
	4	1789.4	0	1878	4.9	267.6	0	3335	86.4	0.0	102	1021762	≥ 100	0.0	101	1011826	≥ 100	0.4
	5	1942.3	0	1985	2.2	219.2	0	4208	≥ 100	0.0	95	951944	≥ 100	0.0	98	982071	≥ 100	0.4