

Submitted to *Operations Research*

manuscript (Please, provide the manuscript number!)

Authors are encouraged to submit new papers to INFORMS journals by means of a style file template, which includes the journal title. However, use of a template does not certify that the paper has been accepted for publication in the named journal. INFORMS journal templates are for the exclusive purpose of submitting to an INFORMS journal and should not be used to distribute the papers in print or online or to submit the papers to another publication.

Vessel Service Planning in Seaports

Lingxiao Wu

HEC Montréal and GERAD, 3000 chemin de la Côte-Sainte-Catherine, Montréal, H3T 2A7, Canada,
lingxiaowu513@gmail.com

Yossiri Adulyasak

HEC Montréal and GERAD, 3000 chemin de la Côte-Sainte-Catherine, Montréal, H3T 2A7, Canada,
yossiri.adulyasak@hec.ca

Jean-François Cordeau

HEC Montréal and GERAD, 3000 chemin de la Côte-Sainte-Catherine, Montréal, H3T 2A7, Canada,
jean-francois.cordeau@hec.ca

Shuaian Wang

Strome College of Business, Old Dominion University, Norfolk, Virginia 23529, USA,
wangshuaian@gmail.com

Berth allocation and pilotage planning are the two most important decisions made by a seaport for serving incoming vessels. Traditionally, the berth allocation problem and the pilotage planning problem are solved sequentially, leading to suboptimal or even infeasible solutions for vessel services. This paper investigates a vessel service planning problem (VSPP) in seaports that addresses berth allocation and pilotage planning in combination. We introduce a compact mixed-integer linear programming formulation for the problem, which can be solved by general-purpose solvers. To solve large-scale instances, we develop an exact solution approach that combines Benders decomposition and column generation in a novel and effective way. Unlike the traditional three-phase Benders decomposition and column generation method, which does not guarantee optimality, we propose a branching scheme that enables the approach to determine an optimal solution to the VSPP. The approach is enhanced through practical acceleration strategies. Extensive computational results using data instances from one of the world's largest seaports show that these acceleration strategies significantly improve the performance of our solution approach and that it can obtain optimal or near-optimal solutions for instances of realistic scale. We perform sensitivity tests to demonstrate the robustness of the approach against variations in problem settings. We also show the benefits brought by integrated optimization by comparing our solution approach with a method that handles berth allocation and pilotage planning sequentially.

Key words: seaport operations, vessel service planning, berth allocation, pilotage planning, Benders decomposition, column generation

History: Received November 2020

1. Introduction

Seaports have long been the key nodes in the global supply chain. In 2018, more than 10 billion tons of cargo were handled by seaports around the world, and the volume is still growing (UNCTAD 2019). Due to the growth of seaborne trade, the number and size of vessels that need to be served at seaports have increased continuously. Seaports and vessels involve huge investments; to recoup these investments, port authorities and shipping companies aim to achieve very high utilization of their capacities, making it critical for vessel services to be efficient (Fransoo and Lee 2013, Roy et al. 2020). In addition, given that vessels are major sources of harmful air pollutants, rapid vessel turnaround at seaports helps to relieve the environmental burden on local communities (Du et al. 2015, Wu and Wang 2020).

Figure 1 shows the layout of a typical seaport, which generally can be divided into three parts: the anchorage, the navigation channel, and the terminals. The quay of a terminal is divided into a set of berths. Vessels coming from the open sea first reach and wait in the anchorage before traveling through the channel and arriving at the designated berths to be handled. After being served at the berths, the vessels leave the berths, sail to the anchorage by passing through the navigation channel and finally return to the open sea. In most seaports, pilotage is compulsory to enhance navigational safety. In particular, whenever a vessel is moving in port waters (including sailing in the channel and berthing into or unberthing from the berths), a pilot must be onboard to provide navigational guidance to the vessel master (Wu et al. 2020).

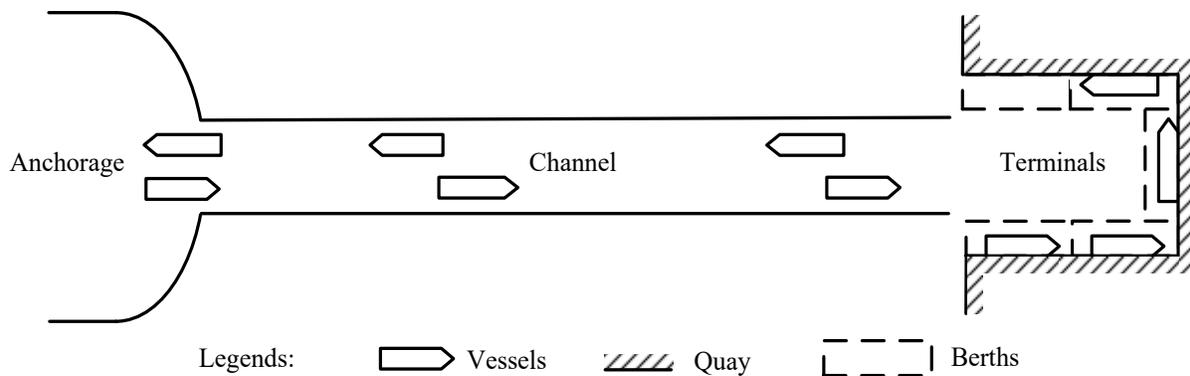


Figure 1 Layout of a Seaport.

Given a set of calling vessels, the service plan made by a seaport includes two main components: berth allocation and pilotage planning. Berth allocation determines the berthing position and time of each vessel, while pilotage planning involves vessel traffic management and pilot scheduling. Berths are the most important resource in a seaport. In busy seaports, vessels often have to wait for

berths to become available due to the limited number of berths compared with the number of calling vessels. Efficient berth allocation promotes efficient cargo dispatch and rapid vessel turnaround. Hence, berth allocation is at the core of seaport operations.

Pilotage planning, another important problem faced by seaport operators, involves the management of vessel traffic in port waters and the assignment of marine pilots for serving the vessels. Vessel movements must be carefully controlled to ensure smooth and collision-free traffic, and pilot assistance is indispensable in this regard. Given their importance, pilots are among the most highly compensated professionals in a seaport. In the U.S., the average yearly salary for marine pilots is more than US\$400,000 (NPR News 2012). Moreover, pilotage is a very demanding job that requires intense concentration and high skill levels. Workload management is thus a key component to improve the health of pilots and mitigate fatigue (International Maritime Organization 2001). Therefore, effective and functional pilotage planning is essential to avoid accidents, secure fluency in port operations, reduce port authorities' operational costs, and relieve work-related anxieties.

Traditionally, the berth allocation problem (BAP) and the pilotage planning problem (PPP) are solved sequentially, with the output of the first problem being an input of the second one. However, this sequential solution approach may lead to suboptimal or even infeasible solutions in vessel services.

In this paper, we consider a vessel service planning problem (VSPP) that addresses berth allocation and pilotage planning in an integrated way. To solve large-scale instances, we develop an exact solution approach that embeds Benders decomposition and column generation in a branch-and-bound framework. To the best of our knowledge, our study is the first to solve the BAP and the PPP jointly. Our study makes three main contributions:

1. We propose a research problem that integrates two important problems in seaport operations and develop a compact mixed-integer linear programming (MILP) formulation for the problem that can be well solved by general-purpose optimization solvers on small-scale instances.
2. We develop an efficient exact approach based on Benders decomposition and column generation that can solve large-scale instances. The structure of the approach is general enough to be applied to other problems that involve synchronized scheduling of multiple resources and can easily be tailored for such purposes. We also propose several acceleration strategies that greatly improve the performance of the approach.
3. We conduct extensive numerical experiments using data instances from one of the world's largest seaports. The results demonstrate the benefits of integrated vessel service planning against the traditional sequential decision method.

The remainder of this paper is structured as follows. We review the relevant studies in Section 2. We formally describe the problem and formulate it as a compact MILP model in Section 3. The

solution approach is described in Section 4. Computational experiments are reported in Section 5, followed by conclusions in Section 6. We provide all mathematical proofs in the EC.1 of the electronic companion.

2. Literature Review

Seaport operations have received considerable attention in the scientific literature. For surveys of studies in this area, we refer readers to Stahlbock and Voß (2008), Bierwirth and Meisel (2010, 2015), and Carlo et al. (2015). Of many problems related to seaport operations that have been studied in the literature, the ones most closely related to the VSPP are the berth allocation problem, the vessel traffic management problem, and the pilot scheduling problem. In the following subsections, we review studies that focus on these problems.

2.1. Review of Studies on Berth Allocation

The berth allocation problem (BAP) has been the subject of intensive research over the past two decades. Broadly, the BAP can be divided into two streams depending on the spatial constraints on berthing positions: the discrete BAP and the continuous BAP (Imai et al. 2005, Xu and Lee 2018). In the discrete BAP, the quay of the seaport is partitioned into a set of discrete berths, and each berth can handle at most one vessel at a time. In the continuous case, vessels can berth at arbitrary positions within the boundaries of the quay. In this study, we consider discrete berth allocation and thus limit our review to studies in this stream.

The discrete BAP was initially studied by Imai et al. (1997), who presented the first MILP formulation for the problem. In the formulation, all berths are available from the beginning of the planning horizon, and all vessels are already waiting in the seaport. The objective considered by the authors was to minimize the total turnaround time of vessels. Imai et al. (2001) extended this study by allowing vessels to arrive dynamically during a planning horizon. A Lagrangian relaxation method was proposed to solve the problem. Imai et al. (2003) considered a BAP with service priorities and solved the problem using a genetic algorithm. Cordeau et al. (2005) studied a BAP with time windows on vessel handling completion; they proposed a multi-depot vehicle routing problem formulation and developed a tabu search algorithm for solving the problem. Monaco and Sammarra (2007) strengthened the formulation provided by Imai et al. (2001) and developed a Lagrangian relaxation method for solving the problem. Buhrkal et al. (2011) presented a general set-partitioning model for the discrete BAP that was shown to outperform all other existing models.

2.2. Review of Studies on Vessel Traffic Management

The aim of traffic management in seaports and other restricted waters is to achieve smooth and collision-free traffic flows. The vessel traffic management problem has gained increasing attention from academia in recent years. Using a simulation, Tang et al. (2016) showed that the handling capacity of a seaport may greatly depend on the configuration and capacity of its navigation channel. Building on Tang et al. (2016), most studies in this area have examined ways to manage traffic in seaports based on predetermined berth allocation plans (Zhang et al. 2016, Lalla-Ruiz et al. 2018, Jia et al. 2019, Li and Jia 2019). Zhang et al. (2016) considered a scheduling problem for vessels entering and leaving a seaport and passing through a navigation channel, the objective being to minimize the total waiting time of vessels at berths or in the anchorage. The problem was solved by a metaheuristic based on simulated annealing and genetic search. Lalla-Ruiz et al. (2018) investigated a similar problem arising in the Port of Shanghai, where vessel traffic has to be scheduled in two parallel waterways. The authors modeled the problem as a multi-depot vehicle routing problem and proved that the problem is NP-hard. A greedy heuristic and a simulated annealing algorithm were proposed to solve the problem. Jia et al. (2019) and Li and Jia (2019) extended the problem studied by Zhang et al. (2016) and Lalla-Ruiz et al. (2018) by considering the use of the anchorage area in the port basin, where vessels can wait near the berths. They proposed Lagrangian relaxation- and column generation-based methods for solving the problem. In addition to vessel traffic management in seaports, Lübbecke et al. (2019) proposed an approach to schedule vessel traffic in the Kiel Canal, a bidirectional waterway in Germany. To mathematically formulate the problem, the authors divided the canal into a set of segments with different nautical rules and showed that the problem shares similarities with *the single-track train scheduling problem*, which can be formulated as a job-shop scheduling problem. A local-search-based heuristic was developed to solve the problem.

The BAP and the traffic management problem in seaports are interconnected and generate more value if solved jointly. The integration of the two problems has been discussed in two studies. Zhen et al. (2017) investigated the BAP in a container terminal and imposed channel capacity limitations such that the total number of vessels simultaneously sailing in a channel cannot exceed a given upper bound. A column generation method was developed to solve this problem. Corry and Bierwirth (2019), noting that channel restrictions based on a maximum throughput capacity may not be sufficient for channel traffic management, proposed and studied a joint BAP and channel traffic control problem. They tackled the problem as a special no-wait flow shop scheduling problem and formulated it as an MILP model. The computational results demonstrated that their model could well solve instances with up to 10 berths and 20 vessels using a general-purpose optimization solver.

2.3. Review of Studies on Pilot Scheduling

Although pilots play a critical role in seaport operations, studies on pilot scheduling are rare. Wermus and Pope (1994) studied a pilot rostering problem to determine work shift start and end times assigned to each pilot, without considering the scheduling of pilotage services. A simple rule-based heuristic was developed to solve the problem. Edwards (2010) investigated a problem for assigning a group of pilots to a set of pilotage tasks, whereby each task corresponds to a vessel entering or leaving a terminal and has a fixed start time. They solved the problem using a heuristic based on column generation.

Two recent works considered the PPP in seaports based on predetermined berth allocation plans. Wu et al. (2020) considered a PPP for jointly scheduling pilots and managing vessel traffic in a seaport. In the problem, work schedules for guiding vessels sailing into and out of berths are generated for pilots working on different shifts. All pilots in the problem were assumed to be homogenous. The authors proved that the PPP is strongly NP-hard and proposed a solution algorithm based on branch-and-price. Jia et al. (2020) investigated an integrated daily vessel traffic and pilot scheduling problem, whereby the seaport has multiple navigation channels and vessels can wait in the anchorage area located near the berths. No working time limits were considered, and pilots were assumed to be able to work all day long. The problem was solved by a Lagrangian relaxation algorithm.

2.4. Review of Other Related Studies

From a modeling perspective, the VSPP jointly solves a resource allocation problem and a personnel scheduling problem. Problems with a similar structure have been studied in other areas. The most well-known examples of such problems include the integrated aircraft routing and crew pairing problem in airlines (Cordeau et al. 2001, Mercier et al. 2005, Sandhu and Klabjan 2007, Ruther et al. 2017), the integrated rolling stock and train driver scheduling problem in railways (Dauzère-Pères et al. 2015), and the integrated vehicle routing and truck driver scheduling problem in road transportation (Goel and Irnich 2017).

The VSPP is different from these problems in multiple ways. First, the problems in airline and railway operations are based on given timetables of flights or trains, and the resources and crew members are scheduled in accordance with these timetables. In the VSPP, however, the times to serve the vessels are also among the decision variables. Second, in the VSPP, pilots can move between the vessels not only through the vessels they serve but also through dedicated transfer vehicles (e.g., pilot boats and helicopters). For airline crews and train or truck drivers, there are no dedicated transfer vehicles that can affect scheduling; rather, personnel movements must conform

to the movements of aircraft, trains, and trucks. For this reason, personnel scheduling has more flexibility in the VSPP than in these other problems. A further distinguishing feature of the VSPP relates to regulations on channel traffic, which lead to precedence and non-simultaneity constraints between vessel movements (discussed in the next section); such constraints generally do not exist in the other aforementioned problems.

3. Problem Statement and Compact Formulation

In this section, we first describe the background of the VSPP in Section 3.1 and then introduce the notation used for formulating the problem in Section 3.2. A compact MILP formulation for the VSPP is presented in Section 3.3. We introduce a series of inequalities for strengthening the formulation in Section 3.4.

3.1. Problem Background

In the VSPP, we are given a set of vessels that dynamically call at a seaport in a planning horizon. These vessels go through a four-stage procedure in their port stay. As shown in Figure 2, after arriving at the anchorage, a vessel waits there until it is ready to sail into the channel (Stage 1). Then, a pilot boards the vessel at the entrance of the channel, navigates it through the channel, and moors it into the designated berth (Stage 2). The pilot leaves the vessel once mooring is completed, and cargo handling at the berth commences (Stage 3). When the vessel is ready to leave the berth, a pilot gets on board and maneuvers the vessel toward the anchorage by passing through the channel (Stage 4). Finally, the pilot debarks the vessel in the anchorage, and the vessel returns to the open sea. Without loss of generality, we assume that each vessel is given time windows for sailing into and out of a berth (i.e., for starting the second and fourth stages).

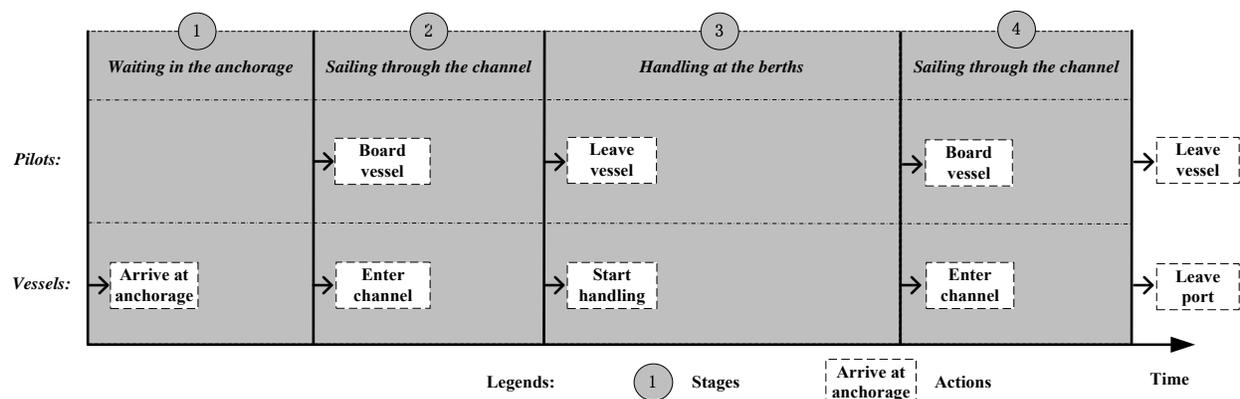


Figure 2 A Vessel's Port Stay.

We consider the following restrictions on vessel traffic in a seaport based on common practices (Corry and Bierwirth 2019, Jia et al. 2019). First, vessels must sail at the same slow and constant speed in the channel. Second, vessels sailing in the same direction (i.e., entering or leaving berths) must queue up in the channel, with no overtaking. Third, there must be a minimum headway between a vessel entering the channel and those following it. Fourth, as the water depth is limited and affected by tides, vessels with large drafts may be able to sail in the channel only in high-tide periods. Fifth, to avoid collisions, vessels with extra widths cannot sail simultaneously in opposite directions in the channel. Finally, as vessels are allowed to moor only in the anchorage or at berths, en-route stops are not allowed during the journey between the anchorage and the berths. That is, a vessel has to sail directly to the berth once it leaves the anchorage and has to sail directly to the anchorage once it leaves the berth.

When moving in port waters, a vessel must be navigated by a pilot. The seaport manages a group of pilots and divides the planning horizon into a set of shifts to coordinate pilot assignments. Pilots are scheduled to work during these shifts and can only serve the vessels within their assigned shifts. Depending on the labor policy in a seaport, shifts may or may not overlap. A pilot can serve in multiple vessel movements in one shift, but there is a minimum setup time between two consecutive services conducted by the same pilot. As the repositioning of pilots through the channel can take a considerable amount of time, the setup time between two pilotage services is longer if the first service terminates and the second service starts at different ends of the channel (and is shorter if they occur at the same end of the channel). For example, after a pilot guides a vessel into a berth, the repositioning time is considerably shorter in the case of guiding another vessel out of a berth than in the case of guiding another vessel into a berth. A typical shift lasts eight hours, and each shift includes a rest period during which there are no assignments. Finally, we assume that pilots are identical and that the number of pilots who can work in each shift is not a binding constraint in the problem (but the seaport has to pay for each pilot assigned to a shift).

In the VSPP, the seaport wants to optimize both the service level and operational costs. We quantify the service level by considering delays in cargo dispatch (i.e., delays at the start of discharging cargo from vessels) and vessel turnaround. Meanwhile, operational costs include vessel handling costs (vessels typically have berth-wise handling costs) and pilot dispatching. The objective of the VSPP is to generate a vessel service plan that achieves the best optimization goal without violating any regulation regarding berth allocation, traffic management, or pilot scheduling.

3.2. Notation

This section describes the notation that we will use to formulate the VSPP as a compact MILP model. For ease of presentation, we use the notation $\mathbb{Z}_{[z, \bar{z}]}$ to denote the set of integers that are no

smaller than \underline{z} and no larger than \bar{z} throughout the paper. In addition, we use \mathbb{Z}^+ to denote the set of nonnegative integers and \mathbb{R} to denote the set of real numbers.

We consider the VSPP with a discrete-time finite planning horizon. We denote by $\mathcal{T} = \mathbb{Z}_{[1, \bar{T}]}$ the set of evenly distributed time steps in the planning horizon, which are indexed by t . The planning horizon is divided into a set \mathcal{S} of shifts, indexed by s . Let $\mathcal{T}_s \subseteq \mathcal{T}$ be the set of time steps in shift s , and $\mathcal{T}_s = \mathbb{Z}_{[\underline{T}_s, \bar{T}_s]}$. We have $\bigcup_{s \in \mathcal{S}} \mathcal{T}_s = \mathcal{T}$. Besides, let \mathcal{S}_t be the set of shifts that contain time step t .

Let \mathcal{K} , indexed by k , be the set of vessels calling at the seaport in the planning horizon. The port has a set \mathcal{B} of berths indexed by b . Let a_b be the available time of berth b , such that vessels in \mathcal{K} can only moor into berth b at time steps in the set $\mathbb{Z}_{[a_b, \bar{T}]}$. Vessel k can be handled at a berth from a subset $\mathcal{B}_k \subseteq \mathcal{B}$. The handling time of vessel k at berth b is denoted by $h_{k,b}$.

We denote by \mathcal{I} , indexed by i , the set of *pilotage tasks* or *tasks* corresponding to pilot services in vessel movements in the seaport. Let $\mathcal{I}^{in} \subseteq \mathcal{I}$ and $\mathcal{I}^{out} \subseteq \mathcal{I}$ be the sets of tasks corresponding to vessels sailing into and out of the berths, respectively. Furthermore, let \mathcal{I}_k be the set of tasks for serving vessel k . Note that $|\mathcal{I}_k| = 2$. The time window to start each task i is given by $\mathcal{E}_i = \{\underline{E}_i, \dots, \bar{E}_i\}$, where \underline{E}_i and \bar{E}_i represent the earliest and latest feasible start times of task i , respectively. The duration of task i is denoted by d_i . For navigational safety, a minimum headway denoted by f_i must be ensured between the start times of task $i \in \mathcal{I}^{in}$ (resp. \mathcal{I}^{out}) and its followers from \mathcal{I}^{in} (resp. \mathcal{I}^{out}). A pair of vessels with extra widths are not be allowed to sail in opposite directions in the channel simultaneously. To impose this restriction, we define a set \mathcal{U} of task pairs such that $\mathcal{U} := \{(i, j) \in \mathcal{I} \times \mathcal{I} \mid i \text{ and } j \text{ cannot be performed simultaneously, } i, j \in \mathcal{I}\}$.

Let \mathcal{P}_s be set of pilots who can work in shift s . To streamline the formulation, in our model, pilots are “shift-specific” such that $p \in \mathcal{P}_s$ does not correspond to a particular pilot in practice, but refers to a generic pilot who is able to work in shift s . By definition, $\mathcal{P}_s \cap \mathcal{P}_{s'} = \emptyset$ if $s \neq s'$. Because we assume that the number of pilots available in each shift is not binding, when implementing the model, we let $|\mathcal{P}_s|$ be equal to the maximum number of tasks that can start in shift s . Given \mathcal{P}_s , we set $\mathcal{P} = \bigcup_{s \in \mathcal{S}} \mathcal{P}_s$.

Any pilot $p \in \mathcal{P}_s$, $\forall s \in \mathcal{S}$, if scheduled to work in this shift, should be awarded a rest period that starts within a predetermined time window and lasts for at least a given minimum duration. For each s , the time window for pilots working in this shift to start the rest period is denoted by \mathcal{L}_s , and $\mathcal{L}_s = \mathbb{Z}_{[\underline{L}_s, \bar{L}_s]}$. Meanwhile, the minimum duration of the rest period in this shift is denoted by g_s . From the modeling perspective, a rest period can also be modeled as a task. Hence, we let \mathcal{J}_p be the set (singleton) of the rest period for pilot p . Corresponding to $i \in \mathcal{J}_p$, we let \mathcal{E}_i and d_i be the time window to start the rest period and its minimum duration, respectively. Here $\mathcal{E}_i = \mathcal{L}_s$ and $d_i = g_s$, $\forall i \in \mathcal{J}_p$, $p \in \mathcal{P}_s$, $s \in \mathcal{S}$. We further define $\mathcal{J} = \bigcup_{p \in \mathcal{P}} \mathcal{J}_p$. For ease of presentation, we use the term *activity* to refer to a task or a rest period in the remainder of this paper.

Pilots may perform multiple activities in a shift. For any two consecutive activities i and j performed by a pilot, we denote by $q_{i,j}$ the minimum setup time between them. In particular, we set $q_{i,j} = 0$ if i or j corresponds to a rest period. We assume that the triangle inequality holds for the setup times between tasks so that $q_{i,j} \leq q_{i,i'} + d_{i'} + q_{i',j}, \forall i, i', j \in \mathcal{I}, i \neq j \neq i'$. Moreover, to tighten the MILP model, $\forall i, j \in \mathcal{I} \cup \mathcal{J}, i \neq j$, we let

$$M_{i,j} = \begin{cases} \max\{0, \bar{E}_i + d_i + q_{i,j} - \underline{E}_j\}, & \text{if } j \in \mathcal{I}, \\ \bar{E}_i + d_i + q_{i,j}, & \text{if } j \in \mathcal{J}. \end{cases}$$

We consider the following cost components in the VSPP. First, given task i and its start time t , we denote by $c_{i,t}^1$ the penalty cost associated with delays in cargo dispatch (if $i \in \mathcal{I}^{in}$) or vessel turnaround (if $i \in \mathcal{I}^{out}$) for starting this task at this time step. In practice, $c_{i,t}^1$ is set by port authorities based on the service priority of task i and the delay between the earliest start time (\underline{E}_i) and time step t . Note that our model can handle any relationship between the penalty cost and the delay. Second, let $c_{k,b}^2$ be the cost of handling vessel k at berth b . Third, the cost of assigning a pilot to work in shift s is denoted by c_s^3 .

The following decision variables are used to formulate the VSPP:

- $u_{k,b}$: binary variable, which takes value 1 if and only if vessel k is handled at berth b ;
- $v_{k,t}^{in}$: binary variable, which takes value 1 if and only if vessel k reaches a berth at time step t ;
- $v_{k,t}^{out}$: binary variable, which takes value 1 if and only if vessel k leaves a berth at time step t ;
- $w_{k,b,t}$: binary variable, which takes value 1 if and only if vessel k occupies berth b at time step t ;
- $x_{i,t}$: binary variable, which takes value 1 if and only if task i starts at time step t ;
- $y_{p,0,j}$: binary variable, which takes value 1 if and only if activity j is the first activity performed by pilot p ;
- $y_{p,i,j}$: binary variable, which takes value 1 if and only if pilot p performs activity j immediately after activity i ;
- $y_{p,i,0}$: binary variable, which takes value 1 if and only if activity i is the last activity performed by pilot p ;
- $z_{s,i}$: binary variable, which takes value 1 if and only if task i is performed by a pilot in shift s ;
- n_s : integer variable, representing the number of pilots working in shift s .

3.3. The VSPP Formulation

We present a compact MILP formulation for the VSPP, which is an extension of the formulations for the BAP, the vessel traffic management problem, and the PPP studied by Zhen et al. (2017), Jia et al. (2019), Xie et al. (2019), Wu et al. (2020), and Jia et al. (2020). We first present the objective function:

$$[\mathbf{M1}] \quad \min \sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{E}_i} c_{i,t}^1 x_{i,t} + \sum_{k \in \mathcal{K}} \sum_{b \in \mathcal{B}_k} c_{k,b}^2 u_{k,b} + \sum_{s \in \mathcal{S}} c_s^3 n_s. \quad (1)$$

This objective function minimizes the total cost of the vessel service plan, which equals the sum of the penalty cost of starting tasks later than their earliest feasible start times (the first term), the cost of handling vessels (the second term), and the cost of assigning pilots (the third term). The constraints can be divided into four blocks: constraints (2)–(8) regulate berth allocation; constraints (9)–(14) control vessel traffic in the channel; constraints (15)–(24) are related to pilot scheduling; and the domains of the variables are defined in constraints (25)–(31). The first block is the following:

$$\sum_{b \in \mathcal{B}_k} u_{k,b} = 1 \quad \forall k \in \mathcal{K} \quad (2)$$

$$\sum_{t \in \mathcal{T}} v_{k,t}^{in} = 1 \quad \forall k \in \mathcal{K} \quad (3)$$

$$\sum_{t \in \mathcal{T}} v_{k,t}^{out} = 1 \quad \forall k \in \mathcal{K} \quad (4)$$

$$\sum_{t \in \mathcal{T}} t v_{k,t}^{in} \geq \sum_{b \in \mathcal{B}_k} a_b u_{k,b} \quad \forall k \in \mathcal{K} \quad (5)$$

$$\sum_{t \in \mathcal{T}} t v_{k,t}^{out} \geq \sum_{t \in \mathcal{T}} t v_{k,t}^{in} + \sum_{b \in \mathcal{B}_k} h_{k,b} u_{k,b} \quad \forall k \in \mathcal{K} \quad (6)$$

$$w_{k,b,t} \geq u_{k,b} + \sum_{t'=1}^t v_{k,t'}^{in} + \sum_{t'=\min\{t+1, \bar{T}\}}^{\bar{T}} v_{k,t'}^{out} - 2 \quad \forall t \in \mathcal{T}, b \in \mathcal{B}, k \in \mathcal{K} \quad (7)$$

$$\sum_{k \in \mathcal{K}} w_{k,b,t} \leq 1 \quad \forall t \in \mathcal{T}, b \in \mathcal{B}. \quad (8)$$

Constraints (2) require that a suitable berth be allocated for handling each vessel. Constraints (3) and (4) require that each vessel reaches and leaves a berth at one and only one time step, respectively. A vessel cannot moor into a berth before the berth becomes available (constraints (5)). Constraints (6) ensure that a vessel has sufficient time for cargo handling when mooring at a berth. Constraints (7) state that berth b is occupied by vessel k at time step t ($w_{k,b,t} = 1$) if (i) the berth is allocated to handle the vessel ($u_{k,b} = 1$), (ii) the vessel has reached the berth at the time step ($\sum_{t'=1}^t v_{k,t'}^{in} = 1$), and (iii) the vessel has not left the berth ($\sum_{t'=\min\{t+1, \bar{T}\}}^{\bar{T}} v_{k,t'}^{out} = 1$). At any time, a berth can handle at most one vessel, as required by constraints (8). The second block contains these constraints:

$$\sum_{t \in \mathcal{E}_i} x_{i,t} = 1 \quad \forall i \in \mathcal{I} \quad (9)$$

$$\sum_{i \in \mathcal{I}^{in}} \sum_{t' \in \mathbb{Z}_{[t-f_i+1, t]} \cap \mathcal{E}_i} x_{i,t'} \leq 1 \quad \forall t \in \mathcal{T} \quad (10)$$

$$\sum_{i \in \mathcal{I}^{out}} \sum_{t' \in \mathbb{Z}_{[t-f_i+1, t]} \cap \mathcal{E}_i} x_{i,t'} \leq 1 \quad \forall t \in \mathcal{T} \quad (11)$$

$$\sum_{t' \in \mathbb{Z}_{[t-d_i+1, t]} \cap \mathcal{E}_i} x_{i, t'} + \sum_{t' \in \mathbb{Z}_{[t-d_j+1, t]} \cap \mathcal{E}_j} x_{j, t'} \leq 1 \quad \forall t \in \mathcal{T}, (i, j) \in \mathcal{U} \quad (12)$$

$$\sum_{t \in \mathcal{T}} tv_{k, t}^{in} = d_i + \sum_{t \in \mathcal{E}_i} tx_{i, t} \quad \forall i \in \mathcal{I}^{in} \cap \mathcal{I}_k, k \in \mathcal{K} \quad (13)$$

$$\sum_{t \in \mathcal{T}} tv_{k, t}^{out} = \sum_{t \in \mathcal{E}_i} tx_{i, t} \quad \forall i \in \mathcal{I}^{out} \cap \mathcal{I}_k, k \in \mathcal{K}. \quad (14)$$

Constraints (9) require that each pilotage starts within a given time window. The minimum headway between any vessel sailing into the channel and those following it is imposed by constraints (10) and (11). Constraints (12) enforce that tasks in any pair in set \mathcal{U} cannot be performed simultaneously. Start times of pilotage tasks associated with a vessel are linked with the times when the vessel enters and leaves a berth in constraints (13) and (14). Note that $\mathcal{I}^{in} \cap \mathcal{I}_k$ and $\mathcal{I}^{out} \cap \mathcal{I}_k$ are singletons, $\forall k \in \mathcal{K}$. The following constraints constitute the third block:

$$\sum_{j \in \mathcal{I} \cup \mathcal{J}_p} y_{p, 0, j} \leq 1 \quad \forall p \in \mathcal{P} \quad (15)$$

$$y_{p, 0, i} + \sum_{j \in \mathcal{I} \cup \mathcal{J}_p \setminus \{i\}} y_{p, j, i} = y_{p, i, 0} + \sum_{j \in \mathcal{I} \cup \mathcal{J}_p \setminus \{i\}} y_{p, i, j} \quad \forall i \in \mathcal{I} \cup \mathcal{J}_p, p \in \mathcal{P} \quad (16)$$

$$\sum_{j \in \mathcal{I} \cup \mathcal{J}_p} y_{p, 0, j} = \sum_{i \in \mathcal{I} \cup \mathcal{J}_p} y_{p, i, 0} \quad \forall p \in \mathcal{P} \quad (17)$$

$$\sum_{p \in \mathcal{P}_s} y_{p, 0, i} + \sum_{p \in \mathcal{P}_s} \sum_{j \in \mathcal{I} \cup \mathcal{J}_p \setminus \{i\}} y_{p, j, i} = z_{s, i} \quad \forall i \in \mathcal{I}, s \in \mathcal{S} \quad (18)$$

$$x_{i, t} \leq \sum_{s \in \mathcal{S}_t} z_{s, i} \quad \forall t \in \mathcal{E}_i, i \in \mathcal{I} \quad (19)$$

$$\sum_{t \in \mathcal{E}_i} x_{i, t} = \sum_{j \in \mathcal{I} \cup \mathcal{J}_p} y_{p, 0, j} \quad \forall i \in \mathcal{J}_p, p \in \mathcal{P} \quad (20)$$

$$y_{p, 0, i} + \sum_{j \in \mathcal{I} \cup \mathcal{J}_p \setminus \{i\}} y_{p, j, i} = \sum_{j \in \mathcal{I} \cup \mathcal{J}_p} y_{p, 0, j} \quad \forall i \in \mathcal{J}_p, p \in \mathcal{P} \quad (21)$$

$$\sum_{t \in \mathcal{E}_j} tx_{j, t} \geq \sum_{t \in \mathcal{E}_i} tx_{i, t} + d_i + q_{i, j} + M_{i, j}(y_{p, i, j} - 1) \quad \forall i, j \in \mathcal{I} \cup \mathcal{J}_p, i \neq j, p \in \mathcal{P} \quad (22)$$

$$\sum_{t \in \mathcal{E}_j} tx_{j, t} \geq \sum_{t \in \mathcal{E}_i} tx_{i, t} + d_i + q_{i, j} + M_{i, j}(y_{p, i, i'} + y_{p, i', j} - 2) \quad (23)$$

$$\forall i, j \in \mathcal{I}, i \neq j, i' \in \mathcal{J}_p, p \in \mathcal{P}$$

$$\sum_{p \in \mathcal{P}_s} \sum_{i \in \mathcal{I} \cup \mathcal{J}_p} y_{p, 0, i} = n_s \quad \forall s \in \mathcal{S}. \quad (24)$$

Each pilot has at most one starting activity in the planning horizon (constraints (15)). Constraints (16) and (17) enforce the activity flow balance for each pilot. Constraints (18) link the z and y variables. Constraints (19) ensure that if a task i starts at time step t , then it should be performed by a pilot who works in a shift that contains the time step. A rest period has to be arranged for a pilot that works in the planning horizon, and this rest period should start within the

allowable time window (constraints (20)). Constraints (21) require that the rest period be included in the activity flow of a pilot if the pilot works in the planning horizon. The minimum setup time between two consecutive activities performed by one pilot is imposed by constraints (22). Constraints (23) avoid infeasible setup times in the event that the duration of a rest period is shorter than the time required to reposition a pilot between two tasks that start right before and immediately after the rest period. We calculate the number of pilots required to work in each shift in constraints (24). The domains for the decision variables are defined in the constraints listed below:

$$u_{k,b} \in \{0, 1\} \quad \forall b \in \mathcal{B}_k, k \in \mathcal{K} \quad (25)$$

$$v_{k,t}^{in}, v_{k,t}^{out} \in \{0, 1\} \quad \forall k \in \mathcal{K}, t \in \mathcal{T} \quad (26)$$

$$w_{k,b,t} \in \{0, 1\} \quad \forall k \in \mathcal{K}, b \in \mathcal{B}, t \in \mathcal{T} \quad (27)$$

$$x_{i,t} \in \{0, 1\} \quad \forall t \in \mathcal{E}_i, i \in \mathcal{I} \cup \mathcal{J} \quad (28)$$

$$y_{p,0,j}, y_{p,i,j}, y_{p,i,0} \in \{0, 1\} \quad \forall i, j \in \mathcal{I} \cup \mathcal{J}_p, i \neq j, p \in \mathcal{P} \quad (29)$$

$$z_{s,i} \in \{0, 1\} \quad \forall i \in \mathcal{I}, s \in \mathcal{S} \quad (30)$$

$$n_s \in \mathbb{Z}^+ \quad \forall s \in \mathcal{S}. \quad (31)$$

Our model is formulated for the VSPP faced by a seaport with a bidirectional channel that has a constant nautical condition in each direction. However, the model can easily be adapted to solve the VSPP under different channel configurations. For example, the channels in some seaport are unidirectional (i.e., vessels must sail in the same direction in these channels). To handle the VSPP under such a channel configuration, one can set \mathcal{U} in the model to include all task pairs that consist of one task corresponding to a vessel entering berths and another corresponding to a vessel leaving berths. In addition, some seaports have channels with multiple segments, some bidirectional and others unidirectional (Corry and Bierwirth 2019). To handle such cases, one can extend \mathcal{U} to be $\mathcal{U}' := \{(i, t, j, t') \in \mathcal{I} \times \mathcal{T} \times \mathcal{I} \times \mathcal{T} \mid i \text{ cannot start at time step } t \text{ if } j \text{ starts at time step } t', i, j \in \mathcal{I}, t, t' \in \mathcal{T}\}$. By carefully setting \mathcal{U}' , one can avoid having two vessels sail in opposite directions in a unidirectional segment at the same time.

3.4. Valid Inequalities

Model M1 can be strengthened by several families of valid inequalities. The first family of inequalities tightens the model by shrinking the time windows for starting the tasks. This is achieved by considering the precedence relationship between tasks and the requirements regarding the rest period in each shift. In particular, a vessel cannot leave its berth before completing cargo handling, and the earliest time step when vessel k can complete cargo handling (denoted by \underline{F}_k^{out}) can be

calculated by $\underline{F}_k^{out} = \min_{b \in \mathcal{B}_k} \{\max\{\underline{E}_i + d_i, a_b\} + h_{k,b}\}$, where i denotes the pilotage task associated with vessel k sailing into berths. Similarly, to ensure that vessels leave their berths within the given time windows, vessel k must start sailing into the berths no later than $\bar{F}_k^{in} = \bar{E}_j - \min_{b \in \mathcal{B}_k} \{h_{k,b}\} - d_i$, where i and j represent the pilotage tasks associated with vessel k sailing into and out of berths, respectively.

In addition, considering the requirements regarding rest periods, any pilot working in shift s cannot start performing task i at time step t if $t < \underline{L}_s + g_s$ (which disallows resting before the task) and $t > \bar{L}_s - d_i$ (which disallows resting after the task). Based on this observation, for each $i \in \mathcal{I}$ and $s \in \mathcal{S}$, we let $\mathcal{Q}_{s,i} := \{t | t < \underline{L}_s + g_s, t > \bar{L}_s - d_i, t \in \mathcal{E}_i\}$, which is the set of time steps in shift s when task i cannot start. Define $\bar{\mathcal{Q}}_i := \bigcap_{s \in \mathcal{S}} \mathcal{Q}_{s,i}$. It can be readily seen that the following inequalities are valid for the VSPP:

$$x_{i,t} = 0 \quad \forall t \in \mathcal{E}_i \cap \mathbb{Z}_{[1, \underline{F}_k^{out} - 1]}, i \in \mathcal{I}^{out} \cap \mathcal{I}_k, k \in \mathcal{K} \quad (32)$$

$$x_{i,t} = 0 \quad \forall t \in \mathcal{E}_i \cap \mathbb{Z}_{[\bar{F}_k^{in} + 1, \bar{T}]}, i \in \mathcal{I}^{in} \cap \mathcal{I}_k, k \in \mathcal{K} \quad (33)$$

$$x_{i,t} = 0 \quad \forall t \in \mathcal{E}_i \cap \bar{\mathcal{Q}}_i, i \in \mathcal{I}. \quad (34)$$

In the second family of valid inequalities, we tighten M1 by using the relationship between x and z in any *optimal* solution to M1. These inequalities are as follows:

$$z_{s,i} \leq \sum_{t \in \mathcal{T}_s \cap \mathcal{E}_i} x_{i,t} \quad \forall i \in \mathcal{I}, s \in \mathcal{S}. \quad (35)$$

The constraints enforce $z_{s,i} = 0$ if i does not start at any time step in shift s , $s \in \mathcal{S}, i \in \mathcal{I}$.

Finally, because all pilots in \mathcal{P}_s are identical $\forall s \in \mathcal{S}$, we can further strengthen the formulation by breaking symmetries in pilot scheduling. To this end, we present \mathcal{P}_s and \mathcal{I} as $\mathcal{P}_s = \{\underline{P}_s, \underline{P}_s + 1, \dots, \bar{P}_s\}$ and $\mathcal{I} = \{1, 2, \dots, |\mathcal{I}|\}$, respectively. By following Jans (2009) and Adulyasak et al. (2014), we introduce the third family of valid inequalities as follows:

$$\sum_{j \in \mathcal{I} \cup \mathcal{J}_p} y_{p,0,j} \geq \sum_{j \in \mathcal{I} \cup \mathcal{J}_{p+1}} y_{p+1,0,j} \quad \forall \underline{P}_s \leq p \leq \bar{P}_s - 1, s \in \mathcal{S} \quad (36)$$

$$\sum_{i'=1}^i (y_{p,0,i'} + \sum_{j \in \mathcal{I} \cup \mathcal{J}_p \setminus \{i'\}} y_{p,j,i'}) 2^{(i-i')} \geq \sum_{i'=1}^i (y_{p+1,0,i'} + \sum_{j \in \mathcal{I} \cup \mathcal{J}_{p+1} \setminus \{i'\}} y_{p+1,j,i'}) 2^{(i-i')} \quad \forall i \in \mathcal{I}, \underline{P}_s \leq p \leq \bar{P}_s - 1, s \in \mathcal{S}. \quad (37)$$

We refer to Model M1 with inequalities (32)–(37) as SM1.

4. A Branch-price-and-Benders-cut Approach

In the previous section, we formulate the VSPP as a compact MILP model. Using a general-purpose optimization solver (e.g., CPLEX), model SM1 can be well solved for instances of small scale. However, this “throwing a model into a solver” method fails to deliver satisfying solutions for large-scale instances. In this section, we propose an exact branch-price-and-Benders-cut (BPBC) approach to solve the VSPP on large instances.

Our approach is based on the set-covering reformulation of SM1 (see Section 4.1). The approach solves the set-covering model by first relaxing the integrality requirements for the columns and gradually imposing these constraints using a branch-and-bound method (see Section 4.4). At each node of the branch-and-bound tree, one thus obtains a linear programming (LP) relaxation of the set-covering problem. We solve the LP relaxation of the set-covering problem at each node by Benders decomposition (Benders 1962).

The master problem in the decomposition can be viewed as the LP relaxation of the *berth allocation problem with vessel traffic flow control*, and the subproblem can be viewed as the LP relaxation of the *pilot scheduling problem* (see Section 4.2). Both the master problem and the subproblem are solved by Dantzig-Wolfe decomposition (Dantzig and Wolfe 1960) through dynamic column(-and-constraint) generation (see Section 4.3).

When a feasible integer solution is generated at a node, an upper bound for the problem is obtained, which can be used to prune branches of the tree. The approach terminates when there are no active nodes in the tree or when the optimality gap is lower than a given threshold. To improve the performance of the approach, we propose a series of acceleration techniques, which are explained in Section 4.5.

Our approach is inspired by the *three-phase Benders decomposition and column generation* method, which was proposed by Cordeau et al. (2001) and has been applied to solve many complicated optimization problems (Mercier et al. 2005, Zeighami and Soumis 2019). Note that this method provides valid lower and upper bounds but does not guarantee to converge to optimality (Cordeau et al. 2001), because the integrality constraints for the subproblem in Benders decomposition are relaxed in the entire process of the method except for the last step, where the integer subproblem is solved only once given a feasible integer solution for the master problem. In many studies (e.g., Mercier et al. 2005, Zeighami and Soumis 2019), however, the optimality gap when the algorithm terminates is sufficiently small to guarantee high-quality solutions.

Our approach extends this method in two directions. First, we design a tailored branching scheme that enables the approach to find an optimal solution to the VSPP. Second, we propose several acceleration strategies that have proven to be very effective in improving the performance of the approach. The framework of the BPBC approach and some acceleration strategies are sufficiently general and adaptable to be applied to solve other similar problems.

4.1. The Set-covering Formulation

In the VSPP, the port stay of a vessel can be depicted as a *vessel route* that records the movements of the vessel in the seaport. Let Ω , indexed by ω , denote the set of feasible vessel routes that satisfy constraints (2)–(6), (9), (13), (14), (25), (26), (28), and (32)–(34). We denote by $\Omega_k \subseteq \Omega$ the routes for vessel k .

For each ω , let $k(\omega)$ denote the associated vessel. We define two binary parameters $\alpha_{\omega,i,t}$ and $\beta_{\omega,b,t,t'}$, $\forall i \in \mathcal{I}, b \in \mathcal{B}, t, t' \in \mathcal{T}, \omega \in \Omega$, where $\alpha_{\omega,i,t}$ is equal to 1 if and only if task i is included in the route and the task starts at time step t , and $\beta_{\omega,b,t,t'}$ is equal to 1 if and only if vessel $k(\omega)$ reaches berth b at time step t and leaves the berth at time step t' . Besides, let \bar{c}_ω denote the cost of the route, which is calculated by $\bar{c}_\omega = \sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{E}_i} \alpha_{\omega,i,t} c_{i,t}^1 + \sum_{b \in \mathcal{B}} \sum_{t \in \mathcal{T}} \sum_{t' \in \mathcal{T}} \beta_{\omega,b,t,t'} c_{k(\omega),b}^2$.

The activities performed by a pilot in a shift can also be represented as a *pilot route* that records the tasks and rest period conducted by the pilot. Let Φ , indexed by ϕ , denote the set of feasible pilot routes that satisfy constraints (15)–(17), (22)–(23), (28) (for all feasible start times of each activity), and (29). Let $\Phi_s \subseteq \Phi$ denote the set of pilot routes for a pilot in shift s . The cost of a pilot route $\phi \in \Phi_s$ is denoted by \bar{c}_ϕ and is equal to c_s^3 . For each ϕ , let $\gamma_{\phi,i,t}$ be a binary parameter set to 1 if and only if task i is covered in the route and the task starts at time step t , $\forall i \in \mathcal{I}, t \in \mathcal{T}$.

In the set-covering model, we use binary decision variables χ_ω to indicate the flows in vessel routes and binary decision variables μ_ϕ to indicate the flows in pilot routes. The model can be formulated as follows:

$$[\mathbf{M2}] \quad \min \sum_{\omega \in \Omega} \chi_\omega \bar{c}_\omega + \sum_{\phi \in \Phi} \mu_\phi \bar{c}_\phi \quad (38)$$

$$\text{s.t.} \quad \sum_{\omega \in \Omega_k} \chi_\omega = 1 \quad \forall k \in \mathcal{K} \quad (39)$$

$$\sum_{\omega \in \Omega} \chi_\omega \sum_{t=1}^{\tau} \sum_{t'=\min\{\tau+1, \bar{T}\}}^{\bar{T}} \beta_{\omega,b,t,t'} \leq 1 \quad \forall \tau \in \mathcal{T}, b \in \mathcal{B} \quad (40)$$

$$\sum_{\omega \in \Omega} \chi_\omega \sum_{i \in \mathcal{I}^{in}} \sum_{t'=\max\{t-f_i+1, 1\}}^t \alpha_{\omega,i,t'} \leq 1 \quad \forall t \in \mathcal{T} \quad (41)$$

$$\sum_{\omega \in \Omega} \chi_\omega \sum_{i \in \mathcal{I}^{out}} \sum_{t'=\max\{t-f_i+1, 1\}}^t \alpha_{\omega,i,t'} \leq 1 \quad \forall t \in \mathcal{T} \quad (42)$$

$$\sum_{\omega \in \Omega} \chi_\omega \sum_{t'=\max\{t-d_i+1, 1\}}^t \alpha_{\omega,i,t'} + \sum_{\omega \in \Omega} \chi_\omega \sum_{t'=\max\{t-d_j+1, 1\}}^t \alpha_{\omega,j,t'} \leq 1 \quad \forall t \in \mathcal{T}, (i, j) \in \mathcal{U} \quad (43)$$

$$\sum_{\phi \in \Phi} \mu_\phi \gamma_{\phi,i,t} - \sum_{\omega \in \Omega} \chi_\omega \alpha_{\omega,i,t} = 0 \quad \forall t \in \mathcal{T}, i \in \mathcal{I} \quad (44)$$

$$\chi_\omega \in \{0, 1\} \quad \forall \omega \in \Omega \quad (45)$$

$$\mu_\phi \in \{0, 1\} \quad \forall \phi \in \Phi. \quad (46)$$

The objective function (38) minimizes the sum of vessel movement scheduling cost and pilot scheduling cost. Each vessel is matched with one and only one vessel route (constraints (39)). The berth capacity limitations are imposed by constraints (40). Constraints (41) and (42) enforce the minimum headway requirements for vessels sailing into the channel when approaching and leaving berths, respectively. The non-simultaneity constraints between tasks are given in (43). Constraints (44) connect vessel routes with pilot routes. Constraints (45) and (46) define the variables to be binary.

The number of feasible vessel routes and pilot routes can be prohibitively large even for small-scale instances, and enumerating all feasible routes is technically impossible. Therefore, our approach solves M2 through column generation (i.e., by gradually generating vessel routes and pilot routes that may be used in an optimal solution). The method starts from a restricted LP relaxation of M2, where only small initial sets of vessel and pilot routes are included (to ensure the feasibility of the model) and the integrality requirements for the routes in constraints (45) and (46) are dropped. Next, vessel routes and pilot routes that may possibly reduce the objective function value are added to the problem, and integrality constraints are imposed gradually by a branch-and-bound method.

At each branch of the branch-and-bound tree, one obtains an LP relaxation of M2. However, due to constraints (44) that couple the vessel routes and the pilot routes, the LP relaxation of M2 is difficult to solve by column generation directly. We observe that once the χ_ω variables are fixed, the problem reduces to a pilot scheduling problem in a (generally) sparse space-time network, which is much easier to solve. Based on this motivation, we propose a Benders decomposition method to solve the LP relaxation of M2. We explain the method in the next section.

4.2. Benders Reformulation

In this section, we explain how to decompose the LP relaxation of M2 at a node in the branch-and-bound tree. We start by formulating the *primal Benders subproblem*, then introduce the dual of it, and finally present the *Benders master problem*.

Given a node in the branch-and-bound tree, let \mathbf{X} be the set of vectors for the χ_ω variables that satisfy $0 \leq \chi_\omega \leq 1, \forall \omega \in \Omega$ and satisfy constraints (39)–(43). For any given vector $\bar{\chi} \in \mathbf{X}$, the LP relaxation of M2 reduces to the following primal Benders subproblem (PBSP) involving only μ_ϕ variables:

$$[\mathbf{PBSP}] \quad \min \sum_{\phi \in \Phi} \mu_\phi \bar{c}_\phi \quad (47)$$

$$\text{s.t.} \quad \sum_{\phi \in \Phi} \mu_\phi \gamma_{\phi,i,t} - \sum_{\omega \in \Omega} \bar{\chi}_\omega \alpha_{\omega,i,t} = 0 \quad \forall i \in \mathcal{I}, t \in \mathcal{T} \quad (48)$$

$$\mu_\phi \leq 1 \quad \forall \phi \in \Phi \quad (49)$$

$$\mu_\phi \geq 0 \quad \forall \phi \in \Phi. \quad (50)$$

When solving the PBSP through column generation, in the pricing subproblem we only need to consider tasks i and time steps t such that $\sum_{\omega \in \Omega} \bar{\chi}_\omega \alpha_{\omega,i,t} > 0$. This enables us to formulate the pricing subproblem in a space-time network with significantly fewer nodes than the original network that contains all time steps for starting a task. Due to this favorable structure, the PBSP can be solved very efficiently.

We let $\boldsymbol{\delta} = (\delta_{i,t} | i \in \mathcal{I}, t \in \mathcal{T})$ and $\boldsymbol{\zeta} = (\zeta_\phi | \phi \in \Phi)$ denote the vectors of dual variables associated with constraints (48) and (49), respectively. The dual of the Benders primal subproblem, called the *dual Benders subproblem* (DBSP), can be formulated as follows:

$$[\text{DBSP}] \quad \max \sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}} \sum_{\omega \in \Omega} \bar{\chi}_\omega \alpha_{\omega,i,t} \delta_{i,t} + \sum_{\phi \in \Phi} \zeta_\phi \quad (51)$$

$$\text{s.t.} \quad \sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}} \gamma_{\phi,i,t} \delta_{i,t} + \zeta_\phi \leq \bar{c}_\phi \quad \forall \phi \in \Phi \quad (52)$$

$$\delta_{i,t} \in \mathbb{R} \quad \forall i \in \mathcal{I}, t \in \mathcal{T} \quad (53)$$

$$\zeta_\phi \leq 0 \quad \forall \phi \in \Phi. \quad (54)$$

PROPOSITION 1. *The DBSP is always feasible and bounded.*

Proposition 1 indicates that it is sufficient to add only *Benders optimality cuts* in the Benders master problem. Let Δ denote the polyhedron defined by constraints (52)–(54), and let Γ_Δ be the set of extreme points of Δ . Introducing the additional variable η , the LP relaxation of M2 can thus be reformulated as the following Benders master problem (BMP):

$$[\text{BMP}] \quad \min \sum_{\omega \in \Omega} \chi_\omega \bar{c}_\omega + \eta \quad (55)$$

$$\text{s.t.} \quad (39)\text{--}(43)$$

$$\eta \geq \sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}} \sum_{\omega \in \Omega} \chi_\omega \alpha_{\omega,i,t} \delta_{i,t} + \sum_{\phi \in \Phi} \zeta_\phi \quad \forall (\boldsymbol{\delta}, \boldsymbol{\zeta}) \in \Gamma_\Delta \quad (56)$$

$$\chi_\omega \leq 1 \quad \forall \omega \in \Omega \quad (57)$$

$$\chi_\omega \geq 0 \quad \forall \omega \in \Omega \quad (58)$$

$$\eta \geq 0. \quad (59)$$

Observe that BMP contains a large number of Benders optimality cuts (56). However, most of these cuts are inactive in an optimal solution. Hence, these constraints need not be generated exhaustively. Instead, an iterative algorithm is used to generate only a subset of cuts that are sufficient for identifying an optimal solution. The algorithm will be explained in Section 4.3.

Algorithm 1 Column and Benders Cut Generation (C&BCG).

-
- 1: Initialize the set of Benders cuts as $\tilde{\Gamma}_\Delta = \Gamma_\Delta^0$. ▷ The algorithm can start with $\Gamma_\Delta^0 = \emptyset$.
 - 2: **while** 1 **do**
 - 3: Solve the BMP with the current set $\tilde{\Gamma}_\Delta$ of Benders cuts by column generation (Section 4.3.1); let Z_{BMP}^* and (χ^*, η^*) denote the optimal objective function value and the optimal solution to the BMP, respectively.
 - 4: Let $\bar{\chi} = \chi^*$, and solve the PBSP by column generation (Section 4.3.2); let Z_{PBSP}^* and μ^* denote the optimal objective function value and the optimal solution to the PBSP, respectively; let (δ^*, ζ^*) denote the vectors of the values of the associated optimal dual variables.
 - 5: **if** $Z_{PBSP}^* > \eta^*$ **then**
 - 6: Update $\tilde{\Gamma}_\Delta = \tilde{\Gamma}_\Delta \cup (\delta^*, \zeta^*)$.
 - 7: **else**
 - 8: Break.
 - 9: **end if**
 - 10: **end while**
 - 11: Return Z_{BMP}^* , χ^* , and μ^* .
-

4.3. Column and Benders Cut Generation

We solve the BMP at each node with a column and Benders cut generation (C&BCG) algorithm. Let Γ_Δ^0 be the initial set of Benders cuts i.e., extreme points (δ, ζ) for the BMP. The framework of the algorithm can be presented in Algorithm 1.

The method to initialize the set of Benders cuts at the beginning of the C&BCG algorithm will be explained in Section 4.5.3. In C&BCG algorithm, both the BMP and the PBSP are solved by Dantzig-Wolfe decomposition (Dantzig and Wolfe 1960), which iterates between a restricted master problem and a pricing subproblem for column generation. We develop efficient polynomial-time algorithms for solving the pricing subproblems in the BMP and PBSP. Details of the column generation methods for the two problems are explained in Sections 4.3.1 and 4.3.2, respectively.

4.3.1. Column Generation for the BMP To define the pricing problems in the Dantzig-Wolfe decomposition of the BMP, we let $\pi = (\pi_k | k \in \mathcal{K})$, $\lambda = (\lambda_{b,t} | t \in \mathcal{T}, b \in \mathcal{B})$, $\theta = (\theta_t | t \in \mathcal{T})$, $\vartheta = (\vartheta_t | t \in \mathcal{T})$, $\kappa = (\kappa_{t,(i,j)} | t \in \mathcal{T}, (i,j) \in \mathcal{U})$, and $\varphi = (\varphi_{(\delta,\zeta)} | (\delta, \zeta) \in \tilde{\Gamma}_\Delta)$ denote the vectors of dual variables associated with constraints (39)–(43) and (56), respectively.

New vessel routes are generated by solving a pricing problem for each vessel $k \in \mathcal{K}$ and each berth that can handle it (i.e., $b \in \mathcal{B}_k$). We denote the pricing problem associated with vessel k and berth b by $\text{MPP}_{k,b}$ (here, the letter “M” represents BMP, and we will use “SPP” to denote pricing problems for the PBSP). The $\text{MPP}_{k,b}$ can be defined as a problem that identifies the arc with the minimum cost in a space-time network $G_{k,b} = (N_{k,b}, A_{k,b})$, where $N_{k,b}$ is the set of nodes and $A_{k,b}$

is the set of arcs. In particular, $N_{k,b} = \{(i, t) | t \in \mathcal{E}_i, i \in \mathcal{I}_k\}$, and $A_{k,b} = \{[(i, t), (i', t')] | t' - t \geq d_i + h_{k,b}, (i, t), (i', t') \in N_{k,b}, i \in \mathcal{I}^{in}, i' \in \mathcal{I}^{out}\}$. The cost of sending a unit flow through arc $[(i, t), (i', t')]$, denoted by $\hat{c}_{[(i,t),(i',t)]}$, is calculated by

$$\begin{aligned} \hat{c}_{[(i,t),(i',t)]} = & c_{i,t}^1 + c_{i',t'}^1 + c_{k,b}^2 - \pi_k - \sum_{\tau=t+d_i}^{\tau=t'-1} \lambda_{\tau,b} - \sum_{\tau=t}^{t+f_i-1} \theta_{\tau} - \sum_{\tau=t'}^{t'+f_{i'}-1} \vartheta_{\tau} - \sum_{j \in \mathcal{I}} \sum_{(i,j) \in \mathcal{U}} \sum_{\tau=t}^{t+d_i-1} \kappa_{\tau,(i,j)} \\ & - \sum_{j \in \mathcal{I}} \sum_{(i',j) \in \mathcal{U}} \sum_{\tau=t'}^{t'+d_{i'}-1} \kappa_{\tau,(i',j)} + \sum_{(\delta,\zeta) \in \bar{\Gamma}_{\Delta}} \varphi_{(\delta,\zeta)}(\delta_{i,t} + \delta_{i',t'}). \end{aligned} \quad (60)$$

Finally, observe that the minimum-cost arc can be found by enumeration, in $\mathcal{O}(|\mathcal{E}_i||\mathcal{E}_{i'}|)$ time, where i and i' denote the tasks associated with vessel k sailing into and out of berth b , respectively. The newly generated columns with negative reduced costs will then be added to the BMP, which will be solved again. This procedure terminates when no routes with negative reduced costs can be detected.

4.3.2. Column Generation for the PBSP In the PBSP, we are given a vector $\bar{\chi}$ that represents the solution of the χ variables for the associated BMP. Given $\bar{\chi}$, let $H = \{(i, t) | \sum_{\omega \in \Omega} \bar{\chi}_{\omega} \alpha_{\omega,i,t} > 0, t \in \mathcal{E}_i, i \in \mathcal{I}\}$, and $H_s = \{(i, t) | s \in \mathcal{S}_t, (i, t) \in H\}, \forall s \in \mathcal{S}$.

New pilot routes for the PBSP are generated by solving $|\mathcal{S}|$ pricing problems, each corresponding to a shift. We denote the pricing problem for shift $s \in \mathcal{S}$ by SPP_s . With a slight abuse of notation, the SPP_s is defined as a shortest-path problem in a space-time network denoted by $G_s = (N_s, A_s)$, where N_s and A_s represent the set of nodes and the set of arcs, respectively. We let $N_s = H_s \cup J_s \cup \{O, D\}$. Here, J_s represents the set of nodes corresponding to starting a rest period. Because the requirements of the rest periods are identical for all pilots working in the same shift, we define J_s as $J_s = \{(i, t), t \in \mathcal{E}_i, i \in \mathcal{J}_p, \exists p \in \mathcal{P}_s\}$. Besides, O and D denote the dummy source and the dummy sink, respectively. Given N_s , we let $A_s^1 = \{[(i, t), (i', t')] | (i, t), (i', t') \in N_s, t + d_i + q_{i,i'} \leq t'\}$, $A_s^2 = \{[O, (i', t')] | (i', t') \in N_s\}$, and $A_s^3 = \{[(i, t), D] | (i, t) \in N_s\}$, and let $A_s = A_s^1 \cup A_s^2 \cup A_s^3$. The SPP_s is equivalent to finding the shortest elementary path in G_s that travels from O to D and contains exactly one node from J_s . The cost of sending a unit flow through any arc $[m, m']$ in G_s (denoted by $\hat{c}_{[m,m']}$) is calculated by

$$\hat{c}_{[m,m']} = \begin{cases} -\delta_{i,t}, & \text{if } [m, m'] \in A_s^1, [m, m'] = [(i, t), (i', t')], (i, t) \in H_s, \\ c_s^3, & \text{if } [m, m'] \in A_s^2, [m, m'] = [O, (i', t')], \\ -\delta_{i,t}, & \text{if } [m, m'] \in A_s^3, [m, m'] = [(i, t), D], (i, t) \in H_s, \\ 0, & \text{otherwise.} \end{cases} \quad (61)$$

To solve the shortest-path problem, we develop a tailored label correcting algorithm that solves the SPP_s in $\mathcal{O}(|H_s|^2)$ time. The algorithm is explained in EC.2 of the electronic companion. The

newly generated columns with negative reduced costs will be added to the PBSP, which will be solved again. This procedure continues until no routes with negative reduced costs can be found.

4.4. Branching Scheme

The optimal solution to the BMP at a node in the tree may have fractional flows in vessel or pilot routes. In this case, the node is branched into two child nodes so that (i) the current fractional solution is infeasible for the BMP at both child nodes and (ii) the overall optimal integer solution for the two child nodes remains the same as that for the BMP at the current node. The detailed branching scheme is explained as follows.

We let χ^* and μ^* denote the optimal solutions to the BMP and the associated PBSP at a node, respectively. To facilitate branching decisions, we make use of the following additional variable vectors: $\Lambda = (\Lambda_{k,b} | b \in \mathcal{B}_k, k \in \mathcal{K})$, $\Xi = (\Xi_{i,t} | t \in \mathcal{E}_i, i \in \mathcal{I})$, $\Pi = (\Pi_{i,s} | s \in \mathcal{S}, i \in \mathcal{I})$, and $\Psi = (\Psi_{i,j} | i, j \in \mathcal{I} \cup \{0\} \cup \mathcal{J}, i \neq j)$. Here, we use 0 to denote the dummy source or sink of a pilot route. The values of these variables are calculated by:

$$\Lambda_{k,b} = \sum_{\omega \in \Omega} \chi_{\omega}^* \sum_{t \in \mathcal{T}} \sum_{t' \in \mathcal{T}} \beta_{\omega,b,t,t'} \quad \forall b \in \mathcal{B}_k, k \in \mathcal{K} \quad (62)$$

$$\Xi_{i,t} = \sum_{\omega \in \Omega} \chi_{\omega}^* \alpha_{\omega,i,t} \quad \forall t \in \mathcal{E}_i, i \in \mathcal{I} \quad (63)$$

$$\Pi_{i,s} = \sum_{\phi \in \Phi_s} \mu_{\phi}^* \sum_{t \in \mathcal{T}_s \cap \mathcal{E}_i} \gamma_{\phi,i,t} \quad \forall s \in \mathcal{S}, i \in \mathcal{I} \quad (64)$$

$$\Psi_{i,j} = \sum_{\phi \in \Phi} \mu_{\phi}^* \sigma_{\phi,i,j} \quad \forall i, j \in \mathcal{I} \cup \{0\} \cup \mathcal{J}, i \neq j, \quad (65)$$

where $\sigma_{\phi,i,j}$ is a binary parameter which equals 1 if and only if route ϕ travels directly from activity i (or the dummy source) to activity j (or the dummy sink).

Let $\Lambda_{\bar{k},\bar{b}}$, $\Xi_{\bar{i},\bar{t}}$, $\Pi_{\bar{i},\bar{s}}$, and $\Psi_{\bar{i},\bar{j}}$ denote the variables that are closest to 0.5 among the variables in vectors Λ , Ξ , Π , and Ψ , respectively. The branching decisions are made as follows:

- If there are fractional $\Lambda_{k,b}$ variables, we separate the current node by generating one child node with the additional constraint $\Lambda_{\bar{k},\bar{b}} = 0$ imposed on the associated BMP and one child node with the additional constraint $\Lambda_{\bar{k},\bar{b}} = 1$ imposed on the associated BMP.
- If all $\Lambda_{k,b}$ variables are integer, but there are fractional $\Xi_{i,t}$ variables, we separate the current node by generating one child node with the additional constraint $\Xi_{\bar{i},\bar{t}} = 0$ imposed on the associated BMP and one child node with the additional constraint $\Xi_{\bar{i},\bar{t}} = 1$ imposed on the associated BMP.
- If all $\Lambda_{k,b}$ and $\Xi_{i,t}$ variables are integer, but there are fractional $\Pi_{i,s}$ variables, we separate the current node by generating one child node with the additional constraint $\Pi_{\bar{i},\bar{s}} = 0$ imposed on the associated PBSP and one child node with the additional constraint $\Pi_{\bar{i},\bar{s}} = 1$ imposed on the associated PBSP.

- If all $\Lambda_{k,b}$, $\Xi_{i,t}$, and $\Pi_{i,s}$ variables are integer, but there are fractional $\Psi_{\bar{i},\bar{j}}$ variables, we separate the current node by generating one child node with the additional constraint $\Psi_{\bar{i},\bar{j}} = 0$ imposed on the associated PBSP and one child node with the additional constraint $\Psi_{\bar{i},\bar{j}} = 1$ imposed on the associated PBSP.

To preserve the structures of the BMP and the PBSP, all branching constraints are imposed on the corresponding pricing problems for generating columns.

In the BPBC approach, we adopt a best-bound strategy for selecting the nodes to solve. In particular, among all unsolved and unfathomed nodes in the branch-and-bound tree, we solve the one that has the minimum objective function value of the associated BMP.

THEOREM 1. *The BPBC approach can generate an optimal solution for model M2.*

4.5. Computational Enhancements

We now describe the computational enhancements that improve the efficiency of the BPBC approach.

4.5.1. Lower-Bound Lifting Inequalities Because parts of the objective function (38) are projected out in the Benders reformulation, the optimality gap of the BMP may be large in the initial stages of the algorithm due to the low quality of the lower bound. A large number of Benders cuts are thus needed to close the gap (Adulyasak et al. 2015). To address this issue, Adulyasak et al. (2015) proposed a method to lift the lower bound of the Benders master problem by using inequalities that contain some information about the parts of the original objective function that were removed. Following this idea, we lift the lower bound of the BMP by using initial cuts, called the lower-bound lifting (LBL) cuts. In particular, we use the LBL cuts to estimate a lower bound of the cost in pilot dispatching and lift the lower bound of the BMP by requiring the variable η to be no smaller than the lower bound of the pilot dispatching cost.

In pilot scheduling, a minimum setup time ($q_{i,j}$) is required when a pilot is repositioned from task i to task j . In the LBL inequalities, we relax the minimum setup time requirement, by setting an identical minimum setup time between task i and any other task (denoted by \underline{q}_i) to be $\underline{q}_i = \min_{j \in \mathcal{I}: j \neq i} \{q_{i,j}\}$. Let $\nu_{i,t,s}$ be a binary variable which is equal to 1 if and only if task i starts at time t and is performed by a pilot in shift s . The following LBL cuts are valid for the BMP:

$$\eta \geq \sum_{s \in \mathcal{S}} c_s^3 n_s \quad (66)$$

$$n_s \geq \sum_{i \in \mathcal{I}} \sum_{t' = \max\{\underline{T}_s, t - d_i - \underline{q}_i + 1\}}^t \nu_{i,t',s} \quad \forall t \in \mathcal{T}_s, s \in \mathcal{S} \quad (67)$$

$$\sum_{s \in \mathcal{S}_t} \nu_{i,t,s} = \sum_{\omega \in \Omega} \chi_{\omega} \alpha_{\omega,i,t} \quad \forall t \in \mathcal{E}_i, i \in \mathcal{I} \quad (68)$$

$$\nu_{i,t,s} \leq 1 \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, s \in \mathcal{S} \quad (69)$$

$$\nu_{i,t,s} \geq 0 \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, s \in \mathcal{S} \quad (70)$$

$$n_s \geq 0 \quad \forall s \in \mathcal{S}. \quad (71)$$

Constraint (66) provides a lower bound for the pilot dispatching cost. Constraints (67) calculate the number of pilots required to work in each shift. Constraints (68) synchronize the scheduling of tasks and the assignment of pilots. The domains of the $\nu_{i,t,s}$ and n_s variables are defined in the last three sets of constraints (they are relaxed to be continuous variables).

We can lift the LBL inequalities by tightening constraints (67) to be

$$n_s \geq \sum_{i \in \mathcal{I}} \sum_{t' = \max\{\underline{T}_s, t - d_i - q'_{i,t,s} + 1\}}^t \nu_{i,t',s} \quad \forall t \in \mathcal{T}_s, s \in \mathcal{S}. \quad (72)$$

For each $i \in \mathcal{I}, t \in \mathcal{T}_s, s \in \mathcal{S}$, $q'_{i,t,s} \geq \underline{q}_i$ and is set as follows:

$$q'_{i,t,s} = \begin{cases} \bar{T}, & \text{if } \bar{\mathcal{I}}_{i,t,s} = \emptyset, \\ \min_{j \in \bar{\mathcal{I}}_{i,t,s}} \min_{t' \in \mathbb{Z}_{[t+d_i+q_{i,j}, \bar{T}_s]} \cap \mathcal{E}_j} t' - d_i, & \text{if } \bar{\mathcal{I}}_{i,t,s} \neq \emptyset, \end{cases}$$

where $\bar{\mathcal{I}}_{i,t,s} = \{j | j \in \mathcal{I}, \mathbb{Z}_{[t+d_i+q_{i,j}, \bar{T}_s]} \cap \mathcal{E}_j \neq \emptyset, j \neq i\}$. Observe that $\bar{\mathcal{I}}_{i,t,s}$ is the set of tasks that can be performed by a pilot working in shift s after completing task i which starts at time step t in this shift.

PROPOSITION 2. *The LBL inequalities (66) and (68)–(72) are valid for the BMP.*

Finally, we note that after adding these inequalities, the BMP has been changed, not only with more decision variables (columns) but also with more constraints. As a sequence, the pricing problems and their solution algorithms should also be modified to incorporate the changes. Consider the pricing problem for vessel k and berth b (denoted by $\text{MPP}'_{k,b}$). The $\text{MPP}'_{k,b}$ can be defined as a problem that identifies the arc with the minimum cost in a three-dimensional task-time-shift network. We present the details of the pricing problems for the BMP with the LBL cuts in EC.3 of the electronic companion.

To solve the $\text{MPP}'_{k,b}$, we extend the enumeration method for the original pricing problem $\text{MPP}_{k,b}$ by not only enumerating all feasible pairs of start times for the vessel to sail into and out of the berth but also enumerating all feasible pilot assignment patterns (in terms of working shifts) for the associated pilotage tasks. Observe that this enumeration method for solving the $\text{MPP}'_{k,b}$ still has polynomial time complexity and solves the problem in $\mathcal{O}(|\mathcal{E}_i| |\mathcal{E}_{i'}| |\mathcal{S}|^2)$ time, where i and i' denote the tasks corresponding to vessel k sailing into and out of berth b , respectively.

4.5.2. Variable Fixing In each iteration of the column generation procedure for solving the BMP, one has to solve $\sum_{k \in \mathcal{K}} |\mathcal{B}_k|$ pricing problems. In this section, we show that the column generation procedure can be accelerated by fixing the value of χ_ω at zero if χ_ω cannot take value 1 in an optimal integer solution for M2.

When solving the BMP at a node in the branch-and-bound tree, at each stage of column generation, we obtain (i) the optimal objective function value denoted by Υ , and (ii) the minimum reduced cost obtained by solving pricing problem $\text{MPP}_{k,b}$ (or $\text{MPP}'_{k,b}$) which is denoted by $r_{k,b}$, where $k \in \mathcal{K}, b \in \mathcal{B}_k$. Then we have:

LEMMA 1. $\Upsilon + \sum_{k \in \mathcal{K}} \sum_{b \in \mathcal{B}_k: r_{k,b} < 0} r_{k,b}$ is a lower bound for the BMP at the current node.

For notational simplicity, let $LB = \Upsilon + \sum_{k \in \mathcal{K}} \sum_{b \in \mathcal{B}_k: r_{k,b} < 0} r_{k,b}$, and let $\Omega_{k,b}$ denote the set of columns corresponding to vessel k being handled at berth b , where $b \in \mathcal{B}_k, k \in \mathcal{K}$. We further have:

PROPOSITION 3. Given a pair of b and k such that $b \in \mathcal{B}_k, k \in \mathcal{K}$ and $r_{k,b} \geq 0$, $LB + r_{k,b}$ gives a lower bound for the BMP with the constraint $\sum_{\omega \in \Omega_{k,b}} \chi_\omega = 1$ at the current node and any child nodes generated from this node.

Suppose that UB is the best (integer) upper bound for model M2. Then, consider a pair of b and k that satisfies $LB + r_{k,b} \geq UB$. In this case, in any optimal solution for the BMP with the constraint $\sum_{\omega \in \Omega_{k,b}} \chi_\omega = 1$ at the current node, the resulting objective function value is no smaller than the upper bound. Note that due to constraints (39), in an integer solution for the BMP at any node, the value of $\sum_{\omega \in \Omega_{k,b}} \chi_\omega$ either takes 1 or 0. Hence, in any integer solution for the BMP with the objective function value smaller than UB at the current node, one must have $\sum_{\omega \in \Omega_{k,b}} \chi_\omega = 0$. To accelerate the column generation procedure, we “fix” $\chi_\omega = 0, \forall \omega \in \Omega_{k,b}$ by not solving the pricing problem $\text{MPP}_{k,b}$ (or $\text{MPP}'_{k,b}$) at the current node and any child nodes generated from this node.

4.5.3. Warm Start for the BMP We warm start the C&BCG algorithm for solving the BMP at a node by using a set of initial Benders cuts in the first iteration of the algorithm and by using a set of initial columns in each subsequent iteration.

Let Θ_ι denote the set of branching constraints regarding the variables in $\mathbf{\Pi}$ and $\mathbf{\Psi}$, as defined in Section 4.4, imposed on the PBSP at a node denoted by ι . The method for initializing Benders cuts for the BMP is based on Proposition 4 and Corollary 1.

PROPOSITION 4. For two nodes ι and ι' , if we have $\Theta_{\iota'} \subseteq \Theta_\iota$, then the Benders cuts that are valid for the BMP at node ι' are also valid for the BMP at node ι .

COROLLARY 1. The Benders cuts that are valid for the BMP at the root node are also valid for the BMP at any other node.

When solving the BMP at node ι , we let the initial set of Benders cuts (denoted by Γ_{Δ}^0 in the C&BCG algorithm) include all Benders cuts that are generated for solving the BMP at any node ι' such that $\Theta_{\iota'} \subseteq \Theta_{\iota}$ (including the root node).

Moreover, in each iteration of the C&BCG algorithm (except for the first iteration), we also initialize a set of columns for the BMP. In particular, the set of initial columns, denoted by Ω_0 , is set as $\Omega_0 = \{\omega | \chi_{\omega}^* > 0, \omega \in \tilde{\Omega}\}$, where $\tilde{\Omega}$ and χ_{ω}^* denote the set of columns (vessel routes) generated for solving the BMP in the previous iteration in the algorithm and their optimal solutions, respectively.

4.5.4. Primal Heuristics A high-quality upper bound for M2 helps prune branches in the branch-and-bound tree in the early stages. It can also accelerate the column generation procedure for solving the BMP through variable fixing. To quickly identify high-quality upper bounds, the approach leverages two methods to generate feasible integer solutions for M2.

The first method detects integer solutions within the C&BCG algorithm for solving the BMP. In particular, in any iteration of the algorithm, let (χ^*, η^*) and Z_1^* be the optimal solution and the optimal objective function value of the BMP, respectively, and let μ^* and Z_2^* be the optimal solution and the optimal objective function value of the associated PBSP, respectively. If all values in χ^* and μ^* are integral, then we obtain a feasible integer solution (χ^*, μ^*) to M2 and a corresponding upper bound equal to $Z_1^* - \eta^* + Z_2^*$.

The second method is used each time the C&BCG algorithm completes solving the BMP and delivers a fractional solution for it. In this case, we run a heuristic that tries to construct a feasible integer solution for M2 based on the current fractional solution. Details of the heuristic are explained in EC.4 of the electronic companion.

4.5.5. Parallel Computing for Pricing Problems We also accelerate the approach by exploiting multi-thread computing. In each iteration of the column generation procedure for the BMP or the PBSP, the approach solves a set of independent pricing problems. Although all of the pricing problems for the BMP and the PBSP have polynomial time complexity, their large number still hinders the solution process. To alleviate this computational burden, our approach solves multiple pricing problems simultaneously by using parallel computing.

5. Computational Experiments

We now perform extensive computational experiments to confirm the applicability and effectiveness of our model and algorithm. The experiments include four parts. In the first part, we examine the impacts of the acceleration techniques on the performance of the BPBC approach. In the second part, we compare the performance of the approach with that of a standard MILP solver applied to models M1 and SM1. In the third part, we evaluate the robustness of the BPBC approach

against variations in problem settings. In the fourth part, we analyze the value of the integration by comparing the results delivered by our approach with those obtained by a method that solves the BAP and the PPP sequentially.

We performed the experiments on an Intel Core i7 2.20 GHz PC with 32 GB RAM, and the pricing problems for the BMP and the PBSP were solved in a six-thread environment (i.e., at most six pricing problems are solved simultaneously). All algorithms were coded in C++ calling CPLEX 12.6. The time limit for any approach to solve any instance was set to 3,600 seconds.

5.1. Instance Generation

We created a set of instances for the experiments based on real operational data from the Hong Kong Container Port, which is the eighth largest container port in the world (The Marine Department of Hong Kong 2020a).

Let L , B , and V be the length of the planning horizon (days), the number of berths, and the number of vessel arrivals in an instance, respectively. In all instances, we let $L \in \{1, 2\}$, $B \in \{10, 15\}$. Then, given each combination of L and B , we set $V = R \cdot L \cdot B$, where R is the vessel arrival rate (vessel arrivals per berth per day) and $R \in \{1.0, 1.2, 1.4\}$. Hence, there are 12 combinations of L , B , and V . For each combination, we created five random instances, yielding 60 instances in total. In all instances, the unit time is set to 10 minutes; i.e., a time step lasts 10 minutes, and a day contains 144 time steps. A day is divided into three shifts, covering 0:00-8:00, 8:00-16:00, and 16:00-24:00. The planning horizon in an instance starts from 0:00 on a given day. Details of the operational data of the Hong Kong Container Port and the settings of other parameters in the instances are explained in EC.5 of the electronic companion.

5.2. Impacts of the Computational Enhancements

We proposed five acceleration strategies in Section 4.5 to improve the performance of the BPBC approach. Table 1 reports the performance of the BPBC approach on the 60 instances when all of these acceleration strategies were used and also when each of them was not applied. The first column shows the settings of (L, B, V) in a group of five instances. Column BPBC indicates the BPBC with all enhancements, and other columns show the results when different computational enhancements were not used. The abbreviations LBL, VF, WS, PH, and PC correspond to the lower-bound lifting inequalities, variable fixing, warm start for the BMP, primal heuristics, and parallel computing for pricing problems, respectively. For each group of instances, we report the number of instances solved to optimality, the average optimality gap (in percentage terms), and the average computational time (in CPU seconds) for each solution method in the corresponding columns Opt, Gap, and CPU, respectively.

Table 1 Average Computational Results of the BPBC Approaches.

(L, B, V)	BPBC			No LBL			No VF			No WS			No PH			No PC		
	Opt	Gap	CPU	Opt	Gap	CPU	Opt	Gap	CPU	Opt	Gap	CPU	Opt	Gap	CPU	Opt	Gap	CPU
(1, 10, 10)	5	0.0	10.1	5	0.0	24.0	5	0.0	10.9	5	0.0	23.3	5	0.0	16.3	5	0.0	11.8
(1, 10, 12)	5	0.0	5.2	5	0.0	26.9	5	0.0	6.0	5	0.0	12.8	5	0.0	5.8	5	0.0	7.2
(1, 10, 14)	5	0.0	35.9	5	0.0	92.2	5	0.0	45.0	5	0.0	76.7	5	0.0	41.8	5	0.0	47.6
(2, 10, 20)	5	0.0	149.6	5	0.0	249.1	5	0.0	220.0	5	0.0	885.2	5	0.0	419.2	5	0.0	177.4
(2, 10, 24)	5	0.0	254.7	5	0.0	455.8	5	0.0	287.0	5	0.0	1084.9	5	0.0	353.7	5	0.0	333.0
(2, 10, 28)	3	0.1	1646.3	3	0.2	1930.6	3	0.2	1798.6	3	0.2	2313.6	2	40.1	2297.1	3	0.1	1702.3
(1, 15, 15)	5	0.0	94.0	5	0.0	160.3	5	0.0	135.8	5	0.0	321.8	5	0.0	130.9	5	0.0	94.8
(1, 15, 18)	5	0.0	26.2	5	0.0	105.0	5	0.0	30.2	5	0.0	83.0	5	0.0	30.5	5	0.0	44.3
(1, 15, 21)	5	0.0	276.6	5	0.0	431.7	5	0.0	351.5	4	0.0	1703.3	5	0.0	378.5	5	0.0	350.8
(2, 15, 30)	5	0.0	1152.8	4	0.0	1591.9	4	0.0	1241.6	3	0.2	2169.7	3	40.0	1634.3	5	0.0	1376.2
(2, 15, 36)	4	0.0	1470.2	4	0.0	1826.8	3	0.0	1620.0	2	0.5	2527.5	3	20.0	1705.1	3	0.0	1649.8
(2, 15, 42)	2	0.8	2548.7	1	1.0	3179.2	2	0.9	2686.7	0	1.5	3600.0	2	60.0	2699.0	2	0.8	2697.2
Total	54	0.1	639.2	52	0.1	839.5	52	0.1	702.8	47	0.2	1233.5	50	13.3	809.4	53	0.1	707.7

Note 1. We take the optimality gap as 100% if a method failed to obtain a feasible solution for an instance.

Note 2. Abbreviations: LBL: lower-bound lifting inequalities; VF: variable fixing; WS: warm start for the BMP; PH: primal heuristics; PC: parallel computing for pricing problems.

The results confirm that all acceleration strategies can significantly improve the performance of the BPBC approach. In particular, the total number of instances solved to optimality increased by 2, 2, 7, 4, and 1, and the average computational time decreased by 23.9%, 9.1%, 48.2%, 21.0%, and 9.7%, respectively, after incorporating LBL, VF, WS, PH, and PC. In addition, the use of primal heuristics enabled the BPBC approach to generate feasible solutions for all instances, which results in a significant decrease in the average optimality gap.

5.3. Comparisons with CPLEX

In this section, we compare the performance of the BPBC approach with methods that use optimization solvers to solve MILP models M1 (the original formulation) and SM1 (M1 with valid inequalities [32]–[37]). We adopted CPLEX 12.6 as the optimization solver.

For the comparisons, we used 15 instances from three groups with $L = 1$, $B = 10$, and $V \in \{10, 12, 14\}$. Table 2 summarizes the results produced by the BPBC approach and CPLEX on M1 and SM1. For each method, we show the number of instances for which feasible solutions were found in column Fs1 and the number of instances solved to optimality in column Opt. Columns Gap and CPU report the average optimality gap in percentage terms and the average computational time for solving the instances in each group, respectively.

For the 15 instances, CPLEX on M1 produced feasible solutions to only one instance within the time limit. The computational time of CPLEX on M1 for each instance reached 3,600 seconds, and the average optimality gap exceeds 93%. In comparison, CPLEX on SM1 found feasible solutions for 13 instances, and four were solved to optimality within the time limit. The average optimality gap of the method is 35.8%, which is significantly smaller than that generated by the solver on M1.

Table 2 Average Computational Results of the BPBC Approach and CPLEX.

(L, B, V)	BPBC				M1				SM1			
	Fsl	Opt	Gap	CPU	Fsl	Opt	Gap	CPU	Fsl	Opt	Gap	CPU
(1, 10, 10)	5	5	0.0	10.1	1	0	81.6	3600.0	5	4	0.4	1688.8
(1, 10, 12)	5	5	0.0	5.2	0	0	100.0	3600.0	5	0	6.8	3600.0
(1, 10, 14)	5	5	0.0	35.9	0	0	100.0	3600.0	3	0	49.4	3600.0
Total	15	15	0.0	17.1	1	0	93.9	3600.0	13	4	35.8	2962.9

Note. We take the optimality gap as 100% if a method failed to obtain a feasible solution for an instance.

For instances with 10 vessels, the average solution time of CPLEX on SM1 is also much shorter than that needed by the solver on M1. These indicate that the valid inequalities (32)–(37) can significantly improve the performance of an MILP solver for solving the instances.

Moreover, as shown in Table 2, the BPBC approach can solve all 15 instances to optimality, and the average computational time is 17.1 seconds, which is 173 times faster than CPLEX on SM1. Hence, the BPBC approach significantly outperforms CPLEX for solving the VSPP instances.

5.4. Sensitivity Analyses

In this section, we evaluate the performance of the BPBC approach and the solution quality when the problem settings are changed.

To this end, we solved the 15 instances with $L = 1$ and $B = 10$ under 20 pilot scheduling settings. Here, a pilot scheduling setting is associated with a shift arrangement and a cost for assigning a pilot to work in a shift. In particular, we considered four shift arrangements in which shifts still last eight hours but the intervals between the start times of two consecutive shifts are set as one, two, four, and eight hours, respectively. Note that a day contains 24, 12, six, and three shifts when the intervals between the start times of two consecutive shifts are one, two, four, and eight hours, respectively, and that shifts can overlap in the first three settings. In addition, we considered five pilot dispatching costs by setting the cost of assigning a pilot to work in a shift (i.e., c_s^3) to 48, 60, 72, 84, and 96, respectively. In all instances, the planning horizons start at 0:00 on a given day, and the first shifts start at the same time.

The results are shown in Table 3. v denotes the interval (hours) between the start times of two consecutive shifts. The second column shows the settings of (L, B, V) in a group of five instances. For each group of instances under each setting (with respect to v and c_s^3), we report the number of instances solved to optimality, the average optimality gap (in percentage terms), and the average computational time (in CPU seconds) obtained by the BPBC approach in the corresponding columns Opt, Gap, and CPU, respectively.

The results demonstrate that the BPBC approach can efficiently solve instances under various pilot scheduling settings. In particular, the algorithm generated optimal solutions in 274 of the

Table 3 Average Computational Results on Instances Under Different Settings.

		$c_s^3 = 48$			$c_s^3 = 60$			$c_s^3 = 72$			$c_s^3 = 84$			$c_s^3 = 96$		
(L, B, V)		Opt	Gap	CPU	Opt	Gap	CPU	Opt	Gap	CPU	Opt	Gap	CPU	Opt	Gap	CPU
$v = 1$	(1, 10, 10)	5	0.0	37.7	5	0.0	53.8	5	0.0	383.1	5	0.0	1195.2	4	0.1	1803.2
	(1, 10, 12)	5	0.0	83.1	5	0.0	573.6	5	0.0	957.5	5	0.0	1559.2	3	0.5	2517.3
	(1, 10, 14)	5	0.0	287.2	5	0.0	618.2	4	0.0	1950.2	2	0.4	2338.1	2	0.5	2519.3
	Total	15	0.0	136.0	15	0.0	415.2	14	0.0	1096.9	12	0.1	1697.5	9	0.4	2279.9
$v = 2$	(1, 10, 10)	5	0.0	15.3	5	0.0	25.2	5	0.0	89.0	5	0.0	357.3	4	0.0	1392.1
	(1, 10, 12)	5	0.0	20.7	5	0.0	51.6	5	0.0	266.1	4	0.0	953.5	4	0.1	1555.6
	(1, 10, 14)	5	0.0	68.3	5	0.0	378.4	4	0.1	1481.3	2	0.4	2426.3	1	1.0	2887.0
	Total	15	0.0	34.8	15	0.0	151.7	14	0.0	612.2	11	0.1	1245.7	9	0.4	1944.9
$v = 4$	(1, 10, 10)	5	0.0	17.6	5	0.0	71.3	5	0.0	194.6	5	0.0	701.8	4	0.0	1595.7
	(1, 10, 12)	5	0.0	17.0	5	0.0	45.2	5	0.0	120.1	5	0.0	436.1	4	0.0	1305.1
	(1, 10, 14)	5	0.0	26.9	5	0.0	107.2	5	0.0	355.1	4	0.0	1266.2	3	0.5	1962.6
	Total	15	0.0	20.5	15	0.0	74.6	15	0.0	223.3	14	0.0	801.4	11	0.2	1621.1
$v = 8$	(1, 10, 10)	5	0.0	10.1	5	0.0	31.5	5	0.0	78.3	5	0.0	215.9	5	0.0	608.4
	(1, 10, 12)	5	0.0	5.2	5	0.0	17.2	5	0.0	47.3	5	0.0	150.9	5	0.0	848.7
	(1, 10, 14)	5	0.0	35.9	5	0.0	71.0	5	0.0	201.7	5	0.0	569.0	5	0.0	1599.2
	Total	15	0.0	17.1	15	0.0	39.9	15	0.0	109.1	15	0.0	311.9	15	0.0	1018.8

300 tests within the time limit (one hour). The average optimality gaps are all within 0.5%. These results indicate that the approach can generate high-quality solutions for instances under different settings within a reasonable time. We can also see from the results that instances with fewer shifts and lower pilot dispatching costs are generally easier to solve. In particular, the approach provided optimality certificates to all instances with $c_s^3 \leq 60$ or $v = 8$.

To further examine the impacts of varying the settings of c_s^3 and v on the solution structure, we compare the average numbers of working pilots and the average objective function values of the instances under different settings. The results are shown in Figure 3.

As shown in Figure 3, the average number of working pilots decreases and the average objective function value (total cost) increases with the increase in dispatching cost. The figure also demonstrates that a smaller v is associated with fewer working pilots and a lower total cost of vessel services.

5.5. Benefits of Integration

The last part of the experiments focuses on the benefits of solving the BAP and the PPP in the VSPP using an integrated optimization method. To this end, we developed a method that solves the VSPP by solving the BAP with vessel traffic flow control and the resulting PPP sequentially. We used this sequential method to solve the 15 instances from three groups with $L = 1$, $B = 10$, and $V \in \{10, 12, 14\}$ under 20 pilot scheduling settings. This led to 300 tests. Table 4 compares the average results obtained by the integrated optimization method (the BPBC approach) and the sequential method for solving instances in each group and under each pilot scheduling setting.

Column #P(S) shows the average numbers of working pilots in the solutions generated by the sequential method. The average numbers of working pilots in the solutions generated by the BPBC

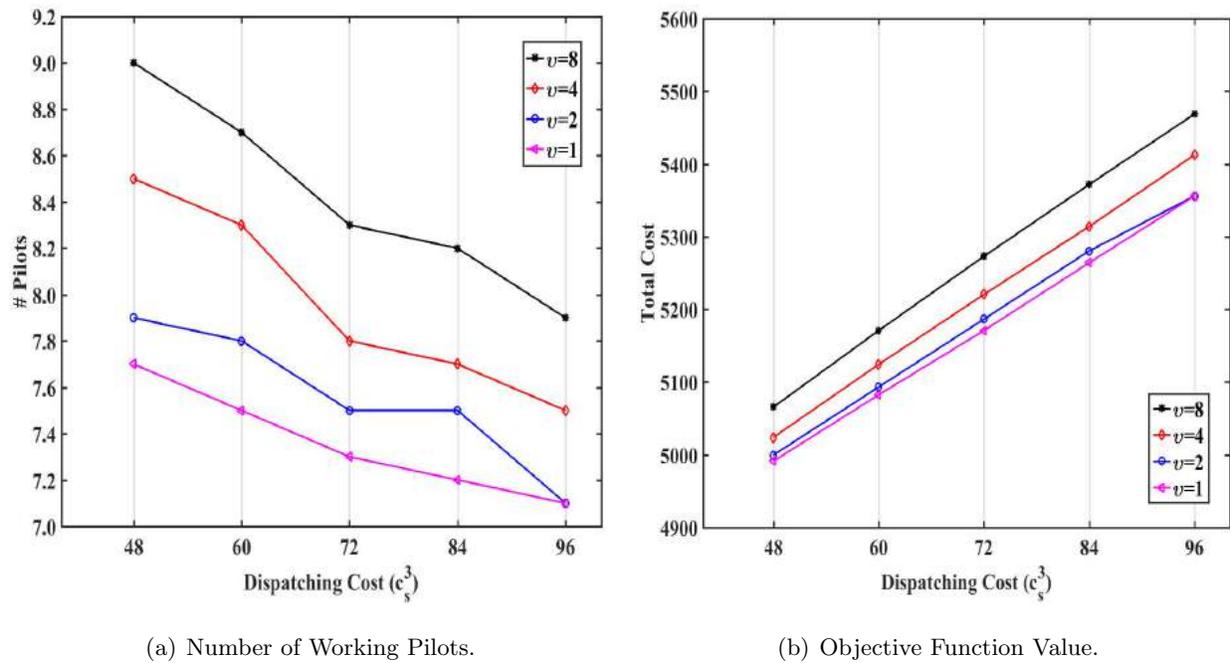


Figure 3 Average Number of Working Pilots and Objective Function Value Under Different Settings.

approach under different settings are presented in columns #P(I). For a certain instance under a certain pilot scheduling setting, let ov_S and ov_I be the objective function values associated with the solutions delivered by the sequential method and the BPBC approach, respectively. We calculated the percentage gap between ov_S and ov_I by $100(ov_S - ov_I)/ov_S$. Columns Gap(%) report the average of these gaps for different instance groups under different settings.

Table 4 Value of Integration under Different Settings.

	(L, B, V)	$c_s^3 = 48$		$c_s^3 = 60$		$c_s^3 = 72$		$c_s^3 = 84$		$c_s^3 = 96$		
		#P(S)	#P(I) Gap(%)									
$v = 1$	(1, 10, 10)	9.8	6.8	1.7	6.8	2.5	6.6	3.3	6.6	4.0	6.6	4.7
	(1, 10, 12)	10.6	8.0	1.6	7.6	2.2	7.3	2.9	7.2	3.6	6.8	3.9
	(1, 10, 14)	11.8	8.2	1.5	8.0	2.2	8.0	2.9	7.8	3.3	8.0	3.9
	Total	10.7	7.7	1.6	7.5	2.3	7.3	3.0	7.2	3.6	7.1	4.2
$v = 2$	(1, 10, 10)	9.8	7.0	1.6	6.8	2.4	6.6	3.2	6.6	3.9	6.6	4.6
	(1, 10, 12)	10.8	8.0	1.7	8.0	2.3	7.8	2.9	7.4	3.6	6.8	4.3
	(1, 10, 14)	12.4	8.6	1.8	8.6	2.5	8.2	3.1	8.4	3.6	8.6	3.8
	Total	11.0	7.9	1.7	7.8	2.4	7.5	3.1	7.5	3.7	7.3	4.2
$v = 4$	(1, 10, 10)	10.2	8.0	1.5	7.7	2.1	7.5	2.7	7.4	3.4	6.8	4.1
	(1, 10, 12)	11.4	8.6	1.6	8.4	2.2	7.6	2.9	7.4	3.7	7.4	4.5
	(1, 10, 14)	12.8	8.8	1.9	8.8	2.7	8.4	3.4	8.2	4.1	8.2	4.5
	Total	11.5	8.5	1.6	8.3	2.3	7.8	3.0	7.7	3.7	7.5	4.4
$v = 8$	(1, 10, 10)	11.2	8.0	2.0	7.6	2.9	7.4	3.7	7.4	4.6	7.4	5.4
	(1, 10, 12)	12.2	8.8	1.9	8.8	2.6	7.8	3.5	7.8	4.3	7.4	5.2
	(1, 10, 14)	15.0	10.2	2.3	9.8	3.2	9.8	4.1	9.4	5.0	8.8	5.8
	Total	12.8	9.0	2.0	8.7	2.9	8.3	3.8	8.2	4.6	7.9	5.5

As shown in Table 4, compared with the sequential method, the solutions generated by the BPBC approach require three to five fewer pilots on average, resulting in a 30% to 40% average reduction in the total pilot dispatching cost. Depending on the pilot scheduling settings, the BPBC approach reduces total vessel service costs by an average of 1.6% to 5.5% compared with the sequential method. These results demonstrate that by solving the BAP and the PPP jointly, the BPBC approach generates significant benefits for port authorities compared with a sequential decision method.

6. Conclusions

In this paper, we have introduced a VSPP in seaports that addresses the BAP and the PPP in combination. We have formulated the problem as a compact MILP model and have proposed several valid inequalities to strengthen the LP relaxation of the model. This model can be solved efficiently by a general-purpose solver for small-scale instances. To solve large-scale instances, we have proposed a BPBC approach. The approach is further improved by several computational enhancements: LBL inequalities, variable fixing, warm start for the BMP, primal heuristics, and parallel computing for pricing problems. The computational results show that these enhancements significantly improve the performance of the BPBC approach.

We have also tested the performance of the BPBC approach with instances under different shift arrangements and pilot dispatching costs. The results show that the approach performs robustly to solve instances under different settings. Finally, we have compared the performances of the BPBC approach and a sequential optimization method, and the results demonstrate that integrating the BAP and the PPP in the VSPP brings significant benefits to port authorities.

The current study can be extended in several ways. First, while we assume deterministic parameters, the arrival and handling times of vessels are uncertain in practice. Thus, a promising extension direction of this research is to design solution methods for the VSPP under conditions of uncertainty.

Second, in our problem, we consider the most common seaport layout, i.e., one consisting of an (uncapacitated) anchorage located at the outermost part of the seaport, a set of berths, and one navigation channel that lies between them. In some large seaports, however, there may be another anchorage that lies between the channel and the berths. Such “inner” anchorages provide temporary mooring areas for vessels before entering the terminals, and they generally have limited capacities. Some large seaports also may have multiple navigation channels. Future studies can consider extending the models and solution approach developed in this paper to solve the VSPP in seaports with different layouts.

Finally, along with pilots, tugboats also play an important role in berthing and unberthing vessels in a seaport. In many seaports, tugboats are managed by companies that are independent of port authorities, e.g., the Port of Hong Kong and the Port of Montreal (The Marine Department of Hong Kong 2020b, The Port of Montreal 2020). Nevertheless, it would be interesting to develop methods to further integrate towage services in the VSPP.

References

- Adulyasak Y, Cordeau JF, Jans R (2014) Formulations and branch-and-cut algorithms for multivehicle production and inventory routing problems. *INFORMS Journal on Computing* 26(1):103–120.
- Adulyasak Y, Cordeau JF, Jans R (2015) Benders decomposition for production routing under demand uncertainty. *Operations Research* 63(4):851–867.
- Benders J (1962) Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik* 4:238–252.
- 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.
- Buhrkal K, Zuglian S, Ropke S, Larsen J, Lusby R (2011) Models for the discrete berth allocation problem: a computational comparison. *Transportation Research Part E: Logistics and Transportation Review* 47(4):461–473.
- Carlo HJ, Vis IF, Roodbergen KJ (2015) Seaside operations in container terminals: literature overview, trends, and research directions. *Flexible Services and Manufacturing Journal* 27(2-3):224–262.
- Cordeau JF, Laporte G, Legato P, Moccia L (2005) Models and tabu search heuristics for the berth-allocation problem. *Transportation Science* 39(4):526–538.
- Cordeau JF, Stojković G, Soumis F, Desrosiers J (2001) Benders decomposition for simultaneous aircraft routing and crew scheduling. *Transportation Science* 35(4):375–388.
- Corry P, Bierwirth C (2019) The berth allocation problem with channel restrictions. *Transportation Science* 53(3):708–727.
- Dantzig GB, Wolfe P (1960) Decomposition principle for linear programs. *Operations Research* 8(1):101–111.
- Dauzère-Pérès S, De Almeida D, Guyon O, Benhizia F (2015) A Lagrangian heuristic framework for a real-life integrated planning problem of railway transportation resources. *Transportation Research Part B: Methodological* 74:138–150.
- 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.

- Edwards H (2010) Pilot assignment to ships in the Sea of Bothnia. *Energy, Natural Resources and Environmental Economics*, 411–425 (Springer).
- Fransoo JC, Lee CY (2013) The critical role of ocean container transport in global supply chain performance. *Production and Operations Management* 22(2):253–268.
- Goel A, Irnich S (2017) An exact method for vehicle routing and truck driver scheduling problems. *Transportation Science* 51(2):737–754.
- Imai A, Nagaiwa K, Tat CW (1997) Efficient planning of berth allocation for container terminals in Asia. *Journal of Advanced Transportation* 31(1):75–94.
- Imai A, Nishimura E, Papadimitriou S (2001) The dynamic berth allocation problem for a container port. *Transportation Research Part B: Methodological* 35(4):401–417.
- Imai A, Nishimura E, Papadimitriou S (2003) Berth allocation with service priority. *Transportation Research Part B: Methodological* 37(5):437–457.
- Imai A, Sun X, Nishimura E, Papadimitriou S (2005) Berth allocation in a container port: using a continuous location space approach. *Transportation Research Part B: Methodological* 39(3):199–221.
- International Maritime Organization (2001) Guidelines on fatigue mitigation and management. URL <http://www.imo.org/en/OurWork/HumanElement/VisionPrinciplesGoals/Documents/1014.pdf>, Accessed September 1, 2020.
- Jans R (2009) Solving lot-sizing problems on parallel identical machines using symmetry-breaking constraints. *INFORMS Journal on Computing* 21(1):123–136.
- Jia S, Li CL, Xu Z (2019) Managing navigation channel traffic and anchorage area utilization of a container port. *Transportation Science* 53(3):728–745.
- Jia S, Wu L, Meng Q (2020) Joint scheduling of vessel traffic and pilots in seaport waters. *Transportation Science* 54(6):1495–1515.
- Lalla-Ruiz E, Shi X, Voß S (2018) The waterway ship scheduling problem. *Transportation Research Part D: Transport and Environment* 60:191–209.
- Li S, Jia S (2019) The seaport traffic scheduling problem: formulations and a column-row generation algorithm. *Transportation Research Part B: Methodological* 128:158–184.
- Lübbecke E, Lübbecke ME, Möhring RH (2019) Ship traffic optimization for the Kiel Canal. *Operations Research* 67(3):791–812.
- Mercier A, Cordeau JF, Soumis F (2005) A computational study of Benders decomposition for the integrated aircraft routing and crew scheduling problem. *Computers & Operations Research* 32(6):1451–1476.
- Monaco MF, Sammarra M (2007) The berth allocation problem: a strong formulation solved by a Lagrangean approach. *Transportation Science* 41(2):265–280.

- NPR News (2012) Harbor Pilots Reap High Rewards For Dangerous Job. <https://www.npr.org/2012/03/21/149091141/harbor-pilots-reap-high-rewards-for-dangerous-job>, Accessed November 18, 2020.
- Roy D, de Koster R, Bekker R (2020) Modeling and design of container terminal operations. *Operations Research* 68(3):686–715.
- Ruther S, Boland N, Engineer FG, Evans I (2017) Integrated aircraft routing, crew pairing, and tail assignment: branch-and-price with many pricing problems. *Transportation Science* 51(1):177–195.
- Sandhu R, Klabjan D (2007) Integrated airline fleetling and crew-pairing decisions. *Operations Research* 55(3):439–456.
- Stahlbock R, Voß S (2008) Operations research at container terminals: a literature update. *OR Spectrum* 30(1):1–52.
- 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.
- The Marine Department of Hong Kong (2020a) Port of Hong Kong in Figures. Technical report, URL https://www.mardep.gov.hk/en/publication/pdf/portstat_pamphlet20.pdf, Accessed November 18, 2020.
- The Marine Department of Hong Kong (2020b) Towage & Salvage in Hong Kong. https://www.mardep.gov.hk/en/pub_services/sdfiles/towasalv.html, Accessed November 18, 2020.
- The Port of Montreal (2020) Services and fees in the Port of Montreal. <https://www.port-montreal.com/en/goods/operations/services-and-fees>, Accessed November 18, 2020.
- UNCTAD (2019) Review of maritime transportation 2019. *Paper presented at the United Nations Conference on Trade and Development* (New York and Geneva), URL http://unctad.org/en/PublicationsLibrary/rmt2019_en.pdf, Accessed September 1, 2020.
- Wermus M, Pope JA (1994) Scheduling harbor pilots. *Interfaces* 24(2):44–52.
- Wu L, Jia S, Wang S (2020) Pilotage planning in seaports. *European Journal of Operational Research* 287(1):90–105.
- Wu L, Wang S (2020) The shore power deployment problem for maritime transportation. *Transportation Research Part E: Logistics and Transportation Review* 135:101883.
- Xie F, Wu T, Zhang C (2019) A branch-and-price algorithm for the integrated berth allocation and quay crane assignment problem. *Transportation Science* 53(5):1427–1454.
- Xu Z, Lee CY (2018) New lower bound and exact method for the continuous berth allocation problem. *Operations Research* 66(3):778–798.
- Zeighami V, Soumis F (2019) Combining Benders’ decomposition and column generation for integrated crew pairing and personalized crew assignment problems. *Transportation Science* 53(5):1479–1499.

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

Electronic Companion

EC.1. Mathematical Proofs

This section presents proofs to lemmas, propositions, theorems, and corollaries in the main text.

EC.1.1. Proof of Proposition 1

Proof of Proposition 1. Given any feasible solution $(\bar{\chi})$ to the BMP, let $H = \{(i, t) \mid \sum_{\omega \in \Omega} \bar{\chi}_\omega \alpha_{\omega, i, t} > 0, t \in \mathcal{E}_i, i \in \mathcal{I}\}$. Given $\bar{\chi}$, the corresponding PBSP is feasible if and only if there exists a solution μ' such that $\sum_{\phi \in \Phi} \mu'_\phi \gamma_{\phi, i, t} = \sum_{\omega \in \Omega} \bar{\chi}_\omega \alpha_{\omega, i, t}, \forall (i, t) \in H$. We can show that such a μ' exists for any $\bar{\chi}$.

Recall that $\bigcup_{s \in \mathcal{S}} \mathcal{T}_s = \mathcal{T}$ and that all vessel routes in model M2 satisfy constraints (34). Hence, corresponding to each $(i, t) \in H$, one can always construct a feasible pilot route that visits i at time t . In particular, the pilot route can consist of only task i (which starts at time t) and a rest period (which starts at a feasible time) and is completed by a pilot in a suitable shift. Then, based on these pilot routes, one can easily construct a feasible solution for the PBSP.

As a result, given any $\bar{\chi}$, the PBSP is always feasible. Furthermore, considering that the cost parameters \bar{c}_ϕ are finite and due to constraints (48)–(50), any feasible solution of the PBSP must be bounded. Therefore, as the dual of the PBSP, the DBSP is also feasible and bounded. \square

EC.1.2. Proof of Theorem 1

Proof of Theorem 1. At each node of the branch-and-bound tree, the BMP and the PBSP are defined on a subset $\Omega' \subseteq \Omega$ of vessel routes and a subset $\Phi' \subseteq \Phi$ of pilot routes, due to the branching constraints. The approach solves the BMP to optimality with column-and-Benders-cut generation. Let (χ^*, η^*) and μ^* denote the optimal solution to the BMP and the associated PBSP, respectively. According to Benders (1962), we have that (i) χ^* and μ^* constitute an optimal solution to the LP relaxation of M2 defined on the same subsets of vessel routes and pilot routes (i.e., the LP relaxation of M2 imposed with the same branching constraints) and that (ii) the BMP and the LP relaxation of M2 have the same optimal objective function value.

Now consider our branching scheme. One can verify the solution (χ^*) generated by the C&BCG algorithm to the BMP is fractional if and only if there is at least one fractional value in \mathbf{A} or \mathbf{E} . Further, given any integer solution $(\bar{\chi})$ for the BMP, the solution generated by column generation to the corresponding PBSP is fractional if and only if there is at least one fractional value in $\mathbf{\Pi}$ or $\mathbf{\Psi}$.

Therefore, by imposing branching constraints on the BMP and the PBSP at the associated nodes of the tree, and by solving the BMP at each node to optimality, the approach gradually

reaches a solution to the BMP that has the minimum objective function value among all integer solutions (i.e., solutions where all values in the solution χ^* are integral and all values in the solution μ^* for the corresponding PBSP are integral as well). Correspondingly, the approach reaches an integer solution that has the minimum cost among all integer solutions for the LP relaxation of M2. Therefore, the BPBC approach delivers an optimal solution for M2. \square

EC.1.3. Proof of Proposition 2

Proof of Proposition 2. Consider the BMP without these LBL inequalities. Given any feasible solution of the χ variables, denoted by $\bar{\chi}$, to the BMP, let $\eta^*(\bar{\chi})$ denote the associated optimal solution of the η variable in the BMP. To prove that the LBL inequalities are valid for the BMP, it suffices to show that the following linear program generates a lower bound to $\eta^*(\bar{\chi})$:

$$[\mathbf{P}_{LBL}(\bar{\chi})] \quad \min \sum_{s \in \mathcal{S}} c_s^3 n_s \quad (\text{EC.1})$$

$$\text{s.t. (69) - (72)}$$

$$\sum_{s \in \mathcal{S}_t} \nu_{i,t,s} = \sum_{\omega \in \Omega} \bar{\chi}_\omega \alpha_{\omega,i,t} \quad \forall t \in \mathcal{E}_i, i \in \mathcal{I}. \quad (\text{EC.2})$$

We have shown in Proposition 1 that the DBSP is always feasible and bounded. Given $\bar{\chi}$, let δ^* and ζ^* denote an optimal solution to the DBSP. Because $(\delta^*, \zeta^*) \in \Gamma_\Delta$, we have $\eta^*(\bar{\chi}) \geq \sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}} \sum_{\omega \in \Omega} \bar{\chi}_\omega \alpha_{\omega,i,t} \delta_{i,t}^* + \sum_{\phi \in \Phi} \zeta_\phi^*$ due to constraints (56). Let μ^* denote the optimal solution to the dual of the DBSP (i.e., the PBSP). Because of strong duality, we have $\sum_{\phi \in \Phi} \mu_\phi^* \bar{c}_\phi = \sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}} \sum_{\omega \in \Omega} \bar{\chi}_\omega \alpha_{\omega,i,t} \delta_{i,t}^* + \sum_{\phi \in \Phi} \zeta_\phi^*$.

We construct the solution (ν', n') for the variables in the $\mathbf{P}_{LBL}(\bar{\chi})$, where

$$\nu'_{i,t,s} = \sum_{\phi \in \Phi_s} \mu_\phi^* \gamma_{\phi,i,t} \quad \forall i \in \mathcal{I}, t \in \mathcal{T}_s, s \in \mathcal{S} \quad (\text{EC.3})$$

$$n'_s = \sum_{\phi \in \Phi_s} \mu_\phi^* \quad \forall s \in \mathcal{S}. \quad (\text{EC.4})$$

Considering the feasibility of pilot routes and the feasibility of μ^* for the PBSP, we have

$$\sum_{i \in \mathcal{I}} \sum_{t'=\max\{\mathcal{T}_s, t-d_i-q'_{i,t,s}+1\}}^t \gamma_{\phi,i,t'} \leq 1, \quad \forall t \in \mathcal{T}_s, \phi \in \Phi_s, s \in \mathcal{S} \quad (\text{EC.5})$$

$$\sum_{s \in \mathcal{S}_t} \sum_{\phi \in \Phi_s} \mu_\phi^* \gamma_{\phi,i,t} = \sum_{\omega \in \Omega} \bar{\chi}_\omega \alpha_{\omega,i,t} \quad \forall t \in \mathcal{E}_i, i \in \mathcal{I}. \quad (\text{EC.6})$$

Because $\mu_\phi^* \geq 0$, $\forall \phi \in \Phi$, by timing each term in inequalities (EC.5) with μ_ϕ^* , and summing them over $\phi \in \Phi_s$ we have

$$\sum_{i \in \mathcal{I}} \sum_{t'=\max\{\mathcal{T}_s, t-d_i-q'_{i,t,s}+1\}}^t \sum_{\phi \in \Phi_s} \mu_\phi^* \gamma_{\phi,i,t'} \leq \sum_{\phi \in \Phi_s} \mu_\phi^*, \quad \forall t \in \mathcal{T}_s, s \in \mathcal{S}. \quad (\text{EC.7})$$

Then, combining (EC.3), (EC.4), and (EC.7) gives

$$\sum_{i \in \mathcal{I}} \sum_{t' = \max\{\underline{T}_s, t - d_i - q'_{i,t,s} + 1\}}^t \nu'_{i,t',s} \leq n'_s \quad \forall t \in \mathcal{T}_s, s \in \mathcal{S}, \quad (\text{EC.8})$$

and from (EC.3) and (EC.6) we have

$$\sum_{s \in \mathcal{S}_t} \nu'_{i,t,s} = \sum_{\omega \in \Omega} \bar{\chi}_\omega \alpha_{\omega,i,t} \quad \forall t \in \mathcal{E}_i, i \in \mathcal{I} \quad (\text{EC.9})$$

$$\nu'_{i,t,s} \leq 1 \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, s \in \mathcal{S} \quad (\text{EC.10})$$

$$\nu'_{i,t,s} \geq 0 \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, s \in \mathcal{S}. \quad (\text{EC.11})$$

Finally, summarizing inequalities (EC.8)–(EC.11), one can conclude that the constructed solution $(\boldsymbol{\nu}', \boldsymbol{n}')$ is feasible for the $P_{LBL}(\bar{\boldsymbol{\chi}})$. Besides, we have $\sum_{s \in \mathcal{S}} c_s^3 n'_s = \sum_{s \in \mathcal{S}} \sum_{\phi \in \Phi_s} \bar{c}_\phi \mu_\phi^* = \sum_{\phi \in \Phi} \bar{c}_\phi \mu_\phi^* \leq \eta^*(\bar{\boldsymbol{\chi}})$. Let $(\boldsymbol{\nu}^*, \boldsymbol{n}^*)$ denote an optimal solution for the $P_{LBL}(\bar{\boldsymbol{\chi}})$. Because the $P_{LBL}(\bar{\boldsymbol{\chi}})$ forms a minimization problem, we have $\sum_{s \in \mathcal{S}} c_s^3 n_s^* \leq \sum_{s \in \mathcal{S}} c_s^3 n'_s \leq \eta^*(\bar{\boldsymbol{\chi}})$.

Therefore, the $P_{LBL}(\bar{\boldsymbol{\chi}})$ generates a lower bound to $\eta^*(\bar{\boldsymbol{\chi}})$, given any feasible solution of the χ variables to the original BMP. This indicates that the LBL inequalities are valid for the BMP. \square

EC.1.4. Proof of Lemma 1

Proof of Lemma 1. We provide a proof to this lemma based on the BMP without the LBL cuts proposed in Section 4.5.1, while the extension to the BMP with these cuts is straightforward.

We start by formulating the Lagrangian dual of the BMP. Let $\bar{\boldsymbol{\pi}} = (\bar{\pi}_k | \bar{\pi}_k \in \mathbb{R}, k \in \mathcal{K})$, $\bar{\boldsymbol{\lambda}} = (\bar{\lambda}_{b,t} | \bar{\lambda}_{b,t} \leq 0, t \in \mathcal{T}, b \in \mathcal{B})$, $\bar{\boldsymbol{\theta}} = (\bar{\theta}_t | \bar{\theta}_t \leq 0, t \in \mathcal{T})$, $\bar{\boldsymbol{\vartheta}} = (\bar{\vartheta}_t | \bar{\vartheta}_t \leq 0, t \in \mathcal{T})$, $\bar{\boldsymbol{\kappa}} = (\bar{\kappa}_{t,(i,j)} | \bar{\kappa}_{t,(i,j)} \leq 0, t \in \mathcal{T}, (i,j) \in \mathcal{U})$, $\bar{\boldsymbol{\varphi}} = (\bar{\varphi}_{(\delta,\zeta)} | \bar{\varphi}_{(\delta,\zeta)} \geq 0, (\delta,\zeta) \in \Gamma_\Delta)$ and $\bar{\boldsymbol{\rho}} = (\bar{\rho}_\omega | \bar{\rho}_\omega \leq 0, \omega \in \Omega)$ be the vectors of the Lagrangian multipliers for constraints (39)–(43), (56), and (57), respectively. Besides, let $\Omega_{k,b}$ denote the set of columns corresponding to vessel k being handled at berth b , where $k \in \mathcal{K}$, $b \in \mathcal{B}_k$. As implied by constraints (39), (57), and (58), the following inequalities are valid for the BMP:

$$\sum_{\omega \in \Omega_{k,b}} \chi_\omega \leq 1, \quad \forall b \in \mathcal{B}_k, k \in \mathcal{K}. \quad (\text{EC.12})$$

Given $\bar{\boldsymbol{\pi}}$, $\bar{\boldsymbol{\lambda}}$, $\bar{\boldsymbol{\theta}}$, $\bar{\boldsymbol{\kappa}}$, $\bar{\boldsymbol{\varphi}}$, and $\bar{\boldsymbol{\rho}}$, the Lagrangian dual of the BMP is as follows:

$$\begin{aligned} \mathcal{D}(\bar{\boldsymbol{\pi}}, \bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\kappa}}, \bar{\boldsymbol{\varphi}}, \bar{\boldsymbol{\rho}}) = \min_{\boldsymbol{\chi}, \boldsymbol{\eta}} & \left\{ \sum_{\omega \in \Omega} \chi_\omega c_\omega + \eta + \sum_{k \in \mathcal{K}} (1 - \sum_{\omega \in \Omega_k} \chi_\omega) \bar{\pi}_k \right. \\ & + \sum_{t \in \mathcal{T}} \sum_{b \in \mathcal{B}} (1 - \sum_{\omega \in \Omega} \chi_\omega \sum_{t=1}^{\tau} \sum_{t' = \min\{\tau+1, \bar{T}\}}^{\bar{T}} \beta_{\omega,b,t,t'}) \bar{\lambda}_{b,t} \\ & \left. + \sum_{t \in \mathcal{T}} (1 - \sum_{\omega \in \Omega} \chi_\omega \sum_{i \in \mathcal{I}^{in}} \sum_{t' = \max\{t-f_i+1, 1\}}^t \alpha_{\omega,i,t'}) \bar{\theta}_t + \sum_{t \in \mathcal{T}} (1 - \sum_{\omega \in \Omega} \chi_\omega \sum_{i \in \mathcal{I}^{out}} \sum_{t' = \max\{t-f_i+1, 1\}}^t \alpha_{\omega,i,t'}) \bar{\vartheta}_t \right\} \end{aligned}$$

$$\begin{aligned}
& + \sum_{t \in \mathcal{T}} \sum_{(i,j) \in \mathcal{U}} \left(1 - \sum_{\omega \in \Omega} \chi_{\omega} \sum_{t' = \max\{t-d_i+1, 1\}}^t \alpha_{\omega, i, t'} - \sum_{\omega \in \Omega} \chi_{\omega} \sum_{t' = \max\{t-d_j+1, 1\}}^t \alpha_{\omega, j, t'} \right) \bar{K}_{t, (i,j)} \\
& - \sum_{(\delta, \zeta) \in \Gamma_{\Delta}} \left(\eta - \sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}} \sum_{\omega \in \Omega} \chi_{\omega} \alpha_{\omega, i, t} \delta_{i,t} - \sum_{\phi \in \Phi} \zeta_{\phi} \right) \bar{\varphi}(\delta, \zeta) \\
& + \sum_{\omega \in \Omega} (1 - \chi_{\omega}) \bar{\rho}_{\omega}, \text{ s.t. (58), (59), (EC.12)} \Big\}. \tag{EC.13}
\end{aligned}$$

Equation (EC.13) is equivalent to

$$\begin{aligned}
\mathcal{D}(\bar{\pi}, \bar{\lambda}, \bar{\theta}, \bar{\kappa}, \bar{\varphi}, \bar{\rho}) & = \sum_{k \in \mathcal{K}} \bar{\pi}_k + \sum_{t \in \mathcal{T}} \sum_{b \in \mathcal{B}} \bar{\lambda}_{b,t} + \sum_{t \in \mathcal{T}} \bar{\theta}_t + \sum_{t \in \mathcal{T}} \bar{\vartheta}_t + \sum_{t \in \mathcal{T}} \sum_{(i,j) \in \mathcal{U}} \bar{K}_{t, (i,j)} + \sum_{(\delta, \zeta) \in \Gamma_{\Delta}} \sum_{\phi \in \Phi} \zeta_{\phi} \bar{\varphi}(\delta, \zeta) \\
& + \sum_{\omega \in \Omega} \bar{\rho}_{\omega} + \min_{\chi, \eta} \left\{ \sum_{\omega \in \Omega} \chi_{\omega} c_{\omega} + \left(1 - \sum_{(\delta, \zeta) \in \Gamma_{\Delta}} \bar{\varphi}(\delta, \zeta) \right) \eta - \sum_{k \in \mathcal{K}} \sum_{\omega \in \Omega_k} \chi_{\omega} \bar{\pi}_k \right. \\
& - \sum_{t \in \mathcal{T}} \sum_{b \in \mathcal{B}} \sum_{\omega \in \Omega} \chi_{\omega} \sum_{t=1}^{\tau} \sum_{t' = \min\{\tau+1, \bar{T}\}}^{\bar{T}} \beta_{\omega, b, t, t'} \bar{\lambda}_{b,t} \\
& - \sum_{t \in \mathcal{T}} \sum_{\omega \in \Omega} \chi_{\omega} \sum_{i \in \mathcal{I}^{in}} \sum_{t' = \max\{t-f_i+1, 1\}}^t \alpha_{\omega, i, t'} \bar{\theta}_t - \sum_{t \in \mathcal{T}} \sum_{\omega \in \Omega} \chi_{\omega} \sum_{i \in \mathcal{I}^{out}} \sum_{t' = \max\{t-f_i+1, 1\}}^t \alpha_{\omega, i, t'} \bar{\vartheta}_t \\
& - \sum_{t \in \mathcal{T}} \sum_{(i,j) \in \mathcal{U}} \sum_{\omega \in \Omega} \chi_{\omega} \sum_{t' = \max\{t-d_i+1, 1\}}^t \alpha_{\omega, i, t'} \bar{K}_{t, (i,j)} - \sum_{\omega \in \Omega} \chi_{\omega} \sum_{t' = \max\{t-d_j+1, 1\}}^t \alpha_{\omega, j, t'} \bar{K}_{t, (i,j)} \\
& \left. + \sum_{(\delta, \zeta) \in \Gamma_{\Delta}} \sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}} \sum_{\omega \in \Omega} \chi_{\omega} \alpha_{\omega, i, t} \delta_{i,t} \bar{\varphi}(\delta, \zeta) - \sum_{\omega \in \Omega} \chi_{\omega} \bar{\rho}_{\omega}, \text{ s.t. (58), (59), (EC.12)} \right\}. \tag{EC.14}
\end{aligned}$$

Let Z^* be the optimal objective function value for the BMP. We have

$$Z^* \geq \max_{\bar{\pi}, \bar{\lambda}, \bar{\theta}, \bar{\kappa}, \bar{\varphi}, \bar{\rho}} \mathcal{D}(\bar{\pi}, \bar{\lambda}, \bar{\theta}, \bar{\kappa}, \bar{\varphi}, \bar{\rho}). \tag{EC.15}$$

Since the BMP is solved by column and Benders cut generation, where columns and Benders cuts are added dynamically, we let $\tilde{\Omega} \subseteq \Omega$ and $\tilde{\Gamma}_{\Delta} \subseteq \Gamma_{\Delta}$ denote the current sets of columns and Benders cuts generated for solving the BMP, respectively. After solving the BMP with columns from $\tilde{\Omega}$ and Benders cuts from $\tilde{\Gamma}_{\Delta}$, one obtains the optimal solutions to the dual variables associated with constraints (39)–(43), (56), and (57) which are denoted by $\boldsymbol{\pi}^* = (\pi_k^* | k \in \mathcal{K})$, $\boldsymbol{\lambda}^* = (\lambda_{b,t}^* | t \in \mathcal{T}, b \in \mathcal{B})$, $\boldsymbol{\theta}^* = (\theta_t^* | t \in \mathcal{T})$, $\boldsymbol{\vartheta}^* = (\vartheta_t^* | t \in \mathcal{T})$, $\boldsymbol{\kappa}^* = (\kappa_{i,(i,j)}^* | t \in \mathcal{T}, (i,j) \in \mathcal{U})$, $\boldsymbol{\varphi}^* = (\varphi_{(\delta, \zeta)}^* | (\delta, \zeta) \in \tilde{\Gamma}_{\Delta})$, and $\boldsymbol{\rho}^* = (\rho_{\omega}^* | \omega \in \tilde{\Omega})$, respectively. We further define vectors $\boldsymbol{\varphi}^+$ and $\boldsymbol{\rho}^+$ such that $\boldsymbol{\varphi}^+ = (\varphi_{(\delta, \zeta)}^+ | \varphi_{(\delta, \zeta)}^+ = \varphi_{(\delta, \zeta)}^*, (\delta, \zeta) \in \tilde{\Gamma}_{\Delta}, \varphi_{(\delta, \zeta)}^+ = 0, (\delta, \zeta) \in \Gamma_{\Delta} \setminus \tilde{\Gamma}_{\Delta})$ and $\boldsymbol{\rho}^+ = (\rho_{\omega}^+ | \rho_{\omega}^+ = \rho_{\omega}^*, \omega \in \tilde{\Omega}, \rho_{\omega}^+ = 0, \omega \in \Omega \setminus \tilde{\Omega})$.

By definition, the BMP is constructed to be feasible and bounded. As a result, the dual of BMP is also feasible and bounded. By the feasibility of the dual of the BMP, one can easily verify that $(\boldsymbol{\pi}^*, \boldsymbol{\lambda}^*, \boldsymbol{\theta}^*, \boldsymbol{\vartheta}^*, \boldsymbol{\kappa}^*, \boldsymbol{\varphi}^+)$ forms a feasible solution for $\mathcal{D}(\bar{\pi}, \bar{\lambda}, \bar{\theta}, \bar{\kappa}, \bar{\varphi}, \bar{\rho})$. It follows that

$$Z^* \geq \mathcal{D}(\boldsymbol{\pi}^*, \boldsymbol{\lambda}^*, \boldsymbol{\theta}^*, \boldsymbol{\vartheta}^*, \boldsymbol{\kappa}^*, \boldsymbol{\varphi}^+, \boldsymbol{\rho}^+), \tag{EC.16}$$

where

$$\begin{aligned}
\mathcal{D}(\boldsymbol{\pi}^*, \boldsymbol{\lambda}^*, \boldsymbol{\theta}^*, \boldsymbol{\vartheta}^*, \boldsymbol{\kappa}^*, \boldsymbol{\varphi}^+, \boldsymbol{\rho}^+) &= \sum_{k \in \mathcal{K}} \pi_k^* + \sum_{t \in \mathcal{T}} \sum_{b \in \mathcal{B}} \lambda_{b,t}^* + \sum_{t \in \mathcal{T}} \theta_t^* + \sum_{t \in \mathcal{T}} \vartheta_t^* + \sum_{t \in \mathcal{T}} \sum_{(i,j) \in \mathcal{U}} \kappa_{t,(i,j)}^* \\
&+ \sum_{(\delta, \zeta) \in \tilde{\Gamma}_\Delta} \sum_{\phi \in \Phi} \zeta_\phi \varphi_{(\delta, \zeta)}^* + \sum_{\omega \in \tilde{\Omega}} \rho_\omega^* + \min_{\chi, \eta} \left\{ \sum_{\omega \in \tilde{\Omega}} \chi_\omega c_\omega + (1 - \sum_{(\delta, \zeta) \in \tilde{\Gamma}_\Delta} \varphi_{(\delta, \zeta)}^*) \eta - \sum_{k \in \mathcal{K}} \sum_{\omega \in \Omega_k} \chi_\omega \pi_k^* \right. \\
&- \sum_{t \in \mathcal{T}} \sum_{b \in \mathcal{B}} \sum_{\omega \in \Omega} \chi_\omega \sum_{t=1}^{\tau} \sum_{t'=\min\{\tau+1, \bar{T}\}}^{\bar{T}} \beta_{\omega, b, t, t'} \lambda_{b, t}^* \\
&- \sum_{t \in \mathcal{T}} \sum_{\omega \in \Omega} \chi_\omega \sum_{i \in \mathcal{I}^{in}} \sum_{t'=\max\{t-f_i+1, 1\}}^t \alpha_{\omega, i, t'} \theta_t^* - \sum_{t \in \mathcal{T}} \sum_{\omega \in \Omega} \chi_\omega \sum_{i \in \mathcal{I}^{out}} \sum_{t'=\max\{t-f_i+1, 1\}}^t \alpha_{\omega, i, t'} \vartheta_t^* \\
&- \sum_{t \in \mathcal{T}} \sum_{(i,j) \in \mathcal{U}} \sum_{\omega \in \Omega} \chi_\omega \sum_{t'=\max\{t-d_i+1, 1\}}^t \alpha_{\omega, i, t'} \kappa_{t, (i,j)}^* - \sum_{\omega \in \Omega} \chi_\omega \sum_{t'=\max\{t-d_j+1, 1\}}^t \alpha_{\omega, j, t'} \kappa_{t, (i,j)}^* \\
&\left. + \sum_{(\delta, \zeta) \in \tilde{\Gamma}_\Delta} \sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}} \sum_{\omega \in \Omega} \chi_\omega \alpha_{\omega, i, t} \delta_{i, t} \varphi_{(\delta, \zeta)}^* - \sum_{\omega \in \tilde{\Omega}} \chi_\omega \rho_\omega^*, \text{ s.t. (58), (59), (EC.12)} \right\}. \quad (\text{EC.17})
\end{aligned}$$

Due to the feasibility of the dual of the BMP, we have $\sum_{(\delta, \zeta) \in \tilde{\Gamma}_\Delta} \varphi_{(\delta, \zeta)}^* \leq 1$ and $\rho_\omega^* \leq 0$, $\forall \omega \in \tilde{\Omega}$. Further, considering that $\eta \geq 0$ and $\chi_\omega \geq 0$, $\forall \omega \in \tilde{\Omega}$, we have $(1 - \sum_{(\delta, \zeta) \in \tilde{\Gamma}_\Delta} \varphi_{(\delta, \zeta)}^*) \eta \geq 0$ and $\sum_{\omega \in \tilde{\Omega}} \chi_\omega \rho_\omega^* \leq 0$. Recall that in this lemma, the optimal objective function value of the BMP is denoted by Υ . By strong duality, we have $\Upsilon = \sum_{k \in \mathcal{K}} \pi_k^* + \sum_{t \in \mathcal{T}} \sum_{b \in \mathcal{B}} \lambda_{b,t}^* + \sum_{t \in \mathcal{T}} \theta_t^* + \sum_{t \in \mathcal{T}} \vartheta_t^* + \sum_{t \in \mathcal{T}} \sum_{(i,j) \in \mathcal{U}} \kappa_{t,(i,j)}^* + \sum_{(\delta, \zeta) \in \tilde{\Gamma}_\Delta} \sum_{\phi \in \Phi} \zeta_\phi \varphi_{(\delta, \zeta)}^* + \sum_{\omega \in \tilde{\Omega}} \rho_\omega^*$. Therefore, we have the following inequality:

$$\begin{aligned}
\mathcal{D}(\boldsymbol{\pi}^*, \boldsymbol{\lambda}^*, \boldsymbol{\theta}^*, \boldsymbol{\vartheta}^*, \boldsymbol{\kappa}^*, \boldsymbol{\varphi}^+, \boldsymbol{\rho}^+) &\geq \mathcal{D}' = \Upsilon + \min_{\chi} \left\{ \sum_{\omega \in \tilde{\Omega}} \chi_\omega (c_\omega - \pi_k^* - \sum_{t \in \mathcal{T}} \sum_{t=1}^{\tau} \sum_{t'=\min\{\tau+1, \bar{T}\}}^{\bar{T}} \beta_{\omega, b, t, t'} \lambda_{b, t}^* \right. \\
&- \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{I}^{in} \cap \mathcal{I}_k} \sum_{t'=\max\{t-f_i+1, 1\}}^t \alpha_{\omega, i, t'} \theta_t^* - \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{I}^{out} \cap \mathcal{I}_k} \sum_{t'=\max\{t-f_i+1, 1\}}^t \alpha_{\omega, i, t'} \vartheta_t^* \\
&- \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{I}_k} \sum_{j \in \mathcal{I}} \sum_{(i,j) \in \mathcal{U}} \sum_{t'=\max\{t-d_i+1, 1\}}^t \alpha_{\omega, i, t'} \kappa_{t, (i,j)}^* \\
&\left. + \sum_{(\delta, \zeta) \in \tilde{\Gamma}_\Delta} \sum_{i \in \mathcal{I}_k} \sum_{t \in \mathcal{T}} \sum_{\omega \in \Omega} \chi_\omega \alpha_{\omega, i, t} \delta_{i, t} \varphi_{(\delta, \zeta)}^* \right\}, \text{ s.t. (58), (EC.12)} \quad (\text{EC.18})
\end{aligned}$$

Considering that $\Omega = \bigcup_{k \in \mathcal{K}} \bigcup_{b \in \mathcal{B}_k} \Omega_{k,b}$ and by decomposing the minimization problem in (EC.18) for each combination of b and k , where $k \in \mathcal{K}$, $b \in \mathcal{B}_k$, \mathcal{D}' can be equivalently calculated by:

$$\mathcal{D}' = \Upsilon + \sum_{k \in \mathcal{K}} \sum_{b \in \mathcal{B}_k} \min_{\chi_\omega: \omega \in \Omega_{k,b}} \sum_{\omega \in \Omega_{k,b}} \left\{ \chi_\omega (c_\omega - \pi_k^* - \sum_{t \in \mathcal{T}} \sum_{t=1}^{\tau} \sum_{t'=\min\{\tau+1, \bar{T}\}}^{\bar{T}} \beta_{\omega, b, t, t'} \lambda_{b, t}^* \right.$$

$$\begin{aligned}
& - \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{I}^{in} \cap \mathcal{I}_k} \sum_{t' = \max\{t-f_i+1, 1\}}^t \alpha_{\omega, i, t'} \theta_t^* - \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{I}^{out} \cap \mathcal{I}_k} \sum_{t' = \max\{t-f_i+1, 1\}}^t \alpha_{\omega, i, t'} \vartheta_t^* \\
& - \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{I}_k} \sum_{j \in \mathcal{I}} \sum_{(i, j) \in \mathcal{U}} \sum_{t' = \max\{t-d_i+1, 1\}}^t \alpha_{\omega, i, t'} \kappa_{t, (i, j)}^* + \sum_{(\delta, \zeta) \in \bar{\Gamma}_\Delta} \sum_{i \in \mathcal{I}_k} \sum_{t \in \mathcal{T}} \sum_{\omega \in \Omega} \chi_\omega \alpha_{\omega, i, t} \delta_{i, t} \varphi^*(\delta, \zeta), \\
& \text{s.t. } \left. \sum_{\omega \in \Omega_{k, b}} \chi_\omega \leq 1, 0 \leq \chi_\omega \leq 1, \forall \omega \in \Omega_{k, b} \right\}. \tag{EC.19}
\end{aligned}$$

Let $r_{k, b}$ be the optimal objective function value of the pricing problem $\text{MPP}_{k, b}$, where $k \in \mathcal{K}$, $b \in \mathcal{B}_k$. It can be readily seen that

$$\mathcal{D}' = \Upsilon + \sum_{k \in \mathcal{K}} \sum_{b \in \mathcal{B}_k: r_{k, b} < 0} r_{k, b}. \tag{EC.20}$$

Summarizing (EC.16), (EC.18), and (EC.20), we have $Z^* \geq \Upsilon + \sum_{k \in \mathcal{K}} \sum_{b \in \mathcal{B}_k: r_{k, b} < 0} r_{k, b}$, which completes the proof. \square

EC.1.5. Proof of Proposition 3

Proof of Proposition 3. Let $LB = \Upsilon + \sum_{k \in \mathcal{K}} \sum_{b \in \mathcal{B}_k: r_{k, b} < 0} r_{k, b}$. From Lemma 1, we have that LB is a lower bound for the BMP at the current node. Given a pair of $b \in \mathcal{B}$ and $k \in \mathcal{K}$ that satisfies the conditions in this proposition, the lower bound (denoted by LB') of the BMP with the constraint $\sum_{\omega \in \Omega_{k, b}} \chi_\omega = 1$ at the current node can be calculated by

$$\begin{aligned}
LB' = LB + \min_{\omega \in \Omega_{k, b}} & \left\{ c_\omega - \pi_k^* - \sum_{t \in \mathcal{T}} \sum_{t=1}^{\tau} \sum_{t' = \min\{\tau+1, \bar{T}\}}^{\bar{T}} \beta_{\omega, b, t, t'} \lambda_{b, t}^* \right. \\
& - \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{I}^{in} \cap \mathcal{I}_k} \sum_{t' = \max\{t-f_i+1, 1\}}^t \alpha_{\omega, i, t'} \theta_t^* - \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{I}^{out} \cap \mathcal{I}_k} \sum_{t' = \max\{t-f_i+1, 1\}}^t \alpha_{\omega, i, t'} \vartheta_t^* \\
& \left. - \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{I}_k} \sum_{j \in \mathcal{I}} \sum_{(i, j) \in \mathcal{U}} \sum_{t' = \max\{t-d_i+1, 1\}}^t \alpha_{\omega, i, t'} \kappa_{t, (i, j)}^* + \sum_{(\delta, \zeta) \in \bar{\Gamma}_\Delta} \sum_{i \in \mathcal{I}_k} \sum_{t \in \mathcal{T}} \sum_{\omega \in \Omega} \chi_\omega \alpha_{\omega, i, t} \delta_{i, t} \varphi^*(\delta, \zeta) \right\},
\end{aligned}$$

which equals $LB + r_{k, b}$.

Therefore, $LB + r_{k, b}$ is a lower bound for the BMP with the constraint $\sum_{\omega \in \Omega_{k, b}} \chi_\omega = 1$ at the current node. It follows easily that $LB + r_{k, b}$ is also a lower bound for the BMP with the constraint $\sum_{\omega \in \Omega_{k, b}} \chi_\omega = 1$ at any node descended from the current node. \square

EC.1.6. Proof of Proposition 4

Proof of Proposition 4. Given any solution to the BMP, denoted by $\bar{\chi}$, the PBSP at node l' is a relaxation of that at node l . Accordingly, given $\bar{\chi}$, the DBSP at node l is a relaxation of that at

node ι' . Hence, any point in the polyhedron defined by (52)–(54) of the DBSP at node ι' is also in the polyhedron defined by (52)–(54) of the DBSP at node ι . As a result, according to Benders (1962), the Benders cuts that are valid for the BMP at node ι' are also valid for the BMP at node ι . \square

EC.1.7. Proof of Corollary 1

Proof of Corollary 1. For the root node (denoted by ι_0), we have $\Theta_{\iota_0} = \emptyset$. Hence, we have that for any node $\iota \neq \iota_0$, $\Theta_{\iota_0} \subseteq \Theta_{\iota}$. Therefore, according to Proposition 4, Benders cuts that are valid for the BMP at the root node are also valid for the BMP at any other node in the branch-and-bound tree. \square

EC.2. The Label Correcting Algorithm for Solving the SPP_s

This section describes the label correcting algorithm for solving the pricing problems in the Dantzig-Wolfe decomposition of the PBSP. Consider the pricing problem corresponding to shift s (SPP_s). The label correcting algorithm aims to identify the path, denoted by \mathcal{P}_s^* in the space-time network G_s that has the minimum cost (denoted by \tilde{c}_s^*).

In the algorithm, H_s is partitioned into two subsets: $H_s^1 = \{(i, t) | t + d_i \leq \bar{L}_s, (i, t) \in H_s\}$ and $H_s^2 = \{(i, t) | t \geq \underline{L}_s + g_s, (i, t) \in H_s\}$ (a node (i, t) can appear in both subsets). Observe that H_s^1 contains all tasks that can start before the rest period in shift s , whereas H_s^2 contains all tasks that can start after the rest period. Each $(i, t) \in H_s^1$ is associated with a label $l_{(i,t)}^1 = (\tilde{c}_{(i,t)}^1, V_{(i,t)}^1)$, where $V_{(i,t)}^1$ is the list of nodes that are visited before or at time t in the label and $\tilde{c}_{(i,t)}^1$ is the associated cost. Similarly, each $(i, t) \in H_s^2$ is associated with a label $l_{(i,t)}^2 = (\tilde{c}_{(i,t)}^2, V_{(i,t)}^2)$, where $V_{(i,t)}^2$ is the list of nodes that are visited after or at time t in the label and $\tilde{c}_{(i,t)}^2$ is the associated cost.

The algorithm generates \mathcal{P}_s^* in three steps. The first step, which is called forward label correcting, generates the $l_{(i,t)}^1$ with the minimum cost $\tilde{c}_{(i,t)}^1$ for each $(i, t) \in H_s^1$. The second step, which is called backward label correcting, generates the $l_{(i,t)}^2$ with the minimum cost $\tilde{c}_{(i,t)}^2$ for each $(i, t) \in H_s^2$. Finally, \mathcal{P}_s^* is constructed in the last step, which is called label joining and route construction. The details of these steps are presented in Sections EC.2.1, EC.2.2 and EC.2.3, respectively. We prove the correctness of the label correcting algorithm for solving the SPP_s and discuss its time complexity in Proposition EC.1 which is presented in Section EC.2.3.

EC.2.1. Forward Label Correcting

In forward label correcting, for each node $(i, t) \in H_s^1$, we find the minimum-cost (partial) path that ends at this node. To this end, we first sequence all nodes in set H_s^1 in non-decreasing order according to their start times (t), where ties are broken arbitrarily. Let (i_h, t_h) , $h = 1, \dots, |H_s^1|$ denote the h -th node in the sequence. The details of this procedure are presented in Algorithm EC.1.

Algorithm EC.1 Forward Label Correcting.

```

1: Initialization:  $l_{(i,t)}^1 = (-\delta_{i,t}, [(i,t)])$ ,  $\forall (i,t) \in H_s^1$ .
2: Sequence  $(i,t) \in H_s^1$  in non-decreasing order of  $t$ .
3: for  $h = 1, \dots, |H_s^1| - 1$  do
4:   for  $h' = h + 1, \dots, |H_s^1|$  do
5:     if  $i_h \neq i_{h'}$  &  $t_h + d_{i_h} + q_{i_h, i_{h'}} \leq t_{h'}$  then
6:       if  $\tilde{c}_{(i_h, t_h)}^1 - \delta_{i_{h'}, t_{h'}} \leq \tilde{c}_{(i_{h'}, t_{h'})}^1$  then
7:          $\tilde{c}_{(i_{h'}, t_{h'})}^1 = \tilde{c}_{(i_h, t_h)}^1 - \delta_{i_{h'}, t_{h'}}$ .
8:          $V_{(i_{h'}, t_{h'})}^1 = V_{(i_h, t_h)}^1 \leftarrow (i_{h'}, t_{h'})$ .  $\triangleright$  Operator “ $\leftarrow$ ” represents adding an element to the rear of
           a list.
9:       end if
10:    end if
11:  end for
12: end for
13: Return  $l_{(i,t)}^1$ ,  $\forall (i,t) \in H_s^1$ .

```

EC.2.2. Backward Label Correcting

In backward label correcting, for each node $(i,t) \in H_s^2$, we find the minimum-cost (partial) path that starts from this node. To this end, we first sequence all nodes in set H_s^2 in non-decreasing order according to their start times (t), where ties are broken arbitrarily. Let (i_h, t_h) , $h = 1, \dots, |H_s^2|$ denote the h -th node in the sequence. The details of this procedure are presented in Algorithm EC.2.

Algorithm EC.2 Backward Label Correcting.

```

1: Initialization:  $l_{(i,t)}^2 = (-\delta_{i,t}, [(i,t)])$ ,  $\forall (i,t) \in H_s^2$ .
2: Sequence  $(i,t) \in H_s^2$  in non-decreasing order of  $t$ .
3: for  $h = |H_s^2|, \dots, 2$  do
4:   for  $h' = h - 1, \dots, 1$  do
5:     if  $i_h \neq i_{h'}$  &  $t_{h'} + d_{i_{h'}} + q_{i_{h'}, i_h} \leq t_h$  then
6:       if  $\tilde{c}_{(i_h, t_h)}^2 - \delta_{i_{h'}, t_{h'}} \leq \tilde{c}_{(i_{h'}, t_{h'})}^2$  then
7:          $\tilde{c}_{(i_{h'}, t_{h'})}^2 = \tilde{c}_{(i_h, t_h)}^2 - \delta_{i_{h'}, t_{h'}}$ .
8:          $V_{(i_{h'}, t_{h'})}^2 = [(i_{h'}, t_{h'})] \leftarrow V_{(i_h, t_h)}^2$ .
9:       end if
10:    end if
11:  end for
12: end for
13: Return  $l_{(i,t)}^2$ ,  $\forall (i,t) \in H_s^2$ .

```

EC.2.3. Label Joining and Route Construction

In the last step, we identify the pilot route with the minimum cost (\mathcal{P}_s^*) by constructing and searching among three sets of pilot routes, including (i) the set of routes that terminate at the rest period, (ii) the set of routes that start from the rest period, and (iii) the set of routes where the rest period serves as an intermediate node. Algorithm EC.3 shows the pseudo-code of this procedure, where we let j denote the rest period.

Algorithm EC.3 Label Joining and Pilot Route Construction.

```

1: Initialization:  $\tilde{c}_s^* = M$ . ▷  $M$  is a sufficiently large constant.
2: for  $(i, t) \in H_s^1$  do
3:   if  $\tilde{c}_s^* > \tilde{c}_{(i,t)}^1 + c_s^3$  then
4:      $\tilde{c}_s^* = \tilde{c}_{(i,t)}^1 + c_s^3$ .
5:      $\mathcal{P}_s^* = V_{i,t}^1 \leftarrow (j, \bar{L}_s)$ . ▷  $(j, \bar{L}_s)$  represents starting the rest period at time step  $\bar{L}_s$ .
6:   end if
7: end for
8: for  $(i, t) \in H_s^2$  do
9:   if  $\tilde{c}_s^* > \tilde{c}_{(i,t)}^2 + c_s^3$  then
10:     $\tilde{c}_s^* = \tilde{c}_{(i,t)}^2 + c_s^3$ .
11:     $\mathcal{P}_s^* = [(j, \underline{L}_s)] \leftarrow V_{i,t}^2$ . ▷  $(j, \underline{L}_s)$  represents starting the rest period at time step  $\underline{L}_s$ .
12:   end if
13: end for
14: for  $(i, t) \in H_s^1$  do
15:   for  $(i', t') \in H_s^2$  do
16:     if  $i \neq i' \ \& \ t + d_i + q_{i,i'} \leq t'$  then
17:        $t_j = \min\{t + d_i, \underline{L}_s\}$ . ▷ Find the earliest feasible start time for the rest period.
18:       if  $t_j + g_s \leq t'$  then ▷ Recall that  $g_s$  denote the minimum duration for a rest period in shift  $s$ .
19:         if  $\tilde{c}_s^* > \tilde{c}_{(i,t)}^1 + \tilde{c}_{(i',t')}^2 + c_s^3$  then
20:            $\tilde{c}_s^* = \tilde{c}_{(i,t)}^1 + \tilde{c}_{(i',t')}^2 + c_s^3$ .
21:            $\mathcal{P}_s^* = V_{(i,t)}^1 \leftarrow (j, t_j) \leftarrow V_{(i',t')}^2$ . ▷  $(j, t_j)$  represents starting the rest period at time step
 $t_j$ .
22:         end if
23:       end if
24:     end if
25:   end for
26: end for
27: Return  $\mathcal{P}_s^*$  and  $\tilde{c}_s^*$ .

```

PROPOSITION EC.1. *The label correcting algorithm solves the SPP_s in $\mathcal{O}(|H_s|^2)$ time.*

Proof. We first prove the correctness of the algorithm. In the first step of the algorithm, we aim to find, for each node $(i, t) \in H_s^1$, the minimum-cost (partial) path that ends at this node. All nodes in H_s^1 have been sequenced in non-decreasing order of t , and it is only possible for a path to travel from node (i, t) to another node (i', t') , if $t' \geq t$. Therefore, for the first node in the sequence, i.e., (i_1, t_1) , the minimum cost and the minimum-cost path have been contained in the label $l_{(i,t)}^1$ after initialization. Next, consider the two-layer for-loop in Algorithm EC.1. Because routes can only travel through the nodes chronologically, when $h = 1$ in the first layer, the algorithm finds the optimal label $l_{(i_2, t_2)}^1$ for node (i_2, t_2) in the second layer. By induction, we can prove that the algorithm finds the best label for node $(i_{\bar{h}+1}, t_{\bar{h}+1})$ when $h = \bar{h}$ in the for-loop. This indicates that Algorithm EC.1 can identify the minimum-cost (partial) path that ends at each node $(i, t) \in H_s^1$ after executing the for-loop.

Following a similar logic, one can prove that Algorithm EC.2 generates the minimum-cost (partial) path that starts from each node $(i, t) \in H_s^2$.

Because H_s^1 (resp. H_s^2) contains all nodes that can be visited before (resp. after) the rest period starts in a pilot route, by comparing the labels corresponding to all nodes in H_s^1 (resp. H_s^2), Algorithm EC.3 finds the pilot route with the minimum cost that terminates at (resp. starts from) the rest period. Finally, by concatenating all compatible pairs of labels corresponding to nodes from H_s^1 and H_s^2 , Algorithm EC.3 finds the pilot route with the minimum cost where the rest period is an intermediate node.

As for the time complexity of the algorithm, in Algorithms EC.1 and EC.2, it takes $\mathcal{O}(|H_s| \log |H_s|)$ time and $\mathcal{O}(|H_s|^2)$ time, respectively, to sequence the nodes in H_s^1 and H_s^2 and correct their labels. In Algorithms EC.3, the optimal pilot routes from the three sets are generated in $\mathcal{O}(|H_s|)$, $\mathcal{O}(|H_s|)$, and $\mathcal{O}(|H_s|^2)$ time, respectively. It can be readily seen that the overall time complexity for the label correcting algorithm is $\mathcal{O}(|H_s|^2)$. \square

EC.3. The Pricing Problems for the BMP with the LBL cuts

In this section, we explain the pricing problems for the BMP with the LBL cuts.

Let $\varpi = (\varpi_{s,t} | t \in \mathcal{T}_s, s \in \mathcal{S})$ denote vector of dual variables associated with constraints (72). New vessel routes are generated by solving a pricing problem (denoted by $MPP'_{k,b}$) for each $k \in \mathcal{K}$ and $b \in \mathcal{B}_k$. The $MPP'_{k,b}$ can be defined as a problem that identifies the arc with the minimum cost in a three-dimensional task-time-shift network $G'_{k,b} = (N'_{k,b}, A'_{k,b})$, where $N'_{k,b}$ and $A'_{k,b}$ denote the sets of nodes and arcs in the network, respectively. Here, $N'_{k,b} = \{(i, t, s) | s \in \mathcal{S}_t, t \in \mathcal{E}_i, i \in \mathcal{I}_k\}$

and $A'_{k,b} = \{[(i, t, s), (i', t', s')] | t' - t \geq d_i + h_{k,b}, (i, t, s), (i', t', s') \in N'_{k,b}, i \in \mathcal{I}^{in}, i' \in \mathcal{I}^{out}\}$. The cost of sending a unit flow through arc $[(i, t, s), (i', t', s')]$, denoted by $\hat{c}'_{[(i,t,s),(i',t',s')]}$, is calculated by

$$\hat{c}'_{[(i,t,s),(i',t',s')]} = \hat{c}_{[(i,t),(i',t')]} + \sum_{\tau=t}^{\min\{t+d_i+q'_{i,t,s}-1, \bar{T}_s\}} \varpi_{s,\tau} + \sum_{\tau=t'}^{\min\{t'+d_{i'}+q'_{i',t',s'}-1, \bar{T}_{s'}\}} \varpi_{s',\tau}, \quad (\text{EC.21})$$

where $\hat{c}_{[(i,t),(i',t')]}$ is calculated by (60).

EC.4. The Heuristic for Generating Integer Solutions

In this section, we explain the heuristic that tries to construct a feasible integer solution for M2 from a fractional solution. Let χ^0 and μ^0 denote the optimal (fractional) solution to the BMP and PBSP at a node in the branch-and-bound tree, respectively. The heuristic consists of three stages. In the first stage, we try to construct a feasible integer solution (denoted by χ^1) for the BMP (without the Benders cuts (56)). If such a solution can be constructed, then in the second step, we solve the associated PBSP using column generation. Let μ^1 denote the solution to the PBSP. If μ^1 is fractional, we construct an integer solution (denoted by μ^2) based on μ^1 in the last step. Note that if χ^0 is integral then we skip the first stage and let $\chi^1 = \chi^0$ and if μ^1 is integral then we skip the third stage and let $\mu^2 = \mu^1$. Corresponding to each feasible integer solution for M2 generated by the heuristic, we obtain a valid upper bound which equals $\sum_{\omega \in \Omega} \chi^1_{\omega} \bar{c}_{\omega} + \sum_{\phi \in \Phi} \mu^2_{\phi} \bar{c}_{\phi}$. Details of the first and the third stages in the heuristic are explained as follows.

EC.4.1. Constructing Vessel Routes

Given a fractional solution (χ^0) to the BMP, we apply a greedy algorithm to construct a feasible integer solution (χ^1) for the BMP without the Benders cuts (56). The algorithm leverages information from the current fractional solution. Let $\Omega^+ = \{\omega | \chi^0_{\omega} > 0, \omega \in \Omega\}$ and let \mathbb{S}_{Ω^+} denote a sequence which lists elements in Ω^+ in non-decreasing order of χ^0_{ω} . We let ω_{ϱ} , $\varrho = 1, 2, \dots, |\Omega^+|$ denote the ϱ -th element in \mathbb{S}_{Ω^+} . The procedure for constructing χ^1 is presented in Algorithm EC.4, and we make use of the following additional notation in the pseudo-code:

- $k(\omega)$: The vessel $k \in \mathcal{K}$ associated with vessel route ω .
- $b(\omega)$: The berth $b \in \mathcal{B}$ used in vessel route ω .
- $i^{in}(\omega)$: The task $i \in \mathcal{I}^{in}$ covered in vessel route ω .
- $i^{out}(\omega)$: The task $i \in \mathcal{I}^{out}$ covered in vessel route ω .
- $t_1(\omega)$: The time step $t \in \mathcal{T}$ when the vessel in ω starts sailing into its berth.
- $t_2(\omega)$: The time step $t \in \mathcal{T}$ when the vessel in ω reaches its berth.
- $t_3(\omega)$: The time step $t \in \mathcal{T}$ when the vessel in ω leaves its berth.
- $\bar{\mathcal{K}}$: Set of vessels that have associated vessel routes in Ω^1 .
- $\bar{\mathcal{T}}^{in}$: Set of time steps at which vessels cannot start sailing into berths.
- $\bar{\mathcal{T}}^{out}$: Set of time steps at which vessels cannot start sailing out of berths.
- $\tilde{\mathcal{T}}_i$: Set of time steps at which task $i \in \mathcal{I}$ cannot start.
- $\hat{\mathcal{T}}_b$: Set of time steps at which berth $b \in \mathcal{B}$ is occupied by vessels.

Algorithm EC.4 Vessel Route Construction.

```

1: Initialization:  $\Omega^1 = \emptyset$ ;  $\bar{\mathcal{K}} = \bar{\mathcal{T}}^{in} = \bar{\mathcal{T}}^{out} = \emptyset$ ;  $\hat{\mathcal{T}}_b = \emptyset$ ,  $b \in \mathcal{B}$ ;  $\tilde{\mathcal{T}}_i = \emptyset$ ,  $i \in \mathcal{I}$ .
2: for  $\varrho = 1, 2, \dots, |\Omega^+|$  do
3:   if  $k(\omega_\varrho) \notin \bar{\mathcal{K}}$  &  $t_1(\omega_\varrho) \notin \bar{\mathcal{T}}^{in}$  &  $t_3(\omega_\varrho) \notin \bar{\mathcal{T}}^{out}$  &  $t_1(\omega_\varrho) \notin \tilde{\mathcal{T}}_{i^{in}(\omega_\varrho)}$  &  $t_3(\omega_\varrho) \notin \tilde{\mathcal{T}}_{i^{out}(\omega_\varrho)}$  &  $\mathbb{Z}_{[t_2(\omega_\varrho), t_3(\omega_\varrho)-1]} \cap \hat{\mathcal{T}}_{b(\omega_\varrho)} = \emptyset$  then
4:      $\bar{\mathcal{K}} = \bar{\mathcal{K}} \cup \{k(\omega_\varrho)\}$ . ▷ Update the set of vessels that have been included in the solution.
5:      $\Omega^1 = \Omega^1 \cup \{\omega_\varrho\}$ . ▷ Update the set of vessel routes that have been included in the solution.
6:     Call sub-procedure P( $\omega$ ) in Algorithm EC.5. ▷ Update the constraints.
7:   end if
8:   if  $|\bar{\mathcal{K}}| = |\mathcal{K}|$  then
9:     Break.
10:  end if
11: end for
12: for  $k \in \mathcal{K} \setminus \bar{\mathcal{K}}$  do ▷ Construct routes for vessels not included in the current solution;  $k$  are enumerated chronologically according to their earliest ready times for entering the channel.
13:   for  $t_1 \in \mathcal{E}_{i^{in}}$  do ▷  $i^{in} \in \mathcal{I}_k \cap \mathcal{I}^{in}$ .  $t_1$  are enumerated chronologically.
14:     for  $t_3 \in \mathcal{E}_{i^{out}}$  do ▷  $i^{out} \in \mathcal{I}_k \cap \mathcal{I}^{out}$ .  $t_3$  are enumerated chronologically.
15:       if  $t_1 \notin \bar{\mathcal{T}}^{in}$  &  $t_3 \notin \bar{\mathcal{T}}^{out}$  &  $t_1 \notin \tilde{\mathcal{T}}_{i^{in}}$  &  $t_3 \notin \tilde{\mathcal{T}}_{i^{out}}$  then
16:         for  $b \in \mathcal{B}_k$  do ▷  $b$  are enumerated in non-increasing order of  $c_{k,b}^2$ .
17:            $t_2 = t_1 + d_{i^{in}}$ .
18:           if  $t_2 \geq a_b$  &  $t_2 + h_{k,b} \leq t_3$  &  $\mathbb{Z}_{[t_2, t_3-1]} \cap \hat{\mathcal{T}}_b = \emptyset$  then
19:             Construct a vessel route  $\omega$  such that  $k(\omega) = k$ ,  $b(\omega) = b$ ,  $t_1(\omega) = t_1$ ,  $t_2(\omega) = t_2$ ,  $t_3(\omega) = t_3$ , and  $\bar{c}_\omega = c_{i^{in}, t_1}^1 + c_{i^{out}, t_3}^1 + c_{k,b}^2$ .
20:              $\bar{\mathcal{K}} = \bar{\mathcal{K}} \cup \{k\}$ .
21:              $\Omega^1 = \Omega^1 \cup \{\omega\}$ .
22:             Call sub-procedure P( $\omega$ ) in Algorithm EC.5.
23:             Break and go to Line 12.
24:           end if
25:         end for
26:       end if
27:     end for
28:   end for
29: end for
30: Return  $\Omega^1$ .

```

Algorithm EC.5 Constraint Update $[\mathbf{P}(\omega)]$.

-
- 1: $\bar{\mathcal{T}}^{in} = \bar{\mathcal{T}}^{in} \cup \mathbb{Z}_{[t_1(\omega), t_1(\omega) + f_{i^{in}(\omega)} - 1]}$. \triangleright Update time steps at which vessels cannot start sailing into the berths.
 - 2: $\hat{\mathcal{T}}_{b\omega} = \hat{\mathcal{T}}_{b\omega} \cup \mathbb{Z}_{[t_2(\omega), t_3(\omega) - 1]}$. \triangleright Update time steps at which the berth is occupied.
 - 3: $\bar{\mathcal{T}}^{out} = \bar{\mathcal{T}}^{out} \cup \mathbb{Z}_{[t_3(\omega), t_3(\omega) + f_{i^{out}(\omega)} - 1]}$. \triangleright Update time steps at which vessels cannot start sailing out of the berths.
 - 4: **for** $j \in I$ **do** \triangleright Update time steps at which a task cannot start due to the non-simultaneity constraints (43).
 - 5: **if** $(i^{in}(\omega), j) \in \mathcal{U}$ **then**
 - 6: $\tilde{\mathcal{T}}_j = \tilde{\mathcal{T}}_j \cup \mathbb{Z}_{[t_1(\omega), t_1(\omega) + d_{i^{in}(\omega)} - 1]}$.
 - 7: **end if**
 - 8: **if** $(i^{out}(\omega), j) \in \mathcal{U}$ **then**
 - 9: $\tilde{\mathcal{T}}_j = \tilde{\mathcal{T}}_j \cup \mathbb{Z}_{[t_3(\omega), t_3(\omega) + d_{i^{out}(\omega)} - 1]}$.
 - 10: **end if**
 - 11: **end for**
 - 12: Return $\bar{\mathcal{T}}^{in}$, $\bar{\mathcal{T}}^{out}$, $\hat{\mathcal{T}}_b$, $b \in \mathcal{B}$, $\tilde{\mathcal{T}}_i$, $i \in \mathcal{I}$.
-

One can easily verify that if $|\bar{\mathcal{K}}| = |\mathcal{K}|$, then by letting $\chi_\omega^1 = 1, \forall \omega \in \Omega^1$, and $\chi_\omega^1 = 0, \forall \omega \in \Omega \setminus \Omega^1$, χ^1 becomes a feasible integer solution for the BMP (without Benders cuts). Meanwhile, observe that it is also possible that Algorithm EC.4 fails to generate a feasible integer solution. In this case, the primal heuristic fails to find a valid upper bound for M2.

EC.4.2. Constructing Pilot Routes

Given a feasible integer solution (χ^1) to the BMP, we solve the associated PBSP using column generation. Suppose the delivered solution to the PBSP, denoted by μ^1 , is fractional. In this case, to construct a feasible integer solution for the PBSP (denoted by μ^2), we use an algorithm that is similar to the algorithm for generating vessel routes in the previous section. The algorithm can generate a feasible integer solution for the PBSP given any feasible integer solution for the BMP (χ^1). The details are explained as follows.

Constructing μ^2 is equivalent to generating a set $\Phi^2 \subseteq \Phi$ such that $\mu_\phi^2 = 1, \forall \phi \in \Phi^2$ and $\mu_\phi^2 = 0, \forall \phi \in \Phi \setminus \Phi^2$. Algorithm EC.6 presents the details of the procedure. In the algorithm, we let $\Phi^+ = \{\phi \mid \mu_\phi^1 > 0, \phi \in \Phi\}$, and let \mathbb{S}_{Φ^+} denote a sequence that lists the elements in Φ^+ in non-decreasing order of μ_ϕ^1 . Besides, let $\phi_\varrho, \varrho = 1, 2, \dots, |\Phi^+|$ denote the ϱ -th element in \mathbb{S}_{Φ^+} . The following notation is also used in the pseudo-code: $s(\phi)$ which denotes the shift associated with pilot route ϕ , $\mathcal{I}(\phi)$ which denotes the set of tasks covered in pilot route ϕ , and $\bar{\mathcal{I}}$ which denotes the set of tasks that have been covered by pilot routes in Φ^2 .

Algorithm EC.6 Pilot Route Construction.

```

1: Initialization:  $\Phi^2 = \emptyset$ ;  $\bar{\mathcal{I}} = \emptyset$ .
2: for  $\varrho = 1, 2, \dots, |\Phi^+|$  do
3:    $\mathcal{I}' = \emptyset$ .
4:   for  $i \in \mathcal{I}(\phi_\varrho)$  do
5:     if  $i \notin \bar{\mathcal{I}}$  then
6:        $\mathcal{I}' = \mathcal{I}' \cup \{i\}$ .
7:     end if
8:   end for
9:   if  $\mathcal{I}' \neq \emptyset$  then
10:     $\bar{\mathcal{I}} = \bar{\mathcal{I}} \cup \mathcal{I}'$ .
11:    Construct a pilot route  $\phi' \in \Phi_{s(\phi)}$  that covers all tasks in  $\mathcal{I}'$ ;  $\triangleright$  Route  $\phi'$  can be generated by
    letting all tasks in  $\mathcal{I}'$  and the rest period start at the same times as they do in route  $\phi_\varrho$ ; The cost of the
    route equals  $\bar{c}_{\phi'} = c_{s(\phi)}^3$ .
12:     $\Phi^2 = \Phi^2 \cup \{\phi'\}$ .
13:   end if
14:   if  $|\bar{\mathcal{I}}| = |\mathcal{I}|$  then
15:     Break.
16:   end if
17: end for
18: Return  $\Phi^2$ .

```

EC.5. Data of the Hong Kong Container Port and Parameter Settings in the Instances

EC.5.1. Data of the Port

We generated the instances based on real operational data from the Hong Kong Container Port. The physical layout of the port is depicted in Figure E.1(a). There are 24 berths in the Kwai Tsing Container Terminals. More than 85% of all vessel arrivals in this port travel through the East Lamma Channel, which is a bidirectional channel with two traffic lanes: one for incoming vessels and the other for outgoing vessels. Due to its sufficient water depth, vessels of all sizes can sail in the channel at any time of the day (i.e., there is no tidal window for vessels in the channel). There is a pilot boarding station at the entrance of the channel where pilots can board or deboard vessels. The port also has an anchorage located near the East Lamma Channel, where arriving vessels can wait before sailing into the channel.

To better explain the movements of vessels in the port, in Figure E.1(b), we illustrate the locations of vessels in the port at a given time and the trajectory of a vessel that was handled in the port. This figure was adapted from a screenshot of a software application that monitors real-time vessel positions and movements in the Hong Kong Container Port.

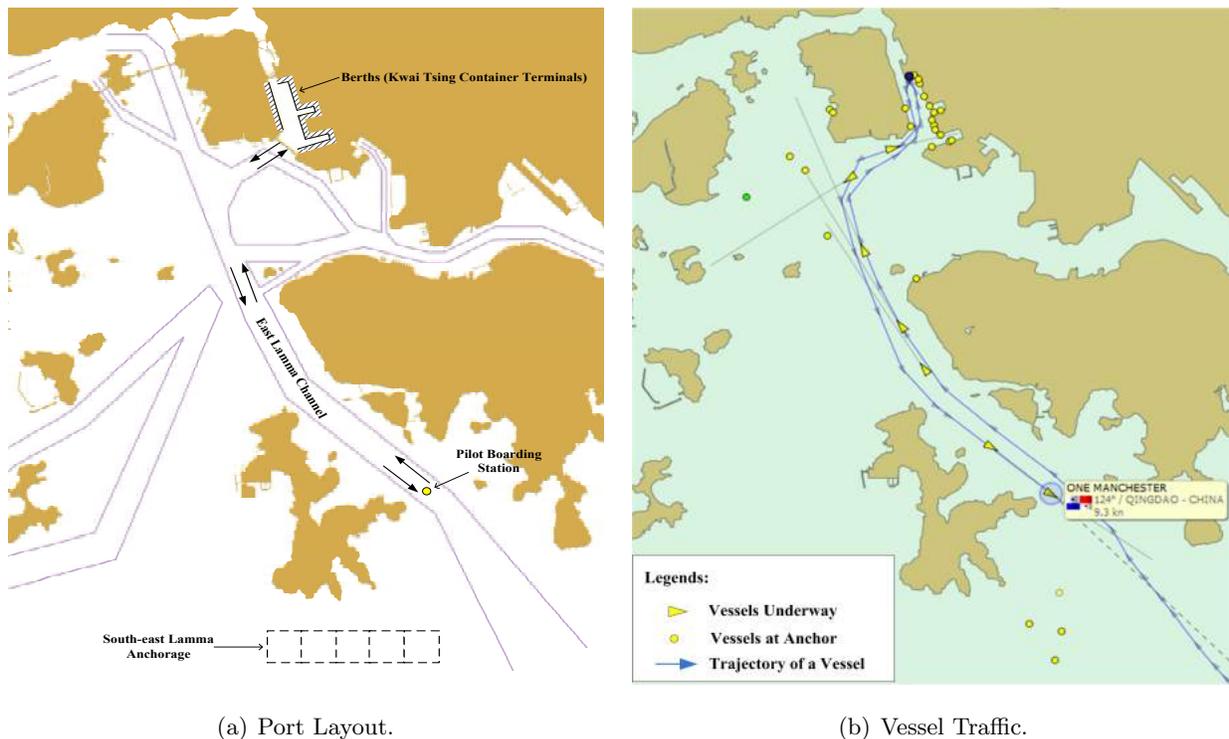


Figure EC.1 Layout of the Hong Kong Container Port and an Illustration of Vessel Traffic in the Port.

In 2017, 10,633 vessel arrivals were handled in the port (The Marine Department of Hong Kong 2018), which translates to approximately 1.21 vessel arrivals per berth per day on average. To derive the vessel service data, we collected messages sent by the onboard Automatic Identification System (AIS) of the vessels handled in the Hong Kong Container Port in 2017. The AIS is to vessels what GPS is to cars. The reader is referred to Yang et al. (2019) for more information on the AIS and AIS messages. By leveraging the AIS data, we obtained 7,062 complete vessel arrival records. Each record depicts the movements of a vessel (accurate to one minute) during its entire port stay. For each of the arrivals, we identified (i) the capacity of the associated vessel, (ii) the handling time at the berths, and (iii) the transit time between the pilot boarding station and the berths.

Table EC.1 reports the distribution of the vessel capacities associated with the 7,062 vessel arrivals. We classify vessels into five groups according to their capacities (in twenty-foot equivalent units, or TEU): feeder, intermediate (S), intermediate (L), large, and jumbo. We present the distribution of handling times (in hours) of different vessels in Figure EC.2. Among the 7,062 vessel arrivals, 6,425 and 5,655 arrivals entered and departed the port through the East Lamma Channel, respectively. The average traveling time between the berths and the pilot boarding station in the channel was 64 minutes, and nearly 80% of transit times lay within the range of [54,72] minutes.

Table EC.1 Distribution of Vessel Capacities.

Type	Size (TEU)	Number	Ratio (%)
Feeder	[500, 3000]	3815	54%
Intermediate (S)	[3001, 6000]	1710	24%
Intermediate (L)	[6001, 9000]	988	14%
Large	[9001, 12000]	229	3%
Jumbo	[12001, 18500]	320	5%

Pilotage is compulsory in Hong Kong; all vessels over 3,000 gross tons must have a pilot on board when navigating in the port (the smallest container vessel in our record is over 8,900 gross tons). There are approximately 100 pilots licensed to serve vessels in Hong Kong (The Marine Department of Hong Kong 2020). To board vessels, pilots travel in the port using pilot boats. Pilot boats are significantly smaller than container vessels; the typical length of a pilot boat is between 12 and 14 meters, whereas the smallest feeder container vessel in our record is 138 meters long. While the speed of container vessels in the channel is normally between 10–15 knots, pilot boats can travel at much higher speeds (up to a maximum of 30 knots) in port waters. Accordingly, pilot boats can overtake container vessels when sailing in the channel. A pilot boat can travel between any two berths in the port within 5 minutes and between berths and the pilot boarding station within 20 minutes.

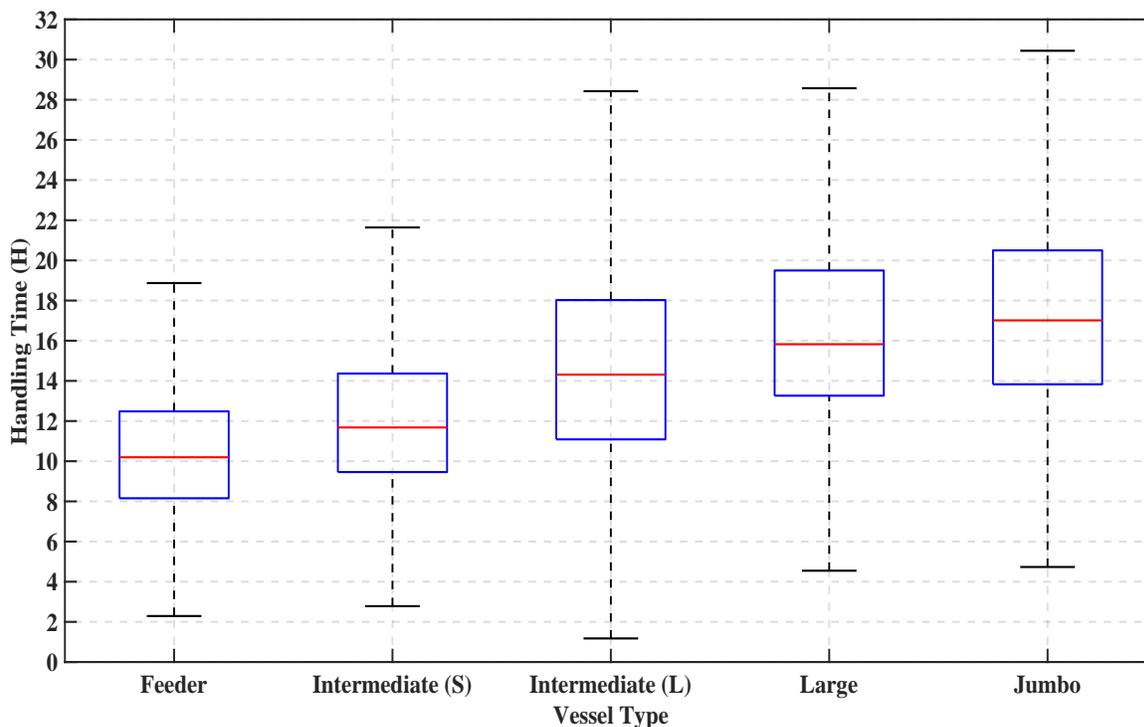


Figure EC.2 Distribution of Vessel Handling Times.

EC.5.2. Parameter Settings in the Instances

Based on the operational data of the Hong Kong Container Port, we generate parameters in our testing instances as follows.

First, corresponding to the five vessel types listed in Table EC.1, we classify vessels in the instances into five grades: Grades I, II, III, IV, and V. A higher grade indicates larger capacities. Among all vessels calling at the seaport in an instance, the probabilities of the vessels being of Grades I, II, III, IV, and V are set to 54%, 24%, 14%, 3%, and 5%, respectively. Vessels arrive at the seaport dynamically and uniformly throughout the planning horizon. In the instances, a vessel can be handled at any berth in the seaport, and the cargo handling times for vessel k at all berths are set to be identical and equal to \hat{h}_k . Here, \hat{h}_k is randomly generated using the uniform distributions $U(8, 12)$, $U(10, 14)$, $U(11, 18)$, $U(13, 20)$, and $U(14, 21)$ (in hours) for vessels of Grades I, II, III, IV, and V, respectively.

Second, requirements regarding channel traffic management in the instances are set as follows. The time for a vessel to travel between the anchorage and berths (including sailing in the channel and berthing into or unberthing from a berth), denoted by \hat{d} , is set to 70 minutes. There should be a minimum headway between any two vessels that sail in the same direction in the channel. Meanwhile, large-sized vessels cannot sail in opposite directions in the channel simultaneously.

However, the exact minimum headway and non-simultaneity requirements in the Hong Kong Container Port are not available to us. In the instances, we set the minimum headway to be 10 minutes, which is in accordance with the setting used by Jia et al. (2019). We also stipulate that vessels of Grade IV or V cannot sail in opposite directions in the channel simultaneously.

Third, the time windows of the pilotage tasks are set in such a way as to fulfill the following two requirements: (i) any incoming vessel should start sailing into the berths no more than eight hours after arrival, and (ii) the total turnaround time of vessel k should not exceed the sum of the sailing time between the anchorage and the berths and the handling time ($2\hat{d} + \hat{h}_k$), plus a total buffer time of 16 hours.

Fourth, given that seaports operate around the clock, we simulate the initial condition in the seaport as follows. We set the probability of a berth being occupied by a vessel to $0.5R$ (R denotes the vessel arrival rate [vessel arrivals per berth per day]). For an occupied berth, the probabilities of it being occupied by a Grade I, II, III, IV, and V vessel are set to 54%, 24%, 14%, 3%, and 5%, respectively. If a berth is occupied, its available time (counting from the beginning of the planning horizon) is generated from $U(1, 10)$ (in hours); otherwise, we set the available time to zero. Meanwhile, to impose the minimum headway requirement, no vessels in the instances can leave their berths within the first 10 minutes after an initially occupied berth becomes available. Moreover, to impose the non-simultaneity requirements between large vessels, if a berth is occupied by a Grade IV or V vessel, no Grade IV or V vessels in the instances can start sailing into the berths within the first 70 minutes after the berth becomes available. To impose these requirements, the time windows of the pilotage tasks for the vessels are adjusted accordingly.

Fifth, in each instance, each pilot working in a shift is awarded a rest period that should last at least an hour and cannot start within the first 30% or the last 30% period in the shift. A pilot can maneuver multiple vessels in a shift, and a minimum setup time is required to reposition a pilot between two tasks. If the repositioning does not necessitate travel through the channel, we set the minimum setup time to 10 minutes; otherwise, we set the minimum setup time to 30 minutes.

Sixth, because we consider a discrete-time problem, the time points (for defining various time windows) and time durations (of tasks, rest periods, and vessel handling) generated in continuous time are translated into parameters depicted by discrete time steps. Such translation is conducted by applying the appropriate ceiling or flooring operators on the continuous-time parameters to ensure the feasibility of the obtained discrete-time solutions for the original continuous-time instances.

Finally, we set the cost components as follows. First, tasks that start later than their earliest possible start times are punished. The unit-time delay cost for tasks corresponding to vessels of Grades I, II, III, IV, and V sailing into berths are set to 2, 3, 4, 5, and 6, respectively. The unit-time

delay cost for tasks corresponding to vessels of Grades I, II, III, IV, and V leaving berths are set to 4, 6, 8, 10, and 12, respectively. Meanwhile, the cost for handling vessel k at berth b is set to $c_{k,b}^2 = \lceil 5o_b \hat{h}_k \rceil$, where o_b is generated from $U(1, 1.5)$. Finally, the cost paid by the port authority for assigning a pilot to work in a shift is set to 48 (i.e., 1 per unit time).

References

- Jia S, Li CL, Xu Z (2019) Managing navigation channel traffic and anchorage area utilization of a container port. *Transportation Science* 53(3):728–745.
- The Marine Department of Hong Kong (2018) Port of Hong Kong Statistical Tables 2017. Technical report, URL https://www.mardep.gov.hk/en/publication/pdf/portstat_ast_2017.pdf, Accessed September 1, 2020.
- The Marine Department of Hong Kong (2020) Pilotage in Hong Kong. https://www.mardep.gov.hk/en/pub_services/ocean/pilot.html, Accessed November 18, 2020.
- Yang D, Wu L, Wang S, Jia H, Li KX (2019) How big data enriches maritime research—a critical review of automatic identification system (AIS) data applications. *Transport Reviews* 39(6):755–773.