https://dx.doi.org/10.1016/j.trb.2023.102851

Transportation Research Part B

The impact of autonomous ships in regional waterways

--Manuscript Draft--

Manuscript Number:	TRB-D-23-00325R1
Article Type:	Research Paper
Keywords:	Autonomous ship; shipping company operations; sample average approximation; Benders decomposition; branch-and-cut
Corresponding Author:	Lu Zhen Shanghai University Shanghai, CHINA
First Author:	Wei Wang
Order of Authors:	Wei Wang
	Shuaian Wang
	Lu Zhen
	Gilbert Laporte
Abstract:	Technological innovation has been reshaping all walks of life, and the marine shipping industry is no exception. Autonomous vessels have gained significant attention due to their numerous advantages. However, regulatory constraints and expensive manufacturing costs are impeding the application of autonomous vessels. To overcome these challenges, this research conducts experiments with autonomous ships on national waterways with less regulation and develops a model to investigate their impact on shipping company operations. The model simultaneously optimizes ship routing, fleet sizing, fleet deployment, and demand fulfillment, taking into account demand uncertainty. Two solution methods, i.e., sample average approximation and a two-phase Benders-based branch-and-cut algorithm, are proposed to solve the problem with acceleration strategies, including column generation and variable fixing. The performance of several solution techniques is tested through numerical experiments using real-world data. Besides, sensitivity analyses are conducted to further discuss the influence of key factors and derive constructive managerial insights for shipping companies.
Suggested Reviewers:	Dong-Ping Song University of Liverpool dongping.song@liverpool.ac.uk
	Kjetil Fagerholt Norwegian University of Science and Technology kjetil.fagerholt@iot.ntnu.no
	ManWo Ng Old Dominion University mng@odu.edu
	Xiangtong Qi The Hong Kong University of Science and Technology ieemqi@ust.hk
	Martin Savelsbergh Georgia Institute of Technology martin.savelsbergh@isye.gatech.edu
Response to Reviewers:	All questions have been answered in the response letter.

- The shipping network consists of both conventional manned ships and autonomous ships.
- This research simultaneously optimizes ship routing, fleet sizing, fleet deployment, and demand fulfillment.
- This study simultaneously considers heterogeneous fleet, multi-trip, split delivery, and uncertainty.
- Two types of solution algorithms are proposed.

The impact of autonomous ships in regional waterways

Wei Wang [a], Shuaian Wang [a], Lu Zhen [b*], Gilbert Laporte [cd]

[a] *Faculty of Business, The Hong Kong Polytechnic University, Hong Kong*

[b] *School of Management, Shanghai University, Shanghai, China*

[c] *Department of Decision Sciences, HEC Montréal, Montréal, Québec, Canada*

[d] *School of Management, University of Bath, Bath, United Kingdom*

[*] Corresponding author. Email addresses: wei-vi.wang@connect.polyu.hk (W Wang), wangshuaian@gmail.com (S Wang), lzhen@shu.edu.cn (L Zhen), gilbert.laporte@cirrelt.net (G Laporte).

The impact of autonomous ships in regional waterways

Wei Wang [a], Shuaian Wang [a], Lu Zhen [b*], Gilbert Laporte [cd]

[a] *Faculty of Business, The Hong Kong Polytechnic University, Hong Kong*

[b] *School of Management, Shanghai University, Shanghai, China*

[c] *Department of Decision Sciences, HEC Montréal, Montréal, Québec, Canada*

[d] *School of Management, University of Bath, Bath, United Kingdom*

Abstract

Technological innovation has been reshaping all walks of life, and the marine shipping industry is no exception. Autonomous vessels have gained significant attention due to their numerous advantages. However, regulatory constraints and expensive manufacturing costs are impeding the application of autonomous vessels. To overcome these challenges, this research conducts experiments with autonomous ships on national waterways with less regulation and develops a model to investigate their impact on shipping company operations. The model simultaneously optimizes ship routing, fleet sizing, fleet deployment, and demand fulfillment, taking into account demand uncertainty. Two solution methods, i.e., sample average approximation and a two-phase Benders-based branch-and-cut algorithm, are proposed to solve the problem with acceleration strategies, including column generation and variable fixing. The performance of several solution techniques is tested through numerical experiments using real-world data. Besides, sensitivity analyses are conducted to further discuss the influence of key factors and derive constructive managerial insights for shipping companies.

Keywords: Autonomous ship, shipping company operations, sample average approximation, Benders decomposition, branch-and-cut

1. Introduction

Technological innovation continues to improve people's lives and solve problems in all walks of life. In marine shipping, researchers and practitioners have become increasingly interested in autonomous vessels over the past two decades (Gu et al., 2021). The popularity of autonomous vessels can be explained by multiple reasons. First, the removal of the deckhouse and accommodation structure provides additional

* Corresponding author. Email addresses: wei-vi.wang@connect.polyu.hk (W Wang), wangshuaian@gmail.com (S Wang), lzhen@shu.edu.cn (L Zhen), gilbert.laporte@cirrelt.net (G Laporte).

space for cargoes, which means that autonomous ships can carry more cargoes than conventional ships of the same size. Second, reduced light displacement (LDT) and air resistance save fuel consumption, thus making autonomous ships more environmentally friendly. Third, autonomous ships are immune to the rising labor costs and the lack of workforce in the industry. Last, autonomous vessels could avoid some human errors, making them safer than conventional ones (Yang et al., 2023).

Although autonomous ships have multiple advantages, regulatory constraints and high manufacturing costs are impeding their application. To overcome regulatory challenges, one could experiment with autonomous ships on national waterways which are more flexible than transnational waterways. For the cost problem, one could redesign the shipping network according to the characteristics of autonomous ships, which may make the shipping network more profitable than with the exclusive use of conventional ships. For example, due to larger capacity, an autonomous ship may visit more ports on a route, hence reducing the number of ships deployed, and ultimately saving operational costs. Therefore, we develop a model to investigate the impact of autonomous ships on shipping company operations.

We consider a problem setting where a container liner shipping company operates a fleet of heterogeneous vessels that consist of both conventional manned ships and autonomous ships on national waterways. Containers are transported along a feeder network with a hub port and many feeder ports, as shown in Figure 1. Due to waterway restrictions on feeder ports, such as draft and width constraints, the types of ships that can travel on this network are limited (Lin et al., 2020). Autonomous ships, as mentioned above, have the advantage of carrying more cargoes while consuming less energy compared to conventional ships of the same size. In this context, "size" refers to the physical dimensions of the vessels, i.e., draft and beam. As a result, ports inaccessible to conventional ships may allow access by autonomous ships of the same size.

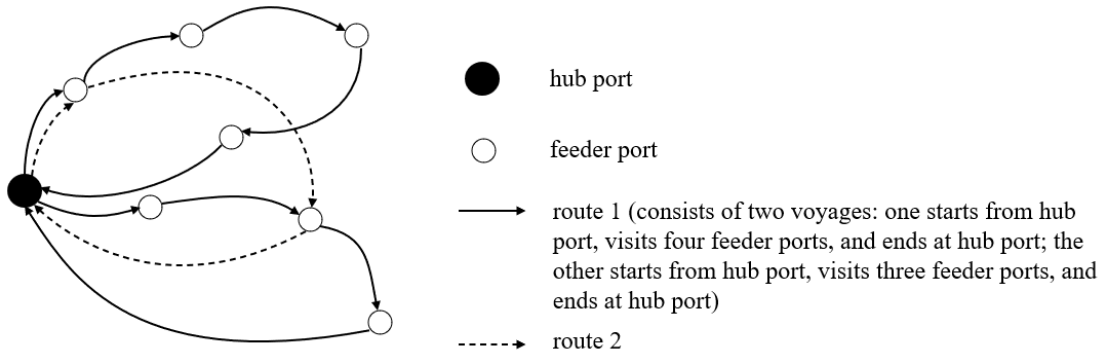


Figure 1. An example of feeder network

Within this problem setting, we calculate the total profits before and after introducing autonomous ships, which involves freight revenue, handling costs, ship operating costs (including labor costs, maintenance costs, insurance, etc.), bunkering costs, and capital costs[1]. Cargo handling profits are determined by the volume of cargoes carried on the network. Ship operating and bunkering costs are influenced by routing and vessel assignment decisions. Capital costs are closely related to the number and the types of ships deployed.

Therefore, we develop in this research a model to simultaneously optimize ship routing, fleet sizing, fleet deployment, and demand fulfillment. We consider heterogenous fleet that contains both autonomous and conventional ships. Besides, due to waterway restrictions, small vessels may be used for cargo delivery, which has two main implications. First, cargo transportation demand at a port may be greater than ship capacity, causing demand to be split and fulfilled by multiple voyages. Second, ships are allowed to sail multiple voyages during the fixed service frequency, which is the time interval between two consecutive visits to the same feeder port. For example, a ship spends three days traveling on the voyage Shanghai–Nanjing–Shanghai, but the service frequency is seven days, which means that there are four days until the next visit to Nanjing. To better utilize the shipping capacity, this ship could make another voyage of at most four days. In addition, shipping demand is not stable. Therefore, we include demand uncertainty in the model. This model extends classic vehicle routing problem (VRP) model by combining a heterogeneous fleet (i.e., conventional and autonomous ships of different sizes), multi-trips (i.e., a ship can sail multiple voyages), split deliveries (i.e., the transportation demand of a port can be satisfied by multiple voyages), and demand uncertainty. Hereafter we refer to it as HFMTSDU-VRP. Since this research involves both strategic decisions that are unlikely to be altered in a short term and operational decisions that are adjusted frequently in response to varying demand, the problem is formulated as a two-stage stochastic programming model. In the first stage, optimal routes, fleet composition, and fleet assignment are determined without realization of demand uncertainty. In the second stage, when demand realization is known, the liner company needs to determine the delivery pattern, i.e., the volume of cargo loaded onto each ship at each port of call. The objective is to maximize

[1] For conventional ships, we only consider ship purchase costs, while for autonomous ships, we need to also consider additional costs related to shore control center, because the operation of autonomous ships requires assistance of shore control center, which performs monitoring and control tasks for autonomous ships and can intervene in autonomous ship operation in unexpected events.

the expected service profit. To solve this challenging problem, we propose a sample average approximation (SAA) heuristic and an exact algorithm based on Benders decomposition (BD) and branch-and-cut (BC). Hereafter we refer to the exact algorithm as two-phase Benders-based branch-and-cut (TPBBC) algorithm. Besides, acceleration technologies, such as column generation and variable fixing, are proposed to strengthen the formulation and reduce computational times. Numerical experiments based on real-world data are conducted to evaluate the performance of different solution methods, explore the impact of introducing autonomous ships, discuss the influences of key factors, and derive valuable managerial insights for liner shipping companies.

The remainder of this study is organized as follows. Related works are reviewed and discussed in Section 2. Section 3 describes the problem and proposes a two-stage stochastic programming model. Section 4 introduces route generation procedure, SAA and TPBBC algorithm with acceleration strategy. Section 5 reports the computational performance of solution techniques and numerical experiments on real-life data. Conclusions are outlined in the last section.

2. Literature Review

The literature related to this study can be divided into three branches. The first branch is research on autonomous ships. The last decade has witnessed a growing interest in autonomous vessels. Liu et al. (2016), Schiaretti et al. (2017a, 2017b), Zolich et al. (2018), and Gu et al. (2021) provide comprehensive reviews of autonomous vessels. Besides, there are also many research projects, such as Maritime Unmanned Navigation through Intelligence in Networks (MUNIN[2]), ReVolt[3], and Yara Birkeland[4]. However, since autonomous vessels are in their early stage, these academic research and research projects mainly focus on economic feasibility (Kretschmann et al., 2017; Ghaderi, 2019; Ziajka-Poznańska and Montewka, 2021), safety issues (Wang et al., 2018; Fan et al., 2020; Goerlandt, 2020; Chang, 2021; de Vos, 2021), laws and regulations (Authority, 2017; Cheng and Ouyang, 2021; Zhu and Xing, 2021), vessel design (Jin, Zhang, and Liu, 2018; Makhsoos et al., 2018), etc. There is a dearth of literature that specifically focuses on the operational optimization of autonomous ships, especially when

[2] MUNIN is a collaborative research project, co-funded by the European Commissions under its Seventh Framework Programme. The goal is to verify the concept for autonomous ships and develop technology for unmanned and autonomous vessels.

[3] ReVolt is a concept ship built and tested by classification society DNV GL. This ship is autonomous and fully battery powered. The ship is assumed to be powered by a 3000 kWh battery and sails at an average speed of 6 knots.

[4] Yara Birkeland is the world's first fully electric and autonomous container vessel built by Yara and Kongsberg. It was put into commercial operation in Porsgrunn in the spring of 2022.

considering the intricate interplay of multiple decisions at both the strategic and the operational levels, such as ship routing, fleet sizing, fleet deployment, and demand fulfillment.

The second branch is related to vehicle routing problem (VRP), which refers to the problems involving routing, fleet sizing, fleet deployment, and demand fulfillment. This decades-old classic research question dates back to Dantzig and Ramser (1959), which found the shortest route that passes through all end points. The most studied variant is the classical capacitated VRP (CVRP) where all vehicles are identical and have the same capacity. The first important variant is called heterogeneous fleet VRP (HFVRP) where multiple types of vehicles are used to fulfill distribution. Branchini, Armentano, and Morabito (2015) solve the integrated problem of ship routing, scheduling, and fleet deployment to serve all contractual voyages and at the same time serve profitable spot voyages if there is room to spare. They consider different ship types that are restricted by the cargoes that can be loaded, the ports that can be visited, and also client contracts. Fadda et al. (2023) design an optimal maritime network considering draft limits, where ship draft is determined by ship type and cargo weight. A second variant is the multi-trip VRP (MTVRP) that extends the classical VRP by allowing vehicles to execute more than one trip during a predetermined service time (Taillard, Laporte, and Gendreau, 1996). In the MTVRP, the limited carrying capacity reduces the number of customers served on a trip and thus some vehicles need to perform several trips during a workday. When solving the last-mile delivery problem, Şahin and Yaman (2022) exploit service route design for a fleet under time window constraints, considering the compatibility of different vehicle types to different routes in a heterogeneous fleet and multi-trip service for each vehicle. A third variant relaxes constraints that each customer must be visited exactly once. This variant is called split delivery VRP (SDVRP). Archetti, Savelsbergh, and Speranza (2008) prove that demand split is most beneficial when demand mean is slightly above half the vehicle capacity and the variance is relatively small. Yoshizaki (2009) solves a distribution problem of a major Brazilian retail group, determining the best route, distribution schedule, vehicle allocation, and delivery amount for each customer. A fourth variant considers uncertainties in the problem, which leads to a version called stochastic VRP (SVRP). Gutierrez et al. (2018) determine schedule and routes for a group of technicians to execute repairing tasks within given time windows. Travel time and service time in the research are assumed to be stochastic and identically gamma and log-normal distributed, respectively.

A single variant usually cannot fully capture the essence of real-life transportation problems, so that multiple variants are often combined. Coelho et al. (2016) and Despaux and Basterrech (2016) investigate good delivery from a central depot to geographically scattered customers, simultaneously considering heterogeneous fleet and multi-trips. Wang, Kinable, and Van Woensel (2020) solve a fuel replenishment problem where tanker trucks carry different types of petrol to refuel petrol stations. Stations that have larger demand than the vehicle capacity may need to be refueled several times. Yang (2022) directs a fleet of identical capacitated vehicles to complete package deliveries. To fully utilize the vehicle mobility during working hours, vehicles are allowed to perform multiple trips.

The third branch is about solution methods for the VRP. VRP is strongly NP-hard, and various solution techniques have been proposed, including metaheuristics and exact algorithms. Here are the most commonly used metaheuristics: iterated local search (Coelho et al., 2016; Accorsi, and Vigo, 2021), memetic algorithms (Mendoza et al., 2010; Gutierrez et al., 2018), large neighborhood search (François et al., 2016; Wang, Kinable, and Van Woensel, 2020). Exact algorithms include column generation (Jin, Liu, and Eksioglu, 2008; Paradiso et al., 2020), branch-and-cut (Archetti, Bianchessi, and Speranza, 2014; Bianchessi, and Irnich, 2019), branch-and-price (Şahin, and Yaman, 2022; Torres, Gendreau, and Rei, 2022), and branch-and-price-and-cut (Desaulniers, 2010; Poggi, and Uchoa, 2014; Pecin et al., 2017; Gschwind, Bianchessi, and Irnich, 2019; Pessoa et al.,2020).

Although there is a substantial body of literature exploring single and multiple variants of the VRP, it is worth noting that existing studies typically consider the combination of only two to three variants at most. To the best of our knowledge, there is a research gap in the literature regarding articles that simultaneously investigate capacity, heterogeneous fleet, multi-trip, split delivery, and stochastic VRP. Moreover, the integration of multiple variants in the problem formulation leads to increased computational complexity, posing significant challenges for solving the problem. As a result, novel approaches and algorithmic adaptations are necessary to effectively address the intricacies introduced by these combined variants and achieve efficient solutions.

Given the existing literature, the main contribution of this study is fourfold. First, this study originally explores whether autonomous ships will replace conventional ones in national waterways by simultaneously optimizing ship routing, fleet sizing, fleet deployment, and demand fulfillment. Second, different from previous models on VRP,

the model proposed in this study simultaneously considers heterogeneous fleet, multi-trip, split delivery, and uncertainty, which has rarely been explored in the existing literature. Third, to solve the problem, we propose two types of solution algorithms. One is based on the commonly used SAA technique and the other is an innovative TPBBC algorithm. In the first phase of TPBBC, we relax integer constraints to add as many Benders cuts as possible, which accelerates convergence and computation speed. In the second phase, we restore integer constraints and take advantage of BC and generic callback to speed up the computation process. The TPBBC can be formulated in two ways, which are compared to find the best fit for the problem. Besides, several acceleration strategies are used, including column generation and variable fixing. Fourth, numerical experiments based on real-life data are conducted to yield managerial insights for shipping companies.

3. Problem Formulation

We consider a feeder network with a hub port p_0 and a set P of feeder ports, where $P = \{1, \dots, |P|\}$. The ships visiting these ports are heterogenous with a type set V. Due to the port restrictions, such as draft and berth constraints, a feeder port $p \in P$ can only be visited by vessels of certain types, denoted by V_p. While hub port p_0 is accessible to all ship types in V, because the hub port usually has deep water and abundant berths of different sizes. Ships sailing on a feeder network usually follow a fixed service frequency α (days), i.e., the time interval between two consecutive visits to the same feeder port. A closed-loop visiting sequence (hereinafter referred to as sequence), denoted by s, is a voyage from hub port to feeder ports and then back to hub port. It is port set made up of a hub port p_0 and a set $P_s \subseteq P$ of feeder ports and is ordered as a vector according to the visiting sequence, such as $(p_0, p_1, p_2, p_3, p_0)$, where $p_1, p_2, p_3 \in P_s$. Since only ships from certain types, namely V_p, are allowed to visit feeder port p, ship types that are allowed to travel on sequence s are denoted by $V_s = \cap_{p \in P_s} V_p$. The time taken to complete the sequence s, denoted by T_s, is the sum of sailing time and dwell time at ports. The sailing time is determined by the sailing distance and the speed. Since the speed of ships is assumed to be a constant, ship sailing time only depends on the travel distance. The dwell time, denoted by t_p, $\forall p \in \{p_0\} \cup P$, is the time a ship spends at port securing the vessel, discharging or loading cargo, and other activities. The set of feasible sequences, denoted by S, consists of those sequences that satisfy $V_s \neq \emptyset$ and $T_s \leq \alpha$. Since some sequences may have a short duration, a ship is allowed to travel several sequences as long as the total duration does not exceed a predetermined

service frequency. Therefore, we define route r as a set of sequences, which contains one or more than one sequence. All the sequences belonging to route $r \in R$ constitute a sequence set $S_r \subseteq S$. The set of ship types that are allowed to be deployed on route r is denoted by $V_r = \cap_{s \in S_r} V_s$. The duration of route r is represented by $T_r = \sum_{s \in S_r} T_s$. A route r is feasible only if $V_r \neq \emptyset$ and $T_r \leq \alpha$. All the feasible routes form a set R. An example in Figure 2 is used to illustrate port, sequence, and route. The solid and hollow circles represent hub port and feeder ports respectively. The capital letters are the names of the ports. The letters in parentheses indicate the ship types that are allowed to visit the corresponding port. Ships from three types, namely $\{a, b, c\}$, are available. The numbers on the arcs are the required travelling time (days). The dwell time at each port is assumed to be 0. All sequences of this example are shown in Table 1. *Ship type* represents the types of ships that are allowed to travel on this sequence. *Duration* is the amount of time to complete this sequence. *Feasible* indicates whether this sequence is feasible or not considering ship type and duration constraints. Table 1 indicates that eight out of 15 sequences are infeasible because no available ship can get access to all ports in each of the eight sequences. Table 2 shows all the feasible routes. As we mentioned, a route consists of one or more than one sequence. Thus, in addition to all the feasible single sequences, the combination of feasible sequences can also be regarded as a feasible route if the sequences in this combination can be visited by the same ship and the duration of this combination does not exceed the preset duration requirement (seven days). *Ship type* in Table 2 represents the ship types that can get access to all ports on the sequences making up this route. *Duration* is the amount of time it takes for a ship to complete this route. If the route is made up of several sequences, the duration is the sum of the sequences' durations. In total, we have 11 feasible routes.

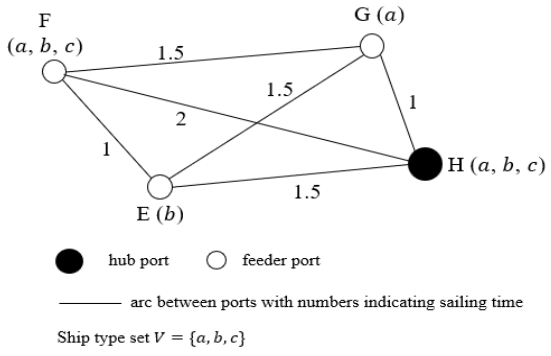


Figure 2. An example to illustrate port, sequence, and route.

257 Table 1. All sequences of the example feeder network

No.	Sequence	Ship type	Duration	Feasible when $\alpha = 7$
1	(H, E, H)	b	3	Yes
2	(H, E, G, H)	-	4	No
3	(H, E, G, F, H)	-	6.5	No
4	(H, E, F, H)	b	4.5	Yes
5	(H, E, F, G, H)	-	5	No
6	(H, G, H)	a	2	Yes
7	(H, G, E, H)	-	4	No
8	(H, G, E, F, H)	-	5.5	No
9	(H, G, F, H)	a	4.5	Yes
10	(H, G, F, E, H)	-	5	No
11	(H, F, H)	$a,\ b,\ c$	4	Yes
12	(H, F, G, H)	a	4.5	Yes
13	(H, F, G, E, H)	-	6.5	No
14	(H, F, E, H)	b	4.5	Yes
15	(H, F, E, G, H)	-	5.5	No

258

259 Table 2. All feasible routes of the example feeder network

No.	Route	Ship type	Duration
1	(H, E, H)	b	3
2	(H, E, F, H)	b	4.5
3	(H, G, H)	a	2
4	(H, G, F, H)	a	4.5
5	(H, F, H)	$a,\ b,\ c$	4
6	(H, F, G, H)	a	4.5
7	(H, F, E, H)	b	4.5
8	(H, E, H, F, H)	b	7
9	(H, G, H, G, F, H)	a	6.5
10	(H, G, H, F, H)	a	6
11	(H, G, H, F, G, H)	a	6.5

260 The feasible routes are regarded as input when formulating the two-stage stochastic
261 programming model. Given the ship routes R, the voyage cost C_r^v (USD/α days),
262 which includes bunkering cost, labor cost, maintenance cost, etc., of using ship type

$v \in V_r$ to serve route $r \in R$ during service frequency α can be obtained. Each ship of type v is associated with an average purchasing cost C_v (USD/α days). When the demand[5] of feeder port p is fulfilled, a handling cost of C_p^h (USD/TEU) will be incurred as well as the freight revenue F_p (USD/TEU). Since demand is uncertain, we use scenarios to represent varying demand, which may lead to different results for demand fulfillment under different scenarios. Hence, we optimize ship routing, fleet sizing, and fleet deployment in the first stage when demand is uncertain. In the second stage, with the realization of demand, the shipping company determines the flow of cargoes. The objective is to maximize expected total profits, i.e., expected revenue minus expected costs. The first stage objective function comprises voyage cost and purchase cost during the service frequency. The second stage objective function equals expected revenue minus handling costs associated with the cargo flow. This research makes three assumptions. First, a sequence can only be contained in one route. Second, a route can be assigned at most one ship and can be traveled at most once during service frequency. Third, the number of ships that can be deployed is unlimited. Notations used to formulate the two-stage stochastic programming model are listed in Table 3.

Table 3. Notations used to formulate the model

Sets	
P	Set of feeder ports to be visited, indexed by p, $p = 1, \dots, \lvert P \rvert$
S	Set of sequences, indexed by s, $s \in S$
R	Set of possible shipping routes, indexed by r, $r \in R$
V	Set of ship types, indexed by v, $v \in V$
R_s	Set of shipping routes that contain sequence s
S_p	Set of sequences that contain port p
P_s	Set of feeder ports that are visited by sequence s
V_r	Set of ship types that can be deployed on route r
Ω	Set of demand scenarios, indexed by ω, $\omega \in \Omega$
Parameters	
C_r^v	Voyage cost (USD) of using ship type v to serve route r during the

[5] We only consider export demand of each feeder port, i.e., the demand that needs to be loaded at a feeder port and unloaded or transshipped at a hub port. If other types of demand need to be considered, the model can be easily extended.

	predetermined time interval, including bunkering cost, labor cost, maintenance cost, etc.
C^v	Average purchasing cost (USD) of a ship from type v during service frequency
C_p^h	Handling cost (USD/TEU) of demand at feeder port p, including loading and unloading costs
F_p	Unit revenue of demand fulfillment of feeder port p (USD/TEU)
Q^v	Capacity (TEUs) of ship type v
T_s	Duration (hour) of a sequence s
D^p	Random demand (TEUs) of feeder port p
$D^p(\omega)$	Demand (TEUs) of feeder port p under scenario ω
α	Service frequency
Decision variables	
x_r^v	Binary variable, equal to 1 if a ship of type v is deployed on route r, and 0 otherwise
$y_s^p(\omega)$	The amount of demand of feeder port p that is served by sequence s under scenario ω

The model is given as follows:

$$\text{[RM] maximize } \mathbb{E}_\Omega[Q(x, D(\omega))] - \sum_{r\in R}\sum_{v\in V_r}(C_r^v + C^v)x_r^v \quad (1)$$

subject to

$$\sum_{r\in R_s}\sum_{v\in V_r} x_r^v \le 1, \ \forall s \in S \quad (2)$$

$$x_r^v \in \{0,1\}, \ \forall v \in V_r, \ r \in R \quad (3)$$

where the recourse function $Q(x, D(\omega))$ denotes the optimal value of the second stage problem under scenario ω:

$$Q(x, D(\omega)) = \text{maximize} \sum_{s\in S}\sum_{p\in P_s}(F_p - C_p^h)y_s^p(\omega) \quad (4)$$

subject to

$$\sum_{s\in S_p} y_s^p(\omega) \le D^p(\omega), \ \forall p \in P \quad (5)$$

$$\sum_{p\in P_s} y_s^p(\omega) \le \sum_{r\in R_s}\sum_{v\in V_r} Q^v x_r^v, \ \forall s \in S \quad (6)$$

$$y_s^p(\omega) \ge 0, \ \forall p \in P_s, \ s \in S. \quad (7)$$

The objective function (1) maximizes the expected total profit, which is the expected second-stage profit associated with cargo flow, minus the first-stage voyage cost and the average purchasing cost. Constraints (2) require that a maximum of one ship of the permitted type can be deployed on each route and each sequence can belong to at most

one route. Constraints (3) define the domains of first-stage decision variables. Equation (4) is the second-stage objective function for one demand scenario. Constraints (5) state that the fulfilled demand at each feeder port cannot exceed its total demand. Constraints (6) illustrate that the total demand served by a sequence cannot exceed the capacity of the ship deployed to travel along this sequence. Constraints (7) are the non-negativity conditions of second-stage decision variables.

4. Solution Algorithm

In this section, we first introduce label setting algorithm that is used to generate feasible routes. We then apply both SAA and TPBBC to solve the two-stage stochastic programming model. To speed up the computation process, we apply some acceleration techniques.

4.1 Generating Feasible Routes

In maritime transportation, the number of ports is limited, especially in a feeder network. Additionally, shipping routes often follow specific directions within a feeder network. For example, the ports along the Yangtze River are almost distributed along a line. Ships usually sail in one direction and then return in the opposite direction. A ship is unlikely to change direction back and forth halfway. It is therefore easy to enumerate all feasible routes as input.

To generate the route set, we use a label setting algorithm to first generate a feasible sequence set and then combine sequences of the same vessel type under the service frequency constraints. The detailed procedure for the label setting algorithm is provided in Appendix A.

4.2 Sample Average Approximation

SAA is often used to solve stochastic programming model by using empirical distribution obtained from samples to approximate the true distribution of the problem. The sample Ω' comprises $|\Omega'|$ scenarios of demand with the same probability of occurrence $\frac{1}{|\Omega'|}$. The SAA formation of the model [RM] is given as follows:

[SAA] maximize $\frac{1}{|\Omega'|}(\sum_{\omega\in\Omega'}\sum_{s\in S}\sum_{p\in P_s}(F_p - C_p^h)y_s^p(\omega)) - \sum_{r\in R}\sum_{v\in V_r}(C_r^v + C^v)x_r^v$ (8)

subject to (2) and (3)

$\sum_{s\in S_p} y_s^p(\omega) \le D^p(\omega),\ \forall p \in P,\ \omega \in \Omega'$ (9)

$$\sum_{p\in P_s} y_s^p(\omega) \le \sum_{r\in R_s}\sum_{v\in V_r} Q^v x_r^v,\ \forall s \in S,\ \omega \in \Omega' \quad (10)$$

$$y_s^p(\omega) \ge 0,\ \forall p \in P_s,\ s \in S,\ \omega \in \Omega'. \quad (11)$$

The sample size $|\Omega'|$ determines the solution quality of SAA. The larger is the sample size, the better is the solution quality, but the longer is the calculation time. To determine the appropriate sample size that could balance between solution quality and computation tractability, we propose Algorithm 1, which calculates confidence intervals (CIs) for lower bound, upper bound, and optimality gap under given sample size. CI is a range of estimates for an unknown parameter. The τ in Algorithm 1 is the level of significance and $1-\tau$ is the confidence level. $|M|$ is the number of samples for CI.

Algorithm 1 Estimate $(1-\tau)$-CI for lower bound, upper bound, and optimality gap under given sample size $|\Omega'|$

1. Generate a set of scenarios Ω'.
2. Solve the SAA with Ω' and obtain the optimal first-stage solution x^*.
3. **for** $m = 1, \dots, M$ **do**
4. Generate a set of new independent scenarios Ω'', $|\Omega''| = |\Omega'|$.
5. Solve SAA with Ω'' and obtain the objective value z_m.
6. Generate a set of new independent scenarios Ω''', $|\Omega'''| \gg |\Omega''|$.
7. Evaluate the quality of the first-stage solution x^* on scenarios in Ω'''. Input x^* into SAA, obtaining cost $z_{x^*}^m$.
8. Let $g_m := z_m - z_{x^*}^m$.
9. **end for**
10. Estimate $(1-\tau)$-CI for the lower bound
11. Let $L := \frac{1}{M}\sum_{m=1}^{M} z_{x^*}^m$ and $S_L := \frac{1}{M-1}\sum_{m=1}^{M}(z_{x^*}^m - L)^2$.
12. The $(1-\tau)$-CI for the lower bound is $\left[L - \frac{t_{M-1,\frac{\tau}{2}}\sqrt{S_L}}{\sqrt{M}}, L + \frac{t_{M-1,\frac{\tau}{2}}\sqrt{S_L}}{\sqrt{M}}\right]$, where $t_{M-1,\frac{\tau}{2}}$ is the t-value obtained from t-distribution with $M-1$ degrees of freedom and confidence level $1-\tau$.
13. Estimate $(1-\tau)$-CI for the upper bound
14. Let U: $= \frac{1}{M}\sum_{m=1}^{M} z_m$ and $S_U := \frac{1}{M-1}\sum_{m=1}^{M}(z_m - U)^2$.
15. The $(1-\tau)$-CI for the upper bound is $\left[U - \frac{t_{M-1,\frac{\tau}{2}}\sqrt{S_U}}{\sqrt{M}}, U + \frac{t_{M-1,\frac{\tau}{2}}\sqrt{S_U}}{\sqrt{M}}\right]$
16. Estimate $(1-\tau)$-CI for the optimality gap

17. Let $G := \frac{1}{M}\sum_{m=1}^{M} g_m$ and $S_G := \frac{1}{M-1}\sum_{m=1}^{M}(g_m - G)^2$.

18. The $(1-\tau)$-CI for the optimality gap is $\left[0, G + \frac{t_{M-1,\tau}\sqrt{S_G}}{\sqrt{M}}\right]$.

342

343

344 **4.3 Exact Algorithm**

345 Although two-stage stochastic programming problems are notorious for being
346 computationally intractable, Benders decomposition (BD) has been widely used to
347 solve the problems efficiently. The main idea of BD is to use delayed constraint
348 generation algorithm to reduce the number of variables and constraints in the problem.
349 It decomposes the stochastic programming model into a master problem and a number
350 of linear subproblems. These two types of problems are then solved iteratively, adding
351 additional constraints, referred to as Benders cuts, to the master problem. However,
352 traditional BD has an obvious limitation. If the master problem contains discrete
353 variables, it will be very time consuming to solve the master problem with an increasing
354 number of Benders cuts. In order to improve efficiency, we propose a two-phase
355 Benders-based branch-and-cut (TPBBC) algorithm, which will be elaborated in the
356 following sections.

357

358 **4.3.1 Benders Decomposition**

359 There are usually two ways to conduct BD for a two-stage model: the first is to regard
360 each scenario as an independent subproblem, as in Peng, Delage, and Li (2020). In this
361 case, many Benders cuts, one for each subproblem, are added at each iteration to
362 accelerate the convergence. The second is to combine all scenarios into a single
363 subproblem, as in Adulyasak, Cordeau, and Jans (2015), to avoid adding too many cuts
364 at each iteration and control the time taken to solve the master problem. The relative
365 performances of these two methods depend on problem structures and numerical
366 instances. In numerical experiments, we will first compare the performances of TPBBC
367 with those of the two BD formulations and then select the better one for the following
368 numerical analysis. The models for these two methods are shown as follows:

369 [MP1] maximize $\frac{1}{|\Omega|}\sum_{\omega\in\Omega}\phi(x,\omega) - \sum_{r\in R}\sum_{v\in V_r}(C_r^v + C^v)x_r^v$ (12)

370 subject to (2) and (3)

[SP1] $\phi(x,\omega)$ =maximize $\sum_{s\in S}\sum_{p\in P_s}(F_p - C_p^h)y_s^p(\omega)$ (13)

subject to (5)–(7)

[MP2] maximize $\phi(x) - \sum_{r\in R}\sum_{v\in V_r}(C_r^v + C^v)x_r^v$ (14)

subject to (2) and (3)

[SP2] $\phi(x)$ =maximize $\frac{1}{|\Omega|}\sum_{\omega\in\Omega}\sum_{s\in S}\sum_{p\in P_s}(F_p - C_p^h)y_s^p(\omega)$ (15)

subject to (9)–(11).

MP1 (resp. MP2) and SP1 (resp. SP2) refer to the master problem and subproblem when each scenario is an independent subproblem (resp. when all scenarios are in one subproblem). In the following analysis, for convenience, we refer to both MP1 and MP2 as master problem (MP), and to SP1 and SP2 as subproblem (SP).

There are two types of cuts in BD, that is feasibility cuts and optimality cuts. It is easy to verify that SP is feasible and bounded (see Appendix B), therefore, we only need to consider optimality cuts as shown in (16) and (17) for independent and aggregated cases, respectively:

$$\phi(x,\omega) \le \sum_{p\in P} D^p(\omega)\,\hat{\lambda}_p + \sum_{s\in S}\sum_{r\in R_s}\sum_{v\in V_r} Q^v x_r^v\,\hat{\mu}_s,\ \forall\omega\in\Omega,\ (\hat{\lambda}_p,\hat{\mu}_s)\in I \quad (16)$$

$$\phi(x) \le \sum_{\omega\in\Omega}\sum_{p\in P} D^p(\omega)\,\hat{\lambda}_{p\omega} + \sum_{\omega\in\Omega}\sum_{s\in S}\sum_{r\in R_s}\sum_{v\in V_r} Q^v x_r^v\,\hat{\mu}_{s\omega},\ \forall(\hat{\lambda}_{p\omega},\hat{\mu}_{s\omega})\in I \quad (17)$$

where λ_p and μ_s (resp. $\lambda_{p\omega}$ and $\mu_{s\omega}$) correspond to the dual variables associated with constraints (5) and (6) (resp. (9) and (10)) respectively. The dual problem of SP has an extreme point set I. For each extreme point $(\hat{\lambda}_p,\hat{\mu}_s)\in I$ (resp. $(\hat{\lambda}_{p\omega},\hat{\mu}_{s\omega})\in I$), Benders cut (16) (resp. (17)) holds.

There will be an exponential number of Benders cut, which will make the problem intractable. BD effectively handles this problem by employing a delayed constraint generation algorithm, which means that BD relaxes the MP with only a subset of these cuts, resulting in a relaxed master problem (RMP).

4.3.2 Two-phase Benders-based Branch-and-cut Algorithm

Unfortunately, BD converges slowly because RMP is an integer problem whose size keeps growing as the iteration count progresses. For this reason, we integrate BD inside branch-and-cut (BC) to make full use of the advantages of BC for solving mixed integer programming (MIP) problems. The basic idea of BC is to divide the entire solution space into multiple subsets that are not constrained by integers, remove infeasible nodes through boundary constraints, and optimize the solution space by adding cuts.

The innovative method of integrating BD inside BC is called two-phase Benders-based branch-and-cut algorithm. At the first phase, we relax integer constraints on RMP and run BD multiple times until no Benders cuts can be added, which speeds up the convergence and saves computation time for the second phase. We then restore integer constraints and keep all generated cuts at first phase. At the second phase, we use BC at each node to find integer solutions. Once an integer solution is found, we use generic callback to solve SP, add Benders cut, and update boundaries. The detailed procedures of TPBBC algorithm are shown in Algorithm 2. First, we define RMP without integer constraints (3) as LRMP. Additional notations for Algorithm 2 are shown in Table 4. The set of active nodes in the branch-and-bound (BB) tree is denoted by N. In the initialization process, it contains only the root node. The value of the best-known feasible solution, called incumbent solution, for the original problem is stored as $\underline{r}$ and provides a lower bound. The upper bound of the objective value is denoted as $\overline{r}$, which is initialized to positive infinity at root node. Each node of the BB tree has an upper bound $ub_n,\ n \in N$, initialized to the value of the parent node, and will be updated if the value of LRMP is lower than initialization value. The maximum upper bound of all active nodes is denoted by $UB_N = \max\{ub_n : n \in N\}$.

Table 4. Additional notations for Algorithm 2

N	Set of active nodes in branch-and-bound (BB) tree, indexed by $n,\ n \in N$
$\underline{r}$	Incumbent solution for the original problem, which provides a lower bound
$\overline{r}$	Upper bound of the original problem
ub_n	Upper bound of node n
UB_N	The maximum upper bound of all active nodes, where $UB_N = \max\{ub_n : n \in N\}$

Algorithm 2: Two-phase Benders-based branch-and-cut algorithm

1. **Input:** A tolerance $\epsilon \geq 0$, maximum first-phase run time T_1, maximum second-phase run time T_2, and sample Ω
2. **Output:** The optimal solutions $\boldsymbol{x}$ and objective value $\underline{r}$
3. **Initialize** $\overline{r} \leftarrow \infty$, $\underline{r} \leftarrow -\infty$, active node set $N \leftarrow${root node}, and $UB_N \leftarrow \infty$
4. First phase:
5. **while** $\overline{r} - \underline{r} > \epsilon$, and first-stage runtime$< T_1$ **do**
6. Solve LRMP: Solve LRMP to obtain optimal solution $\hat{\boldsymbol{x}}$ and optimal objective value is z_n
7. Input $\hat{\boldsymbol{x}}$ to SP to check whether Benders cut is needed
8. **if** needed **then**

9. Add Benders cut (16) or (17) to LRMP
10. **else**
11. **end while**
12. **end if**
13. Second phase: Restore integer constraints and imbed BD into BC
14. **while** $N \neq \emptyset$, $\overline{r} - \underline{r} > \epsilon$, and second-phase run time $< T_2$ **do**
15. Node selection: Select a node n from N
16. **if** $ub_n < \underline{r}$ **then**
17. Prune node n (prune by bound)
18. Continue with the next iteration (back to line 15)
19. **end if**
20. Solve LRMP: Solve LRMP to obtain optimal solution $\hat{\boldsymbol{x}}$ and optimal objective value is z_n
21. **if** $z_n < \underline{r}$ **then**
22. Prune node n (prune by bound)
23. Continue with next iteration (back to line 15)
24. **else if** $z_n < ub_n$ **then**
25. $ub_n \leftarrow z_n$
26. Update UB_N
27. Update $\overline{r} \leftarrow UB_N$
28. **end if**
29. **if** $\hat{\boldsymbol{x}}$ satisfies integer constraints **then**
30. Input $\hat{\boldsymbol{x}}$ to SP and obtain optimal solution $\hat{\boldsymbol{y}}$
31. Calculate $lb_n \leftarrow \frac{1}{|\Omega|}\sum_{\omega\in\Omega}\sum_{s\in S}\sum_{p\in P_s}(F_p - C_p^h)\hat{y}_s^p(\omega) - \sum_{s\in S}\sum_{v\in V_s} C_s^v\,\hat{x}_s^v$
32. $\underline{r} \leftarrow max(\underline{r}, lb_n)$
33. **if** add Bender cut **then**
34. Add Benders cut (16) or (17) to LRMP
35. Back to solve LRMP (line 20)
36. **else**
37. Prune node n (prune by integer)
38. Continue with next iteration (back to line 15)
39. **end if**
40. **else**
41. Branch: add two new nodes into N
42. Remove node n from N
43. **end if**
44. **end while**
45. **Return** $\underline{r}$ and corresponding optimal solution $\boldsymbol{x}$

Since we have already described the procedure of the first phase above, we only introduce the second phase. For the second phase, a node is first selected. It is verified whether the upper bound of the node is less than the lower bound of the original problem. If yes, the node is pruned. Otherwise, solve LRMP to obtain optimal first-stage solutions and objective value. Conduct a bound check again. If the objective value is lower than the inherited upper bound, update the upper bound of the node and at the same time update the upper bound of the problem if the condition is met. Then check whether integer constraints are satisfied. If the solution is integer, solve the subproblems and update the lower bound. Then check whether the optimality constraints are violated. Add Benders cuts once a violation is found. Otherwise, prune the node. If the solution is fractional, branch the node into two new nodes. Repeat the iteration until active node set is empty, the gap between upper and lower bounds is below a predetermined tolerance level, or the time exceeds a preset value.

4.3.3 Acceleration Techniques

With the increase of the number of variables and constraints, the problem will become computationally difficult, and the computer could run out of memory. Therefore, in this section, we propose several acceleration techniques that can greatly reduce the number of variables and constraints, thus speeding up the computation process.

We know that only a small proportion of x_r^v will equal 1 in the optimal solution. Therefore, we apply variable fixing (VF) to assign zero to part of the variables, which can reduce the search space and simplify the computation. To set the value of x_r^v, we make use of Proposition 1.

Proposition 1. *Let* $\boldsymbol{P}$ *be an MIP defined as* $z(\boldsymbol{P}) = \max\{\boldsymbol{cx} + \boldsymbol{dy} | \boldsymbol{Ax} + \boldsymbol{By} \leq \boldsymbol{h}, \boldsymbol{x} \in \{0,1\}^{n_1}, \boldsymbol{y} \in \mathbb{R}_+^{n_2}\}$. *Let* $\boldsymbol{x}'$ *and* $\boldsymbol{y}'$ *be a feasible solution, and* $\boldsymbol{\omega}$ *be a feasible dual solution of the linear relaxation of* $\boldsymbol{P}$. *Any optimal solution* $\boldsymbol{x}^*$ *cannot contain a variable* $x_i^* = 1$ *if the reduced cost* c_i *is less than* $\boldsymbol{cx}' + \boldsymbol{dy}' - \boldsymbol{\omega h}$.

The proof of Proposition 1 is in Appendix C. According to Proposition 1, we need to identify a feasible solution $\boldsymbol{x}'$ and $\boldsymbol{y}'$, and a feasible dual solution $\boldsymbol{\omega}$. Since finding a good feasible solution of MIP is not easy in a large-size problem, we can exploit the property that column generation (CG) only uses a subset of variables to solve the problem to simply computation process. The CG algorithm will decompose the problem

into a master problem and a subproblem. The master problem is MP1 (resp. MP2) with a relaxation of integer constraints (3) and only a subset of variables. The subproblem is PP1 (resp. PP2) that is used to identify the variables that could improve objective function. At each iteration, we solve master problem to optimum under the given subset of variables, obtaining optimal solution $\boldsymbol{x}^*$. Then input $\boldsymbol{x}^*$ into SP1 (resp. SP2) to check whether to add Benders cuts. If no Benders cut can be added, run PP1 (resp. PP2) to check whether to add new variables to MP1 (resp. MP2). If new variables could be added, repeat the above process, otherwise, a good subset of variables $\boldsymbol{x}_{sub}$ for feasible solution of original MIP and the optimal dual objective $z(\boldsymbol{\omega})$ of linear relaxation of MIP are obtained, and the CG computes.

[PP1] $\sum_{s\in S_r}\eta_s^* - \sum_{k\in K}\sum_{s\in S_r} Q^v\hat{\mu}_s^k\xi_k^*,\ \forall v\in V_r,\ r\in R\backslash R'$ (18)

[PP2] $\sum_{s\in S_r}\eta_s^* - \sum_{k\in K}\sum_{\omega\in\Omega}\sum_{s\in S_r} Q^v\hat{\mu}_{s\omega}^k\xi_k^*,\ \forall v\in V_r,\ r\in R\backslash R'$ (19)

where η_s^* is the dual of Constraints (2), ξ_k^* is the dual of k^{th} Benders cut added to master problem, $\hat{\mu}_s^k$ (resp. $\hat{\mu}_{s\omega}^k$) is the dual of Constraints (6) (resp. Constraints (10)) in the k^{th} Benders cut.

Having the subset $\boldsymbol{x}_{sub}$, we restore integer constraints and use procedures in the second phase of TPBBC to generate a feasible objective $z(MIP)$ for original MIP. We then use variable fixing for all $\boldsymbol{x}$. If the reduced cost $c_r^v,\ \forall v\in V_r,\ r\in R$ satisfies $c_r^v < z(MIP) - z(\boldsymbol{\omega})$, we have $c_r^v = 0$. Setting part of the variables to 0 and lower bound of MIP to $z(MIP)$, we run the second phase of TPBBC to obtain optimum solution of MIP.

In addition to Proposition 1, we also propose two valid constraints that could strengthen the formulation of the problem and reduce the search space. The first valid constraint (20) is based on comparison of cost and profit of using the vehicle type v to serve route r:

$x_r^v = 0,\ \forall v\in V_r,\ r\in R$ if $C_r^v + C^v \geq UQ^v$ (20)

where $U = \max\{F_p - C_p^h | \forall p\in P\}$. This means that if the cost of using ship type v on route r is no less than the maximum profits can be earned on this trip, the ship type v cannot be deployed on route r.

The second valid constraint is for split delivery. Dror and Trudeau (1990) propose a theorem for split delivery stating that if the costs satisfy triangular inequality, there exists an optimal solution in which no two routes have more than one split customer in common. Inspired by this theorem, Archetti, Bianchessi, and Speranza (2014) propose

Corollary 1 that is more suitable for solving our problem.

Corollary 1. *If the costs satisfy triangle inequality, then there exists an optimal solution such that the total number of splits is lower than the number of routes*.

According to Corollary 1, we can obtain the following valid constraint:

$$x_r^v = 0,\ \forall v \in V_r,\ r \in R \ \text{ if } \ \sum_{p \in P_r} n_{pr} \geq N_r \tag{21}$$

where n_{pr} is the number of splits port p experiences on route r. For example, if two sequences of route r visit port p, then $n_{pr} = 2 - 1$. N_r is the number of sequences route r contains.

5. Numerical Experiment

In this section, we use real-world data to test the performances of the solution methods. We first use label setting to generate feasible routes. We then compare the performances of solution algorithms mentioned above. We also conduct sensitivity analysis to test the impact of some key parameters, which provides valuable managerial insights for practical implement. All the experiments are carried out on a laptop with i9-12900K CPU, 3.20 GHz processing speed and 32 GB of memory. The model was implemented in C++ programming and solved by CPLEX 12.10.

5.1 Parameter Setting

The numerical experiment uses real world data along the Yangtze River, as shown in Figure 3. A total of 13 ports are considered where Shanghai port is a hub port while the remaining ports are feeder ports. The distance between ports is given in Table 5. The number in each cell represents the distance from the port in the first row to the leftmost port in the first column. For example, it is 53 nautical miles from SH to NT. Six ship types are considered in the numerical experiment, as shown in Table 6. Capital letters S, M, and L represent small-, medium-, and large-size ships, respectively. We assume that conventional and autonomous ships of the same size have different capacities because the removal of deckhouse and hotel system in autonomous vessels leaves more room for cargoes. The same is true for medium- and large-size ships. The ship operational cost includes labor cost, maintenance cost, insurance, etc. The capital cost includes the purchase cost and the cost related to shore control center for autonomous ships (Notteboom and Cariou, 2013; Kretschmann, Burmeister, and Jahn, 2017). Demands of feeder ports are based on data provided by Ministry of Transport of the People's Republic of China. Demand dataset contains 216 scenarios where each

scenario contains weekly demand of all feeder ports. The revenue and handling costs for serving a container from a feeder port are the real-time prices obtained from the logistics website. The port dwelling time, including loading, unloading, waiting, etc., is calculated based on Tan et al. (2021). Parameters related to feeder ports are shown in Table 7. The symbol "-" in the second and third columns means that the revenue and handling cost of SH are not considered because we do not consider the demand of hub port. Meanwhile, "-" in port dwell time columns means the corresponding ship type is not allowed to berth at the feeder port because of the waterway restrictions, such as draft and width constraints (Li et al., 2019). The service frequency α is set to seven days and the sailing speed is 15 knots.

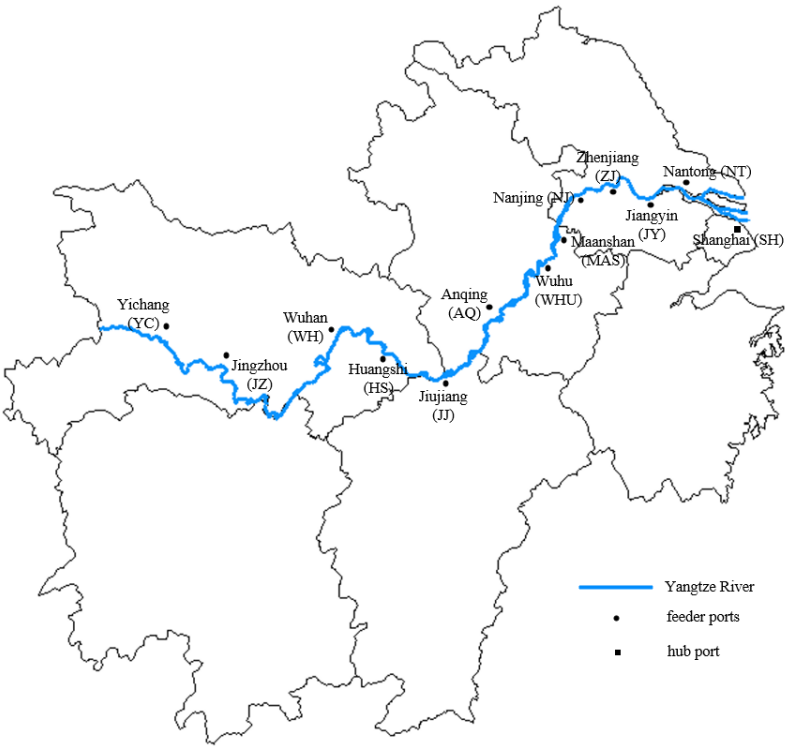


Figure 3. Ports along the Yangtze River

Table 5. Distance between ports (nautical mile)

	SH	NT	JY	ZJ	NJ	MAS	WHU	AQ	JJ	HS	WH	JZ
NT	53											
JY	85	31										
ZJ	147	94	63									
NJ	187	134	103	40								
MAS	214	160	129	66	26							

WHU	239	186	154	92	52	25						
AQ	345	292	260	198	158	131	106					
JJ	428	375	343	281	241	214	189	83				
HS	492	439	408	345	305	279	253	147	64			
WH	563	510	478	416	376	349	324	218	135	71		
JZ	787	733	702	639	599	573	548	442	359	294	224	
YC	901	848	816	754	714	687	662	556	473	409	338	114

Note: The distance is obtained from the website http://ports.com/sea-route/.

Table 6. Ship type information

	Conventional vessels			Autonomous vessels		
	C-S	C-M	C-L	A-S	A-M	A-L
Capacity (TEUs)	300	400	500	420	560	700
Unit bunkering cost (USD/nautical mile)	25	29	33	23	27	31
Ship operational cost (USD/hour)	119	144	160	71	86	96
Capital cost (million USD/year)	1.36	1.64	1.82	1.49	1.80	2.01

Note: The data have been adjusted based on the data used in the studies by Notteboom and Cariou (2013) and Kretschmann, Burmeister, and Jahn (2017).

Table 7. Parameters related to feeder ports

Ports	Revenue (USD/TEU)	Handling cost (USD/TEU)	Port dwell time (hours)					
			C-S	C-M	C-L	A-S	A-M	A-L
SH	-	-	8.9	9.4	9.9	8.9	9.4	9.9
NT	154	39	5.2	5.7	6.2	5.2	5.7	6.2
JY	168	28	4.0	4.5	5.0	4.0	4.5	5.0
ZJ	168	37	4.7	5.2	5.7	4.7	5.2	5.7
NJ	154	33	4.2	4.7	5.2	4.2	4.7	5.2
MAS	168	56	6.8	7.3	-	6.8	7.3	-
WHU	168	88	6.7	7.2	-	6.7	7.2	-
AQ	280	74	4.2	4.7	-	4.2	4.7	-
JJ	168	54	6.5	7.0	-	6.5	7.0	-
HS	210	60	6.7	7.2	-	6.7	7.2	-
WH	168	82	5.6	6.1	-	5.6	6.1	-
JZ	168	34	4.1	-	-	4.1	-	-

YC	168	30	3.6	-	-	3.6	-	-

Note: The data have been adjusted based on the data used in the studies by Li et al. (2019) and Tan et al. (2021).

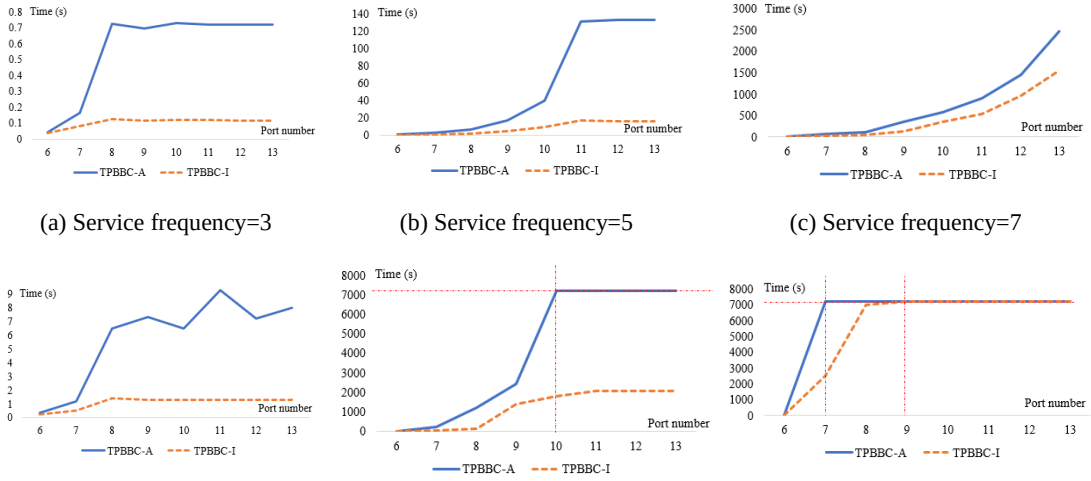
5.2 Computational Performances

In this section, we use parameters provided in Section 5.1 to test the performances of the solution techniques mentioned in Section 4. We generate test instances of varying sizes by adjusting two parameters: the number of ports and the service frequency. The number of ports ranges from six to 13, while the service frequency is set to three, five, or seven days. To avoid computation burden of large sample size, in this test, we set the sample size to be 10. The influences of sample size will be explored in the following section.

A total of five solution methods, i.e., SAA, TPBBC with each scenario being an independent subproblem (TPBBC-I), TPBBC with all scenarios in one subproblem (TPBBC-A), branch-and-Benders-cut (BBC), and TPBBC-I with CG and VF (TPBBCCGVF-I), were tested. BBC is a commonly used branch-and-cut method that only has the second phase of TPBBC. By comparing the performances of TPBBC and BBC, we can know whether it is necessary to conduct first phase of TPBBC before the classic branch-and-cut method. For TPBBCCGVF-I, we first apply valid constraints before solving the problem to see if some variables can be eliminated. Then in the first phase of TPBBC, we iteratively use CG and add Benders cut to obtain a subset of routes for a feasible solution and an optimal dual solution to the original MIP without integer constraints. Having the subset, we restore the integer constraints, and obtain a feasible solution to MIP. With a feasible solution and a dual solution, variable fixing is applied to assign a value of zero to part of the variables. With reduced variables and constraints, the second phase of TPBBC is conducted.

We first compare performances of TPBBC-I and TPBBC-A. Results are shown in Figure 4, where (a), (b) and (c) show the time spent for the first-phase when service frequency is three, five, or seven days, respectively; (d), (e), and (f) show the total time when the service frequency is three, five, or seven days, respectively. The horizontal bar indicates the number of ports considered, while the vertical bar is the solution time. The dashed red horizontal line is the time limit (i.e., 7200s). When the point reaches the line, the computation time of this algorithm exceeds the limit and we thus do not have the exact time for them. The dashed red vertical line represents the turning point when computation time of the algorithm exceeds the time limit. For example, in Fig. 4(e),

when the number of ports is 10, the total time of TPBBC-A exceeds 7200s. We can see that TPBBC-I always outperforms TPBBC-A for two main reasons. First, the former adds more Benders cuts and thus makes the computation converge more quickly. Second, we can only obtain at most one Benders cut after solving the subproblem under all scenarios of a sample for TPBBC-A, while for each scenario, we have about one Benders cut for TPBBC-I, which means the time spent for obtaining the Benders cut is much shorter for TPBBC-I. The results mean that TPBBC-I is more suitable for our problem and hence we only consider the case for which each scenario corresponds to an independent subproblem in the following analysis.

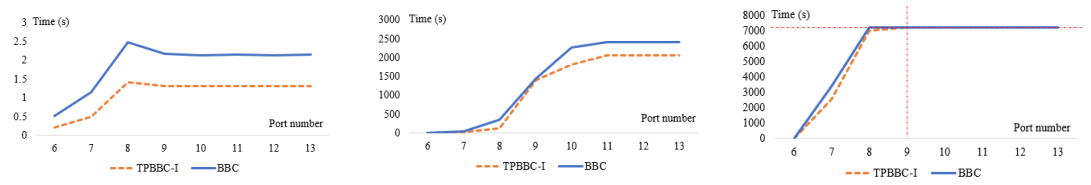


(a) Service frequency=3 (b) Service frequency=5 (c) Service frequency=7

(d) Service frequency=3 (e) Service frequency=5 (f) Service frequency=7

Figure 4. Comparison of computational times (second) between TPBBC-I and TPBBC-A under different service frequency and the number of ports. (a), (b) and (c) show the first-phase time when service frequency is three, five, or seven days, respectively. (d), (e), and (f) show total time when service frequency is three, five, or seven days, respectively.

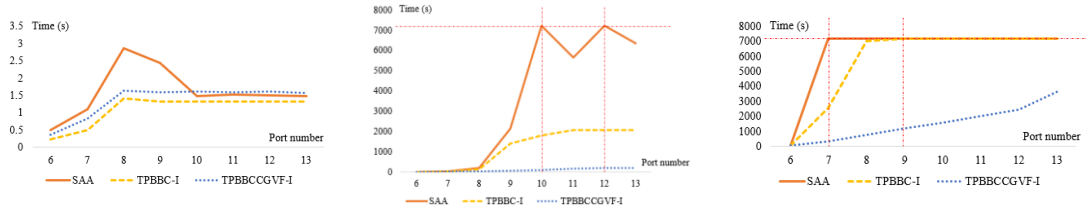
We then compare TPBBC-I with BBC to determine whether it is more efficient to add some Benders cuts at root node before branch-and-cut process. The results shown in Figure 5 indicate that modifying BBC to TPBBC-I speeds up convergence and saves computation time.



(a) Service frequency=3 (b) Service frequency=5 (c) Service frequency=7

Figure 5. Comparison of computational times between TPBBC-I and BBC under different service frequency and the number of ports

Knowing that TPBBC-I outperforms TPBBC-A and BBC in all test instances, we then compare SAA, TPBBC-I, and TPBBCCGVF-I. From Figure 6, we can see that TPBBC-I spends less time in small size instances and the gap between these methods is small, while for medium and large instances, there are significant differences between the solution times of these methods. When the service frequency is five days, SAA takes over an hour and a half in 50% of the instances and fails to solve the problem optimally within the time limit for two instances. Meanwhile, TPBBC-I finds an optimal solution for all instances within one hour, and TPBBCCGVF-I does this within three minutes. When the service frequency is seven days, SAA can only obtain an optimal solution for one instance. TPBBC-I can solve the problem to optimality only when the number of ports is less than nine. TPBBCCGVF-I finds an optimal solution for all instances in about one hour. The reason for the high efficiency of TPBBCCGVF-I can be seen from Table 8. SF and NP are the service frequency and the number of ports, respectively. Variables and Constraints indicate the number of variables and constraints in the model. For SAA, these are the number of columns and rows after MIP presolve. For TPBBC-I and TPBBCCGVF-I, these are the corresponding values for the second phase after restoration of the integer constraints. Variable fixing shows the number of variables that were set to 0 during variable fixing procedure in TPBBCCGVF-I. Valid 1 and 2 are the number of variables set to 0 by valid constraints (20) and (21), respectively. Since only TPBBCCGVF-I uses variable fixing and valid constraints, the values for SAA and TPBBC-I are not available in the corresponding columns, so we use "-" instead. At the beginning of TPBBCCGVF-I, some variables were eliminated by valid constraints, which shortens the time for first phase. Then during variable fixing, other variables were set to 0, greatly accelerating the computation at the second phase. This is particularly significant in large instances where a large number of variables were eliminated in the first phase.



(a) Service frequency=3 (b) Service frequency=5 (c) Service frequency=7

Figure 6. Comparison of computational times between SAA, TPBBC-I, and TPBBCCGVF-I under different service frequencies and number of ports

Table 8. Comparison of SAA, TPBBC-I, and TPBBCCGVF-I in terms of variables and constraints

SF	NP	Algorithm	Variables	Constraints	Variable fixing	Valid 1	Valid 2
3	6	SAA	958	390	-	-	-
		TPBBC-I	218	89	-	-	-
		TPBBCCGVF-I	217	89	0	0	1
3	7	SAA	2108	741	-	-	-
		TPBBC-I	341	171	-	-	-
		TPBBCCGVF-I	340	171	0	0	1
3	8	SAA	2691	1000	-	-	-
		TPBBC-I	401	208	-	-	-
		TPBBCCGVF-I	400	284	0	0	1
3	9	SAA	2691	1000	-	-	-
		TPBBC-I	401	208	-	-	-
		TPBBCCGVF-I	400	284	0	0	1
3	10	SAA	2691	1000	-	-	-
		TPBBC-I	401	208	-	-	-
		TPBBCCGVF-I	400	284	0	0	1
3	11	SAA	2691	1000	-	-	-
		TPBBC-I	401	208	-	-	-
		TPBBCCGVF-I	400	284	0	0	1
3	12	SAA	2691	1000	-	-	-
		TPBBC-I	401	208	-	-	-
		TPBBCCGVF-I	400	284	0	0	1
3	13	SAA	2691	1000	-	-	-
		TPBBC-I	401	208	-	-	-
		TPBBCCGVF-I	400	284	0	0	1
5	6	SAA	3404	391	-	-	-
		TPBBC-I	2719	89	-	-	-
		TPBBCCGVF-I	688	89	1376	16	639
5	7	SAA	7950	753	-	-	-
		TPBBC-I	6197	190	-	-	-
		TPBBCCGVF-I	1203	197	3562	20	1411

5	8	SAA	12644	1467	-	-	-
		TPBBC-I	8497	271	-	-	-
		TPBBCCGVF-I	2542	321	4348	0	1560
5	9	SAA	19524	2885	-	-	-
		TPBBC-I	9794	410	-	-	-
		TPBBCCGVF-I	3331	399	4749	0	1572
5	10	SAA	33176	5692	-	-	-
		TPBBC-I	11049	699	-	-	-
		TPBBCCGVF-I	3749	466	5351	0	1573
5	11	SAA	52613	9819	-	-	-
		TPBBC-I	12323	1009	-	-	-
		TPBBCCGVF-I	3749	495	6318	0	1573
5	12	SAA	52613	9819	-	-	-
		TPBBC-I	12327	1020	-	-	-
		TPBBCCGVF-I	3749	494	6321	0	1573
5	13	SAA	52613	9819	-	-	-
		TPBBC-I	12327	1020	-	-	-
		TPBBCCGVF-I	3749	494	6321	0	1573
7	6	SAA	25769	391	-	-	-
		TPBBC-I	25160	93	-	-	-
		TPBBCCGVF-I	3031	92	2506	15150	4473
7	7	SAA	92222	753	-	-	-
		TPBBC-I	90599	179	-	-	-
		TPBBCCGVF-I	5624	202	9724	63465	11786
7	8	SAA	167204	1467	-	-	-
		TPBBC-I	163683	283	-	-	-
		TPBBCCGVF-I	27301	371	64707	678	70997
7	9	SAA	231421	2885	-	-	-
		TPBBC-I	222781	425	-	-	-
		TPBBCCGVF-I	37474	546	95321	2394	87575
7	10	SAA	294915	5711	-	-	-
		TPBBC-I	274567	706	-	-	-
		TPBBCCGVF-I	41017	668	128231	6222	98848
7	11	SAA	364565	11353	-	-	-
		TPBBC-I	316569	1229	-	-	-
		TPBBCCGVF-I	49482	834	152217	10028	104184
7	12	SAA	431234	22555	-	-	-
		TPBBC-I	323515	2240	-	-	-
		TPBBCCGVF-I	49482	822	156579	11599	104195
7	13	SAA	521678	38763	-	-	-
		TPBBC-I	328508	3420	-	-	-
		TPBBCCGVF-I	84115	1198	124477	13265	104195

Based on the above analysis, the following conclusions can be drawn within the

context of this study: 1) When implementing Benders decomposition, treating each scenario as an independent subproblem proves to be a more efficient approach; 2) The two-phase branch-and-Benders-cut method demonstrates superior performance compared with the classic one; 3) The acceleration techniques significantly enhance computational speed. In summary, the proposed solution algorithm can efficiently solve the problem, especially for large-size instances.

5.3 Determination of the Sample Size

As we mentioned in Section 4.2, the larger the sample size, the better the solution quality but the longer the computation time. To identify a sample size that can achieve a balance between solution quality and computation complexity, Algorithm 1 was used in this section under the given parameter setting. From the previous experiments we know that the instance size that SAA can solve to optimality is limited. Considering the computation time, we selected 10 instances shown in Figure 7 and 8. Each instance was tested under five different sample sizes, i.e., 10, 30, 50, 100, and 150. Figure 7 and Figure 8 show the ratio between the 95% confidence interval for the optimality gap and a point estimate of lower bound and computation time for 10 test instances under different sample sizes, respectively. The dashed line in Figure 7 is the 1% limit. It can be seen that when sample size reaches 100, the ratio falls below 1% in all cases. Although the computation time is slightly longer than for smaller sample sizes, it is acceptable given the size of the instance. Therefore, considering both solution quality and computation time, we set the sample size to 100 for the following analysis.

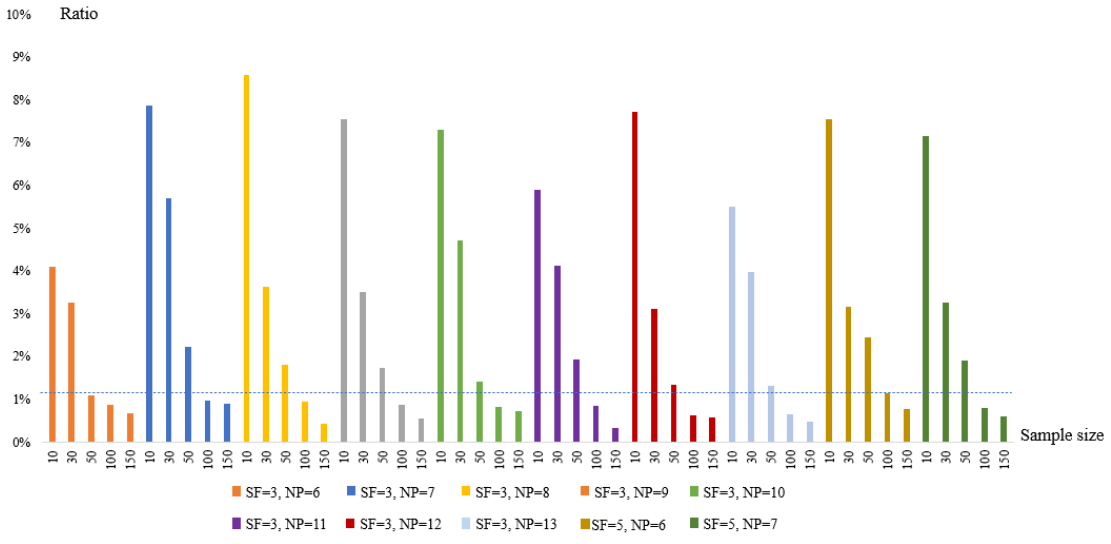


Figure 7. Ratio between the 95% confidence interval for optimality gap and the point estimate of lower bound under different sample size and problem setting

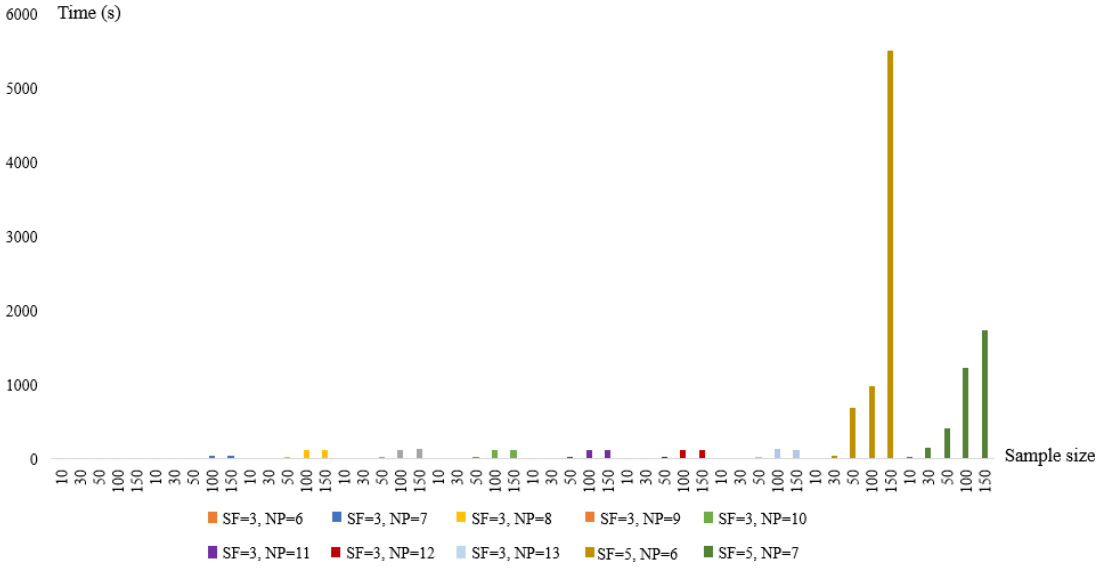


Figure 8. Computation time under different sample size and problem setting

5.4 Sensitivity Analysis

In this section, we investigate the influence of key factors and derive constructive managerial insights for shipping companies.

5.4.1 The Impact of Autonomous Ships

We design two scenarios where in the first scenario only conventional ships are used, while in the second scenario we consider all types of ships. The comparison between these two scenarios is shown in Table 9 and 10. The ship fleet in Table 9 is the fleet composition under the corresponding scenarios. C and A represent conventional and autonomous ships, respectively. S, M, and L represent small, medium, and large ships, respectively. Route in Table 9 provides information on the optimal route, including the number of routes selected, and the number of sequences that make up of each route. The first row of Table 10 contains the names of six feeder ports. We find that when autonomous ships are added to the network, profits increase while costs decrease. The increase in profit comes from the larger capacity of autonomous ships to serve more demand, which could be reflected from the demand satisfaction level in Table 10. The cost reduction is the result of two main causes. First, ships with larger capacity can visit more or even farther ports in one voyage, thus reducing the number of sequences and reducing voyage cost. Second, more port demands can be fulfilled in one visit, reducing

the additional cost of multiple visits to the same port. Therefore, although autonomous ships are more expensive, the benefits far outweigh the costs.

Table 9. Comparison between the cases with and without autonomous ships

	Total profit	Voyage cost	Capital cost	Service profit	Ship fleet	Route
Scenario 1	123959	65996	4356	194312	1 C-S, 1 C-L	two routes: one sequence three sequences
Scenario 2	150573	42839	5219	198631	1 A-M, 1 A-L	two routes: one sequence two sequences

Note: Scenario 1 denotes the condition where only conventional ships are permissible for use, while Scenario 2 denotes the condition where both conventional and autonomous ships are allowed.

Table 10. Port demand satisfaction level under two scenarios

	NT	JY	ZJ	NJ	MAS	WHU
Scenario 1	99.4%	100.0%	100.0%	100.0%	96.0%	79.6%
Scenario 2	100.0%	100.0%	100.0%	100.0%	99.0%	94.9%

Note: Scenario 1 denotes the condition where only conventional ships are permissible for use, while Scenario 2 denotes the condition where both conventional and autonomous ships are allowed.

5.4.2 The Impact of Port Restriction

As mentioned before, due to waterway restrictions on feeder ports, such as draft and width constraints, some ports inaccessible to conventional ships may be visited by lighter autonomous ships of the same size. In this section, we want to check whether the results would change if ports treated both types of ships equally. The results are shown in Tables 11 and 12. We find that port restrictions do not influence the outcome in this research. Because autonomous ships in this research are more advantageous. Even though we relaxed port restrictions, the same ship type was chosen. In a situation where conventional ships would be more beneficial, the port restriction may have impacts on system performance.

Table 11. Comparison between the cases with and without port restriction

Restriction	Total profit	Voyage cost	Capital cost	Service profit	Ship fleet	Route
Yes	150573	42839	5219	198631	1 A-M, 1 A-L	two routes: one sequence two sequences
No	150573	42839	5219	198631	1 A-M, 1 A-L	two routes: one sequence

Table 12. Port demand satisfaction level with and without port restriction

	NT	JY	ZJ	NJ	MAS	WHU
With restriction	100.0%	100.0%	100.0%	100.0%	99.0%	94.9%
Without restriction	100.0%	100.0%	100.0%	100.0%	99.0%	94.9%

5.4.3 The Impact of Cost Structure

The previous analysis was carried out in the case where autonomous ships are more advantageous. In this section, we explore what happens if the cost advantage of autonomous ships wanes. We scaled the capital cost and operational cost of autonomous ships by different coefficients in Table 13 and Table 14, respectively. The results in Table 13 suggest that autonomous ships will only lose their competitive advantage if their capital costs are sufficiently large. In this research, autonomous ships will partially and completely lose their competitive advantage if the capital cost is four and 10 times the current value, respectively. Table 14 suggests that if the operational cost increases to four times its current value, the conventional ships will outperform autonomous ones. The findings from both tables indicate that operational costs have a significant impact on ship operations due to their substantial proportion. If the technology and skilled technicians required for autonomous ships become excessively expensive, shipping companies may opt for conventional ships. However, as technology continues to advance, the cost of autonomous ships is expected to decrease gradually, making them increasingly competitive in the future. Therefore, once they are allowed to be widely adopted in the maritime industry and their costs become manageable, autonomous ships have the potential to replace traditional ships.

Table 13. The impact of capital cost

Coefficient	Total profit	Voyage cost	Capital cost	Profit	Ship fleet
1	150573	42839	5219	198631	1 A-M, 1 A-L
2	145353	42839	10439	198631	1 A-M, 1 A-L
4	140198	43913	11726	195837	1 C-S, 1 A-M
6	135266	43913	16658	195837	1 C-S, 1 A-M
8	130335	43913	21589	195837	1 C-S, 1 A-M
10	123959	65996	4356	194312	1 C-S, 1 C-L

Table 14. The impact of operational cost

Coefficient	Total profit	Voyage cost	Capital cost	Profit	Ship fleet

1	150573	42839	5219	198631	1 A-M, 1 A-L
2	137724	54969	5219	197912	1 A-M, 1 A-L
3	126166	62196	4795	193156	1 A-S, 1 A-L
4	123959	65996	4356	194312	1 C-S, 1 C-L
6	123959	65996	4356	194312	1 C-S, 1 C-L
8	123959	65996	4356	194312	1 C-S, 1 C-L

5.4.4 The Impact of Demand Satisfaction

In the previous analysis, we found that in order to maximize total profit, not all demands are satisfied. In this section, we explore the impact of satisfying all demands. Results are shown in Table 15. The first column indicates whether the constraints by which all demands should be satisfied are considered. We find that when all demands are required to be fulfilled, the total profit decreases. The reason is that the number of ports that can be served during a voyage is reduced. Therefore, more sequences must be traveled, increasing voyage costs.

Table 15. Comparison between the cases with and without constraints of demand satisfaction

All demand	Total profit	Voyage cost	Capital cost	Service profit	Ship fleet	Route
No	150573	42839	5219	198631	1 A-M, 1 A-L	two routes: one sequence two sequences
Yes	143933	50822	5219	199975	1 A-M, 1 A-L	two routes: two sequences two sequences

5.4.5 The Impact of technology development and government intervention

The level of automation in autonomous ships is continuously evolving. The IMO has classified them into four levels based on their degree of automation: crewed ship with automated processes and decision support (degree one); remotely controlled ship with seafarers on board (degree two); remotely controlled ship without seafarers on board (degree three); and fully autonomous ship (degree four). Based on whether there are seafarers on board, we can further divide all the ships into two categories: low degree and high degree. For the low degree ships, we assume that they have the same exterior design as the conventional ships, while for high degree ships, we assume that the deckhouse and accommodation structures are removed so that they have greater capacity for cargoes. Table 16 and 17 illustrate the impact of capital and operational

costs when considering both conventional ships and low-degree autonomous ships. We assume that low-degree autonomous ships have the same capacity as conventional ships, but their cost structure aligns with that of the autonomous ships discussed in Section 5.1. The results in Table 16 and 17 align with the trends observed in Section 5.4.3. Since low-degree autonomous ships do not possess a capacity advantage, their adoption will primarily depend on the level of capital and operational costs. Only when these costs are sufficiently low will low-degree autonomous ships be chosen. For high-degree autonomous ships, where autonomous ships have greater capacity, the optimization results align with those discussed in Section 5.4.3.

Table 16. The impact of capital cost when both conventional ships and low degree autonomous ships are involved

Coefficient	Total profit	Voyage cost	Capital cost	Profit	Ship fleet
1	136553	52964	4795	194312	1 A-S, 1 A-L
2	131758	52964	9589	194312	1 A-S, 1 A-L
3	127217	56971	10123	194312	1 C-S, 1 A-L
4	124464	56971	12877	194312	1 C-S, 1 A-L
6	123959	65996	4356	194312	1 C-S, 1 C-L
8	123959	65996	4356	194312	1 C-S, 1 C-L

Table 17. The impact of operational cost when both conventional ships and low degree autonomous ships are involved

Coefficient	Total profit	Voyage cost	Capital cost	Profit	Ship fleet
1	136553	52964	4795	194312	1 A-S, 1 A-L
2	123959	65996	4356	194312	1 C-S, 1 C-L
3	123959	65996	4356	194312	1 C-S, 1 C-L

Another crucial factor that influences the adoption of autonomous ships is government policies. The government can stimulate the adoption of autonomous ships by offering subsidies or establishing regulations on the proportion of autonomous ships owned by shipping companies. Subsidies can effectively reduce either the capital or the operational costs, and their impacts align with those discussed in Section 5.4.3. Setting the proportion of autonomous ships owned by shipping companies can yield different outcomes, as demonstrated in Table 18. In the previous analysis, conventional ships were considered only when the operational costs of low-degree autonomous ships and high-degree autonomous ships respectively increased to two and four times their original values. Therefore, we use the results obtained by scaling the operational costs

to the corresponding multiples as the reference for comparison. Table 18 indicates that when autonomous ships are advantageous, there is no need to set a proportion because the optimal results will not include conventional ships. However, if conventional ships are preferred, setting the proportion will reduce the profits of shipping companies. In such cases, the government would need to provide additional benefits to compensate for the profit losses.

In summary, the level of automation and government policies, such as subsidies and regulations on the proportion of autonomous ships owned by shipping companies, play a significant role in shaping the application of autonomous ships. Higher levels of automation can provide autonomous ships with a capacity advantage. However, when making decisions, shipping companies must also consider the impact of costs. Government intervention can potentially result in reduced profits for shipping companies. Therefore, it is necessary to provide compensation measures to mitigate any financial losses incurred by these companies.

Table 18. The impact of setting proportion

Case	Proportion	Total profit	Voyage cost	Capital cost	Profit	Ship fleet
I	none	123959	65996	4356	194312	1 C-S, 1 C-L
	10%	123272	66505	4534	194312	1 C-L, 1 A-S
	30%	123272	66505	4534	194312	1 C-L, 1 A-S
	50%	123272	66505	4534	194312	1 C-L, 1 A-S
	100%	121064	68453	4795	194312	1 A-S, 1 A-L
II	none	123959	65996	4356	194312	1 C-S, 1 C-L
	10%	121408	65951	5000	192359	1 C-M, 1 A-L
	30%	121408	65951	5000	192359	1 C-M, 1 A-L
	50%	121408	65951	5000	192359	1 C-M, 1 A-L
	100%	114702	73660	4795	193156	1 A-S, 1 A-L

Notes: Case I denotes the condition where conventional ships and autonomous ships with seafarers on board are considered, while Case II denotes the condition where conventional ships and autonomous ships without seafarers on board are considered. The “none” in the “Proportion” column means there is no restriction on the proportion of the autonomous ships.

6. Conclusions

Although autonomous ships may be the future of the shipping industry, they are currently in the early stages. Very little research has been conducted on the influence of autonomous ships on conventional ones in national waterways. In this paper, we have developed a two-stage stochastic programming model to investigate the impact of autonomous ships on shipping company operations. In the first stage, optimal routes,

fleet composition, and fleet assignment are determined without the realization of demand uncertainty. In the second stage, when demand realization becomes known, the liner company determines the delivery pattern, i.e., the volume of cargo loaded on each ship from each feeder port. The objective is to maximize the expected total profit. An approximation algorithm SAA and an exact solution method TPBBC were proposed to solve the problem. To speed up the computations, we have used acceleration strategies, such as column generation and variable fixing.

The solution methods were validated by numerical experiments based on real-world data. After comparison, TPBBCCGVF-I was proved to be more suitable and efficient for this research. Sensitivity analyses are conducted to further explore the influences of key factors and derive valuable managerial insights to guide practical implementation in shipping companies. Our results show that autonomous ships are competitive under most of the scenarios considered. Besides, it was proved that the satisfaction of all demands will lead to a decrease in total profits.

Despite the contributions and insights provided by this study, there are certain limitations that should be acknowledged. Firstly, this research assumes full automation of ships that can operate without human intervention. However, in reality, autonomous ship technology is still in the research and development stage, and achieving full automation requires multiple stages of development and testing. Therefore, to better reflect the evolutionary process of autonomous ships, future research can consider developing a multi-period model that incorporates the characteristics of autonomous ships at different stages of development. Secondly, the current study addresses demand uncertainty by using scenarios, which may not capture the full range of demand characteristics. To overcome this limitation, future research can explore the use of advanced data analytics and machine learning techniques to enhance the accuracy of demand modeling.

Appendix A. Label setting algorithm to generate feasible route set

Label setting algorithm is a common approach used to solve minimum-cost network flow problems. The efficiency of label setting in this research is mainly affected by two main factors. The first is the number of ports. The second is two types of constraints used during sequence and route generation process, namely the ship type constraint and the duration constraint. To speed up the computation process, we first group feeder ports by vessel type. Each port p can be visited by a ship in ship type set V_p. For each vessel type $v \in V$, we have a set of feeder ports $P_v = \{1, \dots, |P_v|\}$, where $v \in V_p,\ \forall p \in P_v$. This grouping procedure has two main benefits. First, it can make the number of ports in each group less than total number of ports, hence reducing the computation time. Second, it eliminates the procedure to judge whether ship type constraint is satisfied.

We define a forward path $f_v = (0, i_1, \dots, i_n)$ for each vessel type v, which is a sequence or part of a sequence. It starts at the hub port p_0 (denoted by 0), visits a set of feeder ports $P^{f_v} = \{i_1, \dots, i_{n-1}\} \subseteq P_v$, and ends at port $i_n \in \{0\} \cup P_v \backslash P^{f_v}$. If it ends at port $\{0\}$, it is a complete sequence, otherwise it is part of a sequence. We associate a label $L^{f_v} = (P^{f_v}, i_n, T^{f_v})$ with each path $f_v = (0, i_1, \dots, i_n)$, where P^{f_v} is the set of feeder ports that this path has already visited, i_n is the last port of the path, T^{f_v} is the duration of the path, including the sailing time and the dwell time. Because of duration constraints, a label is feasible only if $T^{f_v} \leq \alpha$. Clearly, each feasible path $f_v = (0, i_1, \dots, i_n)$, where $i_n = 0$ corresponds to a feasible sequence. To generate a feasible sequence set, we initialize a label with $P_v^f = \emptyset,\ i_n = 0,\ T^{f_v} = 0$. This means that this path has just started from the hub port. In the first extension, the path must visit a feeder port $i_n' \in P_v$. In the next extensions, it can either visit a feeder port $i_n' \in P_v$ or back to hub port $i_n' = 0$. For the sake of clarity, we explain the first and next iterations separately as follows:

First iteration: Let $L^{f_v'}$ represents the label generated by extending L^{f_v} with $i_n' \in P_v$. It is constructed as follows:

- $P^{f_v'} = P^{f_v} \cup \{i_n'\}$;
- $T^{f_v'} = T^{f_v} + t_{i_n} + t_{i_n, i_n'}$;

where t_{i_n} is the dwell time at port i_n, $t_{i_n, i_n'}$ is the travel time from port i_n to port i_n'.

Next iterations:

(1) If $i'_n \in P_v \backslash P^{f_v}$, the label $L^{f'_v}$ is constructed the same as the first iteration.

(2) If $i'_n = 0$, the label $L^{f'_v}$ is constructed as follows:

- $P^{f'_v} = P^{f_v}$;
- $T^{f'_v} = T^{f_v} + t_{i_n} + t_{i_n,0}$.

To speed up the solution process and reduce the number of labels generated, the following dominance rule can be applied. Given two labels $L^{f_v^1} = (P^{f_v^1}, i_n^1, T^{f_v^1})$ and $L^{f_v^2} = (P^{f_v^2}, i_n^2, T^{f_v^2})$, $L^{f_v^1}$ dominates $L^{f_v^2}$ if (i) $P^{f_v^1} = P^{f_v^2}$, (ii) $i_n^1 = i_n^2$, (iii) $T^{f_v^1} < T^{f_v^2}$.

Proof: If relations (i)–(iii) hold for $L^{f_v^1}$ and $L^{f_v^2}$, then for any port $i'_n \in P_v \backslash P^{f_v^2}$ such that $L^{f_v^2}$ can be extended toward i'_n to form a feasible label $L^{f_v^{2'}} = (P^{f_v^{2'}}, i'_n, T^{f_v^{2'}})$, $L^{f_v^1}$ can be extended toward j to form a feasible label $L^{f_v^{1'}} = (P^{f_v^{1'}}, i'_n, T^{f_v^{1'}})$ with relations (i)–(iii) still holding for $L^{f_v^{1'}}$ and $L^{f_v^{2'}}$.

Thus, by applying the above argument repeatedly until $i'_n = 0$, $L^{f_v^2}$ can be extended toward $i'_n = 0$ to form a feasible label $L^{f_v^{2'}} = (P^{f_v^{2'}}, 0, T^{f_v^{2'}})$, $L^{f_v^1}$ can be extended toward $i'_n = 0$ to form a feasible label $L^{f_v^{1'}} = (P^{f_v^{1'}}, 0, T^{f_v^{1'}})$ with relations (i)–(iii) still holding for $L^{f_v^{1'}}$ and $L^{f_v^{2'}}$.

Since $i'_n = 0$, both $L^{f_v^{1'}}$ and $L^{f_v^{2'}}$ are feasible and complete labels and feasible sequences. We can conclude that for every feasible and complete label $L^{f_v^{2'}}$ extending from $L^{f_v^2}$, there exists a feasible and complete label $L^{f_v^{1'}}$ extending from $L^{f_v^1}$ that has an equal or smaller duration. Therefore, $L^{f_v^1}$ dominates $L^{f_v^2}$.

The dominance rule can delete dominated subsequences and sequences during sequence generating process, which improves solution efficiency. Using a label setting algorithm augmented by dominance rules, we obtain all feasible sequences for each vessel type $v \in V$. We then use sequences of the same vessel type as an input to the dynamic programming model in order to generate feasible routes. Finally, we obtain all feasible routes.

Appendix B. Proof of feasibility and boundedness of subproblem

It is easy to show that $y_s^p(\omega) = 0,\ \forall p \in P_s,\ s \in S,\ \omega \in \Omega$ is feasible solution regardless of first-stage solution $\boldsymbol{x}$, which proves the feasibility. We construct two

expressions $\sum_{s\in S}\sum_{p\in P_s}(F_p - C_p^h)D^p(\omega)$ and $\frac{1}{|\Omega|}\sum_{\omega\in\Omega}\sum_{p\in P}\left(F_p - C_p^h\right)D^p(\omega)$, which are the maximum value that SP (13) and (15) can achieve, respectively. This proves the boundness.

Appendix C. Proof of Proposition 1

Proof. Given a feasible solution $\boldsymbol{x}'$ and $\boldsymbol{y}'$ and the optimal solution $\boldsymbol{x}^*$ and $\boldsymbol{y}^*$ of $\boldsymbol{P}$, and a feasible dual solution $\boldsymbol{\omega}$ of the linear relaxation of $\boldsymbol{P}$, we deduct:

$$z(\boldsymbol{P}) = \sum_{i=1}^{n_1} c_i x_i^* + \sum_{j=1}^{n_2} d_j y_j^* \tag{22}$$

$$z(\boldsymbol{P}) \le \sum_{i=1}^{n_1} c_i x_i^* + \sum_{j=1}^{n_2} d_j y_j^* + \boldsymbol{\omega}(\boldsymbol{h} - \sum_{i=1}^{n_1} a_i x_i^* - \sum_{j=1}^{n_2} b_j y_j^*) \tag{23}$$

$$z(\boldsymbol{P}) \le \sum_{i=1}^{n_1}(c_i - \boldsymbol{\omega} a_i)x_i^* + \sum_{j=1}^{n_2}(d_j - \boldsymbol{\omega} b_j)y_j^* + \boldsymbol{\omega h}, \tag{24}$$

where a_i (resp. b_j) is the i^{th} (resp. j^{th}) column coefficient vector of matrix $\boldsymbol{A}$ (resp. $\boldsymbol{B}$) and c_i (resp. d_j) is the i^{th} (resp. j^{th}) coefficient of vector $\boldsymbol{c}$ (resp. $\boldsymbol{d}$). Since $\boldsymbol{\omega}$ is a feasible dual solution, we have $d_j - \boldsymbol{\omega} b_j \le 0$ for all $j \in \{1, \dots, n_2\}$. Therefore,

$$z(\boldsymbol{P}) \le \sum_{i=1}^{n_1}(c_i - \boldsymbol{\omega} a_i)x_i^* + \boldsymbol{\omega h} \tag{25}$$

after transformation, and we have $(c_i - \boldsymbol{\omega} a_i)x_i^* \ge \sum_{i=1}^{n_1}(c_i - \boldsymbol{\omega} a_i)x_i^* \ge z(\boldsymbol{P}) - \boldsymbol{\omega h}$ for all $i \in \{1, \dots, n_1\}$. Since $z(\boldsymbol{P}) \ge \boldsymbol{cx}' + \boldsymbol{dy}'$, it holds that $(c_i - \boldsymbol{\omega} a_i)x_i^* \ge \boldsymbol{cx}' + \boldsymbol{dy}' - \boldsymbol{\omega h}$ for all $i \in \{1, \dots, n_1\}$, which completes the proof.

Reference

Accorsi, L., & Vigo, D. (2021). A fast and scalable heuristic for the solution of large-scale capacitated vehicle routing problems. *Transportation Science*, 55(4), 832-856.

Adulyasak, Y., Cordeau, J.-F., & Jans, R. (2015). Benders decomposition for production routing under demand uncertainty. *Operations Research*, 63(4), 851-867.

Archetti, C., Bianchessi, N., & Speranza, M. G. (2014). Branch-and-cut algorithms for the split delivery vehicle routing problem. *European Journal of Operational Research*, 238(3), 685-698.

Archetti, C., Savelsbergh, M. W. P., & Speranza, M. G. (2008). To split or not to split: That is the question. *Transportation Research Part E: Logistics and Transportation Review*, 44(1), 114-123.

Authority, D. M. (2017). Analysis of regulatory barriers to the use of autonomous ships final report. *Kopenhag: Danish Maritime Authority*.

Bianchessi, N., & Irnich, S. (2019). Branch-and-cut for the split delivery vehicle routing problem with time windows. *Transportation Science*, 53(2), 442-462.

Branchini, R. M., Armentano, V. A., & Morabito, R. (2015). Routing and fleet deployment in liner shipping with spot voyages. *Transportation Research Part C: Emerging Technologies*, 57, 188-205.

Chang, C. H., Kontovas, C., Yu, Q., & Yang, Z. (2021). Risk assessment of the operations of maritime autonomous surface ships. *Reliability Engineering & System Safety*, 207, 107324.

Cheng, H. H., & Ouyang, K. (2021). Development of a strategic policy for unmanned autonomous ships: a study on Taiwan. *Maritime Policy & Management*, 48(3), 316-330.

Coelho, V. N., Grasas, A., Ramalhinho, H., Coelho, I. M., Souza, M. J., & Cruz, R. C. (2016). An ILS-based algorithm to solve a large-scale real heterogeneous fleet VRP with multi-trips and docking constraints. *European Journal of Operational Research*, 250(2), 367-376.

Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, 6(1), 80-91.

Desaulniers, G. (2010). Branch-and-price-and-cut for the split-delivery vehicle routing problem with time windows. *Operations Research*, 58(1), 179-192.

Despaux, F., & Basterrech, S. (2016). Multi-trip vehicle routing problem with time windows and heterogeneous fleet. *International Journal of Computer Information Systems and Industrial Management Applications*, 8, 355-363.

de Vos, J., Hekkenberg, R. G., & Banda, O. A. V. (2021). The impact of autonomous ships on safety at sea–a statistical analysis. *Reliability Engineering & System Safety*, 210, 107558.

Dror, M., & Trudeau, P. (1990). Split delivery routing. *Naval Research Logistics*, 37(3), 383-402.

Fadda, P., Mancini, S., Serra, P., & Fancello, G. (2023). The heterogeneous fleet vehicle routing problem with draft limits. *Computers & Operations Research*, 149, 106024.

Fan, C., Wróbel, K., Montewka, J., Gil, M., Wan, C., & Zhang, D. (2020). A framework to identify factors influencing navigational risk for Maritime Autonomous Surface Ships. *Ocean Engineering*, 202, 107188.

François, V., Arda, Y., Crama, Y., & Laporte, G. (2016). Large neighborhood search for multi-trip vehicle routing. *European Journal of Operational Research*, 255(2), 422-441.

Ghaderi, H. (2019). Autonomous technologies in short sea shipping: trends, feasibility and implications. *Transport Reviews*, 39(1), 152-173.

Goerlandt, F. (2020). Maritime autonomous surface ships from a risk governance perspective: Interpretation and implications. *Safety Science*, 128, 104758.

Gschwind, T., Bianchessi, N., & Irnich, S. (2019). Stabilized branch-price-and-cut for the commodity-constrained split delivery vehicle routing problem. *European Journal of Operational Research*, 278(1), 91-104.

Gu, Y., Goez, J. C., Guajardo, M., & Wallace, S. W. (2021). Autonomous vessels: state of the art and potential opportunities in logistics. *International Transactions in Operational Research*, 28(4), 1706-1739.

Gutierrez, A., Dieulle, L., Labadie, N., & Velasco, N. (2018). A multi-population algorithm to solve the VRP with stochastic service and travel times. *Computers & Industrial Engineering*, 125, 144-156.

Jin, J., Zhang, J., & Liu, D. (2018). Design and verification of heading and velocity coupled nonlinear controller for unmanned surface vehicle. *Sensors*, 18(10), 3427.

Jin, M., Liu, K., & Eksioglu, B. (2008). A column generation approach for the split delivery vehicle routing problem. *Operations Research Letters*, 36(2), 265-270.

Kretschmann, L., Burmeister, H. C., & Jahn, C. (2017). Analyzing the economic benefit of unmanned autonomous ships: An exploratory cost-comparison between an autonomous and a conventional bulk carrier. *Research in Transportation Business & Management*, 25, 76-86.

Li, F., Yang, D., Wang, S., & Weng, J. (2019). Ship routing and scheduling problem for steel plants cluster alongside the Yangtze River. *Transportation Research Part E: Logistics and Transportation Review*, 122, 198-210.

Lin, Y., Wang, X., Hu, H., & Zhao, H. (2020). Research on feeder network design: a case study of feeder service for the port of Kotka. *European Transport Research Review*, 12(1), 1-13.

Liu, Z., Zhang, Y., Yu, X., & Yuan, C. (2016). Unmanned surface vehicles: An overview of developments and challenges. *Annual Reviews in Control*, 41, 71-93.

Makhsoos, A., Mousazadeh, H., Mohtasebi, S. S., Abdollahzadeh, M., Jafarbiglu, H., Omrani, E., Salmani, Y., & Kiapey, A. (2018). Design, simulation and experimental evaluation of energy system for an unmanned surface vehicle. *Energy*, 148, 362-372.

Mendoza, J. E., Castanier, B., Guéret, C., Medaglia, A. L., & Velasco, N. (2010). A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands. *Computers & Operations Research*, 37(11), 1886-1898.

Munim, Z. H. (2019). Autonomous ships: A review, innovative applications and future maritime business models. *Supply Chain Forum: An International Journal*, 20(4), 266-279.

Notteboom, T., & Cariou, P. (2013). Slow steaming in container liner shipping: is there any impact on fuel surcharge practices? *The International Journal of Logistics Management*, 24(1), 73-86.

Paradiso, R., Roberti, R., Laganá, D., & Dullaert, W. (2020). An exact solution framework for multitrip vehicle-routing problems with time windows. *Operations Research*, 68(1), 180-198.

Pecin, D., Pessoa, A., Poggi, M., & Uchoa, E. (2017). Improved branch-cut-and-price for capacitated vehicle routing. *Mathematical Programming Computation*, 9, 61-100.

Peng, C., Delage, E., & Li, J. (2020). Probabilistic envelope constrained multiperiod stochastic emergency medical services location model and decomposition scheme. *Transportation Science*, 54(6), 1471-1494.

Pessoa, A., Sadykov, R., Uchoa, E., & Vanderbeck, F. (2020). A generic exact solver for vehicle routing and related problems. *Mathematical Programming*, 183, 483-523.

Poggi, M., & Uchoa, E. (2014). New exact algorithms for the capacitated vehicle routing problem. *Vehicle Routing: Problems, Methods, and Applications* (pp. 59-86).

Şahin, M. K., & Yaman, H. (2022). A branch and price algorithm for the heterogeneous

fleet multi-depot multi-trip vehicle routing problem with time windows. *Transportation Science*, 56(6), 1636-1657.

Schiaretti, M., Chen, L., & Negenborn, R. R. (2017a). Survey on autonomous surface vessels: Part I-a new detailed definition of autonomy levels. *In International Conference on Computational Logistics* (pp. 219-233). Springer, Cham.

Schiaretti, M., Chen, L., & Negenborn, R. R. (2017b). Survey on autonomous surface vessels: Part II-categorization of 60 prototypes and future applications. *In International Conference on Computational Logistics* (pp. 234-252). Springer, Cham.

Taillard, E. D., Laporte, G., & Gendreau, M. (1996). Vehicle routeing with multiple use of vehicles. *Journal of the Operational Research Society*, 47(8), 1065-1070.

Tan, Z., Liu, Q., Song, J., Wang, H., & Meng, Q. (2021). Ship choice and shore-power service assessment for inland river container shipping networks. *Transportation Research Part D: Transport and Environment*, 94, 102805.

Torres, F., Gendreau, M., & Rei, W. (2022). Crowdshipping: An open VRP variant with stochastic destinations. *Transportation Research Part C: Emerging Technologies*, 140, 103677.

Wang, L., Kinable, J., & Van Woensel, T. (2020). The fuel replenishment problem: A split-delivery multi-compartment vehicle routing problem with multiple trips. *Computers & Operations Research*, 118, 104904.

Wang, Y., Yu, X., Liang, X., & Li, B. (2018). A COLREGs-based obstacle avoidance approach for unmanned surface vehicles. *Ocean Engineering*, 169, 110-124.

Yang, X., Gu, W., Wang, W., & Wang, S. (2023). Optimal scheduling of autonomous vessel trains in a hub-and-spoke network. *Ocean & Coastal Management*, 231, 106386.

Yang, Y. (2022). An exact price-cut-and-enumerate method for the capacitated multitrip vehicle routing problem with time windows. *Transportation Science*, 57(1), 230-251.

Yoshizaki, H. T. Y. (2009). Scatter search for a real-life heterogeneous fleet vehicle routing problem with time windows and split deliveries in Brazil. *European Journal of Operational Research*, 199(3), 750-758.

Zhu, L., & Xing, W. (2021). Policy-oriented analysis on the navigational rights of unmanned merchant ships. *Maritime Policy & Management*, 49(3), 447-462.

Ziajka-Poznańska, E., & Montewka, J. (2021). Costs and benefits of autonomous shipping—a literature review. *Applied Sciences*, 11(10), 4553.

Zolich, A., Palma, D., Kansanen, K., Fjørtoft, K., Sousa, J., Johansson, K.H., Jiang, Y., Dong, H. (2018). Survey on communication and networks for autonomous

marine systems. *Journal of Intelligent & Robotic Systems*, 95, 789-813.

Wei Wang: Conceptualization, Methodology, Programming, Writing—Original draft preparation. Shuaian Wang: Methodology, Writing—Reviewing and Editing. Lu Zhen: Methodology, Writing—Reviewing and Editing. Gilbert Laporte: Methodology, Writing—Reviewing and Editing.