PUBLISHED IN Transportation Research Part B, 120, 15-32, 2019

Fleet deployment and demand fulfillment for container shipping liners

Lu Zhen[a], Yi Hu[a], Shuaian Wang[b,*], Gilbert Laporte[c], Yiwei Wu[a]

[a]School of Management, Shanghai University, Shanghai, China
[b]The Hong Kong Polytechnic University Shenzhen Research Institute, Nanshan District, Shenzhen, China
Department of Logistics & Maritime Studies, The Hong Kong Polytechnic University, Hung Hom, Hong Kong
[c]Department of Decision Sciences, HEC Montréal, Montréal, Canada

Abstract

This paper models and solves a fleet deployment and demand fulfillment problem for container shipping liners with consideration of the potential overload risk of containers. Given the stochastic weights of transported containers, chance constraints are embedded in the model at the strategic level. Several realistic limiting factors such as the fleet size and the available berth and yard resources at the ports are also considered. A non-linear mixed integer programming (MIP) model is suggested to optimally determine the transportation demand fulfillment scale for each origin-destination pair, as well as the ship deployment plan along each route, with an objective incorporating revenue, fixed operation cost, fuel consumption cost, holding cost for transhipped containers, and extra berth and yard costs. Two efficient algorithms are then developed to solve the non-linear MIP model for different instance sizes. Numerical experiments based on real-world data are conducted to validate the effectiveness of the model and the algorithms. The results indicate the proposed methodology yields solutions with an optimality gap less than about 0.5%, and can solve realistic instances with 19 ports and four routes within about one hour.

Keywords: Demand fulfillment; fleet deployment; transsphipment; port capacity; stochastic container weight.

*Corresponding author. lzhen@shu.edu.cn (L. Zhen), wangshuaian@gmail.com (S. Wang)

1. Introduction

Shipping liners play an important role in today's economy which is becoming increasingly global, and more operations are being outsourced and moved offshore (Fransoo and Lee, 2013). Shipping liners run weekly-serviced ship routes with fixed schedules to transport containers for customers. Each shipping company operates its own shipping network covering a number of routes (services) and ports. A shipping liner cannot usually fulfill all customer demands in a given container transportation market due to the limitations of its fleet size and of the available port resources (e.g., berths and yard space), and because of some other unforeseen factors (Zhen, 2015, 2016). The transportation demand is usually characterized by the number of containers that need to be transported between the origin-destination (OD) pairs of the shipping network. Given the data on the full-size market demand, a shipping liner needs to determine an economic fulfillment scale for each OD pair's transportation demand, as well as the number of ships deployed on each route of its shipping network so as to maximize its profit. This is an important strategic decision for the managers of shipping liners.

The above strategic level problem involves intertwined decisions as well as numerous complex factors. While it is easy to understand that the demand fulfillment scale is positively related to the number of deployed ships, the optimal allocation of the available ships along the routes is not a straightforward decision because of the different unit transportation fees among OD pairs, the different cost configurations among the routes, and the complex underlying relationship between the OD pairs and the routes. A liner may not always fulfill as much transportation demand as possible by using all its available ships because the port resources reserved for the liner in the shipping network are fixed. Moreover, several features proper to the ocean shipping industry must also be considered in this strategic level decision problem. For example, the number of ships deployed on a route affects the ships' speed on each leg of a route, which further influences fuel consumption and cost. These costs and the fixed operation costs of the deployed ships jointly constitute the bulk of the cost for a shipping liner. In addition, the ship schedule of each route (service) affects the containers' storage time at the transshipment hubs which connect the routes in the shipping network. The holding cost of the transshipped containers should therefore also be taken into account. Finally, the potential overload risk of containers should not be ignored since the weights of the transported containers are stochastic. Wang et al. (2016) state that almost all the existing literature regards the weights of containers as constants and few existing studies consider the problem of container overload. However, the potential overload risk of containers occurs frequently and has irreparable consequences. Indeed, ship overload accidents account for 60 percent of accidents on inland waterways and up to 70 percent in some areas. Therefore, studying the overload risk of containers is practical. For example, a shipping liner may promise a quota of 1,000 twenty-foot equivalent units (TEUs) to a customer with respect to an OD pair, but when the shipping liners make long term decisions on the demand fulfillment scale, the cargo types and weights in the containers are unforeseen. For example, the weights of 1,000 TEUs of plastic and of metal are significantly different. The overload risk should therefore also be controlled.

This paper provides a comprehensive study of this complex decision problem. Given a shipping network with multiple routes connected by transshipment hubs and the transportation demand information, we propose a non-linear chance-constrainted mathematical integer program (MIP) to optimally determine the transportation demand fulfillment scale for each OD pair, as well as the ship deployment plan along each route in order to maximize the total profit, equal the revenue earned by fulfilling the demand, minus four types of cost: the fixed operation cost of the deployed ships, the fuel cost, the cost for storing transshipped containers at ports, and the cost of using extra port resources. The chance constraints embedded in the model control the potential overload risk resulting from random container weights. In addition to the chance constraints, the model contains other non-linear components. Some new techniques are suggested to linearize the model into a mixed integer second-order cone programming (MISOCP) model that can be tractable for some commercial solvers such as CPLEX.

The remainder of this paper is organized as follows. Section 2 provides an overview of the related literature. Section 3 describes the problem. A mathematical model is proposed in Section 4, followed by a linearization scheme in Section 5. Two heuristics are developed to solve the model in Section 6. Section 7 reports the results of our computational experiments. Closing remarks and conclusions follow in Section 8.

2. Related works

There exist numerous related studies on fleet deployment. Readers interested in overviews can refer to Ronen (1993), Christiansen et al. (2004), Christiansen et al. (2013), and Meng et al. (2014). At the strategic decision level, the fleet deployment problem (FDP) consists of assigning available vessels to predetermined voyages (Fagerholt et al., 2009) in order to maximize profit or minimize cost.

Several linear programming and mixed integer linear programming models for the FDP have been put forward. Perakis and Jaramillo (1991) were the first to develop a linear programming model for the FDP, which takes account of ship capacity, and minimizes service frequency requirements as well as ship charter cost. However, this model works with continuous decision variables for the allocation of ships to shipping routes, instead of integer variables. To remedy this problem, Jaramillo and Perakis (1991) proposed an integer programming model. Cho and Perakis (1996) formulated a MIP model for the FDP, where the demand of containers between two given ports can be served by any shipping route passing through the two ports. Powell and Perkins (1997) extended the model of Jaramillo and Perakis (1991) by adding ship lay-out costs to the objective function. Álvarez (2009) proposed a MIP formulation for the integrated optimization of vessel routing and fleet deployment. Based on previous works, Gelareh and Meng (2010) developed a MIP model for the FDP in which speed is a decision variable, and investigated the problem of ship speed optimization to obtain optimal sailing speeds through a non-linear model, which can be approximated as a MIP model. This model was later improved by Wang et al. (2011). Meng and Wang (2011) investigated a multi-period fleet planning and FDP with a known container demand for each OD pair and each period. Meng and Wang (2010) proposed a chance-constrained model for the FDP under uncertain demand, but ignored transshipment activities. Because the speed of ships has an impact on fuel consumption cost, Zacharioudakis et al. (2011) developed a practical methodology that considers the effect of speed on fuel consumption for shipping companies to solve FDPs. Andersson et al. (2015) put forward an integrated model to optimize fleet deployment and sailing speed for RoRo shipping companies. Zheng et al. (2015) set up a shipping network for liner shipping alliances, and proposed a model with consideration of ship deployment, cargo allocation, and container routing. Xia et al.

(2015) developed a comprehensive model to simultaneously and optimally determine ship deployment, sailing speed, and container allocation in order to maximize profit at the strategic level. Zhao et al. (2016) designed a novel method of fleet deployment based on risk evaluation so as to take advantage of resources for navigation and reduce risks. Monemi and Gelareh (2017) provided an integrated model considering shipping network design, FDP and empty container repositioning. The number of routes and their design play an endogenous role in their problem. Wang et al. (2017) proposed a two-stage stochastic programming model to optimally solve the FDP and compute the sailing speeds with the consideration of market uncertainties. Some studies have incorporated container transshipment in FDPs. Wang and Meng (2012) developed an MIP model for the FDP in which containers can be transshipped at any port, which was extended by Meng and Wang (2012) by adding transit time constraints.

There also exist some studies on FDPs that consider the uncertainties of liner service schedule or container shipment demand. Wang and Meng (2012), Qi and Song (2012) and Bell et al. (2011) considered uncertainty in the liner service schedule but ignored uncertainty in container shipment demand. In order to tackle demand uncertainty, Meng and Wang (2010) proposed a chance-constrained model, which extends the deterministic FDP to a FDP under uncertainty. Meng et al. (2012) assumed that the container shipment demand is a random variable, and hence formulated a two-stage stochastic integer programming model, and developed an algorithm integrating sample average approximation with a dual decomposition and Lagrangian relaxation method. Wang et al. (2012) further extended the model of Meng et al. (2012) by adding the expectation and variance of the cost in the objective function.

In conclusion, several related studies on the FDP have not taken transshipment activities into account. Although some authors did consider these, they did not incorporate the demand fulfillment decision and the potential overload risk of containers due to their stochastic weights. Moreover, some port resources such as berths and yard space, which are crucial in maritime activities, have also been ignored. (Liu et al., 2016) conducted an integrated planning of the berth allocation and the yard allocation in container terminals.

Our paper proposes an integrated decision model that compounds ship fleet deployment and demand fulfillment decisions by considering crucial factors such as transshipment activities, the stochastic weight of containers, port resources, the timetabling of ship visits at each port of call, and the demand fulfillment scale for each OD pair. There is no doubt that these factors complicate this already difficult fleet deployment and demand fulfillment problem. We propose a comprehensive model and we develop some techniques to handle the complexity resulting from the chance constraints. We believe the problem features considered in our study are realistic and new with respect to previous research.

3. Problem description

We consider a shipping liner operating on a network containing a set R of container shipping routes (services), which cover a set P of ports. Figure 1 depicts a shipping network with four routes and 19 ports. Each ship route r is described as (port p_{r1}, port $p_{r2}, \cdots$,port $p_{ri}, \cdots$,port p_{rN_r}, port p_{r1}), which implies that ship route r has N_r ports of call as well as N_r legs. Let leg i denote the voyage from port p_{ri} to port $p_{r,i+1}$, where $p_{r,N_r+1} = p_{r1}$. We denote by I_r the set of legs in ship route r. The details on the objective and key constraints considered in this study are provided in the following subsections.

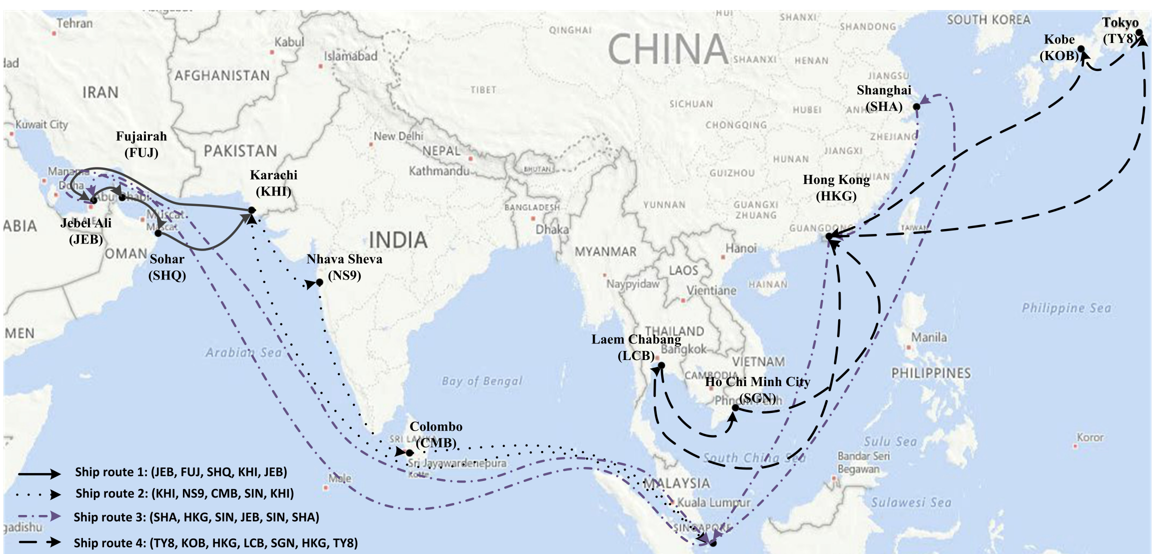


Figure 1: A shipping network with four routes

3.1. Revenue of the demand fulfillment

Container transportation demand is usually described by OD pairs indexed by $\varepsilon \in \Omega$. The number of containers requesting transport for each OD pair during a week can be estimated according to historical data. Given the unit fee for transporting a TEU container, we can compute the maximum revenue V_ε that can be earned if all the transportation demand of OD pair ε is fulfilled. We define a variable π_ε equal to the percentage of OD pair ε's transportation demand fulfilled by the shipping liner. Then the total revenue can be calculated as $\sum_{\varepsilon \in \Omega} V_\varepsilon \pi_\varepsilon$.

3.2. Fixed operation cost of deployed ships

A fleet of homogeneous ships is deployed on each route to maintain a weekly service frequency. If the number of ships deployed on route r is β_r, then the total fixed operation cost for all the deployed ships in all the routes during one week can be calculated as $\sum_{r \in R} C_r^{Opr} \beta_r$, where C_r^{Opr} is the weekly operation cost for deploying one ship on route r.

3.3. Fuel cost depending on sailing speed

The total time for a ship completing the travel along a route is $7\beta_r$ days. More specifically, $\sum_{i \in I_r} (d_{ri} + \delta_{ri}) = 7\beta_r$, where d_{ri} is the dwell time of ships at the i^{th} port of call on ship route r, and δ_{ri} is the sailing time of ships on the i^{th} leg on ship route r. In reality, the port dwell time d_{ri} is usually predetermined according to some contracts between the shipping liner and port operators, but the sailing time δ_{ri} of each leg can be a decision variable for the shipping liner, which can be used to modify the value of δ_{ri} by updating the ships' speed on each leg.

A ship's unit fuel consumption significantly depends on its sailing speed. In this study, we assume that the unit fuel consumption function on sailing speed y is calculated as $y = kx^a$ (USD per nautical mile), where x is the speed, and k and a are positive coefficients. More specifically, the fuel cost for the i^{th} leg on ship route r is $l_{ri} k_{ri} (l_{ri}/\delta_{ri})^{a_{ri}}$, where l_{ri} is the leg's length, and k_{ri} and a_{ri} are coefficients that can be estimated according to historical data. The total fuel cost is then calculated as $\sum_{r \in R} \sum_{i \in I_r} l_{ri} k_{ri} (l_{ri}/\delta_{ri})^{a_{ri}}$.

3.4. Holding cost for storing transshipped containers

The above decisions on ship deployment and sailing speeds influence the cost related to each route which are inter-route costs. Decisions made on the arrival time of ships at each port of call in each route affect the storing time and cost of the containers at transshipment hubs, which are inter-route costs.

We define a quadruple (r, i, s, j) to denote that the i^{th} port of call on ship route r and the j^{th} port of call on ship route s are the same physical port in the network, where $r, s \in R, i \in I_r$ and $j \in I_s$. Hence $Q = \{(r, i, s, j)|p_{ri} = p_{sj}\}$. Let $m_{risj\varepsilon}$ be the maximum number of TEUs transshipped at hub (r, i, s, j) for OD pair ε if all the transportation demand for the OD pair is fulfilled. Then the number of transshipped containers at the hub is $\pi_\varepsilon m_{risj\varepsilon}$. We define a parameter C^{Hold} equal to the unit holding cost (USD per TEU per day), and a variable γ_{risj} to denote the difference in days between the time a ship visits the port of call (r, i) and the time a ship visits (s, j). Then the total holding cost for storing transshipped containers is $C^{Hold} \sum_{(r,i,s,j)\in Q} \sum_{\varepsilon\in\Omega} \pi_\varepsilon m_{risj\varepsilon} \gamma_{risj}$.

3.5. Cost for using extra berth or yard space

Each port has a certain yard space reserved for storing transshipped containers, and a certain number of berths for the shipping liner, booked in advance according to contracts. If the yard space and berth capacity limitations at ports are violated, then some extra costs are incurred (Petering et al., 2017).

In this study, we define B_p as the set of berths b in port p reserved for the shipping liner. Another index $\hat{b}$ is defined as a dummy berth, which is used when there are no available berths in the reserved berth set B_p when a ship arrives at port p. From the perspective of modeling, if the dummy berth $\hat{b}$ is used by a ship, then an extra cost is incurred. Here we define binary decision variables θ_{rib} to denote whether the ship arrives at berth b in the port of call (r, i), and we define a parameter $C^{Berth}_{p_{ri}}$ as the penalty cost incurred when the dummy berth $\hat{b}$ is used in the port of call (r, i). Then the total cost for extra berth usage is $\sum_{r\in R} \sum_{i\in I_r} C^{Berth}_{p_{ri}} \theta_{ri\hat{b}}$.

For the yard resource, we also define an auxiliary variable λ_{pw} as the extra used yard space (measured in number of TEUs) for storing transshipped containers at port p on day $w \in W$ of a week. The formula for computing the variable λ_{pw} will be explained in the Section 4. Let C_p^{Yard} be the penalty cost for using one unit of extra yard space (TEUs), beyond the agreed reserved yard space, in port p to store the transshipped containers for one day. Then the total cost for extra yard space usage is $\sum_{p\in P}\sum_{w\in W} C_p^{Yard}\lambda_{pw}$.

3.6. Risk of overload due to random container weight

This study also considers the potential overload risk of ships due to the stochastic weight of transported containers. To illustrate this, suppose a liner promises a customer or an agency a quota of one thousand TEUs for an OD pair ε. When the liner makes the long term decision on the demand fulfillment scale for that OD pair, the weights of the cargos in the containers are unforeseen and may create an overload. We define a parameter $n_{ri\varepsilon}$ as the maximum number of containers transported on leg (r, i) for OD pair ε if all the transportation demand for the OD pair is fulfilled. Thus there will be $\lceil \pi_\varepsilon n_{ri\varepsilon} \rceil$ containers be transported on leg (r, i) for the OD pair ε. A stochastic parameter $\tilde{c}_{ri\varepsilon u}$ is defined as the random weight of the containers on leg (r, i) for OD pair ε, where u is the index of the container. Suppose the maximum load capacity (in tons) of a ship on leg (r, i) is A_{ri}^{Load}, and the probability of overload should be constrained to lie under a level α (e.g., 1%, 0.1%), then the constraint $\text{Prob}(\sum_{\varepsilon\in\Omega}\sum_{u=1}^{\lceil \pi_\varepsilon n_{ri\varepsilon} \rceil} \tilde{c}_{ri\varepsilon u} > A_{ri}^{Load}) \leq \alpha$ should hold for each leg (r, i).

3.7. Assumptions and data preparation before using the model

Based on the above analysis on the revenue, on the various types of costs considered in the objective function and on the chance constraints controlling the overload risk, we will formulate a mathematical model in the next section. We first clarify the assumptions of this study:

(1) the shipping network of the ports and routes (voyages) is already determined;

(2) the ships are homogenous on each route in terms of capacity and cost structure;

(3) the ships' dwell time at each port of call is deterministic.

Finally, we provide some explanation on how to prepare some key input data for the decision model. First, a shipping liner should collect the historical data on the weekly demand for each OD pair (Fagerholt et al., 2009; Bell et al., 2013). Based on the estimated unit price for shipping one TEU for each OD pair ε, the liner can calculate the V_ε values (i.e., the maximum revenue that can be earned if all the transportation demand of the OD pair ε is fulfilled). Moreover, the mapping from an OD pair to a set of its covered legs as well as a set of transshipment ports is also deterministic. Given this mapping information, the liner can also estimate the parameters $n_{ri\varepsilon}$ (i.e., the maximum number of containers transported on each leg (r, i) for each OD pair ε) and the parameters $m_{risj\varepsilon}$ (i.e., the maximum number of containers transshipped from the port of call (r, i) to (s, j) for OD pair ε, if all the demand is fulfilled).

Another important input value is the stochastic parameter $\tilde{c}_{ri\varepsilon u}$ about the random container weight on the leg (r, i) for the OD pair ε. The liner could collect the historical data on the weights of containers transported for each OD pair ε, and then calibrate the expected value and standard deviation. Given the mapping information between an OD pair and the set of its covered legs, one can obtain the expected value $\mu_{ri\varepsilon}$ and the standard deviation $\sigma_{ri\varepsilon}$ for the random weights of containers transported on each leg (r, i) for each OD pair ε. These two parameters will be used in Section 5 to linearize the chance constraints in the model.

4. Model formulation

We now introduce a non-linear chance-constrained MIP model for the problem. We first define some indices, sets, input parameters and decision variables.

Indices and sets

ε	index of an OD pair;
Ω	set of all the OD pairs;
r (or s)	index of a ship route;
R	set of all the ship routes;
i (or j)	index of port of call (or leg) on a ship route (leg i is from port of call i to i+1);
I_r	set of the ports of call (or legs) on ship route r;
p	index of a physical port, which is different from the port of call (defined as i);

P — set of all the ports;

p_{ri} — index of the port, which corresponds to the port of call (r, i);

I'_{rp} — set of the ports of call (or legs) on ship route r; these port of calls are the same physical port p;

R'_p — set of ship routes that include port p;

Q — set of quadruples (r, i, s, j), where $r, s \in R; i \in I_r, j \in I_s$; a (r, i, s, j) means the ports of call (r, i) and (s, j) are the same physical port in shipping network. $Q = \{(r, i, s, j) | p_{ri} = p_{sj}\}$;

Q_p — a subset of Q; $Q_p = \{(r, i, s, j) | p_{ri} = p_{sj} = p\}$;

w — index of a day in a week, i.e., $0 =$ Sun, $1 =$ Mon, $2 =$ Tue,$\cdots$, $6 =$ Sat;

W — set of days in a week, $W = \{0, 1, 2, \cdots, 6\}$;

b — index of a berth;

B_p — set of berths in port p; these berths are reserved for the shipping liner;

$\hat{b}$ — index of a dummy berth, used when there are not available berths in the reserved berth set B_p when a ship arrives at port p;

$\mathbb{Z}$ — set of integers;

$\mathbb{Z}_+$ — set of non-negative integers.

Parameters

V_ε — maximum revenue if all the transportation demand of OD pair ε is fulfilled;

$n_{ri\varepsilon}$ — maximum number of containers (TEUs) transported on leg (r, i) for the OD pair ε if all the demand is fulfilled;

$m_{risj\varepsilon}$ — maximum number of containers (TEUs) transshipped from the port of call (r, i) to (s, j) for the OD pair ε if all the demand is fulfilled; here $(r, i, s, j) \in Q$;

N_r^{Ship} — maximum number of ships that can be deployed on ship route r;

291 T_{ri}^{Leg} minimum sailing time on leg (r, i), which is determined by ships' maximum speed;

292 A_p^{Port} capacity (TEUs) of port p for storing the transshipped containers;

293 A_{ri}^{Vol} maximum volume capacity (in TEUs) of a ship on leg (r, i);

294 A_{ri}^{Load} maximum load capacity (in tons) of a ship on leg (r, i);

295 α probability limit of overload risk for ships (e.g., 1%, 0.1%);

296 $\tilde{c}_{ri\varepsilon u}$ stochastic parameter, the weight of the u^{th} container on the leg (r, i) for OD pair ε;

297 $\mu_{ri\varepsilon}$ the expected value for the random weight $\tilde{c}_{ri\varepsilon u}$;

298 $\sigma_{ri\varepsilon}$ the standard deviation for the random weight $\tilde{c}_{ri\varepsilon u}$;

299 C_r^{Opr} weekly operation cost of one ship deployed on ship route r;

300 C^{Hold} unit holding cost (USD per TEU per day) of transshipped containers storing at ports;

301 C_p^{Berth} penalty cost each time the dummy berth $\hat{b}$ is used at the port p;

302 C_p^{Yard} penalty cost for using one TEU extra yard space for transshipped containers in port p;

303 d_{ri} duration (days) of a ship dwells at the port of call (r, i);

304 $\bar{D}$ maximum value of d_{ri} for all the ports of call;

305 l_{ri} length of the leg (r, i);

306 k_{ri}, a_{ri} coefficients to calculate the unit fuel cost for travelling per nautical mile on leg (r, i);

307 g_{bw} equals one if berth b is available on day w in a week, otherwise equals zero;

$f_{\dot{w}\ddot{w}w}$ — equals one if day w is in time interval from day $\dot{w}$ to $\ddot{w}$; otherwise equals zero. Here $\dot{w}, \ddot{w}, w \in W, W = \{0, 1, 2, \cdots, 6\}$. For example, if $\dot{w} = 1, \ddot{w} = 3$, then $f_{\dot{w}\ddot{w}1} = f_{\dot{w}\ddot{w}2} = f_{\dot{w}\ddot{w}3} = 1$; if $\dot{w} = 3, \ddot{w} = 1$, then $f_{\dot{w}\ddot{w}3} = f_{\dot{w}\ddot{w}4} = f_{\dot{w}\ddot{w}5} = f_{\dot{w}\ddot{w}6} = f_{\dot{w}\ddot{w}0} = f_{\dot{w}\ddot{w}1} = 1$.

Decision variables

(1) Binary variables

η_{riw} — binary variable equal to one if and only if the ship arrives at the port of call (r, i) on day w of a week;

θ_{rib} — binary variable equal to one if and only if the ship uses berth b (including $\hat{b}$) in the port of call (r, i).

(2) General integer variables

β_r — number of ships deployed on ship route r; here $\beta_r \in \{1, 2, 3, \cdots, N_r^{Ship}\}$;

δ_{ri} — sailing time (days) of leg (r, i);

τ_{ri} — time (day) when a ship arrives at the port of call (r, i), where $i = 1, 2, 3, \cdots, |I_r| + 1$; $\tau_{r1} \in \{0, 1, 2, \cdots, 6\}$; $\tau_{r,|I_r|+1}$ denotes the time at which the ship completes a round trip journey;

ζ_{ri} — auxiliary variable associated with τ_{ri}, used to transform τ_{ri} into a day in one week;

γ_{risj} — arrival time difference in days of a ship visiting (r, i) and a ship visiting (s, j);

ξ_{risj} — auxiliary variable associated with γ_{risj} to transform γ_{risj} into an integer less than seven;

λ_{pw} — extra used yard space (TEUs) for storing transshipped containers at port p on day w.

(3) Continuous variables

π_ε — percentage of OD pair ε's transportation demand fulfilled by the shipping liner.

Mathematical model

The model is then as follows:

$$[\boldsymbol{M1}] \quad \text{Maximize Z} = \underbrace{\sum_{\varepsilon\in\Omega} V_\varepsilon \pi_\varepsilon}_{Revenue} - \underbrace{\sum_{r\in R} C_r^{Opr}\beta_r}_{Ship\ operation\ cost} - \underbrace{\sum_{r\in R}\sum_{i\in I_r} l_{ri}k_{ri}(l_{ri}/\delta_{ri})^{a_{ri}}}_{Fuel\ cost}$$

$$-\underbrace{C^{Hold}\sum_{(r,i,s,j)\in Q}\sum_{\varepsilon\in\Omega}\pi_\varepsilon m_{risj\varepsilon}\gamma_{risj}}_{Holding\ cost\ of\ transshipment} - \underbrace{\sum_{r\in R}\sum_{i\in I_r} C_{p_{ri}}^{Berth}\theta_{ri\hat{b}}}_{Berth\ cost\ for\ extra\ usage} - \underbrace{\sum_{p\in P}\sum_{w\in W} C_p^{Yard}\lambda_{pw}}_{Yard\ cost\ for\ extra\ usage} \tag{1}$$

subject to

$$1 \le \beta_r \le N_r^{Ship} \quad r\in R \tag{2}$$

$$0 \le \tau_{r1} \le 6 \quad r\in R \tag{3}$$

$$\delta_{ri} \ge T_{ri}^{Leg} \quad r\in R, i\in I_r \tag{4}$$

$$\tau_{r,i+1} = \tau_{ri} + d_{ri} + \delta_{ri} \quad r\in R, i\in I_r \tag{5}$$

$$\tau_{r,|I_r|+1} = \tau_{r1} + 7\beta_r \quad r\in R \tag{6}$$

$$\sum_{w\in W}\eta_{riw} = 1 \quad r\in R, i\in I_r \tag{7}$$

$$\tau_{ri} = \sum_{w\in W} w\eta_{riw} + 7\zeta_{ri} \quad r\in R, i\in I_r \tag{8}$$

$$0 \le \zeta_{ri} \le \beta_r - 1 \quad r\in R, i\in I_r \tag{9}$$

$$\tau_{sj} - \tau_{ri} + 7\xi_{risj} = \gamma_{risj} \quad (r,i,s,j)\in Q \tag{10}$$

$$0 \le \gamma_{risj} \le 6 \quad (r,i,s,j)\in Q \tag{11}$$

$$\sum_{b\in B_{p_{ri}}\bigcup\{\hat{b}\}} \theta_{rib} = 1 \quad r\in R, i\in I_r \tag{12}$$

$$\sum_{r\in R'_p}\sum_{v=1}^{\bar{D}}\sum_{i\in I'_{rp}:d_{ri}=v}\sum_{k=0}^{v-1}\theta_{rib}\eta_{r,i,(w-k+7)\mod 7} \le g_{bw} \quad p\in P, b\in B_p, w\in W \tag{13}$$

$$(\sum_{(r,i,s,j)\in Q_p}\sum_{\varepsilon\in\Omega}\pi_\varepsilon m_{risj\varepsilon}\sum_{\dot{w},\ddot{w}\in W}\eta_{ri\dot{w}}\eta_{sj\ddot{w}}f_{\dot{w}\ddot{w}w}-A_p^{Port})^+=\lambda_{pw}\quad p\in P,w\in W \tag{14}$$

$$\sum_{\varepsilon\in\Omega}\pi_\varepsilon n_{ri\varepsilon}\le A_{ri}^{Vol}\quad r\in R,i\in I_r \tag{15}$$

$$Prob(\sum_{\varepsilon\in\Omega}\sum_{u=1}^{\lceil\pi_\varepsilon n_{ri\varepsilon}\rceil}\tilde{c}_{ri\varepsilon u}>A_{ri}^{Load})\le\alpha\quad r\in R,i\in I_r \tag{16}$$

$$0\le\pi_\varepsilon\le1\quad\varepsilon\in\Omega \tag{17}$$

$$\beta_r\in\mathbb{Z}_+\quad r\in R \tag{18}$$

$$\tau_{ri}\in\mathbb{Z}_+\cup\{0\}\quad r\in R,i\in I_r\cup\{|I_r|+1\} \tag{19}$$

$$\eta_{riw}\in\{0,1\}\quad r\in R,i\in I_r,w\in W \tag{20}$$

$$\delta_{ri}\in\mathbb{Z}_+\cup\{0\}\quad r\in R,i\in I_r \tag{21}$$

$$\zeta_{ri}\in\mathbb{Z}_+\cup\{0\}\quad r\in R,i\in I_r \tag{22}$$

$$\gamma_{risj}\in\mathbb{Z}_+\cup\{0\}\quad(r,i,s,j)\in Q \tag{23}$$

$$\xi_{risj}\in\mathbb{Z}\quad(r,i,s,j)\in Q \tag{24}$$

$$\theta_{rib}\in\{0,1\}\quad r\in R,i\in I_r,b\in B_{p_{ri}}\cup\{\hat{b}\} \tag{25}$$

$$\lambda_{pw}\ge0\quad p\in P,w\in W. \tag{26}$$

The objective (1) is to maximize the revenue, minus the five types of cost described in Section 3. Constraints (2) state that at least one ship and at most N_r^{Ship} ships should be deployed on each route. Constraints (3) ensure the start time of each route (service) occurs in the first week. Constraints (4) relate to the minimum required sailing time T_{ri}^{Leg} for each leg, which depends on the maximum speed of ships. Constraints (5) link the arrival time τ_{ri} of a port of call with the arrival time $\tau_{r,i+1}$ of the next port of call on a route. Constraints (6) guarantee that the total number of days $\tau_{r,|I_r|+1}-\tau_{r1}$ for a ship completing its travel on a route is the number of ships deployed on the route times seven, because all the services follow weekly arrival pattern and one week has seven days. Constraints (7)–(9) link the binary variable η_{riw} and the integer variable τ_{ri}, both of which denote the arrival time of the i^{th}

port of call on ship route r. The difference is that τ_{ri} denotes the arrival time on a universal time axis, while μ_{ri}^{w} denotes the arrival time in one of the seven days in a week. The former is from the perspective of port arrival time in one ship's itinerary (e.g., day 2 at port 1, day 11 at port 2), while the latter is from the perspective of the port arrival time of a fleet of ships deployed on a route (e.g., Mon at port 1, Wed at port 2). Constraints (10)–(11) transfer the absolute time gap (days) $\tau_{sj} - \tau_{ri}$ between two ports of call (r, i) and (s, j) to a time difference γ_{risj} in days within one week. Similarly, the former is from the perspective of port arrival time in two ship's itineraries for two routes (e.g., a ship in route 1 arrives at port p on day 2, a ship in route 2 arrives at the port p on day 11, and the absolute time difference is nine days), while the latter is from the perspective of the port arrival time of two fleets of ships deployed on two routes (e.g., route 1's fleet arrives at the port on Mon, route 2's fleet arrives at the port on Wed, and the time difference is two days, which is the waiting time for transshipment from route 1 to route 2). Constraints (12) guarantee that each port of call of a route should be assigned a berth (one of reserved berths or the dummy berth $\hat{b}$). The berth availability limitation is ensured by Constraints (13), which are not straightforward and will be explained later. Constraints (14) calculate the extra used yard space (TEUs) for storing transshipped containers at each port on each day. Constraints (15) define the limitation of the ship capacity with respect to its available space during each leg. Constraints (16) mean that the overload probability is lower than a threshold α. Constraints (17)–(26) state the ranges of the defined decision variables.

More explanations are required for Constraints (13). In the simplest case where all ships dwell at ports for only one day, the left-hand side of the constraint is $\sum_{r \in R'_p} \sum_{i \in I'_{rp}} \theta_{rib} \eta_{riw}$, which denotes whether or not one of the reserved berths b is used by a ship on day w in a week. This value should not be greater than g_{bw}, which is the availability of the berth. If some ships dwell at a port for one day (i.e., $d_{ri} = 1$), and some ships dwell for two days (i.e., $d_{ri} = 2$), the calculation on whether or not berth b is used by the i^{th} port of call on ship route r is as follows: (1) if $w = 1, 2, 3, \cdots, 6$, then $\sum_{r \in R'_p} [\sum_{i \in I'_{rp}: d_{ri}=1} \theta_{rib} \eta_{riw} + \sum_{i \in I'_{rp}: d_{ri}=2} (\theta_{rib} \eta_{r,i,w-1} + \theta_{rib} \eta_{riw})]$; (2) if $w = 0$, then $\sum_{r \in R'_p} [\sum_{i \in I'_{rp}: d_{ri}=1} \theta_{rib} \eta_{riw} + \sum_{i \in I'_{rp}: d_{ri}=2} (\theta_{rib} \eta_{r,i,w-1+7} + \theta_{rib} \eta_{riw})]$. In what follows, subscripts $w - 1$ and $w - 1 + 7$ are interpreted as $(w - 1 + 7)$ mod 7. Then suppose the ships' dwell time can be one, two,$\cdots$, or at most $\bar{D}$ days, then the above formula becomes $\sum_{r\in R'_p}\sum_{v=1}^{\bar{D}}\sum_{i\in I'_{rp}:d_{ri}=v}\sum_{k=0}^{v-1}(\theta_{rib}\,\eta_{r,i,(w-k+7)\mod 7})$. This value does not exceed g_{bw} by Constraints (13).

5. Linearization of the model

The above model $[\boldsymbol{M1}]$ is an optimization problem with integer decision variables and non-linear terms that are non-convex. It is difficult to solve it using off-the-shelf solvers because (i) it contains a large number of discrete variables and (ii) it has a non-linear objective function and non-linear constraints. To solve this model, we first linearizate it, and we then develop a sequential optimization algorithm.

5.1. Linearization of Objective (1)

Objective (1) contains a non-linear part $\sum_{r\in R}\sum_{i\in I_r} l_{ri}k_{ri}(l_{ri}/\delta_{ri})^{a_{ri}}$, which can be rewritten as $\sum_{r\in R}\sum_{i\in I_r} l_{ri}k_{ri}l_{ri}^{a_{ri}}\delta_{ri}^{-a_{ri}}$. The key is to transform $\delta_{ri}^{-a_{ri}}$ into a linear form. We adopt the linearization method used by Wang et al. (2013). We first redefine δ_{ri} as a new binary variable δ'_{rit}, which denotes whether or not the sailing time for the i^{th} leg of ship route r equals t days, $t\in T$, where T is the set of integers denoting the possible sailing times (in days) for all legs; for example $T\in\{1,\cdots,15\}$. The non-linear form $\delta_{ri}^{-a_{ri}}$ can then be replaced with $\sum_{t\in T}\delta'_{rit}t^{-a_{ri}}$, subject to $\sum_{t\in T}\delta'_{rit}$=1 for all $r\in R, i\in I_r$.

Objective (1) contains another non-linear part $\sum_{(r,i,s,j)\in Q}\pi_\varepsilon m_{risj\varepsilon}\gamma_{risj}$, which can be linearized as follows Alharbi et al. (2015). We first transform the integer variable γ_{risj} into a binary variable. Since $\gamma_{risj}\in W$, we redefine γ_{risj} as a binary variable γ'_{risjw}, equal to one if and only if the time gap between ports of call (r,i) and (s,j) is w days. Then γ_{risj} is replaced with $\sum_{w\in W} w\gamma'_{risjw}$, subject to $\sum_{w\in W}\gamma'_{risjw}=1$ for all $(r,i,s,j)\in Q$. Here both π_ε and γ'_{risjw} are binary variables; therefore, the value of M is 1.

Based on the above linearization, Objective (1) becomes

$$\text{Maximize Z} = \underbrace{\sum_{\varepsilon\in\Omega} V_\varepsilon \pi_\varepsilon}_{Revenue} - \underbrace{\sum_{r\in R} C_r^{Opr}\beta_r}_{Ship\ operation\ cost} \underbrace{- \sum_{r\in R}\sum_{i\in I_r} l_{ri}k_{ri}l_{ri}^{a_{ri}} \sum_{t\in T}\delta'_{rit}t^{-a_{ri}}}_{Fuel\ cost}$$
$$\underbrace{- C^{Hold}\sum_{(r,i,s,j)\in Q}\sum_{w\in W} m_{risj\varepsilon} w \varrho_{risjw\varepsilon}}_{Holding\ cost\ of\ transshipment} - \underbrace{\sum_{r\in R}\sum_{i\in I_r} C_{p_{ri}}^{Berth}\theta_{ri\hat{b}}}_{Berth\ cost\ for\ extra\ usage} - \underbrace{\sum_{p\in P, w\in W} C_p^{Yard}\lambda_{pw}}_{Yard\ cost\ for\ extra\ usage} \tag{27}$$

The newly defined variables and constraints needed for this linearization are summarized as follows:

Newly defined indices, sets and parameters:

t	index of the number of days;
T	set of possible numbers of days for a leg's sailing time, $T = \{1, \cdots, \lvert T\rvert\}$;
M	a sufficiently large positive number.

Newly defined variables:

δ'_{rit}	a binary variable equal to one if and only if the sailing time of the leg (r, i) is t;
γ'_{risjw}	a binary variable equal to one if and only if the time gap between the ports of call (r, i) and (s, j) (i.e., γ_{risj}) is w days;
$\varrho_{risjw\varepsilon}$	continuous variable equal to $\pi_\varepsilon\gamma'_{risjw}$ if $\gamma'_{risjw} = 1$; otherwise zero.

Newly defined constraints:

Constraints (11) are removed. Constraints (5), (10), (21), (23) are replaced with the following four constraints, respectively.

$$\tau_{r,i+1} = \tau_{ri} + d_{ri} + \sum_{t\in T} t\delta'_{rit} \quad r \in R, i \in I_r \tag{28}$$

$$\tau_{sj} - \tau_{ri} + 7\xi_{risj} = \sum_{w\in W} w\gamma'_{risjw} \quad (r, i, s, j) \in Q \tag{29}$$

$$\delta'_{rit} \in \{0, 1\} \quad r \in R, i \in I_r, t \in T \tag{30}$$

$$\gamma'_{risjw} \in \{0,1\} \quad (r,i,s,j) \in Q, w \in W. \tag{31}$$

In addition, four new constraints are defined:

$$\sum_{t \in T} \delta'_{rit} = 1 \quad r \in R, i \in I_r \tag{32}$$

$$\sum_{w \in W} \gamma'_{risjw} = 1 \quad (r,i,s,j) \in Q \tag{33}$$

$$0 \le \varrho_{risjw\varepsilon} \le 1 \quad (r,i,s,j) \in Q, w \in W, \varepsilon \in \Omega. \tag{34}$$

5.2. Linearization of Constraints (13)

Constraints (13) contain a non-linear part $\theta_{rib}\eta_{r,i,(w-k+7) \mod 7}$, which is the product of two binary variables. Following the method used by Yi et al. (2018), we define a new binary variable φ_{ribw} to replace the non-linear part.

Newly defined variables:

φ_{ribw} binary variable equal to one if and only if the ship arrives at the berth b on the day w of a week in the i^{th} port of call on ship route r.

Then Constraints (13) become

$$\sum_{r \in R'_p} \sum_{v=1}^{\bar{D}} \sum_{i \in I'_{rp}: d_{ri}=v} \sum_{k=0}^{v-1} \theta_{r,i,b,(w-k+7) \mod 7} \le g_{bw} \quad p \in P, b \in B_p, w \in W. \tag{35}$$

In addition, some more constraints need to be defined so that the newly defined variable φ_{ribw} can replace the function of $\theta_{rib}\eta_{r,i,(w-k+7) \mod 7}$.

$$\varphi_{ribw} \ge \theta_{rib} + \eta_{riw} - 1 \quad r \in R, i \in I_r, b \in B_{p_{ri}}, w \in W \tag{36}$$

$$\varphi_{ribw} \le \theta_{rib} \quad r \in R, i \in I_r, b \in B_{p_{ri}}, w \in W \tag{37}$$

$$\varphi_{ribw} \le \eta_{riw} \quad r \in R, i \in I_r, b \in B_{p_{ri}}, w \in W \tag{38}$$

$$\varphi_{ribw} \in \{0,1\} \quad r \in R, i \in I_r, b \in B_{p_{ri}}, w \in W. \tag{39}$$

5.3. Linearization of Constraints (14)

Constraints (14) contain the product of three variables $\pi_\varepsilon, \eta_{ri\ddot{w}}$ and $\eta_{sj\ddot{w}}$, In addition, the form $\lambda_{pw} = (\cdot)^+$ is also non-linear. In the first case, we use an approach similar to that of Section 5.2 to handle it. This approach was used by Wang and Meng (2012). We define some more decision variables and constraints:

Newly defined variables:

$\psi_{risj\dot{w}\ddot{w}}$ binary variable equal to one if and only if both variables $\eta_{ri\dot{w}}$ and $\eta_{sj\ddot{w}}$ are equal to one;

$\phi_{risj\dot{w}\ddot{w}\varepsilon}$ binary variable equal to π_ε if and only if $\psi_{risj\dot{w}\ddot{w}} = 1$.

Then Constraints (14) become

$$\lambda_{pw} = (\sum_{(r,i,s,j)\in Q_p} \sum_{\varepsilon\in\Omega} \sum_{\dot{w},\ddot{w}\in W} m_{risj\varepsilon} f_{\dot{w}\ddot{w}w} \phi_{risj\dot{w}\ddot{w}\varepsilon} - A_p^{Port})^+ \quad p \in P, w \in W. \tag{40}$$

In addition, some more constraints need to be defined as follows so that the newly defined variable $\psi_{risj\dot{w}\ddot{w}}$ can replace the function of $\eta_{ri\dot{w}}\eta_{sj\ddot{w}}$:

$$\psi_{risj\dot{w}\ddot{w}} \geq \eta_{ri\dot{w}} + \eta_{sj\ddot{w}} - 1 \quad (r,i,s,j) \in Q; \dot{w}, \ddot{w} \in W \tag{41}$$

$$\psi_{risj\dot{w}\ddot{w}} \leq \eta_{ri\dot{w}} \quad (r,i,s,j) \in Q; \dot{w}, \ddot{w} \in W \tag{42}$$

$$\psi_{risj\dot{w}\ddot{w}} \leq \eta_{sj\ddot{w}} \quad (r,i,s,j) \in Q; \dot{w}, \ddot{w} \in W \tag{43}$$

$$\psi_{risj\dot{w}\ddot{w}} \in \{0, 1\} \quad (r,i,s,j) \in Q; \dot{w}, \ddot{w} \in W \tag{44}$$

$$\phi_{risj\dot{w}\ddot{w}\varepsilon} \geq \pi_\varepsilon + (\psi_{risj\dot{w}\ddot{w}} - 1)M \quad (r,i,s,j) \in Q; \dot{w}, \ddot{w} \in W, \varepsilon \in \Omega \tag{45}$$

$$0 \leq \phi_{risj\dot{w}\ddot{w}\varepsilon} \leq 1 \quad (r,i,s,j) \in Q; \dot{w}, \ddot{w} \in W, \varepsilon \in \Omega. \tag{46}$$

For the non-linear part $\lambda_{pw} = (\cdot)^+$, we adopt the linearization method used by Wang and Meng (2015). We define two more non-negative variables λ_{pw}^+ and λ_{pw}^-, and Constraints (40) are changed into

$$\sum_{(r,i,s,j)\in Q_p} \sum_{\varepsilon\in\Omega} \sum_{\dot{w},\ddot{w}\in W} m_{risj\varepsilon} f_{\dot{w}\ddot{w}w} \phi_{risj\dot{w}\ddot{w}\varepsilon} - A_p^{Port} = \lambda_{pw}^+ - \lambda_{pw}^- \quad p \in P, w \in W. \tag{47}$$

Then Constraints (26) are replaced with

$$\lambda_{pw}^+, \lambda_{pw}^- \geq 0 \quad p \in P, w \in W. \tag{48}$$

Moreover, Objective (27) is further restated by replacing λ_{pw} with λ_{pw}^{+}. Then the final version of the objective becomes

$$\text{Maximize Z} = \underbrace{\sum_{\varepsilon\in\Omega} V_{\varepsilon}\pi_{\varepsilon}}_{\textit{Revenue}} - \underbrace{\sum_{r\in R} C_r^{Opr}\beta_r}_{\textit{Ship operation cost}} - \underbrace{\sum_{r\in R}\sum_{i\in I_r} l_{ri}k_{ri}l_{ri}^{a_{ri}}\sum_{t\in T}\delta'_{rit}t^{-a_{ri}}}_{\textit{Fuel cost}}$$
$$- \underbrace{C^{Hold}\sum_{\langle r,i,s,j\rangle\in Q}\sum_{w\in W} m_{risj\varepsilon}w\varrho_{risjw\varepsilon}}_{\textit{Holding cost of transshipment}} - \underbrace{\sum_{r\in R}\sum_{i\in I_r} C_{p_{ri}}^{Berth}\theta_{ri\hat{b}}}_{\textit{Berth cost for extra usage}} - \underbrace{\sum_{p\in P, w\in W} C_p^{Yard}\lambda_{pw}^{+}}_{\textit{Yard cost for extra usage}}. \tag{49}$$

Lemma 1. *Because the weights of the* $\lceil\pi_{\varepsilon}n_{ri\varepsilon}\rceil$ *containers are independent and identically distributed random variables with expected values* $u_{ri\varepsilon}$ *and variances* $\sigma_{ri\varepsilon}^2$*, the classical central limit theorem (CLT) states that since* $\lceil\pi_{\varepsilon}n_{ri\varepsilon}\rceil$ *is very large, the distribution of the total weight* $\sum_{u=1}^{\lceil\pi_{\varepsilon}n_{ri\varepsilon}\rceil}\tilde{c}_{ri\varepsilon u}$ *is approximately normal with mean* $\lceil\pi_{\varepsilon}n_{ri\varepsilon}\rceil\mu_{ri\varepsilon}$ *and variance* $\lceil\pi_{\varepsilon}n_{ri\varepsilon}\rceil\sigma_{ri\varepsilon}^2$*.*

Lemma 2. *When* $\lceil\pi_{\varepsilon}n_{ri\varepsilon}\rceil$ *is very large,* $r\in R, i\in I_r, \varepsilon\in\Omega$*, since the containers weights are independent, the total weight* $\sum_{\varepsilon\in\Omega}\sum_{u=1}^{\lceil\pi_{\varepsilon}n_{ri\varepsilon}\rceil}\tilde{c}_{ri\varepsilon u}$ *of all the carried containers approximately follows a normal distribution* $N(\sum_{\varepsilon\in\Omega}\lceil\pi_{\varepsilon}n_{ri\varepsilon}\rceil\mu_{ri\varepsilon}, \sum_{\varepsilon\in\Omega}\lceil\pi_{\varepsilon}n_{ri\varepsilon}\rceil\sigma_{ri\varepsilon}^2)$*.*

In reality, the number of containers is large. According to Lemma (2), Constraints (16) can be approximated by

$$\sum_{\varepsilon\in\Omega}\lceil\pi_{\varepsilon}n_{ri\varepsilon}\rceil\mu_{ri\varepsilon} + z_{1-\alpha}\Big(\sum_{\varepsilon\in\Omega}\lceil\pi_{\varepsilon}n_{ri\varepsilon}\rceil\sigma_{ri\varepsilon}^2\Big)^{\frac{1}{2}} \le A_{ri}^{Load} \quad r\in R, i\in I_r, \tag{50}$$

where $z_{1-\alpha}$ is the $100(1-\alpha)$ percentile of the standard normal distribution.

Proposition 1. *The left-hand sides of Constraints* (50) *are in general non-convex in* π_{ε}*.*

Proof. To prove the proposition, we just need to provide a non-convex example. Consider a simple case with only one OD pair, i.e., $|\Omega| = 1$. Suppose that for this OD pair ε, we have $\mu = 1$, $\sigma^2 = 0.25$. Suppose further than $z = 1$ and $n_{ri\varepsilon} = 10$. Then the left-hand side of the constraint becomes $10\pi + 0.25\sqrt{10\pi}$. Consider three values of

π: $\pi_1 = 0$, $\pi_2 = 1$, and $\pi_3 = 2$. Then $10\pi_1+0.25\sqrt{10\pi_1} = 0$, $10\pi_2+0.25\sqrt{10\pi_2} = 1.25$, $10\pi_3+0.25\sqrt{10\pi_3} = 2.35$. In other words, $\pi_2 = (\pi_1+\pi_3)/2 = (0+2)/2 = 1$, however, $10\pi_2 + 0.25\sqrt{10\pi_2} > (10\pi_1 + 0.25\sqrt{10\pi_1} + 10\pi_3 + 0.25\sqrt{10\pi_3})/2$. Therefore, the left-hand side of the constraint in this case is non-convex. $\square$

In order to handle the non-convex Constraints (50), we propose a second-order cone programming (SOCP)-based algorithm, which will be elaborated in Section 6.

6. Algorithmic strategy

We now present an SOCP-based algorithm to handle non-convex constraints in the model. A dynamic linearization algorithm and a tabu search algorithm are applied to solve the model under different scales of route networks.

6.1. SOCP transformation

We use SOCP to transfer Constraints (50) to a convex one. We first define a new binary variable $\kappa_{ri\varepsilon h}$ to represent the integer $\lceil \pi_\varepsilon n_{ri\varepsilon} \rceil$:

$$\lceil \pi_\varepsilon n_{ri\varepsilon} \rceil = \sum_{h=0}^{H_{ri\varepsilon}} 2^h \kappa_{ri\varepsilon h} \qquad r \in R, i \in I_r, \varepsilon \in \Omega, h = 0, 1, \cdots, H_{ri\varepsilon} \tag{51}$$

$$\sum_{h=0}^{H_{ri\varepsilon}} 2^h \kappa_{ri\varepsilon h} \leq n_{ri\varepsilon} \qquad r \in R, i \in I_r, \varepsilon \in \Omega, h = 0, 1, \cdots, H_{ri\varepsilon} \tag{52}$$

$$\kappa_{ri\varepsilon h} \in \{0, 1\} \qquad r \in R, i \in I_r, \varepsilon \in \Omega, h = 0, 1, \cdots, H_{ri\varepsilon}, \tag{53}$$

where $H_{ri\varepsilon} := \lfloor \log_2 n_{ri\varepsilon} \rfloor$. Then Constraints (50) become

$$\sum_{\varepsilon\in\Omega} \mu_{ri\varepsilon} \sum_{h=0}^{H_{ri\varepsilon}} 2^h \kappa_{ri\varepsilon h} + z_{1-\alpha}\Big(\sum_{\varepsilon\in\Omega} \sigma_{ri\varepsilon}^2 \sum_{h=0}^{H_{ri\varepsilon}} 2^h \kappa_{ri\varepsilon h}\Big)^{\frac{1}{2}} \leq A_{ri}^{Load} \qquad r \in R, i \in I_r. \tag{54}$$

Since $\kappa_{ri\varepsilon h}$ is binary, we have $\kappa_{ri\varepsilon h} = \kappa_{ri\varepsilon h}^2$. Using this property, Constraints (54) become

$$\Big(\sum_{\varepsilon\in\Omega} \sum_{h=0}^{H_{ri\varepsilon}} 2^h \sigma_{ri\varepsilon}^2 \kappa_{ri\varepsilon h}^2\Big)^{\frac{1}{2}} \leq \Big(A_{ri}^{Load} - \sum_{\varepsilon\in\Omega} \sum_{h=0}^{H_{ri\varepsilon}} 2^h \mu_{ri\varepsilon} \kappa_{ri\varepsilon h}\Big)/z_{1-\alpha} \qquad r \in R, i \in I_r. \tag{55}$$

Constraints (55) are convex and now the following [$\boldsymbol{M2}$] is a mixed integer SOCP (MISOCP) model, which can be solved by off-the-shelf solvers such as CPLEX.

[$\boldsymbol{M2}$] An MISOCP model: Objective (49)

subject to Constraints (2)–(4), (6)–(9), (11)–(12), (15)–(20), (22), (24)–(25), (28)–(39), (41)–(48), (52)–(53), (55).

6.2. Dynamic linearization for solving [$\boldsymbol{M2}$]

We propose solving the MISOCP model [$\boldsymbol{M2}$] by integer linear programming. The core idea is as follows: since Constraints (55) are convex, if we know an infeasible solution $\check{\boldsymbol{y}} := (\check{\kappa}_{ri\varepsilon h}, r \in R, i \in I_r, \varepsilon \in \Omega, h = 0, 1, 2, \cdots, H_{ri\varepsilon})$ that violates the non-linear Constraints (55), we can linearize the left-hand side $(\sum_{\varepsilon\in\Omega}\sum_{h=0}^{H_{ri\varepsilon}} 2^h \sigma_{ri\varepsilon}^2 \kappa_{ri\varepsilon h}^2)^{\frac{1}{2}}$ of the constraint at $\check{\boldsymbol{y}}$. Note that $\frac{\partial(\sum_{\varepsilon\in\Omega}\sum_{h=0}^{H_{ri\varepsilon}} 2^h\sigma_{ri\varepsilon}^2\kappa_{ri\varepsilon h}^2)^{\frac{1}{2}}}{\partial \kappa_{ri\varepsilon h}} = \frac{2^h\sigma_{ri\varepsilon}^2\check{\kappa}_{ri\varepsilon h}}{(\sum_{\varepsilon\in\Omega}\sum_{h=0}^{H_{ri\varepsilon}} 2^h\sigma_{ri\varepsilon}^2\check{\kappa}_{ri\varepsilon h}^2)^{\frac{1}{2}}}$ at $\check{\boldsymbol{y}}$. Hence, we can add the resulting linear constraint to the model in order to cut off the infeasible solution $\check{\boldsymbol{y}}$, as well as some other infeasible solutions. We propose the following Algorithm 1 to solve model [$\boldsymbol{M2}$] and we then prove its correctness.

Algorithm 1 Dynamic linearization algorithm for solving [$\boldsymbol{M2}$]

Step 1. Define a set Ψ of generated intermediate infeasible solutions of $\boldsymbol{y} := (\kappa_{ri\varepsilon h}, r \in R, i \in I_r, \varepsilon \in \Omega, h = 0, 1, 2, \cdots, H_{ri\varepsilon})$. Initialize $\Psi \leftarrow \varnothing$.

Step 2. Solve model [$\boldsymbol{M3}$] whose objective function is Eq. (49) subject to Constraints (2)–(4), (6)–(9), (11)–(12), (15)–(20), (22), (24)–(25), (28)–(39), (41)–(48), (52)–(53) and the following constraints:

$$\frac{\sum_{\varepsilon\in\Omega}\sum_{h=0}^{H_{ri\varepsilon}} 2^h\sigma_{ri\varepsilon}^2\check{\kappa}_{ri\varepsilon h}(\kappa_{ri\varepsilon h} - \check{\kappa}_{ri\varepsilon h})}{(\sum_{\varepsilon\in\Omega}\sum_{h=0}^{H_{ri\varepsilon}} 2^h\sigma_{ri\varepsilon}^2\check{\kappa}_{ri\varepsilon h}^2)^{\frac{1}{2}}} + (\sum_{\varepsilon\in\Omega}\sum_{h=0}^{H_{ri\varepsilon}} 2^h\sigma_{ri\varepsilon}^2\check{\kappa}_{ri\varepsilon h}^2)^{\frac{1}{2}} \leq \frac{A_{ri}^{Load} - \sum_{\varepsilon\in\Omega}\sum_{h=0}^{H_{ri\varepsilon}} 2^h\mu_{ri\varepsilon}\kappa_{ri\varepsilon h}}{z_{1-\alpha}}, \check{\boldsymbol{y}} \in \Psi, r \in R, i \in I_r. \tag{56}$$

Let $\hat{\boldsymbol{y}}$ be the optimal solution to model [$\boldsymbol{M3}$].

Step 3. Check whether $\hat{\boldsymbol{y}}$ satisfies Constraints (55). If yes, then $\hat{\boldsymbol{y}}$ is the optimal solution to [$\boldsymbol{M2}$] and stop. Otherwise, set $\Psi \leftarrow \Psi \cup \{\hat{\boldsymbol{y}}\}$ and go to **Step 1**.

Proposition 2. *No solution will be generated twice in Algorithm 1.*

Proof. If a generated solution $\hat{\boldsymbol{y}}$ is feasible with respect to Constraints (55), then the algorithm stops and hence it will not be generated twice. If it is infeasible, then it will become an element of Ψ at the next iteration and we denote it by $\check{\boldsymbol{y}}$. Since $\check{\boldsymbol{y}}$ is infeasible, we have

$$(\sum_{\varepsilon\in\Omega}\sum_{h=0}^{H_{ri\varepsilon}} 2^h \sigma_{ri\varepsilon}^2 \check{\kappa}_{ri\varepsilon h}^2)^{\frac{1}{2}} > \frac{A_{ri}^{Load} - \sum\limits_{\varepsilon\in\Omega}\sum\limits_{h=0}^{H_{ri\varepsilon}} 2^h \mu_{ri\varepsilon}\check{\kappa}_{ri\varepsilon h}}{z_{1-\alpha}} \qquad r\in R, i\in I_r. \tag{57}$$

Inequality (57) implies that $\hat{\boldsymbol{y}} = \check{\boldsymbol{y}}$ violates the added Constraints (56). Hence, $\hat{\boldsymbol{y}} = \check{\boldsymbol{y}}$ will not be generated again. $\square$

Proposition 3. *Algorithm 1 terminates in a finite number of iterations.*

Proof. Since all $\kappa_{ri\varepsilon h}$ variables are binary for $r \in R, i \in I_r, \varepsilon \in \Omega$, and $h = 0, 1, \cdots, H_{ri\varepsilon}$, the number of solutions feasible to Constraints (2)–(4), (6)–(9), (11)–(12), (15)–(20), (22), (24)–(25), (28)–(39), (41)–(48) and (52)–(53) is at most $2^{\sum_{r\in R}\sum_{i\in I_r}\sum_{\varepsilon\in\Omega}(1+H_{ri\varepsilon})}$. Proposition 2 implies that at least one solution is excluded at each iteration. Hence, Algorithm 1 terminates in at most $2^{\sum_{r\in R}\sum_{i\in I_r}\sum_{\varepsilon\in\Omega}(1+H_{ri\varepsilon})}$ iterations. $\square$

Proposition 4. *An optimal solution is obtained when Algorithm 1 terminates.*

Proof. Model [$\boldsymbol{M3}$] is a relaxation of the original model [$\boldsymbol{M2}$], because the linearization on the left-hand side of inequality (56) underestimates the convex function $(\sum_{\varepsilon\in\Omega}\sum_{h=0}^{H_{ri\varepsilon}} 2^h \sigma_{ri\varepsilon}^2 \kappa_{ri\varepsilon h}^2)^{\frac{1}{2}}$. Since [$\boldsymbol{M2}$] and [$\boldsymbol{M3}$] have the same objective function, the value of $\hat{\boldsymbol{y}}$ generated in **Step 1** is at least equal to that of the optimal value of [$\boldsymbol{M2}$]. If $\hat{\boldsymbol{y}}$ is feasible for [$\boldsymbol{M2}$], then the objective value of the feasible solution $\hat{\boldsymbol{y}}$ to [$\boldsymbol{M2}$] is equal to an upper bound (the optimal objective value of [$\boldsymbol{M3}$]), meaning that $\hat{\boldsymbol{y}}$ is optimal for [$\boldsymbol{M2}$]. $\square$

6.3. Tabu search algorithm for solving [$\boldsymbol{M2}$]

We now propose a tabu search algorithm to solve [$\boldsymbol{M2}$]. Tabu search algorithm, introduced by Glover (1986), is an adaptive local iterative search that operates within a solution space. It moves from one solution to another and diversifies solutions so as to find a better one (Vivaldini et al., 2016). At each iteration, the search process is applied to explore the neighborhood of the current optimal solution. Tabu search algorithm has often been applied to problems solving in the maritime industry. Cordeau et al. (2005) applied a tabu search algorithm to the berth allocation problem (BAP). Tirado et al. (2013) solved a dynamic and stochastic cargo transportation problem by means of tabu search. Nikolopoulou et al. (2017) used tabu search to compare two kinds of cargo transportation methods in the shipping industry.

6.3.1. Local optimization using tabu search

Given a neighborhood structure ($N(p_c)$) and an initial solution p, the tabu search algorithm iteratively replaces the incumbent solution p_c by a best eligible neighbor solution ($\hat{p} \in N(p_c)$) until a stopping criterion is met, i.e., the current optimal solution p^* has not been improved for T_{max} consecutive iterations. At each iteration, the best movement is recorded in the tabu list to prevent the reverse movement in the next iterations. A movement is eligible if it is not in the tabu list or if it results in a better solution than the current optimal solution. The general tabu search framework is described in Algorithm 2 and the details are explained in subsequent sections.

6.3.2. Population initialization

The population initialization is obtained by generating 10 random solutions using a uniform probability distribution. The component $x_{r,i}$ of each solution is randomly assigned a value from $[T_{ri}^{min}, T_{ri}^{max}]$, where $r \in R, i \in I_r \backslash \{|I_r|\}$. The minimum value T_{ri}^{min} and maximum value T_{ri}^{max} refer to the minimum sailing time and the maximum sailing time of each leg of each route according to the maximum speed and minimum sailing speed, respectively. Moveover, we should guarantee that the sum of the sailing time on each leg plus the duration time at each port is the multiple of seven by adjusting the time of the last component $x_{r,|I_r|}$, where $r \in R$. We then select the best solution p_0 among the 10 random solutions as the initial solution p.

6.3.3. Neighborhood structure and movement

The neighborhood structure is the crucial component of the algorithm. The neighborhood $N(p_c)$ contains all solutions in which the value of one component is changed to its immediate adjacent values. The neighborhood $N(p_c)$ is defined by the one-change movement operator which consists of changing the current solution p_c of a single component either from $x_{r,i}$ to $x_{r,i}+1$ or from $x_{r,i}$ to $x_{r,i}-1$, where $r \in R, i \in I_r \backslash \{|I_r|\}$. Meanwhile, we should guarantee that the sum of the sailing time on each leg plus the duration time on each port is the multiple of seven by adjusting the time of the last component $x_{r,|I_r|}$, where $r \in R$. Given an incumbent solution p_c, the one-change movement operator is composed of all possible solutions that can be obtained by applying the one-change movement to p_c.

6.3.4. Sorted candidate solutions

The candidate solutions $(SCS_1, SCS_2, \cdots, SCS_l, \cdots, SCS_{C_{max}})$ are generated after the movement is achieved, where C_{max} is the number of candidate solutions, and the fitness values of the candidate solutions $(SCF_1, SCF_2, \cdot\cdot\cdot, SCF_l, \cdot\cdot\cdot, SCF_{C_{max}})$ are sorted in non-increasing order by using the bubble sorting method. Bubble sorting is a simple sort algorithm. It compares two adjacent elements SCF_l and SCF_{l+1}. If SCF_l is less than SCF_{l+1}, which means their order is opposite, the two adjacent element positions are exchanged and their corresponding candidate solution positions are also updated. If SCF_l is greater than or equal to SCF_{l+1}, no transformation operation is taken.

Algorithm 2 Tabu search algorithm for the fleet deployment and demand fulfillment for container shipping liners

Input: parameters T_{ri}^{min}, T_{ri}^{max}, T_{max}, C_{max}, L_{max}, D_{max}, GBF //P_{ri}^{min}, P_{ri}^{max} are the minimum and maximum values of the initial solution with respect to r, i; T_{max} is the given number of iterations for t; C_{max} is the number of candidate solutions; L_{max} is the tabu list size; D_{max} is the given number of iterations for d; GBF is the best fitness of all solutions

Output: the objective value

1: initialization: initial solution $p = p_0$ //p_0 is the best solution among the t random solutions
2: neighborhood structure $N(p)$
3: tabu list $L = \varnothing$
4: $GBF \leftarrow f(p)$
5: $f(p^*) \leftarrow GBF$ //p^* is the current optimal solution
6: $p_c \leftarrow p_0$ //p_c is the incumbent solution
7: $d \leftarrow 0$ //d counts the consecutive number of iterations in which p^* is not improved
8: $t \leftarrow 0$ //t counts the consecutive number of iterations where p^* is not updated
9: **while** $t < T_{max}$ **do**
10: find a best solution $\hat{p} \in \text{argmax}_{N(p_c)} [f(p_c)]$ //$\hat{p}$ keeps the best solution found

11: record the movement in the tabu list
12: **if** $\hat{p} \notin L$ **then**
13: move to the best neighbor $p_c \leftarrow \hat{p}$
14: update tabu list
15: **else**
16: **if** $f(\hat{p}) > f(p^*)$ **then**
17: move to the best neighbor $p_c \leftarrow \hat{p}$
18: $GBF \leftarrow f(\hat{p})$, $f(p^*) \leftarrow GBF$
19: $p^* \leftarrow \hat{p}, d \leftarrow 0, t \leftarrow 0$
20: clean tabu list
21: **else if** $f(\hat{p}) \leq f(p^*)$ **then**
22: $d \leftarrow d + 1$
23: $t \leftarrow t + 1$
24: **if** $d = D_{max}$ **then**
25: clean tabu list
26: *sum*$\leftarrow 0$
27: **for** $r \in R$
28: **for** $i \in I_r \backslash \{|I_r|\}$
29: generate a solution sol_{ri}, whose value is allocated from T_{ri}^{min} to T_{ri}^{max}
30: *sum* $\leftarrow$ *sum*+sol_{ri}
31: **end for**
32: adjust $sol_{r,|I_r|}$ to guarantee *sum* is the multiple of seven days
33: **end for**
34: save the incumbent solution $p_c \leftarrow (sol_{ri}, r \in R, i \in I_r)$
35: $d \leftarrow 0$
36: **end if**
37: **end if**
38: **end if**
39: **end while**
40: **return** the objective value

6.3.5. Intensification and diversification strategies

The use of memory structures within a tabu search meta-heuristic has been proven to create a flexible search behavior. A key element of the proposed framework is to achieve a balance between search intensification and diversification. The intensification strategy encourages move combinations and solution features that have appeared to be effective during the search. In contrast, diversification is used to broaden the exploration of the solution space. In our algorithm, the diversification strategy cleans the tabu list and then randomly generates a new solution. In lines 20 and 25-34 of Algorithm 2, we provide a description of our intensification and diversification strategies.

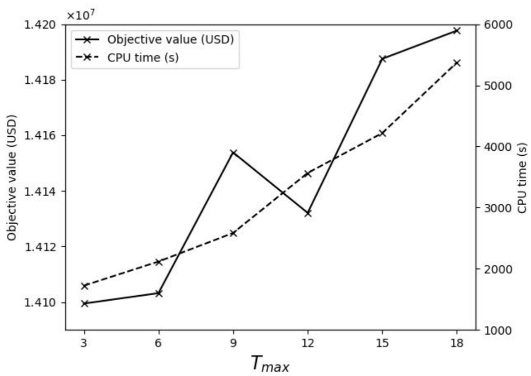
6.3.6. Sensitivity analysis of the parameters

To study the effectiveness of the proposed algorithm, we performed sensitivity analyses to determine the optimal combination of heuristic parameters. The chosen four parameters are the consecutive number of iterations where the current optimal solution is not updated (T_{max}), the number of candidate solutions (C_{max}), the tabu list size (L_{max}) and the consecutive number of iterations where the current optimal solution is not improved (D_{max}). These parameters are key parameters which may significantly affect the performance of the tabu search algorithm.

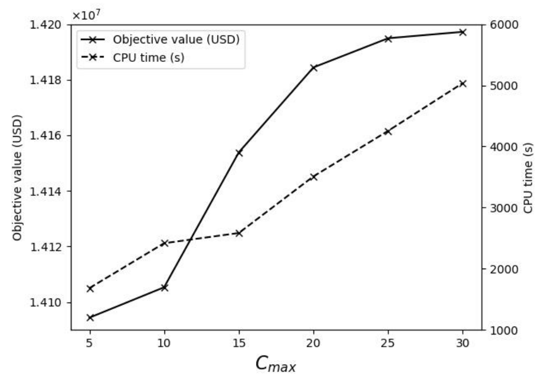
To show how the objective value and the computation time are influenced by parameters T_{max}, C_{max}, L_{max} and D_{max}, we designed four test schemes. The outputs consist of the computation time and the objective value. When we conduct sensitivity analysis for one parameter, the values of the other three parameters are fixed. Figure 2-(a) illustrates the interrelation between the value of parameter T_{max} and the objective value as well as the computation time, with the value of T_{max} varying in $\{3, 6, \ldots, 18\}$. The same method is applied to parameters C_{max}, L_{max} and D_{max}, varying in $\{5, 10, \ldots, 30\}$, $\{10, 20, \ldots, 60\}$, and $\{3, 4, \ldots, 8\}$, respectively.

The performance of tabu search algorithm is evaluated based on both the objective value and the computation time. The results in Figure 2 show that with increases in the values of parameters T_{max}, C_{max}, L_{max} and D_{max}, the computation times of the tabu search algorithm rise considerably, which indicates that the computation times are sensitive to the setting of parameters T_{max}, C_{max}, L_{max} and D_{max}. Interestingly, the objective values of tabu search algorithm grow considerably with the values of parameters T_{max} and C_{max}, but they fluctuate moderately as a function of L_{max} and D_{max}, which illustrates that the objective values of tabu search algorithm are sensitive to the setting of parameters T_{max}, C_{max}, but not to L_{max} and D_{max}.

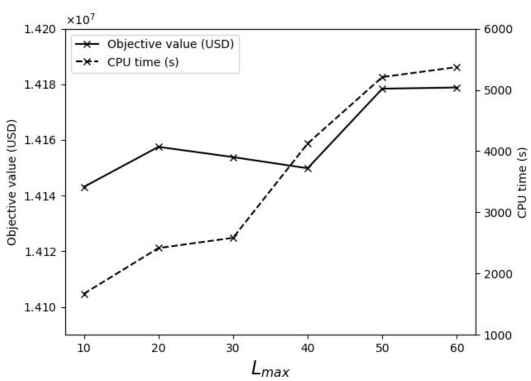
We then evaluated the performance of the tabu search algorithm over 36 instances with fixed $L_{max} = 20$, $D_{max} = 4$ and different values of T_{max} and C_{max}. For each test instance, several combinations of the two parameters T_{max} and C_{max} were used. The objective values (left column) and the computational time (right column) of each test



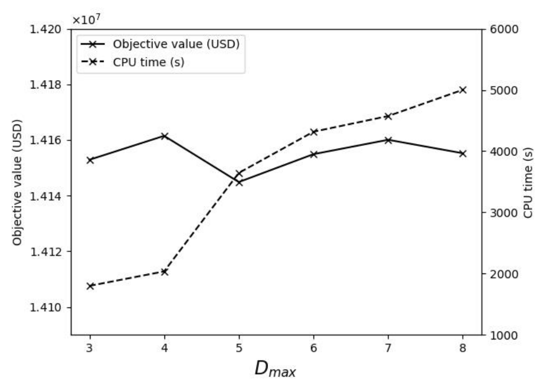
(a)



(b)



(c)



(d)

Figure 2: Sensitivity analysis of the parameters

Table 1: Influence of the parameters T_{max} and C_{max} on the performance of the tabu search heuristic

obj and time — C_{max} / T_{max}	5		10		15		20		25		30	
4	14,102,397	1,568	14,150,993	1,733	14,159,445	2,518	14,160,098	3,022	14,178,442	3,426	14,192,053	3,677
6	14,140,967	1,821	14,137,802	2,200	14,104,409	2,212	14,135,676	3,454	14,170,465	3,672	14,200,567	4,348
8	14,130,444	1,577	14,143,956	2,417	14,143,855	2,584	14,192,930	3,510	14,137,544	4,255	14,179,581	5,037
10	14,184,778	2,055	14,167,990	2,343	**14,210,744**	**2,555**	14,193,398	4,022	14,202,056	4,709	14,188,033	5,633
12	14,150,376	3,734	14,180,336	3,723	14,192,098	3,554	14,197,544	4,392	14,179,002	4,430	14,207,834	5,722
14	14,192,804	4,271	14,192,003	3,356	14,199,581	4,598	14,190,095	4,765	14,201,566	5,221	14,198,892	5,576

Notes: (1) The objective values and the CPU time are recorded in the left column and right column, respectively. (2) The objective values and the CPU time are denoted by obj and time, respectively. (3) The CPU time is in seconds.

instance are recorded in Table 1. It can be seen that when $T_{max} \doteq 10$ and $C_{max} \doteq 15$, we can obtain the best results. Therefore, the four values T_{max}=10, $C_{max} = 15$, $L_{max} = 20$ and $D_{max} = 4$ will be used in the next experiments.

7. Computational experiments

In order to assess the effectiveness of the proposed decision model and the efficiency of our algorithms, we have carried out several computational experiments on a LENOVO P910 workstation with 28 cores of CPUs, 2.4 GHz processing speed and 256 GB of memory. All of the models and algorithms proposed in this article were implemented in C# programming. The MIP models (the original model and the submodels embedded in algorithms) were solved by CPLEX 12.5.1.

7.1. Instance setting

We first detail the setting of the model parameters. The value of V_ε relates to the sailing distance and to the number of containers transported between an OD pair. The sailing distance data can be obtained on the Internet websites, and the unit container revenue data can be acquired on some logistics companies' official websites. The average of C_r^{Opr} is set to 180,000 USD (Wang and Meng, 2015; Wang et al., 2015; Alharbi et al., 2015). The average of N_r^{Ship} which depends on the length of one cycle time is set to 20. This is consistent with the parameter setting used in previous works (Wang and Xu, 2015; Yao et al., 2012). The average of k_{ri} is set to 0.25, and the average of a_{ri} is set to 2.6, which are basically the same as in previous works (Wang et al., 2015; Bell et al., 2013; Yao et al., 2012; Wang and Meng, 2015; Meng et al., 2016). The average of C^{Hold} is set to 20 USD per day per TEU (Zheng et al., 2015; Wen et al., 2017; Wang and Meng, 2015; Bell et al., 2013). The value of α is set to 1%. The maximum value of sailing speed is set to 22 knots, which is also in line with the setting used in related works (Jiang and Jin, 2017; Wang et al., 2015; Yao et al., 2012; Aydin et al., 2017). The average of C_p^{Berth} is set to 3000 per berth (Chen et al., 2012) and the average of C_p^{Yard} is set to 200 USD per TEU (Jiang and Jin, 2017). The value of $\overline{D}$ is two days, which is consistent with realistic data from the APL company.

The shipping network investigated in the numerical experiments is depicted in Figure 1. The numbers of routes are three and four in the two different scales of experiments, and the numbers of ports of call are four, four, five and six in route 1, 2, 3, and 4, respectively. The experimental instances are generated on the basis of a specific rule. Taking the small-scale route network for example, the number of routes is three and the numbers of ports of call are four, four, and five in route 1, 2, and 3, respectively. We can then generate four cases in route 1, which differ from each other only with respect to the ports of call. Each of the four cases uses three ports of call among the four ports of call in the original route 1 shown in Figure 1. Analogously, more sets of cases can be generated through different selections of ports of call in other routes.

Thus for the small-scale without all ports of call network with three routes, there are four sets of cases including three sets without all the ports of call, and an integrated case with all of them. Similarly, as for the large-scale route network consisting of four routes (as shown in Figure 1), there are four sets of cases without all ports of call, and an integrated case with all of them.

7.2. Investigating the efficiency of the proposed methods

Here we apply the dynamic linearization algorithm to solve the model [$\boldsymbol{M2}$]. A large number of numerical experiments on small-scale cases were carried out to validate this algorithm by comparing the values of its solutions with the optimal results obtained by CPLEX.

From the results shown in Table 2, the objective values obtained by the dynamic linearization algorithm are equal to the optimal results, but this algorithm is faster on the small-scale route network. Based on these observations, we can confirm the efficiency of dynamic linearization algorithm. Table 2 also provides an upper bound (UB) obtained by relaxing Constraints (15), and it shows the gap between the UB and the optimal solution value, which is used to evaluate the efficiency of tabu search algorithm in the large-scale route network. To generate a more complex shipping network, we increase the number of routes from the three to four, which yields a large-scale route network. The results of the experiments show that it is difficult to obtain an optimal solution on this network within a reasonable time.

Table 2: Performance of the dynamic linearization (three routes)

Cases		CPLEX		Dynamic linearization					Upper Bound	
Num. of ports in three routes	ID	Z_C	T_C	π_ε	Z_D	T_D	GAP_C	$\frac{T_D}{T_C}$	Z_{UB}	GAP_{UB}
3-4-5 (Cases differ on the ports in **route 1**)	Case 1	2,550,670	43	90.34%	2,550,670	13	0.00%	0.30	2,557,281	0.26%
	Case 2	2,592,150	59	92.83%	2,592,150	11	0.00%	0.19	2,605,755	0.52%
	Case 3	2,450,207	28	91.90%	2,450,207	12	0.00%	0.43	2,463,843	0.56%
	Case 4	2,729,982	21	94.55%	2,729,982	8	0.00%	0.38	2,743,593	0.50%
4-**3**-5 (Cases differ on the ports in **route 2**)	Case 1	2,766,213	48	95.28%	2,766,213	12	0.00%	0.25	2,779,856	0.49%
	Case 2	2,959,825	73	94.46%	2,959,825	17	0.00%	0.23	2,969,885	0.34%
	Case 3	2,307,711	58	93.17%	2,307,711	10	0.00%	0.17	2,308,947	0.05%
	Case 4	2,648,636	27	93.35%	2,648,636	9	0.00%	0.33	2,652,364	0.14%
4-4-**4** (Cases differ on the ports in **route 3**)	Case 1	2,354,829	30	91.96%	2,354,829	10	0.00%	0.33	2,368,568	0.58%
	Case 2	2,571,288	56	92.03%	2,571,288	12	0.00%	0.21	2,584,892	0.53%
	Case 3	2,667,825	28	93.30%	2,667,825	9	0.00%	0.32	2,671,536	0.14%
	Case 4	2,570,305	57	92.27%	2,570,305	12	0.00%	0.21	2,576,964	0.26%
	Case 5	2,664,537	35	94.82%	2,664,537	11	0.00%	0.31	2,668,272	0.14%
4-4-5	Case 1	3,905,795	75	93.11%	3,905,795	15	0.00%	0.20	3,921,398	0.40%
	Average			**93.10%**			**0.00%**	**0.28**		**0.35%**

Notes: (1) The optimal objective values and the CPU time are denoted by Z_C and T_C, respectively. (2) The objective values and the CPU time of the dynamic linearization algorithm are denoted by Z_D and T_D, respectively. (3) $GAP_C = (Z_D - Z_C)/Z_C$, $GAP_{UB} = (Z_{UB} - Z_C)/Z_C$.

Table 3: Comparing dynamic linearization with tabu search (four routes)

Cases		Dynamic linearization			Tabu search		Comparison	
Num. of ports in four routes	ID	Z_D	$Time_D$	π_ε	Z_T	$Time_T$	GAP_{TD}	$\frac{Time_T}{Time_D}$
3-4-5-6 (Cases differ on the ports in **route 1**)	Case 1	11,038,551	3,180	92.02%	11,008,751	1,489	0.27%	0.47
	Case 2	11,874,243	2,692	91.44%	11,809,739	1,803	0.54%	0.67
	Case 3	11,728,095	3,034	93.56%	11,708,295	1,865	0.17%	0.61
	Case 4	12,133,870	1,872	94.47%	12,114,073	1,174	0.16%	0.63
4-**3**-5-6 (Cases differ on the ports in **route 2**)	Case 1	12,170,101	2,845	93.23%	12,124,114	1,687	0.38%	0.59
	Case 2	12,470,813	2,923	94.34%	12,421,430	2,075	0.40%	0.71
	Case 3	12,117,924	2,900	93.54%	12,098,124	2,126	0.16%	0.73
	Case 4	12,039,317	2,879	90.86%	11,999,518	2,020	0.33%	0.70
4-4-**4**-6 (Cases differ on the ports in **route 3**)	Case 1	11,893,770	3,045	92.88%	11,867,079	1,988	0.22%	0.65
	Case 2	12,178,122	3,300	94.32%	12,145,369	1,723	0.27%	0.52
	Case 3	12,058,507	2,954	95.63%	12,018,702	1,636	0.33%	0.55
	Case 4	11,051,646	2,326	92.32%	11,018,431	1,702	0.30%	0.73
	Case 5	12,069,020	3,875	93.64%	12,029,216	1,734	0.33%	0.45
4-4-5-**5** (Cases differ on the ports in **route 4**)	Case 1	10,947,598	3,004	90.32%	10,927,768	1,556	0.18%	0.52
	Case 2	11,120,029	3,154	91.75%	11,080,223	2,164	0.36%	0.69
	Case 3	11,998,000	2,934	94.92%	11,958,243	2,171	0.33%	0.74
	Case 4	11,594,584	3,357	93.33%	11,554,784	1,947	0.34%	0.58
	Case 5	11,424,056	3,011	92.44%	11,363,650	1,436	0.53%	0.48
	Case 6	11,996,883	2,173	92.46%	11,957,174	1,309	0.33%	0.60
4-4-5-6	Case 1	14,284,795	4,503	90.02%	14,210,744	2,555	0.52%	0.57
	Average			**92.87%**			**0.32%**	**0.61**

Notes: (1) $Time_D$ and $Time_T$ denote the CPU time of the dynamic linearization algorithm and tabu search algorithm, respectively. (2) $GAP_{TD} = (Z_D - Z_T)/Z_D$. (3) The CPU time is in seconds.

Therefore, we suggest applying tabu search algorithm to solve the model, and we compare its objective value with that obtained by the dynamic linearization algorithm. The results in the rightmost two columns of Table 3 demonstrate that the average gap between dynamic linearization and tabu search algorithm is about 0.32%, but the average ratio of the CPU time of tabu search algorithm to that of the dynamic linearization algorithm is only 0.61, which indicates that tabu search may not only obtain near-optimal objective function values, but can also solve the model in a much faster way. These results confirm the effectiveness of the dynamic linearization algorithm and of the tabu search algorithm. They demonstrate that tabu search is an effective method for solving the proposed model.

8. Conclusions

We have proposed an integrated optimization model for the fleet deployment and demand fulfillment problem, with the consideration of overload risk of containers, vessel size and port resources (e.g., berths, yard space). The objective was to jointly optimize the number of ships in each route, the ship speed on each leg, the visiting time of ships at each port of call, and the fulfillment scale of each OD pair's demand. Since the proposed model is a chance-constrained non-linear MIP model, we have suggested some novel techniques to linearize it into a tractable MISOCP model for some commercial solvers such as the CPLEX. Two efficient algorithms were then suggested to solve the model under different scales of route networks. The proposed model as well as the algorithms can help shipping liners plan the deployment and scheduling of ships along each route. Numerical experiments based on real-word data were conducted to validate the effectiveness of our decision model and the efficiency of the proposed solution methods. With respect to the large body of research on liner ship fleet deployment, we have made three main new contributions:

(1) Few of the previous fleet deployment related studies have considered the demand fulfillment decisions. However, both the fleet deployment and the demand fulfillment decisions are strategic in nature and are intertwined. This study proposed an integrated decision model for optimizing the ship fleet deployment, the scheduling of ship visits at each port of call, and the demand fulfillment scale for each OD pair. The objective was to maximize the total benefit of shipping liners by considering various

types of operation costs for running shipping networks.

(2) The overload risk of transported containers has seldom been considered in the FDP related literature, but this issue should not be ignored given the stochastic weights of containers. Our study takes stochasticity into account by embedding chance constraints in the decision model so as to control the overload risk under a certain threshold probability. Some tactics were also suggested to handle the model's non-linearity as well the complexity yielded by the chance constraints.

(3) Several realistic factors ignored in previous studies were considered in our decision model, but solving them proved to be difficult. We have developed two algorithms to solve the proposed non-linear chance-constrained MIP on large-scale instances. Experiments conducted on real-world data demonstrate that our methodology yields solutions with an optimality gap less than about 0.5%, and can solve realistic instances with 19 ports and four routes within about one hour.

Acknowledgment

This work was supported by the National Natural Science Foundation of China [grant numbers 71831008, 71671107, 71422007, 71701178] and the Canadian Natural Sciences and Engineering Research Council [grant number 2015-06189]. Thanks are due to the reviewers for their valuable comments.

References

Alharbi, A., Wang, S., Davy, P., 2015. Schedule design for sustainable container supply chain networks with port time windows. Advanced Engineering Informatics 29 (3), 322–331.

Álvarez, J. F., 2009. Joint routing and deployment of a fleet of container vessels. Maritime Economics & Logistics 11 (2), 186–208.

Andersson, H., Fagerholt, K., Hobbesland, K., 2015. Integrated maritime fleet deployment and speed optimization: Case study from roro shipping. Computers & Operations Research 55, 233–240.

Aydin, N., Lee, H., Mansouri, S. A., 2017. Speed optimization and bunkering in liner shipping in the presence of uncertain service times and time windows at ports. European Journal of Operational Research 259 (1), 143–154.

Bell, M. G., Liu, X., Angeloudis, P., Fonzone, A., Hosseinloo, S. H., 2011. A frequency-based maritime container assignment model. Transportation Research Part B: Methodological 45 (8), 1152–1161.

Bell, M. G., Liu, X., Rioult, J., Angeloudis, P., 2013. A cost-based maritime container assignment model. Transportation Research Part B: Methodological 58, 58–70.

Chen, C., Zeng, Q., Zhang, Z., 2012. An integrating scheduling model for mixed cross-operation in container terminals. Transport 27 (4), 405–413.

Cho, S.-C., Perakis, A., 1996. Optimal liner fleet routeing strategies. Maritime Policy and Management 23 (3), 249–259.

Christiansen, M., Fagerholt, K., Nygreen, B., Ronen, D., 2013. Ship routing and scheduling in the new millennium. European Journal of Operational Research 228 (3), 467–483.

Christiansen, M., Fagerholt, K., Ronen, D., 2004. Ship routing and scheduling: Status and perspectives. Transportation Science 38 (1), 1–18.

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

Fagerholt, K., Johnsen, T. A., Lindstad, H., 2009. Fleet deployment in liner shipping: a case study. Maritime Policy and Management 36 (5), 397–409.

Fransoo, J. C., Lee, C.-Y., 2013. The critical role of ocean container transport in global supply chain performance. Production and Operations Management 22 (2), 253–268.

Gelareh, S., Meng, Q., 2010. A novel modeling approach for the fleet deployment problem within a short-term planning horizon. Transportation Research Part E: Logistics and Transportation Review 46 (1), 76–89.

Glover, F., 1986. Future paths for integer programming and links to artificial intelligence. Computers & Operations Research 13 (5), 533–549.

Jaramillo, D., Perakis, A. N., 1991. Fleet deployment optimization for liner shipping part 2. implementation and results. Maritime Policy and Management 18 (4), 235–262.

Jiang, X. J., Jin, J. G., 2017. A branch-and-price method for integrated yard crane deployment and container allocation in transshipment yards. Transportation Research Part B: Methodological 98, 62–75.

Liu, M., Lee, C.-Y., Zhang, Z., Chu, C., 2016. Bi-objective optimization for the container terminal integrated planning. Transportation Research Part B: Methodological 93, 720–749.

Meng, Q., Du, Y., Wang, Y., 2016. Shipping log data based container ship fuel efficiency modeling. Transportation Research Part B: Methodological 83, 207–229.

Meng, Q., Wang, S., 2012. Liner ship fleet deployment with week-dependent container shipment demand. European Journal of Operational Research 222 (2), 241–252.

Meng, Q., Wang, S., Andersson, H., Thun, K., 2014. Containership routing and scheduling in liner shipping: overview and future research directions. Transportation Science 48 (2), 265–280.

Meng, Q., Wang, T., 2010. A chance constrained programming model for short-term liner ship fleet planning problems. Maritime Policy and Management 37 (4), 329–346.

Meng, Q., Wang, T., 2011. A scenario-based dynamic programming model for multi-period liner ship fleet planning. Transportation Research Part E: Logistics and Transportation Review 47 (4), 401–413.

Meng, Q., Wang, T., Wang, S., 2012. Short-term liner ship fleet planning with container transshipment and uncertain container shipment demand. European Journal of Operational Research 223 (1), 96–105.

Monemi, R. N., Gelareh, S., 2017. Network design, fleet deployment and empty repositioning in liner shipping. Transportation Research Part E: Logistics and Transportation Review 108, 60–79.

Nikolopoulou, A. I., Repoussis, P. P., Tarantilis, C. D., Zachariadis, E. E., 2017. Moving products between location pairs: Cross-docking versus direct-shipping. European Journal of Operational Research 256 (3), 803–819.

Perakis, A. N., Jaramillo, D., 1991. Fleet deployment optimization for liner shipping part 1. background, problem formulation and solution approaches. Maritime Policy and Management 18 (3), 183–200.

Petering, M. E., Wu, Y., Li, W., Goh, M., de Souza, R., Murty, K. G., 2017. Real-time container storage location assignment at a seaport container transshipment terminal: dispersion levels, yard templates, and sensitivity analyses. Flexible Services and Manufacturing Journal 29 (3–4), 369–402.

Powell, B., Perkins, A., 1997. Fleet deployment optimization for liner shipping: An integer programming model. Maritime Policy and Management 24 (2), 183–192.

Qi, X., Song, D.-P., 2012. Minimizing fuel emissions by optimizing vessel schedules in liner shipping with uncertain port times. Transportation Research Part E: Logistics and Transportation Review 48 (4), 863–880.

Ronen, D., 1993. Ship scheduling: The last decade. European Journal of Operational Research 71 (3), 325–333.

Tirado, G., Hvattum, L. M., Fagerholt, K., Cordeau, J.-F., 2013. Heuristics for dynamic and stochastic routing in industrial shipping. Computers & Operations Research 40 (1), 253–263.

Vivaldini, K., Rocha, L. F., Martarelli, N. J., Becker, M., Moreira, A. P., 2016. Integrated tasks assignment and routing for the estimation of the optimal number of agvs. The International Journal of Advanced Manufacturing Technology 82 (1–4), 719–736.

Wang, C., Xu, C., 2015. Sailing speed optimization in voyage chartering ship considering different carbon emissions taxation. Computers & Industrial Engineering 89, 108–115.

Wang, H., Zhang, X., Wang, S., 2016. A joint optimization model for liner container cargo assignment problem using state-augmented shipping network framework. Transportation Research Part C: Emerging Technologies 68, 425–446.

Wang, S., Meng, Q., 2012. Liner ship fleet deployment with container transshipment operations. Transportation Research Part E: Logistics and Transportation Review 48 (2), 470–484.

Wang, S., Meng, Q., 2015. Robust bunker management for liner shipping networks. European Journal of Operational Research 243 (3), 789–797.

Wang, S., Meng, Q., Liu, Z., 2013. Bunker consumption optimization methods in shipping: A critical review and extensions. Transportation Research Part E: Logistics and Transportation Review 53, 49–62.

Wang, S., Wang, T., Meng, Q., 2011. A note on liner ship fleet deployment. Flexible Services and Manufacturing Journal 23 (4), 422–430.

Wang, T., Meng, Q., Wang, S., 2012. Robust optimization model for liner ship fleet planning with container transshipment and uncertain demand. Transportation Research Record: Journal of the Transportation Research Board (2273), 18–28.

Wang, X., Fagerholt, K., Wallace, S. W., 2017. Planning for charters: A stochastic maritime fleet composition and deployment problem. Omega, DOI 10.1016/j.omega.2017.07.007.

Wang, Y., Meng, Q., Du, Y., 2015. Liner container seasonal shipping revenue management. Transportation Research Part B: Methodological 82, 141–161.

Wen, M., Pacino, D., Kontovas, C. A., Psaraftis, H. N., 2017. A multiple ship routing and speed optimization problem under time, cost and environmental objectives. Transportation Research Part D: Transport and Environment 52, 303–321.

Xia, J., Li, K. X., Ma, H., Xu, Z., 2015. Joint planning of fleet deployment, speed optimization, and cargo allocation for liner shipping. Transportation Science 49 (4), 922–938.

Yao, Z., Ng, S. H., Lee, L. H., 2012. A study on bunker fuel management for the shipping liner services. Computers & Operations Research 39 (5), 1160–1172.

Yi, W., Chi, H.-L., Wang, S., 2018. Mathematical programming models for construction site layout problems. Automation in Construction 85, 241–248.

Zacharioudakis, P. G., Iordanis, S., Lyridis, D. V., Psaraftis, H. N., 2011. Liner shipping cycle cost modelling, fleet deployment optimization and what-if analysis. Maritime Economics & Logistics 13 (3), 278–297.

Zhao, Y., Jia, R., Jin, N., He, Y., 2016. A novel method of fleet deployment based on route risk evaluation. Information Sciences 372, 731–744.

Zhen, L., 2015. Tactical berth allocation under uncertainty. European Journal of Operational Research 247 (3), 928–944.

Zhen, L., 2016. Modeling of yard congestion and optimization of yard template in container ports. Transportation Research Part B: Methodological 90, 83–104.

Zheng, J., Gao, Z., Yang, D., Sun, Z., 2015. Network design and capacity exchange for liner alliances with fixed and variable container demands. Transportation Science 49 (4), 886–899.