Elsevier Editorial System(tm) for Transportation Research Part B
Manuscript Draft

Manuscript Number: TRB-D-14-00456R2

Title: Joint service capacity planning and dynamic container routing in shipping network with uncertain demands

Article Type: Research Paper

Keywords: service capacity planning, dynamic container routing, container shipping, stochastic

Corresponding Author: Dr. Jing-Xin Dong,

Corresponding Author's Institution:

First Author: Jing-Xin Dong

Order of Authors: Jing-Xin Dong; Chung-Yee Lee; Dong-Ping SONG

Statement of Contributions

1) We consider a novel and challenging problem in liner shipping concerning service capacity planning and dynamic shipment routing with uncertain demands, container transhipment, and delivery time constraints.
2) The joint optimisation problem has been resolved rigorously. This involves the formulation of the problem as a two-stage stochastic programming model and the implementation of three solution strategies.
3) An extension of PHA method based on Lagrangian relaxation method, APHA, has been proposed. It can be used to solve large-scale problems that are not tractable using the existing methods such as SAA and PHA.
4) A link-based dynamic container routing model is applied to formulate the second stage problem. According to Wang (2014), "the number of variables in link-based models increases polynomially with the size of the liner shipping network". Therefore, the model has good tractability. Furthermore, the container routing model considers the dynamics of container shipping system on a daily basis or even shorter, thus it has the merit of modelling the container waiting time more accurately.

Authors' responses to reviewers' comments on the manuscript Ref. No.: TRB-D-14-00456

We highly appreciate the kind support for publication of our manuscript from the reviewers and the editor. We also would like to take the opportunity to thank the reviewers and the editor for taking the time to process our manuscript.

Responses to Reviewer #1's comments:

Reviewer #1: The paper has addressed my concerns and I recommend it for publication.

Response: we really appreciate your kind support during the review process, and we are very grateful for your early comments which significantly improve the quality of our manuscript.

Responses to Reviewer #2's comments:

Reviewer #2: I am generally happy with the authors' respond to my comments. Just a small note for the authors to consider: I think it will be good if the authors can also discuss some limitation on the work, e.g. the use of the distribution to describe the demand and the consideration of empty container repositioning.

Response: we really appreciate your kind support and your early comments. We have further addressed the limitation of our work by re-writing the last paragraph in conclusion section.

Highlights

- Model the joint service capacity planning and dynamic container routing for stochastic customer demands with day-to-day changes
- Apply SAA and PHA to solve small-scale problems
- Develop a new APHA(Adapted PHA) to solve the problems for large shipping network in reality
- Illustrate the relative merits of the three solution strategies on both hypothetic and realistic shipping networks

Joint service capacity planning and dynamic container routing in shipping network with uncertain demands

Jing-Xin DONG
Business School, Newcastle University, 5 Barrack Road, Newcastle upon Tyne, NE1 4SE, UK
Email: jingxin.dong@ncl.ac.uk

Chung-Yee LEE
Department of Industrial Engineering and Logistics Management, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong
Email : cylee@ust.hk

Dong-Ping SONG
School of Management, University of Liverpool, Chatham Street, Liverpool, L69 7ZH, United Kingdom
Email: dongping.song@liverpool.ac.uk

Abstract: Service capacity planning is a key tactic decision in container shipping, which has a significant impact on daily operations of shipping company. On the other hand, operational decisions such as demand fulfilment and shipment routing will impact on service capacity requirements and utilisation, particularly in the presence of demand uncertainty. This article proposes a two stage stochastic programming model with recourse to deal with the problem of joint service capacity planning and dynamic container routing in liner shipping. The first stage of the model concerns how to determine the optimal service capacity, and the second focuses on the optimal routing of shipments in stochastic and dynamic environments under a given service capacity plan. Initially, SAA (Sample Average Approximation) is employed to solve the model. Noting the computational complexity of the problem, Progressive Hedging Algorithm (PHA) is employed to decompose the SAA model into a number of scenario-based models so that reasonably large scale problems can be solved. To handle larger scale problems, we develop a new solution procedure termed as APHA (Adapted Progressive Hedging Algorithm) that further decomposes the scenario-based model into job (customer order) based models with measurable error bounds. Numerical experiments are conducted to illustrate the effectiveness of the proposed APHA in solving the problems under consideration.

Keywords: service capacity planning, dynamic container routing, container shipping, stochastic

1 Introduction

Container shipping industry plays a very important role in world economy. Each year container shipping industry transports two-thirds of the value of total global trade, which equals more than US$ 4 trillion. It also has direct gross output or GDP contribution -- US$ 183.3 Billion per year (http://www.worldshipping.org/benefits-of-liner-shipping/global-economic-engine). Improving the efficiency of container transport system would benefit not only the shipping industry itself but also other broad industrial sectors and the general public.

One of the key decisions in container shipping is to determine the service capacity (i.e. supply) to meet fluctuating trade (i.e. demand). Basically, the issue concerns how to determine the capacity of each vessel deployed on the shipping service network, which includes the decisions on chartering in slot capacities from other companies' vessels (e.g. the slot exchange and purchase between members of a shipping alliance). The importance of the problem can be evidenced from several aspects. Firstly, the purchase of container vessel involves huge capital investment, e.g., in the current ship markets, one 4,000-TEUs vessel costs $60 million roughly, and a 12,000-TEUs vessel costs $120 million. Secondly, it has a medium/long-term and significant impact on the operations of shipping companies, e.g., a container ship's life span can be as long as some 30 years. Thirdly, nowadays shipping alliance is becoming increasingly popular in shipping practice, which involves vessel sharing and slot chartering between different companies, e.g., CKYHE Alliance, G6 Alliance, and the recent proposals of 2M alliance (Maersk and MSC) and Ocean Three alliance (CMA CGM, UASC and CSCL). As the members of an alliance are independent from the financial and market perspective, it is vital for them to determine how much capacity of their own vessels should be kept and how much capacity of other members' vessels should be chartered in by considering their own market demands. Fourthly, a service capacity planning problem can also be regarded as a part of liner service network design problem, in which the shipping line needs to determine its service capacity (and vessel deployment) in the service network (that may consist of existing service routes and new candidate service routes). For example, Maersk uses the term 'network management' to describe the adjustment of their service routes and service capacity in response to the change of demand patterns and/or the deployment of new ships (e.g. the delivery of Triple-E vessels in 2013), and regards it as the heart of their business.

Determining service capacity is interwoven with the routing of container shipments on shipping network. The optimal service capacity can only be obtained when container flow is distributed in the best way. In shipping practice, container flows are driven by uncertain and dynamic customer demands. It is a challenging task to find the optimally distributed container flows and consequently the optimal service capacity in a stochastic and dynamic environment. In the paper, we will use a two-stage stochastic model with recourse to tackle the challenge. In shipping practice, container flows are driven by uncertain and dynamic customer demands. It should be pointed out that forecasting the market demand is difficult due to many external factors including the potential competitors and their behaviours. However, as most shipping lines have been running business for many years and their historical data could be used as reference data to fit into a probability distribution. In fact, probability distribution is a common approach to represent uncertain demands in the literature, e.g. Meng and Wang 2010; Meng et al. 2012. Furthermore, our model uses the average value of sample processes to approximate the expected value of the random variables, which essentially just takes historical demand information as input without the need to determine the distribution function of demand.

Many studies in relation to service capacity planning and container routing have been conducted. In previous studies, service capacity planning is partially dealt with under the name of Liner Ship Fleet Deployment (LSFD). LSFD aims to decide how many vessels for a specific type should be deployed to each service route on container shipping network. The solution to LSFD implies the capacities that a service route should have. Service capacity planning is significantly different from LSFD. LSFD normally selects vessels from a given set of vessel types and the vessels deployed on each service route are homogeneous, whereas service capacity planning in our context concerns more about the amount of TEU slots on each vessel rather than the vessel type, which implies that the available capacities could vary vessel by vessel even they belongs to the same service route. With regard to LSFD, the studies can be classified as deterministic models and stochastic models. The deterministic models have been proposed in Perakis and Jaramillo (1991), Jaramillo and Perakis (1991), Cho and Perakis (1996), Powell and Perakis (1997), Gelareh and Meng (2010), Wang et al.

(2011), Meng and Wang (2011a, b), and Zacharioudakis, et al (2011). These models consider either direct shipping service or single service route, and therefore, transhipment issues are not concerned. Some other deterministic models have been designed for multiple service routes where transhipments have been considered, e.g., Mourão et al (2010), Liu et al (2011), Wang and Meng (2012a), Meng and Wang (2012), Fagerholt et al (2009). The research methods adopted in the deterministic models are mainly Linear Programming (LP), Integer Linear Programming(ILP) or Mixed Integer Linear Programming(MILP). The research community has also recognised the stochastic nature of the issue, and developed a number of stochastic models. Meng and Wang (2010) perhaps is the first study considering stochastic demands in containership fleet planning. The study focuses on the vessel deployment on a single service route with uncertain demands. A more complex model has been presented in Meng et al (2012), which considers both transhipment and uncertain demands. Wang et al (2012) have made some extension to the study by incorporating risk oriented costs into the objective function.

With regard to container routing problems in liner shipping, there was very little research before 2004 (Christiansen et al., 2004). In the last decade, it has attracted a lot of attention. The existing studies can be classified as link-based routing (Alvarez, 2009; Agarwal & Ergun, 2008; Bell et al. , 2011, 2013; Meng & Wang, 2012; Yan et al., 2009; Song et al. , 2005;) and path-based routing (Brouer et al., 2011; Song and Dong, 2012; Wang et al., 2013; Wang and Meng, 2012b). In general, the scale of link-based routing model is smaller than that of path-based routing model as path-based model is based on the enumeration of all possible paths or dynamical generation of the profitable paths (Wang, 2014). However, the majority of the existing studies tackle the container routing problems at the tactic level without considering the detailed operations, e.g. assuming that containers' travelling time on a path and waiting time at transhipment ports are fixed and known input data, and irrelevant to the container routing decisions; there are fixed weekly demands without uncertainties; there are no constraints on the delivery time.

In the study, we will consider service capacity planning and shipment routing with uncertain demands, container transhipment, and delivery time constraints. A two-stage stochastic model with recourse will be developed. The first stage centres on minimising the acquisition costs of service capacity, and the second stage is to seek the optimal dynamic routing plan of container flows with uncertainty. Our second stage model is a dynamic link-based container routing model in which waiting-time at transhipment ports is dependent on the routing plan, and can only be revealed in the execution of the routing plan. Moreover, the waiting-times at transhipment ports are measured on a daily basis or even shorter.

The way we model the problem can provide good accuracy as it models the operational details of a realistic container shipping system. However, the formulation can lead to very large-scale problems, which is computationally challenging to find the optimal solutions. In this study, we propose a solution procedure termed as Adapted Progressive Hedging Algorithm (APHA). The APHA is developed by tailoring Progressive Hedging Algorithm (PHA) (Rockafellar & Wets, 1991) to our specific problem using Lagrangian relaxation method. The numerical experiments show that the proposed solution method has good performance in solving large-scale problem.

The contributions of the article are summarised as follows.

1) We consider a novel and challenging problem in liner shipping concerning service capacity planning and dynamic shipment routing with uncertain demands, container transhipment, and delivery time constraints.
2) The joint optimisation problem has been resolved rigorously. This involves the formulation of the problem as a two-stage stochastic programming model and the implementation of three solution strategies.
3) An extension of PHA method based on Lagrangian relaxation method, APHA, has been proposed. It can be used to solve large-scale problems that are not tractable using the existing methods such as SAA and PHA.
4) A link-based dynamic container routing model is applied to formulate the second stage problem. According to Wang (2014), “the number of variables in link-based models increases polynomially with the size of the liner shipping network”. Therefore, the model has good tractability. Furthermore, the container routing model considers the dynamics of container shipping system on a daily basis or even shorter, thus it has the merit of modelling the container waiting time more accurately.

The rest of the article is structured as follows. In Section 2, the problem of joint service capacity planning and dynamic shipment routing with uncertain demands will be formulated as a two-stage stochastic programming model with recourse. In Section 3, we will develop three solutions including SAA, PHA, and APHA for solving the problem. Numerical examples are given to illustrate the effectiveness of the three solution methods in Section 4. Lastly, concluding remarks are made in Section 5.

2 Model formulation

In the section, we firstly define the notations to be used in the remainder of the articles, and then we give the formulation of our problem. In the literature, the space-time network model is often used to formulate the container flows in a shipping network (e.g. Brouer et al. 2011). We present a slightly different model in the following, which offers a more intuitive view of the evolution of the jobs' status over space and time.

2.1 Notations

Index and sets

P	the set of ports
V	the set of vessels
$\mathbf{\Omega}$	the entire populations of customer demands
$\omega(n)$	a sample process of customer demands, $1 \leqslant n \leqslant N$, where N represents the number of samples.
$\boldsymbol{J}(\omega(n))$, $\boldsymbol{J}$	the set of transportation jobs for a sample process of customer demands $\omega(n)$. To simplify our narrative, we drop off $\omega(n)$ and just use $\boldsymbol{J}$ when our discussion is limited for a given $\omega(n)$.
j	an individual transportation job, $j \in \boldsymbol{J}$ or $j \in \boldsymbol{J}(\omega(n))$. The important information associated with job j is its original and destination port, generation time (the time that job j is available to be serviced), the promised delivery time for job j, and its amount in TEUs.
$p \in P$	a port
i	a port-of-call (or portcall), and i+1 represents the next portcall after i. In the study, the first portcall is numbered as 0. $p(v,i)$ denote its port that vessel v calls at in its ith portcall in a round-trip.
l	a loop (round-trip or voyage) that vessel v sails along the service route.

$v \in V$	a vessel
t	a decision period
P_v	the set of ports that vessel v calls at in the service
$V_p^a(t)$	the set of vessels that arrive at port p at beginning of period t
$V_p^d(t)$	the set of vessels that depart from port p at the beginning of period t

Parameters

o_j	original port of job j
d_j	destination port of job j
D_j	transportation volume of job j in TEUs, which is a random variable in a certain range. For a realised customer demand $\omega(n)$, it is a known number.
t^0_j	The generation time period of job j
T_j	The promised delivery time for job j
$t^a_{v,l,i}$	the time period that vessel v arrives at portcall i in its l^{th} loop (round-trip)
$t^d_{v,l,i}$	the time period that vessel v departs from portcall i in its l^{th} loop (round-trip)
C_v	the unit cost of the shipping capacity for vessel v per period
c_t^j	The waiting cost per unit per period of job j during the delivery from the original port to the destination port
c_p^f	the lifting-off costs per unit of shipment at port p
c_p^o	the lifting-on costs per unit of shipment at port p
L_v	the minimum vessel capacity that the shipping company has to charter or purchase from vessel v
U_v	the maximum vessel capacity that the shipping company can charter and purchase from vessel v
T	the planning time horizon

Decision variables

y_v	the shipping service capacity on vessel v
$x^j_v(t)$	1, if job j is on board of vessel v during period t; otherwise, 0
$z^j_p(t)$	1, if job j is at port p during period t; otherwise, 0
$u^j_p(t)$	1, if job j is loaded onto a vessel at port p at time t
$v^j_p(t)$	1, if job j is unloaded from a vessel at port p at time t

Y	**Y**={y_1, …, y_v, …, $y_{\|V\|}$}, a vector consisting of all vessel shipping capacities
X	**X** ={$x^j_v(t)$, $z^j_p(t)$, $u^j_p(t)$, $v^j_p(t)$ | $j \in J$, $v \in V$, $p \in P$, $0 < t < T$}, which denotes all the second-stage decision variables

2.2 Two stage stochastic programming model

We consider a container shipping system comprising a set of ports *P*, a set of vessels *V*, a container shipping network, and a set of transportation jobs *J* that involves moving customers' cargoes from the original ports to the destination ports in *P* using the vessels in *V*. Each route on the given container shipping network comprises a number of ports in a fixed sequence. Normally, some common ports are shared by different shipping service routes, which become transhipment ports to link different shipping service routes to form an interconnected shipping network. The interconnection of shipping service routes enables container shipping company to move containers across shipping service routes, consequently provides much wider coverage of customer demands. The vessels in ***V*** are scheduled in a way that they repetitively make round trips on their deployed service routes on a weekly basis. The capacity of each vessel in *V* is treated as a decision variable in our suggested model. Additionally, in the process of serving customer demands, a very important decision that the container shipping company needs to make in their daily operation is which route is the best for a customer order. Their routing decisions are subject to vessel capacity constraint aimed at minimising transportation costs and transhipment costs. In this study, the transportation costs are assumed positively proportional to travelling times. An unfulfilled customer order will have a 'travelling time' equal to the difference between planning horizon and the generation time of the job, and will incur a cost in proportional to the 'travelling time'. This will serve as a penalty costs for not serving a job. We adopt this penalty mechanism to simplify the cost structure and the model development. It is noted that such a penalty may lead to rejecting servicing jobs near the end of the planning period *T* if the transportation costs exceed the penalty cost. This drawback can be overcome by appropriately selecting the job list and the planning horizon, e.g. using a cut-off time to exclude those jobs. The transhipment costs are incurred for lifting-on and lifting-off the containers at transhipment ports in the process of transferring them from one service route to another. When the vessel capacities are sufficiently big, the routes with the lowest transportation costs and transhipment costs can be selected for each order. However, this may lead to excessive investment on the vessel capacity. Our research question is how to achieve the best balance among the investment on vessel capacity, the operational costs including transportation costs and transhipments costs, and the unfulfilled job penalty costs in the stochastic demand situations.

Our problem is formulated based on the following assumptions.

Assumption 1: A shipment has to be at a port at least one period earlier before loading onto a vessel.

Assumption 2: The empty container repositioning is not considered explicitly.

Assumption 3: Container lifting-off from a vessel is performed in the vessel's arrival period; and lifting-on is done in the vessel's departure period. The vessel arrival and departure periods are different for each portcall.

Assumption 4: The supply of vessel capacities that container shipping companies can obtain by purchasing new ships, and charting in slots from the other shipping companies are sufficiently large. In other words, U_v is sufficiently large.

Assumption 1 is in line with the shipping practice as containers must be ready prior to the vessel arrival. Assumption 2 is common in the literature on container shipping network design and ship fleet deployment, e.g., Meng et al (2012), Wang et al (2012). The rationales for Assumption 2 may be explained as follows: (i) empty container repositioning does not generate revenue directly, and therefore laden container transportation usually has priority over empty container repositioning; (ii) liner service routes are cyclic. This implies that the service capacity into and out of a port is the same. In theory, the shipping line should have the shipping capacity to reposition empty containers (although in reality it is difficult to achieve); in that sense, empty container repositioning can be treated as a separate problem under the constraints of service network and capacity; (iii) incorporating empty container repositioning into our problem would be mathematically more complicated and difficult to solve. Assumption 3 ensures that container lifting-on/off activities are modelled. By setting the length of a decision stage reasonably short, e.g., 1 day or half a day, vessel arrival and departure are guaranteed to be distinguishable. Assumption 4 ensures shipping companies can acquire adequate vessel capacities if they need. It should be noted that Assumption 4 is only needed when constructing an upper bound in Proposition 5.

We model the problem as a two stage stochastic programming model. Its objective function is given below, in which the first term of the right-hand-side represents the total service capacity cost per period, and the second term represents the job-related costs per period):

$$\textbf{P0} \qquad \min Z(\mathbf{Y}, \mathbf{X}) = \sum_{v} C_v \cdot y_v + \frac{1}{T} E_{\Omega} Q(\mathbf{Y}, \mathbf{X}) \tag{1}$$

The first stage is to minimise the capacity investment, and the second stage is to minimise the expectation of the sum of the shipment transportation costs and transhipment costs and the unfulfilled job penalty costs with respect to random customer demands. For a given realisation of customer demands $\omega(n)$, $Q(\mathbf{Y},\mathbf{X},\omega(n))$ is the optimal value of a linear programming problem. The objective function of the linear programming is to find the cheapest route for each realised customer order (or transportation job) subject to the vessel capacity constraints given in **Y**.

$$Q(\mathbf{Y},\mathbf{X},\omega(n)) = \sum_{j\in J(\omega(n))} D_j \cdot c_t^j \cdot [T - t_j^0 - \sum_t z_{d_j}^j(t)] + \sum_{j\in J(\omega(n))} \sum_p \sum_t D_j \cdot [c_p^o \cdot u_p^j(t) + c_p^f \cdot v_p^j(t)] \tag{2}$$

In Eq. (2), the first term represents the transportation costs that are in proportion to travelling times and the unfulfilled job penalty costs that are in proportion to $T - t^0_j$, and the second term is total lifting-on/off costs.

Constraints

Constraint 1: Constraints related to each $v \in V$;

During the time at port $p(v,i)$, job j's status on vessel v will not change in this duration.

$$x^j_v(t^a_{v,l,i}) = \ldots = x^j_v(t^d_{v,l,i} - 1) \tag{3}$$

During the time at sea between portcall i and portcall i+1, job j's status on vessel v will not change.

$x^j_v(t^d_{v,l,i}) = \ldots = x^j_v(t^a_{v,l,i+1} - 1)$, if portcall i is not the vessel v's final portcall in the loop;

$x^j_v(t^d_{v,l,i}) = \ldots = x^j_v(t^a_{v,l+1,0} - 1)$, if portcall i is vessel v's final portcall in the loop; (4)

Constraint 2: Constraints related to vessel v's each portcall;

At vessel v's arrival period at port $p(v,i)$, i.e. $t^a_{v,l,i}$, the following constraints should be met.

$$x^j_v(t^a_{v,l,i}-1) \geq x^j_v(t^a_{v,l,i}) \qquad \forall\, t^a_{v,l,i} > t^0_j \tag{5}$$

$$\sum_{u\in V^a_{p(v,i)}(t^a_{v,l,i})} x_u^j(t^a_{v,l,i}-1) + \sum_{u\in V^d_{p(v,i)}(t^a_{v,l,i})} x_u^j(t^a_{v,l,i}-1) + z^j_{p(v,i)}(t^a_{v,l,i}-1) = \sum_{u\in V^d_{p(v,i)}(t^a_{v,l,i})} x_u^j(t^a_{v,l,i}) + \tag{6}$$

$$\sum_{u \in V_i^a(t^a_{v,l,i})} x_u^j(t^a_{v,l,i}) + z^j_{p(v,i)}(t^a_{v,l,i}) \qquad \forall\, t^a_{v,l,i} > t^0_j$$

Eq. (5) represents that a shipment on a vessel will remain on board or unloaded from the vessel when the vessel arrives at a port. Eq. (6) represents that the state relationship of shipment j between the time periods $t^a_{v,l,i} - 1$ and $t^a_{v,l,i}$ when the vessel v arrives at port $p(v,i)$. For example, if shipment j is located at port $p(v,i)$ at time period $t^a_{v,l,i} - 1$, then it will either remain at the port $p(v,i)$ or be loaded on one of the departing vessel at time period $t^a_{v,l,i}$, which is reflected by Eqs. (6) and (5). On the other hand, if shipment j is on board of one of the arriving vessel at time period $t^a_{v,l,i} - 1$, then it will either remain on the vessel or be unloaded to the port $p(v,i)$ at time period $t^a_{v,l,i}$.

At vessel v's departure period at port $p(v,i)$, i.e. $t^d_{v,l,i}$:

$$x^j_v(t^d_{v,l,i} - 1) \le x^j_v(t^d_{v,l,i}) \qquad \forall\, t^d_{v,l,i} > t^0_j \tag{7}$$

$$\sum_{u \in V^a_{p(v,i)}(t^d_{v,l,i})} x_u^j(t^d_{v,l,i} - 1) + \sum_{u \in V^d_{p(v,i)}(t^d_{v,l,i})} x_u^j(t^d_{v,l,i} - 1) + z^j_{p(v,i)}(t^d_{v,l,i} - 1) = \sum_{u \in V^d_{p(v,i)}(t^d_{v,l,i})} x_u^j(t^d_{v,l,i}) + \sum_{u \in V_i^a(t^d_{v,l,i})} x_u^j(t^d_{v,l,i}) + z^j_{p(v,i)}(t^d_{v,l,i}), \qquad \forall\, t^d_{v,l,i} > t^0_j \tag{8}$$

Constraint 3: <u>Constraints related to port $p \in P$ at the periods without vessel arrivals or departures:</u>

Suppose t_p is the first event epoch (time period) that a vessel arrives at or departs from port p after the time t^0_j. Then, job j's status at port p will not change before t_p.

$$z^j_p(t) = z^j_p(t^0_j) \qquad \forall\, t^0_j < t < t_p \tag{9}$$

Suppose t_1 and t_2 are two consecutive vessel arrival or vessel departure event epochs at port p. In other words, there is no vessel arrival or departure in the time interval (t_1, t_2). Then, job j's status at port p will not change in this interval:

$$z^j_p(t_1) = z^j_p(t_1+1) = \ldots = z^j_p(t_2 - 1) \qquad \forall\, t_1 > t^0_j; \tag{10}$$

Constraint 4: <u>Constraints of vessel capacity</u>

$$\sum_{j \in J} x_v^j(t) D_j \le y_v \qquad \forall\; v, t \tag{11}$$

Constraint 5: <u>Constraints of job status</u>

$$\sum_{v \in V} x_v^j(t) + \sum_{p \in P} z_p^j(t) = 1, \qquad \forall\, t \ge t^0_j$$
$$x^j_v(t) = 0, \qquad \forall\, j, v, t \le t^0_j \tag{12}$$

$z^j_p(t) = 0, \quad \forall\, j, p, t < t^0_j$

$z^j_{o_j}(t^0_j) = 1;\ z^j_p(t^0_j) = 0$, if $p \neq o_j \quad \forall\, j$

Constraint 6: Constraints of vessel chartering market

$$L_v \leq y_v \leq U_v \quad \forall v \tag{13}$$

Constraint 7: Constraints of promised delivery time of job j (i.e. the fulfilled job must be delivered within T_j time period after its generation),

$$(T - t^0_j) \cdot z^j_{d_j}(T) - \sum_t z^j_{d_j}(t) \leq T_j \quad \forall\, j \tag{14}$$

Constraint 8: Constraints of decision variables;

$$\begin{aligned} & u^j_p(t) + v^j_p(t) \leq 1; && \forall\, t, j, p \\ & v^j_p(t) - u^j_p(t) = z^j_p(t+1) - z^j_p(t) && \forall\, t < \mathrm{H}, j, p \\ & u^j_p(t) = 0 \text{ or } 1 && \forall\, t, j, p \\ & v^j_p(t) = 0 \text{ or } 1 && \forall\, t, j, p \\ & x^j_v(t) = 0 \text{ or } 1 && \forall\, t, j, v \\ & z^j_p(t) = 0 \text{ or } 1 && \forall\, t, j, p \end{aligned} \tag{15}$$

Proposition 1: **P0** is an NP-complete problem.

This can be proved by simplifying the problem **P0** to a knapsack problem.

3 Solution strategy

In the section, three solution methods including SAA (Sample Average Approximation), PHA (Progressive Hedging Algorithm) and APHA (Adapted Progressive Hedging Algorithm) will be proposed to solve the aforementioned model. SAA and PHA are mature methods to solve stochastic programming problems, while APHA is our proposed method tailored for our specific research question based on Lagrangian relaxation.

3.1 SAA method

In the above formulation, $E_\Omega Q(\mathbf{Y}, \mathbf{X})$ is very difficult to calculate. Actually, even the closed form of $E_\Omega Q(\mathbf{Y}, \mathbf{X})$ is hard to obtain. In the study, we use SAA (Sample Average Approximation) to cope with the problem. In SSA scheme, $E_\Omega Q(\mathbf{Y}, \mathbf{X})$ is approximated by $N^{-1}\sum_{n=1}^{N} Q(\mathbf{Y}, \mathbf{X}(\omega(n)), \omega(n))$ that comprises N realised sample processes of customer demands: $\{\omega(1), \omega(2), \ldots, \omega(n), \ldots, \omega(N)\}$, and scenario-dependent decision variables $\mathbf{X}(\omega(n))$.

$N^{-1}\sum_{n=1}^{N} Q(\mathbf{Y},\mathbf{X}(\omega(n)),\omega(n))$ is an unbiased estimator of $E_\Omega Q(\mathbf{Y},\mathbf{X})$ (Dantzig and Thapa, 2003), and will converge to $E_\Omega Q(\mathbf{Y},\mathbf{X})$ with probability 1 as the sample size N goes to infinity, i.e., $P\{\lim_{N\to\infty} N^{-1}\sum_{n=1}^{N} Q(\mathbf{Y},\mathbf{X}(\omega(n)),\omega(n)) = E_\Omega Q(\mathbf{Y},\mathbf{X})\} = 1$ (Ruszczynski and Shapiro, 2003). This result is obtained based on the law of Large Numbers. By substituting $N^{-1}\sum_{n=1}^{N} Q(\mathbf{Y},\mathbf{X}(\omega(n)),\omega(n))$ into **P0**, we can get the following linear programming model:

$$\textbf{P1} \qquad \min_{\mathbf{Y},\mathbf{X}(\omega(n))} Z(\mathbf{Y},\mathbf{X}(\omega(n))) = \sum_v C_v \cdot y_v + \frac{1}{N \cdot T}\sum_{n=1}^{N} Q(\mathbf{Y},\mathbf{X}(\omega(n)),\omega(n)) \tag{16}$$

s.t.

$$\mathbf{AX}(\omega(n)) = \mathbf{B}(\omega(n),\mathbf{Y}) \qquad \text{for } n = 1\ldots N \tag{17}$$

Eq. (16) is the objective function to minimise capacity investment and the average of operational costs related to N different demand realisations. Eq. (17) comprises N copies of Eqs. (3) - (15). Apart from the first stage decision variables $\mathbf{Y}=\{y_v|\forall v\in V\}$, the decision variables in each copy become scenario-dependent decision variables such as $x^j_v(t,\omega(n))$, $z^j_p(t,\omega(n))$, $u^j_p(t,\omega(n))$ and $v^j_p(t,\omega(n))$ and relate to a given sample process of customer demands $\omega(n)$.

If the scale of problem **P1** is not large, it can be solved using standard integer programming method such as branch and cut, which has been well implemented in the commercial optimisation software such as IBM CPLEX or Matlab. In general, however, **P1** unfortunately has a very large number of decision variables and constraints. This is because the scale of **P1** is positively proportional to the sample size N, and N has to be sufficiently large to ensure $N^{-1}\sum_{n=1}^{N} Q(\mathbf{Y},\mathbf{X}(\omega(n)),\omega(n))$ close enough to $E_\Omega Q(\mathbf{Y},\mathbf{X})$. Additionally, each scenario in **P1** has a formulation similar to Eqs.(3) - (15) , which is in fact a capacitated dynamic container routing problem. In other words, **P1** is a combination of ***N*** capacitated dynamic container routing problems. Considering that dynamic routing problem is hard to solve, there is a need to develop efficient solution methods for our problem.

3.2 Progressive Hedging Algorithm (PHA)

An idea to solve the problem like **P1** is to decompose it to a number of smaller problems that are easier to solve. Some methods have been proposed, e.g., L-shaped method (Slyke & Wets, 1969), PHA (Progressive Hedging Algorithm) (Rockafellar & Wets, 1991). As L-shaped method needs to compute the duals of the second stage problem, it would not be suitable for our case because our second stage problem is a standard 0-1 programming. Therefore, we choose PHA to solve our problem.

The logic behind PHA is to decompose problem **P1** into N independent scenario based problems with each modelling container routing problem for a given sample process. In PHA, Lagrangian relaxation is employed to decompose the problem. Prior to the implementation of Lagrangian relaxation, we introduce scenario-dependent decision variables $\mathbf{Y}(\omega(n))=\{y_1(\omega(n)), \ldots, y_v(\omega(n)), \ldots, y_{|V|}(\omega(n))\}(1 < n < N)$, and re-write the original problem.

$$\textbf{P2} \quad \min_{\mathbf{Y},\mathbf{Y}(\boldsymbol{\omega}(n)),\mathbf{X}(\boldsymbol{\omega}(n))} Z(\mathbf{Y},\mathbf{Y}(\boldsymbol{\omega}(n)),\mathbf{X}(\boldsymbol{\omega}(n))) = \frac{1}{N}\sum_{n=1}^{N}[\sum_{v} C_v \cdot y_v(\omega(n)) + \frac{1}{T}Q(\mathbf{Y}(\omega(n)),\mathbf{X}(\omega(n)),\omega(n))] \tag{18}$$

s.t.

$$\mathbf{AX}(\omega(n)) = \mathbf{B}(\omega(n),\mathbf{Y}(\omega(n))) \qquad \forall n \tag{19}$$

$$y_v(\omega(n)) = y_v \qquad \forall\ n, v \tag{20}$$

$$L_v \le y_v(\omega(n)) \le U_v \qquad \forall n, v \tag{21}$$

It should be noted that the newly added variables do not affect the optimal solution, thus **P2** is equivalent to **P1**.

By dropping off the constant coefficient $1/N$, and moving nonanticipativity constraints into the objective function based on Lagrangian relaxation method, we can have

$$\textbf{P3} \quad \max_{\lambda} \min_{\mathbf{Y},\mathbf{Y}(\boldsymbol{\omega}(n)),\mathbf{X}(\boldsymbol{\omega}(n))} Z(\mathbf{Y},\mathbf{Y}(\boldsymbol{\omega}(n)),\mathbf{X}(\boldsymbol{\omega}(n)),\lambda) = \sum_{n=1}^{N}\sum_{v=1}^{|V|}\lambda(n,v)\cdot\left|y_v(\omega(n)) - y_v\right| + + \sum_{n=1}^{N}\left[\sum_{v} C_v \cdot y_v(w(n)) + \frac{1}{T}Q(\mathbf{Y}(\omega(n)),\mathbf{X}(\omega(n)),\omega(n))\right] \tag{22}$$

s.t.

$$\lambda(n, v) \ge 0 \qquad \forall\ n, v \tag{23}$$

Eq. (19) – (21)

In the above formulation, to simplify the computer programming, we use the absolute value of the difference between scenario-dependent variables and first-stage decision variables times Lagrangian multipliers to relax non-anticipativity constraints instead of Augmented

Lagrangian method suggested by Rockafellar & Wets (1991) who firstly proposed PHA. The method has been used in another study by Long et al. (2012)

P3 is separable on a scenario base. As it contains N scenarios, it can be broken down into N individual sub-problems. An arbitrary sub-problem indexed by $n \in (1, N)$ has the following form,

$$\textbf{P4} \quad \max_{\boldsymbol{\lambda}} \min_{\mathbf{Y},\mathbf{Y}(\boldsymbol{\omega}(n)),\mathbf{X}(\boldsymbol{\omega}(n))} Z_n(\mathbf{Y},\mathbf{Y}(\boldsymbol{\omega}(n)),\mathbf{X}(\boldsymbol{\omega}(n)),\boldsymbol{\lambda}) = \sum_{v=1}^{|V|} \lambda(n,v)\cdot\left|y_v(\omega(n)) - y_v\right| + \sum_v C_v \cdot y_v(\omega(n)) + \frac{1}{T} Q(\mathbf{Y}(\omega(n)),\mathbf{X}(\omega(n)),\omega(n)) \tag{24}$$

s.t.

$$\mathbf{AX}(\omega(n)) = \mathbf{B}(\omega(n),\mathbf{Y}(\omega(n))) \tag{25}$$

$$\lambda(n, v) \geq 0 \qquad \forall\, v \tag{26}$$

It is noted that **P4** is nonlinear due to the first term in the objective function. We introduce auxiliary variables, $\mathbf{a} = \{a_v \mid v \in V\}$ and $\mathbf{a}' = \{a_v' \mid v \in V\}$ to linearise the absolute value in Eq. (24), we can get the following problem

$$\textbf{P5} \quad \max_{\boldsymbol{\lambda}} \min_{\mathbf{Y},\mathbf{Y}(\boldsymbol{\omega}(n)),\mathbf{X}(\boldsymbol{\omega}(n))} Z_n(\mathbf{Y},\mathbf{Y}(\boldsymbol{\omega}(n)),\mathbf{X}(\boldsymbol{\omega}(n)),\boldsymbol{\lambda},\mathbf{a},\mathbf{a}') = \sum_{v=1}^{|V|} \lambda(n,v)\cdot(a_v + a_v') + \sum_v C_v \cdot y_v(\omega(n)) + \frac{1}{T} Q(\mathbf{Y}(\omega(n)),\omega(n)) \tag{27}$$

s.t.

$$AX(\omega(n)) = B(\omega(n),\mathbf{Y}(\omega(n))) \tag{28}$$

$$y_v(\omega(n)) - y_v = a_v - a_v' \qquad \forall\, v \tag{29}$$

$$a_v \geq 0,\ a_v' \geq 0 \qquad \forall\, v \tag{30}$$

$$\lambda(n, v) \geq 0 \qquad \forall\, v \tag{31}$$

According to the solution to **P5**, an approximated costs for **P1** can be calculated, i.e.,

$$\widehat{Z}(\mathbf{Y},\mathbf{X}(\omega(n)) = Z(\mathbf{Y},\mathbf{Y}(\boldsymbol{\omega}(n)),\mathbf{X}(\boldsymbol{\omega}(n)),\boldsymbol{\lambda}) = \frac{1}{N}\sum_{n=1}^{N} Z_n(\mathbf{Y},\mathbf{Y}(\boldsymbol{\omega}(n)),\mathbf{X}(\boldsymbol{\omega}(n)),\boldsymbol{\lambda},\mathbf{a},\mathbf{a}') \tag{32}$$

Proposition 2: (i) When $\lambda(n,v)=0\,(\forall n,\ v)$, $\frac{1}{N}\sum_{n=1}^{N} Z_n(\mathbf{Y},\mathbf{Y}(\boldsymbol{\omega}(n)),\mathbf{X}(\boldsymbol{\omega}(n)),\lambda)$ is a lower bound to $Z(\mathbf{Y},\mathbf{X}(\omega(n)))$ in **P1**; (ii) $\frac{1}{N}\sum_{n=1}^{N} Z_n(\mathbf{Y},\mathbf{Y}(\boldsymbol{\omega}(n)),\mathbf{X}(\boldsymbol{\omega}(n)),\lambda)$ converges to an upper bound to **P1** as $\lambda(n,v)$ is sufficiently large. There exists the following relationship:

$$\begin{aligned}&\frac{1}{N}\sum_{n=1}^{N}\min\{Z_n(\mathbf{Y},\mathbf{Y}(\boldsymbol{\omega}(n)),\mathbf{X}(\boldsymbol{\omega}(n)),\mathbf{0})\}\le Z^*(\mathbf{Y},\mathbf{X}(\omega(n)))\\ &\le\frac{1}{N}\sum_{n=1}^{N}\min\{Z_n(\mathbf{Y},\mathbf{Y}(\boldsymbol{\omega}(n)),\mathbf{X}(\boldsymbol{\omega}(n)),\lambda')\}\ (\lambda'\text{ represents a sufficiently large }\lambda)\end{aligned} \tag{33}$$

Proof: When $\lambda(n,v)=0\,(\forall n,\ v)$, each scenario can choose the best vessel capacity for itself, therefore, the sum of the minimised costs over all the scenarios will be lower than the original problem **P1** where all the scenarios must have the same vessel capacity. When λ is sufficiently large, it forces $\sum_{v=1}^{|V|}\lambda(n,v)\cdot(a_v+a'_v)$ to be zero. Thus we can have $|\mathbf{Y}-\mathbf{Y}(\boldsymbol{\omega}(n))|=0$, which is a feasible solution to **P1**, and consequently lead to the upper bound. This completes the proof.

According to Proposition 2, an efficient way to update λ can be designed. Initially, we set $\lambda(n,v)=0\ (\forall n,\ v)$, and then increase the value of $\lambda(n,v)$. The increment of $\lambda(n,v)$ is positively proportional to the absolute value of the difference between $\mathbf{Y}(\boldsymbol{\omega}(n))$ and its average value of $\bar{\mathbf{Y}}$. A more detailed description of the algorithm is described as follows.

Algorithm 1: Progressive Hedging Algorithm

Step 1: Initialisation. Set $\lambda(n,v)=0$, $(\forall\ n,\ v)$; iteration number $k=0$; and assign a constant to $\rho^{(0)}$, and another constant greater than 1 to α;

Step 2: Solve **P5** for each scenario, and obtain the scenario dependant solution for the k^{th} iteration, $\mathbf{Y}^{(k)}(\omega(n))=\{\,y_v^{(k)}(\omega(n))|\ v=1,\ \ldots,\ |V|\}$, and the corresponding optimal value of objective function, $Z_n^{(k)}$;

Step 3: Compute the reference point, $\bar{\mathbf{Y}}^{(\boldsymbol{k})}=\{\ \bar{y}_v^{(k)}(\omega(n))|\ v=1,\ \ldots,\ |V|\}$, where $\bar{y}_v^{(k)}=\frac{1}{N}\sum_{n=1}^{N} y_v^k(\omega(n))$;

Step 4: The algorithm stops if either of the following criteria is satisfied:

a. $\sum_{n=1}^{N}\sum_{v=1}^{|V|}\left|y_v^{(i)}(\omega(n))-\bar{y}_v^{(i)}\right| \le \eta$, where η is a small positive number.

b. There is no improvement in recent L steps

Where, η and L are the pre-specified control parameters.

Step 5: Update Lagrangian multipliers using the following equation:

$$\lambda^{(i+1)}(n,v)= \lambda^{(i)}(n,v) + \rho^{(i+1)}|\, y_v^{(i)}(\omega(n)) - y_v^{(i)}\,| \qquad \text{where, } \rho^{(i+1)} = \alpha\, \rho^{(i)} \tag{34}$$

Step 6: i = i+1, and go to (2)

It should be noted that Algorithm 1 decomposes a large-scale problem into a number of smaller scenario-based problems, which can produce near-optimal solutions (Rockafellar & Wets 1991). Our numerical experiments also confirm that the PHA can achieve a very high accuracy when the decomposed problems are solvable.

3.3 Adapted Progressive Hedging Algorithm (APHA)

The above progressive hedging strategy can decompose a large stochastic programming problem (e.g. when there are many samples in the SSA model) into a number of smaller scenario-based problems. Therefore, it is very helpful to solve the problem that contains a large number of samples. However, in many cases, even the problem for an individual sample has a large number of variables and constraints that are beyond the capability of PHA. The problem we are dealing with actually is one of them. Each decomposed problem, i.e., **P5**, still contains a capacitated dynamic routing problem, which can be difficult to solve for large shipping networks. Unfortunately, the existing literature in relation to stochastic programming does not give a solution to the issue as they mainly focuses on how to decompose SSA model into scenario-based sub-problems, e.g., the aforementioned PHA and the famous L-Shaped method (Slyke & Wets, 1969). In this section, we will develop a new approach to cope with the issue. Our approach is along the same line as PHA. Its main idea is to decompose the scenario-based problem obtained in PHA into smaller job (customer order) based problems using Lagrangian relaxation once again. Therefore, we term the approach as Adapted Progressive Hedging Algorithm (APHA). APHA can be used for the situation where PHA cannot work due to the large-scale of a single scenario or sample process.

The key issue in APHA is to determine the tight lower bound and upper bound to original problem **P1**. The overall procedure of the APHA can be regarded as a two-phase procedure. In the first phase, we focus on the lower bound. The way to obtain the lower bound in APHA is slightly different from that in PHA. In PHA, only non-anticipativity constraints are relaxed whereas, in APHA, both the capacity constraint and non-anticipativity constraints will be relaxed. Initially, arbitrary Lagrangian multipliers, e.g., 0, are used to obtain a loose lower bound. By updating the Lagrangain multipliers using the subgradient procedure (Fisher, 2004), the lower bound will become tighter. When changing the Lagrangian multipliers cannot improve the lower bound any more, the searching procedure for lower bound stops. The finally obtained lower bound can be used as an estimate of the optimal value of **P1**.

However, the lower bound may not provide a feasible solution since some constraints have been relaxed and moved to the objective function. Therefore, we need to search for a good feasible solution and obtain a tight upper bound, which is the focus of the second phase of the procedure. Our approach here is to tweak the solution corresponding to the lower bound to make it feasible. During the process, we will follow some mathematically proved principles. If the obtained feasible solution is not good enough, a special procedure called Lagrangian Costs Guided Gradient Search (LCGGS) will be followed to further improve the quality of feasible solution and seek a tighter upper bound. The LCGGS is similar to normal gradient search method except that the Lagrangian-relaxed problems instead of the original problem will be used to calculate the gradients. The LCGGS will stop when there is no improvement in a certain number of iterations. After the procedure described above, we can obtain both upper bound and lower bound, and calculate the gap between them and measure the performance of our algorithm.

3.3.1 The relaxed problems

To simplify the narrative, we drop n and $\omega(n)$ from **P4**, introduce y'_v and $\mathbf{Y}'$ to replace the scenario-dependent symbol $\mathbf{Y}(\omega(n))$ and $y(\omega(n))$, and substitute $\boldsymbol{Q(\cdot)}$ with Eq.(2), then we get

P6

$$\max_{\boldsymbol{\lambda}} \min_{\mathbf{Y}',\mathbf{Y},\mathbf{X}} Z(\mathbf{Y}',\mathbf{Y},\mathbf{X},\lambda) = \sum_{v} C_v \cdot y'_v + \sum_{v=1}^{|V|} \lambda(v)\cdot \left|y'_v - y_v\right| + \frac{1}{T}\{\sum_j D_j \cdot c_t^j \cdot [T - t_j^0 - \sum_t z_{d_j}^j(t)] + \sum_j \sum_p \sum_t [c_p^o \cdot u_p^j(t) + c_p^f \cdot v_p^j(t)]\} \tag{35}$$

s.t.

$$\mathbf{AX} = \mathbf{B}(\mathbf{Y}') \tag{36}$$

$$\lambda(v) \geq 0 \qquad \forall\, v \tag{37}$$

We move the capacity constraints in Eq.(36) whose explicit form was given in Eq. (11) into the objective function of **P6**, then we can have,

P7

$$\max_{\boldsymbol{\lambda},\boldsymbol{\gamma}} \min_{\mathbf{Y}',\mathbf{Y},\mathbf{X}} Z(\mathbf{Y}',\mathbf{Y},\mathbf{X},\boldsymbol{\lambda},\boldsymbol{\gamma}) = \sum_{v} C_v \cdot y'_v + \sum_{v=1}^{|V|} \lambda(v) \cdot \left| y'_v - y_v \right| + \sum_{v}\sum_{t=1}^{T} \gamma(v,t) \cdot (\sum_{j\in J} x_v^j(t) D_j - y'_v)$$
$$+ \frac{1}{T}\{\sum_j D_j \cdot c_t^j \cdot [T - t_j^0 - \sum_t z_{d_j}^j(t)] + \sum_j\sum_p\sum_t [c_p^o \cdot u_p^j(t) + c_p^f \cdot v_p^j(t)]\} \tag{38}$$

s.t.

$$\mathbf{A}'\mathbf{X} = \mathbf{B}' \tag{39}$$

$$L_v \leq y'_v \leq U_v \qquad \forall\, v \tag{40}$$

$$L_v \leq y_v \leq U_v \qquad \forall\, v \tag{41}$$

$$\lambda(v) \geq 0 \qquad \forall\, v \tag{42}$$

$$\gamma(v, t) \geq 0 \qquad \forall\, v, t \tag{43}$$

It is noted that $\mathbf{A}'$ and $\mathbf{B}'$ were introduced in Eq.(39) to reflect the change of relaxing the vessel capacity constraints, and that $\boldsymbol{B}'$ is not dependent on $\mathbf{Y}'$ as the constraints related to $\mathbf{Y}'$ have been either moved to the objective function or written explicitly in Eq.(40). $\gamma(v,t)$ ($\forall v$, t) are the corresponding Lagrangian multipliers for a given single scenario. To be more accurate, the Lagrangian multipliers corresponding to capacity constraints should be denoted as $\gamma(n, v, t)$. Here, to simplify our narrative, we have dropped off n and limit our discussion in a single scenario.

After removing the constants $\frac{1}{T}\{\sum_j D_j \cdot c_t^j \cdot [T - t_j^0]\}$ in Eq. (38) , we will have the following problem.

P8

$$\max_{\boldsymbol{\lambda},\boldsymbol{\gamma}} \min_{\mathbf{Y}',\mathbf{Y},\mathbf{X}} Z'(\mathbf{Y}',\mathbf{Y},\mathbf{X},\boldsymbol{\lambda},\boldsymbol{\gamma}) = \sum_{v} C_v \cdot y'_v + \sum_{v=1}^{|V|} \lambda(v) \cdot \left| y'_v - y_v \right| - \sum_{v}\sum_{t=1}^{T} \gamma(v,t) \cdot y'_v$$
$$+ \sum_{v}\sum_{t=1}^{T}\sum_{j\in J} \gamma(v,t) x_v^j(t) D_j - \frac{1}{T}\sum_j\sum_t c_t^j \cdot D_j \cdot z_{d_j}^j(t) + \frac{1}{T}\sum_j\sum_p\sum_t D_j \cdot [c_p^o \cdot u_p^j(t) + c_p^f \cdot v_p^j(t)] \tag{44}$$

s.t. (39) – (43)

It can be observed that Eq.(44) can be divided into two groups: **X** related terms, and **Y** and **Y**' related items, thus **P8** can be rewritten as:

$$\max_{\boldsymbol{\lambda},\boldsymbol{\gamma}} \min_{\mathbf{Y}',\mathbf{Y},\mathbf{X}} Z'(\mathbf{Y}',\mathbf{Y},\mathbf{X},\boldsymbol{\lambda},\boldsymbol{\gamma}) = \max_{\boldsymbol{\lambda},\boldsymbol{\gamma}} \{ \min_{\mathbf{Y}',\mathbf{Y}} Z_{(1)}(\mathbf{Y}',\mathbf{Y},\boldsymbol{\lambda},\boldsymbol{\gamma}) + \min_{\mathbf{X}} Z_{(2)}(\mathbf{X},\boldsymbol{\gamma}) \} \tag{45}$$

The explicit forms of $\min_{\mathbf{Y}',\mathbf{Y}} Z_{(1)}(\mathbf{Y}',\mathbf{Y})$ and $\min_{\mathbf{X}} Z_{(2)}(\mathbf{X})$ will lead to two independent set of optimisation problems, **P9** and **P10**, as described below.

P9 $$\min_{\mathbf{Y}',\mathbf{Y}} Z_{(1)}(\mathbf{Y}',\mathbf{Y},\boldsymbol{\lambda},\boldsymbol{\gamma}) = \min_{\mathbf{Y}',\mathbf{Y}} \sum_{v=1}^{|V|} \lambda(v) \cdot |y'_v - y_v| + \sum_v y'_v \cdot [C_v - \sum_{t=1}^{T} \gamma(v,t)] \tag{46}$$

s.t. (40) – (43)

As $\lambda(v)$ increases, $\sum_{v=1}^{|V|} \lambda(v) \cdot |y'_v - y_v|$ in Eq. (46) will approach to 0 eventually, which will ensure that all the scenarios have the same vessel capacities. P9 can be solved using the same solution strategy introduced in Section 3.2. The main idea of the strategy is adding auxiliary variables like $\mathbf{a} = \{a_v \mid v \in V\}$ and $\mathbf{a}' = \{a_v' \mid v \in V\}$ to linearise **P9,** and using $\frac{1}{N}\sum_{n=1}^{N} y'_v$ to estimate y_v.

P10 $$\min_{\mathbf{X}} Z_{(2)}(\mathbf{X},\boldsymbol{\gamma}) = \min_{\mathbf{X}} \sum_v \sum_{t=1}^{T} \sum_{j \in J} \gamma(v,t) x_v^j(t) D_j - \frac{1}{T}\sum_j \sum_t c_t^j \cdot D_j \cdot z_{d_j}^j(t) + \frac{1}{T}\sum_j \sum_p \sum_t D_j \cdot [c_p^o \cdot u_p^j(t) + c_p^f \cdot v_p^j(t)] \tag{47}$$

s.t. (39) and (43)

P10 can be broken down into $|J|$ independent sub-problems as there are no correlations between jobs (customer demands) in Eq.(39). Each individual sub-problem has the following structure,

P11 $$\min_{\mathbf{X}} Z^j(\mathbf{X},\boldsymbol{\gamma}) = \sum_v \sum_{t=1}^{T} \gamma(v,t) x_v^j(t) - \frac{1}{T}\sum_t c_t^j \cdot z_{d_j}^j(t) + \frac{1}{T}\sum_p \sum_t [c_p^o \cdot u_p^j(t) + c_p^f \cdot v_p^j(t)] \tag{48}$$

s.t. (3) – (10), (12),(14), (15)

It should be noted that D_j has been removed from the objective function in **P11** as it is the common coefficient for each item in the objective function.

P11 is a dynamic shortest path problem for a given set of $\{\gamma(v,t) | \forall v, t\}$ if $\gamma(v,t)$ is treated as the cost for using vessel v at time t. It has the following properties.

Proposition 3: If r is a possible path for a transportation job j, then it is always not optimal to use part of path r and leave transportation job j halfway unfinished.

Proof: Let $Z^j(0,\gamma)$ denote the value of objective function when job j is not serviced; and $Z^j(\mathbf{X}_r,\gamma)$ the value of objective function when path r is selected to transport job j. Clearly, we have $Z^j(0,\gamma)$ = 0 from (48). If the job j carried on the path r did not reach the final destination port at the end of planning horizon, we would have $z^j_{d_j}(r,t)=0$ for any t. It follows that $Z^j(\mathbf{X}_r,\gamma) > 0$ by (48). Therefore, leaving the transportation job j unfinished en route is worse than not servicing it in the first place. This completes the proof.

Proposition 3 reveals that partial use of a path and uncompleted transportation job should not be included in the optimal solution, and job j should be either left at the original port or be delivered to the destination port before the planning horizon. By excluding the partial use of a path that can serve job j, the space of feasible solutions can be significantly reduced.

Proposition 4: If path r is chosen to serve job j in the optimal solution to **P11**, then r satisfies the following conditions:

(i) $Z^j(\mathbf{X}_r,\gamma) = \sum_v \sum_{t=1}^{H} \gamma(v,t) x^j_v(p,t) - c_t \cdot \sum_t z^j_{d_j}(p,t) + \frac{1}{T}\sum_p \sum_t [c^o_p \cdot u^j_p(p,t) + c^f_p \cdot v^j_p(p,t)] \le 0$

(ii) $r = argmin_r\{ \sum_v \sum_{t=1}^{H} \gamma(v,t) x^j_v(p,t) - c_t \cdot \sum_t z^j_{d_j}(p,t) + \frac{1}{T}\sum_p \sum_t [c^o_p \cdot u^j_p(p,t) + c^f_p \cdot v^j_p(p,t)] \}$

Condition (i) follows from Proposition 3, which ensures that choosing path r outperforms not servicing the job; and condition (ii) ensures that path r minimises the objective function of **P11** among all the paths.

Let λ^* and γ^* be the optimal Lagrange multipliers of **P8**. Let $Z^*_{(1)}(\mathbf{Y}',\mathbf{Y},\boldsymbol{\lambda}^*,\boldsymbol{\gamma}^*)$ be the optimal cost of **P9**, $Z^{j^*}(\mathbf{X},\boldsymbol{\gamma}^*)$ denote the optimal cost of problem **P11,** and $x^{j^*}_v(t)$ be the corresponding optimal value of $x^j_v(t)$ ($v \in V$, $1 \le t \le T$). In addition, let $y_v{}^U$ = $\max\{\max[\sum_{j\in J(\omega(n))} x^{j^*}_v(t)\cdot D_j \mid 1\le n \le N, 1\le t \le T\], L_v\}$. A lower bound and an upper bound to the original problem **P1**, $Z^L(\mathbf{Y},\mathbf{X}(\omega(n)))$ and $Z^U(\mathbf{Y},\mathbf{X}(\omega(n)))$, respectively, can be obtained using the following proposition.

Proposition 5: $Z^L(\mathbf{Y},\mathbf{X}(\omega(n))) = \frac{1}{N}\ [\ \sum_{\omega(n)} Z^*_{(1)}(\mathbf{Y}',\mathbf{Y},\boldsymbol{\lambda}^*,\boldsymbol{\gamma}^*) + \sum_{\omega(n)} \sum_{j\in J_{\omega(n)}} D_j \cdot [Z^{j^*}(\mathbf{X},\gamma^*) + \frac{1}{T} c_t^j (T - t_j^0)]\ ]$ is a lower bound for **P1**; $Z^U(\mathbf{Y},\mathbf{X}(\omega(n))) = \frac{1}{N}\sum_{n=1}^{N}\sum_{j\in J_{\omega(n)}} D_j \cdot [Z^j(\mathbf{X}^*,\gamma^*) + \frac{1}{T} c_t^j (T - t_j^0)] + \sum_v C_v \cdot y_v^U$ is an upper bound for **P1**.

Proof: For the first part, as the average value of all the optimal solutions to **P9** and **P11** will be the optimal solution to the Lagrangian relaxation based problem **P8**, it will then construct a lower bound to the original problem after adding the constants $\frac{1}{T} D_j c_t^j (T - t_j^0)$ that has been removed from **P7**. For the second part, according to the definition of y_v^U ($\forall\ v\in V$), they are the minimum sufficient capacities that can ensure all the optimal solutions to **P11** to be served, hence $Z^U(\mathbf{Y},\mathbf{X}(\omega(n))) = \frac{1}{N}\sum_{n=1}^{N}\sum_{j\in J_{\omega(n)}} D_j \cdot Z^j(\mathbf{X}^*,\gamma^*) + \sum_v C_v \cdot y_v^U$ is an upper bound. This completes the proof.

Following Proposition 5, we can construct a good estimate of the optimal value of **P1** as $\hat{Z}(\mathbf{Y},\mathbf{X}(\omega(n))) = \frac{1}{2}\ [\ Z^L(\mathbf{Y},\mathbf{X}(\omega(n))) + Z^U(\mathbf{Y},\mathbf{X}(\omega(n)))\ ]$.

Lemma 1: When $\boldsymbol{\gamma} = 0$, $Z^{L-SP}(\mathbf{Y},\mathbf{X}(\omega(n))) = \frac{1}{N}\ [\ \sum_{\omega(n)}\sum_v C_v \cdot L_v + \sum_{\omega(n)}\sum_{j\in J_{\omega(n)}} D_j \cdot [Z^{j^*}(\mathbf{X},\mathbf{0}) + \frac{1}{T} c_t^j (T - t_j^0)]\ ]$ and $Z^{U-SP}(\mathbf{Y},\mathbf{X}(\omega(n))) = \frac{1}{N}\sum_{n=1}^{N}\sum_{j\in J_{\omega(n)}} D_j \cdot [Z^j(\mathbf{X}^*,0) + \frac{1}{T} c_t^j (T - t_j^0)] + \sum_v C_v \cdot y_v^{U-SP}$ is a lower bound and an upper bound for P1, respectively .

Proof: $\boldsymbol{\gamma} = \mathbf{0}$ is a special case for Proposition 5. Since $C_v - \sum_{t=1}^{T}\gamma(v,t) = C_v > 0$, $y_v^* = L_v$. Therefore, $Z^L(\mathbf{Y},\mathbf{X}(\omega(n)))$ will be reduced to $Z^{L-SP}(\mathbf{Y},\mathbf{X}(\omega(n))) = \frac{1}{N}\ [\ \sum_{\omega(n)}\sum_v C_v \cdot L_v + \sum_{\omega(n)}\sum_{j\in J_{\omega(n)}} D_j \cdot [Z^{j^*}(\mathbf{X},\mathbf{0}) + \frac{1}{T} c_t^j (T - t_j^0)]\ ]$. In the situation, finding the solution to **P11** is equivalent to obtaining the shortest path for all the jobs in ***J*** without capacity constraint. Therefore, $Z^{U-SP}(\mathbf{Y},\mathbf{X}(\omega(n))) = \frac{1}{N}\sum_{n=1}^{N}\sum_{j\in J_{\omega(n)}} D_j \cdot [Z^j(\mathbf{X}^*,0) + \frac{1}{T} c_t^j (T - t_j^0)] + \sum_v C_v \cdot y_v^{U-SP}$ is an upper bound to

P1, where $y_v^{U-SP} = \max\{\max[\sum_{j\in J(\omega(n))} x_v^{j^*}(t)\cdot D_j$ $(1< n <N,\ 1< t <T)],\ L_v\}$, is the minimum sufficient capacities when $\boldsymbol{\gamma} = \mathbf{0}$. This completes the proof.

3.3.2 The lower bound of relaxed problems

To determine the lower bound specified in Proposition 5, we need to find the optimal Lagrangian multipliers λ^* and γ^*. We use subgradient procedure (Fisher, 2004) to update the Lagrangian multipliers. The detailed algorithm procedure is described below.

Algorithm 2: Lower Bound of P2

Step 1: Initialisation. Set $\boldsymbol{\gamma} = \mathbf{0}$, i.e., $\gamma(n,v)=0(\forall\ n, v)$; a constant $\alpha(0<\alpha<1)$;

Step 2: Set iteration number $k = 0$. Allocate constants to $\tau^{(0)}(v,t)(\forall\ v, t)$ and $\rho^{(0)}$. Solve **P9** $(\forall\omega(n))$ and **P11**$(\forall\ j\in \boldsymbol{J}(\omega(n)),\ 1 \leqslant n \leqslant N)$ and obtain $Z^{L^{(k)}}(\mathbf{Y},\mathbf{X}(\omega(n))) = Z^{L-SP}(\mathbf{Y},\mathbf{X}(\omega(n)))$, and $Z^{U-SP}(\mathbf{Y},\mathbf{X}(\omega(n)))$ according Proposition 5 and Lemma 1.

Step 3: $k = k+1$; $\rho^{(k)} = \alpha\,\rho^{(k-1)}$; $\tau^k(v,t) = \alpha\,\tau^{k-1}(v,t)\cdot$

Step 4: Update Lagrangian multipliers $\gamma(v,t)$ and $\lambda(n,v)$

$$\gamma^{(k)}(v,t) = \max[0,\ \gamma^{(k-1)}(v,t) + t^{(k)}(v,t)\cdot(\sum_{j\in J} x_v^j(t)D_j - y_v)] \qquad \forall\ v, t$$

$$\text{where, } t^k = \tau^k(v,t)\cdot \frac{Z^{U-SP}(\mathbf{Y},\mathbf{X}(\boldsymbol{\omega}(n))) - Z^{L^{(k-1)}}(\mathbf{Y},\mathbf{X}(\boldsymbol{\omega}(n)))}{\sum_{n=1}^{N}\sum_{v\in V}\sum_{t=1}^{T}\left\|\sum_{j\in J} x_v^j(t)D_j - y_v\right\|^2} \tag{49}$$

$$\lambda^{(k)}(n,v) = \lambda^{(k-1)}(n,v) + \rho^{(k)}\left|\,y_v^{(k-1)}(\omega(n)) - y_v^{(k-1)}\right| \tag{50}$$

Step 5: Solve **P9**$(\forall\omega(n))$ and **P11**$(\forall j\in \boldsymbol{J}(\omega(n)))$ based on new updated $\gamma^{(k)}(v,t)$ and $\lambda^{(k)}(n,v)$, and obtain updated $Z^{L^{(k)}}(\mathbf{Y},\mathbf{X}(\omega(n)))$.

Step 6: Go to Step 3 unless one of the following termination criteria is satisfied:

a. $|Z^{L^{(k)}}(\mathbf{Y},\mathbf{X}(\omega(n))) - Z^{L^{(k-1)}}(\mathbf{Y},\mathbf{X}(\omega(n)))| < \epsilon^1$, where ϵ^1 is a pre-determined error bound;

b. Any $\tau^k(v,t) < \epsilon^2$, where ϵ^2 is a small positive number;

c. There is no improvement in recent L consecutive iterations, where L is predetermined control parameters.

3.3.3 The upper bound of relaxed problems

In the section, we will firstly present an upper bound, and then discuss how to improve the upper bound when the performance of heuristics bound is not satisfactory.

Upper bound

After solving the relaxed problem **P11**, apply the following procedure to obtain a heuristic upper bound:

1) According to Proposition 3, remove the jobs which have not arrived at destination ports from the solution to **P11**;
2) According to Proposition 4, remove the jobs which do not satisfy condition (1) in Proposition 4;
3) Derive an upper bound for **P1** based on the rest of solutions to **P11** according to Proposition 5.

Note that the above upper bound is obtained by tweaking the solution to the Lagrangian relaxation based problem so that it becomes a good feasible solution to the original problem, which is a common approach in the literature. The heuristics method has advantage on computational time. However, its gap to the lower bound might not be satisfactory in some cases. In the section, we will propose a procedure to further reduce the gap when it is not satisfactory.

Lagrangian Costs Guided Gradient Search (LCGGS)

The procedure was inspired by the stochastic quasigradient methods (Ermoliev, 1983; Gaivoronski, 1988; Birge & Louveaux, 2011). We made some changes to the original quasigradient procedure to avoid solving the large ILP model comprising Eqs. (2) – (15) as it is quite difficult to solve for the large shipping network. Our method is to relax the capacity constraint, and use the maximised Lagrangian costs to estimate true costs and then descent gradient with respect to y_v ($\forall\ v$).

LCGGS starts from a known position k denoted by {$\mathbf{Y}^k$, $Z^k(\mathbf{Y}^k, \mathbf{X}^k(\omega(n)))$}, and searches for the next point with lower costs. The gradient at position k will be needed to search for the next position. This involves calculating the partial derivative at position {$\mathbf{Y}^k$, $Z^k(\mathbf{Y}^k, \mathbf{X}^k(\omega(n)))$}, denoted by

$$\nabla Z^k(\mathbf{Y}, \mathbf{X}(\omega(n))) = \frac{Z^k(\mathbf{Y}^k + \Delta\mathbf{Y}, \mathbf{X}^k(\omega(n))) - Z^k(\mathbf{Y}^k, \mathbf{X}^k(\omega(n)))}{\Delta\mathbf{Y}} \quad (51)$$

This formula requires to calculate $Z^k(\mathbf{Y}^k+\Delta\mathbf{Y},\mathbf{X}^k(\omega(n)))$ } for perturbed $\mathbf{Y}^k$. As $(\mathbf{Y}^k+\boldsymbol{\Delta}\mathbf{Y})$ is known in this situation, the problem **P1** is reduced to a set of separated ILP problems. Each problem has a formulation comprising Eqs. (2)-(15) but with different realised demand data. For the small-scale problem, the exact solution to the scenario level model can be obtained, hence, $Z^k(\mathbf{Y}+\Delta\mathbf{Y},\mathbf{X}(\omega(n)))$ can be measured accurately. However, when shipping network is large, the scenario level model cannot be solved. As the paper aims to solve relatively large scale of shipping network for which the exact solution cannot be obtained using standard ILP solution method, we adopt Lagrangian relaxation to decompose the scenario level model comprising Eqs. (2) – (15) into job based problems. The relaxed problem can be formulated as (assume $y_{v'}$ is perturbed to be $y_{v'} + \Delta$ in $\mathbf{Y} + \Delta\mathbf{Y}$),

P12

$$\begin{aligned}\max_{\gamma}\min_{\mathbf{X}} Z(\mathbf{Y}+\Delta\mathbf{Y},\mathbf{X},\boldsymbol{\gamma}) = {} & \sum_{v\neq v'} C_v\cdot y_v + C_{v'}\cdot(y_{v'}+\Delta) - \sum_{v\neq v'}\sum_{t=1}^{T}\gamma(v,t)\cdot y_v \\ & - \sum_{t=1}^{T}\gamma(v',t)\cdot(y'_v+\Delta) + \frac{1}{T}\sum_j c_t^j\cdot D_j\cdot(T-t_j^0) - \frac{1}{T}\sum_j\sum_t c_t^j\cdot D_j\cdot z_{d_j}^j(t) \\ & + \frac{1}{T}\sum_j\sum_p\sum_t D_j\cdot[c_p^o\cdot u_p^j(t)+c_p^f\cdot v_p^j(t)] + \sum_v\sum_{t=1}^{T}\sum_{j\in J}\gamma(v,t)x_v^j(t)D_j\end{aligned} \quad (52)$$

In the formulation, the first five terms are constants. The rest of items can be decomposed into a number of job-based problems, and each of them will have the same formulation as **P11**. We use the optimal solution to **P12**, $Z(\mathbf{Y}+\Delta\mathbf{Y},\mathbf{X}^*,\boldsymbol{\gamma}^*)$ to estimate $Z^k(\mathbf{Y}+\Delta\mathbf{Y},\mathbf{X}(\omega(n)))$, Eq. (51) can be rewritten as,

$$\nabla Z^k(\mathbf{Y},\mathbf{X}(\omega(n))) = \frac{Z^k(\mathbf{Y}+\Delta\mathbf{Y},\mathbf{X}^*,\boldsymbol{\gamma}^*) - Z^k(\mathbf{Y},\mathbf{X}^*,\gamma^*)}{\Delta\mathbf{Y}} \quad (53)$$

Once the gradient is determined, the next searching position can be easily determined. The details of the LCGGS procedure are described in Algorithm 3.

Algorithm 3: LCGGS

Step 1: Initialisation. Set iteration number $k = 0$; $y_v{}^k = y_v{}^U$ $(\forall v \in V)$; $Z^k(\mathbf{Y}^k,\mathbf{X}^k(\omega(n)))\,|_{k=0} = Z^U(\mathbf{Y},\mathbf{X}(\boldsymbol{\omega}(n)))$; the best-so-far solution $\mathbf{Y}^{best} = \{y_v{}^{best} = y_v{}^U \mid \forall v \in V\}$, $Z^{best}(\mathbf{Y},\mathbf{X}(\omega(n))) = Z^k(\mathbf{Y}^k,\mathbf{X}^k(\omega(n)))\,|_{k=0}$.

Step 2: Calculate $\nabla Z^k(\mathbf{Y}^k,\mathbf{X}^k(\omega(n)))$

(a). Add a positive small variation Δ onto an element $y_{v'}^k$ in $\mathbf{Y}^k$, then the vessel capacity

---

vector becomes, $\mathbf{Y}^k + \Delta\mathbf{Y}_{v'}^k = \{y_1{}^k, \ldots, y_{v'}^k + \Delta, \ldots, y_{|V|}^k\}$

(b). Set inner loop number $m = 0$, $\boldsymbol{\gamma}^m = \mathbf{0}$, a positive constant α;

(c). Solve the problem **P11** for the given $\boldsymbol{\gamma}^m$; obtain an estimate of $Z^k(\mathbf{Y}^k + \Delta\mathbf{Y}_{v'}^k, \mathbf{X}^k(\omega(n)))$:

$$\begin{aligned}\widehat{Z}^m(\mathbf{Y}^k + \Delta\mathbf{Y}_{v'}^k, \mathbf{X}^k(\omega(n))) &= \sum_{v \neq v'} C_v \cdot y_v^k + C_{v'} \cdot (y_{v'}^k + \Delta) + \frac{1}{N \cdot T}\sum_{n=1}^{N}\sum_{j \in J_{\omega(n)}} c_t^j \cdot D_j \cdot (T - t_j^0) \\ &- \sum_{v \neq v'}\sum_{t=1}^{H} \gamma^m(v,t) \cdot y_v^k - \sum_{t=1}^{H} \gamma^m(v',t) \cdot (y_{v'}^k + \Delta) + \frac{1}{N}\sum_{n=1}^{N}\sum_{j \in J_{\omega(n)}} Z^j(\mathbf{X}^*, \boldsymbol{\gamma}^m)\end{aligned} \tag{54}$$

(d). Set $m = m + 1$; update the Lagrangian multipliers using the following equation:

$$\gamma^{m+1}(v,t) = \max[0, \gamma^m(v,t) + t^{m+1}(v,t) \cdot (\sum_{j \in J} x_v^j(t) D_j - y_v)] \qquad \forall\, v \neq v', t$$

$$\gamma^{m+1}(v,t) = \max[0, \gamma^m(v,t) + t^{m+1}(v,t) \cdot (\sum_{j \in J} x_v^j(t) D_j - y_{v'} - \Delta)] \qquad \text{for } v = v', t$$

$$\text{where, } t^{m+1} = \alpha \cdot \frac{\widehat{Z}^m(\mathbf{Y}^k + \Delta\mathbf{Y}_{v'}^k, \mathbf{X}^k(\omega(n))) - Z^L(\mathbf{Y}, \mathbf{X}(\omega(n)))}{\sum_{n=1}^{N}\sum_{v \neq v'}\sum_{t=1}^{T}\left\| \sum_{j \in J(\omega(n))} x_v^j(t) D_j - y_v \right\|^2 + \sum_{n=1}^{N}\sum_{t=1}^{T}\left\| \sum_{j \in J(\omega(n))} x_v^j(t) D_j - y_{v'} - \Delta \right\|^2} \tag{55}$$

$Z^L(\mathbf{Y}, \mathbf{X}(\omega(n))$ is the lower bound obtained from Algorithm 2.

(e). Go to sub-step (b) unless $|\widehat{Z}^{m+1}(\mathbf{Y}^k + \Delta\mathbf{Y}_{v'}^k, \mathbf{X}^k(\omega(n))) - \widehat{Z}^m(\mathbf{Y}^k + \Delta\mathbf{Y}_{v'}^k, \mathbf{X}^k(\omega(n)))| < \epsilon$;

(f). Let $Z^k(\mathbf{Y}^k + \Delta\mathbf{Y}_{v'}^k, \mathbf{X}^k(\omega(n))) = \widehat{Z}^{m+1}(\mathbf{Y}^k + \Delta\mathbf{Y}_{v'}^k, \mathbf{X}^k(\omega(n)))$, then the partial derivative for $y_{v'}$ can be estimated as follows:

$$\frac{\partial Z^k(\mathbf{Y}^k, \mathbf{X}^k(\omega(n)))}{\partial y_{v'}} = \frac{Z^k(\mathbf{Y}^k + \Delta\mathbf{Y}_{v'}^k, \mathbf{X}^k(\omega(n))) - Z^k(\mathbf{Y}^k, \mathbf{X}^k(\omega(n)))}{\Delta}$$

(g). Go back to sub-step (a) until all the elements in **Y** have been perturbed and $\nabla Z^k(\mathbf{Y}^k, \mathbf{X}^k(\omega(n)))$ has been determined.

Step 3: Determine next searching point.

$$\mathbf{Y}^{k+1} = \mathbf{Y}^k + \frac{Z^k(\mathbf{Y}^k, \mathbf{X}^k(\omega(n)) - Z^L(\mathbf{Y}, \mathbf{X}(\omega(n))))}{\nabla Z^k(\mathbf{Y}^k, \mathbf{X}^k(\omega(n)))}$$

Step 4: Evaluate the costs $Z^{k+1}(\mathbf{Y}^{k+1}, \mathbf{X}^{k+1}(\omega(n)))$ using the procedure similar to (b) − (e) in Step 2 for the new position $\mathbf{Y}^{k+1}$.

Step 5: Obtain an upper bound $Z^U(\mathbf{Y}^{k+1}, \mathbf{X}^{k+1}(\omega(n)))$ using the method described in Proposition 5 for $Z^{k+1}(\mathbf{Y}^{k+1}, \mathbf{X}^{k+1}(\omega(n)))$. If $Z^U(\mathbf{Y}^{k+1}, \mathbf{X}^{k+1}(\omega(n))) < Z^{best}(\mathbf{Y}, \mathbf{X}(\omega(n)))$, then $\mathbf{Y}^{best} = \mathbf{Y}^{k+1}$, $Z^{best}(\mathbf{Y}, \mathbf{X}(\omega(n))) = Z^U(\mathbf{Y}^{k+1}, \mathbf{X}^{k+1}(\omega(n)))$; otherwise, go to next step.

Step 6: $k = k+1$, and go to Step 2 unless one of the following condition is met:

(a). $Z^U(\mathbf{Y}^{k+1}, \mathbf{X}^{k+1}(\omega(n)))$ is close enough to the estimated value $\widehat{Z}(\mathbf{Y}, \mathbf{X}(\omega(n)))$ denoted by

$$\widehat{Z}(\mathbf{Y},\mathbf{X}(\omega(n))) = \frac{Z^U(\mathbf{Y}^{k+1},\mathbf{X}^{k+1}(\omega(n))) + Z^L(\mathbf{Y},\mathbf{X}(\omega(n)))}{2}$$

(b). There is no update for $Z^{best}(\mathbf{Y},\mathbf{X}(\omega(n)))$ in a certain number of iterations.

4 Numerical experiments

The three solution strategies including SAA, PHA, and APHA were coded using Visual C++ 2010 and IBM CPLEX 12.5 library functions. Around 7000 lines of C++ codes have been written excluding the functions for processing data files. Additionally, both Linux and Microsoft Windows version have been developed. We use Windows version to test the algorithm for small-scale shipping network on laptops and desktops, and the Linux version for practical shipping network on high performance server.

The implemented three algorithms have been experimented on two datasets detailed in Song and Dong (2012). The two datasets involve a hypothetical small-scale shipping network and a realistic shipping network. We will examine the solution accuracy and computational times of the three algorithms for the two shipping systems, and then discuss their strengths and weaknesses and possible further improvements in future. It should be noted that although in the two datasets below, there is only one job for each port-pair on a particular day, our model and programme is able to deal with multiple jobs for each port-pair per day as long as each job has a unique index. In addition, our programme can also process the customer orders/jobs (which may have seasonality) as input data from a stored text file.

4.1 The small-scale shipping network

The small shipping network comprises 5 ports, 3 shipping services routes, and 3 vessels. Each day there will be 5 × 5 =25 jobs generated. The amount of containers required for each job varies on a daily basis and generated from Normal distribution with average values and standard deviations as detailed below.

Table 1 The average values of daily demands

	5001	5002	5003	5004	5005
5001	0	0	0	0	0
5002	10	0	0	0	10
5003	5	0	0	0	5
5004	10	0	0	0	10
5005	0	0	0	0	0

Table 2 The standard deviations of daily demands

	5001	5002	5003	5004	5005
5001	0	0	0	0	0
5002	2	0	0	0	2
5003	1	0	0	0	1
5004	2	0	0	0	2
5005	0	0	0	0	0

We set C_v = 1000 British Pounds per day, and the waiting costs c_t^j = 100 British Pounds per day. The planning horizon considered is 5 weeks. The other parameters are the same as those in Song and Dong (2012).

We ran our programme on a Windows desktop with an INTEL I7 3.4G HZ CPU and 8GB RAM, and obtained the outputs of the three algorithms as shown below.

Table 3 The results of SAA, PHA, and APHA for a Small-Scale Shipping Network

N	SAA $Z(\mathbf{Y},\mathbf{X}(\omega(n)))$		**PHA** $Z(\mathbf{Y},\mathbf{Y}(\boldsymbol{\omega}(n)),\mathbf{X}(\boldsymbol{\omega}(n)),\lambda)$			**APHA** upper Bound $Z^U(\mathbf{Y},\mathbf{X}(\omega(n)))$			**APHA** LCGGS $Z^{best}(\mathbf{Y},\mathbf{X}(\omega(n)))$			**APHA** Lagrangian Lower Bound $Z^L(\mathbf{Y},\mathbf{X}(\omega(n)))$		
	Costs	**Time (s)**	**Costs**	**Time (s)**	**Gap to SAA**	**Costs**	**Time (s)**	**Gap to SAA**	**Costs**	**Time (s)**	**Gap to SAA**	**Costs**	**Time (s)**	**Gap to SAA**
5	338129	8	339075	78	0.28%	341008	1	0.84%	—	—	—	315910	512	-6.57%
10	347206	20	347869	139	0.19%	359273	1	3.36%	356720	2225	2.67%	322116	848	-7.23%
20	348678	101	349990	259	0.37%	363376	1	4.04%	353118	4851	1.26%	321453	2436	-7.81%
40	346003	606	346329	1029	0.09%	364586	1	5.10%	360945	19759	4.14%	313600	4172	-9.36%
60	344427	1465	344913	1508	0.14%	365951	1	5.9%	355305	16529	3.06%	316000	7243	-8.25%
80	—	—	343246	1987	—	366975	1	—	360148	74109	—	313416	14017	—
100	—	—	—	—	—	347733	1	—	364236	103441		312701	19443	—

In Table 3, SAA is solved using the standard branch-and-cut solution algorithm implemented in IBM CPLEX. The algorithm in CPLEX can provide exact solution with the shortest computational time. However, when the sample size N increases to 80 scenarios or above, SAA cannot produce any result due to the large scale of the problem.

PHA needs longer computation time than SAA as it needs to iterate the Lagrangian multipliers corresponding to nonanticipativity constraints. For each iteration, it requires to solve N scenario based ILP using CPLEX. PHA will converge to a feasible solution with a small gap to the exact solution of SAA. However, it should be noted that PHA is not able to solve the problems as the size increases, e.g. $N \geq 100$.

APHA requires the longest computational time among three algorithms as it involves more iterations of Lagrangian multipliers. It can be observed that the majority of computational times were spent on calculating the lower bound $Z^L(\mathbf{Y}, \mathbf{X}(\omega(n)))$ and LCGGS, whereas the time taken to obtain an upper bound from the solutions to **P9** and **P10** was less than 1 second. The average upper bound has an average gap 3.85% above the optimal costs (from the exact solution) according to the results for the problems with sample size 5 – 80; and LCGGS can further narrow the average gap down to 2.78%. In the experiment, we terminate the LCGGS procedure when the best-so-far solution is close to the estimated optimal value of **P1** (within 5%). Mathematically, the criterion we adopt to stop LCGGS is when $| Z^{best}(\mathbf{Y}, \mathbf{X}(\omega(n))) - \frac{Z^U(\mathbf{Y}, \mathbf{X}(\omega(n))) + Z^L(\mathbf{Y}, \mathbf{X}(\omega(n)))}{2} | < 5\%$. It should be pointed out that the LCGGS has the potential to find better solution if the acceptable gap is further reduced at the expense of more computational time.

4.2 A practical sized shipping network

We now experiment the algorithms on a realistic shipping network that contains 25 ports, 24 vessels, and 5 shipping service routes. Everyday there are 25× 25 = 625 jobs generated, i.e., $|\boldsymbol{J}(\omega(n))| = 625$. The amount of containers required for each job follows normal distribution. The coefficient of variation (i.e. the ratio of the standard deviation to the average value) is 0.2. To save the space, we do not list the average value and standard deviation of each OD pair here. The planning horizon is 77 days (11 weeks), thus the number of jobs that need to be processed in a single scenario is 48125. For the case with a sample size of 10 ($N = 10$), the number of the variables $x_v^i(t)$ in SSA, in PHA (scenario based model) and in APHA (job-based model), will be $10\times 625 \times 24 \times 77 \approx 1.12 \times 10^7$; $625 \times 24 \times 77 \approx 1.12 \times 10^6$; and $24 \times 77 = 1848$, respectively.

We used a Linux server with 4 AMD 2.3 GHz CPU and 64GB memory to do the experiments. The maximum memory usage allocated by the server administrator is 16 GB out of the 64GB. Unfortunately, neither PHA nor SAA can produce any result due to the large scale of the problem. APHA is the only method that can produce result. The obtained results are given in Table 4.

Table 4 The results of APHA for a realistic Shipping Network

N	CPU Time (s)	Heuristics upper Bound $Z^U(\mathbf{Y},\mathbf{X}(\omega(n)))$	CPU Time (s)	Lagrangian Lower Bound $Z^L(\mathbf{Y},\mathbf{X}(\omega(n)))$	Estimated true value $\hat{Z}(\mathbf{Y},\mathbf{X}(\omega(n)))$	Gaps between Upper bound & Estimated true value
10	1	2616560	81950	2748190	2682375	2.39%
20	1	2554590	305236	2689420	2622005	2.51%
30	1	2508080	545796	2653620	2580850	2.74%
40	1	2503270	883176	2677370	2590320	3.25%

It can be observed that the solution generated by APHA has good performance since the average gaps between the heuristic upper bounds and the estimated true values are 2.72% for the sample size ranging from 10 to 40 scenarios. The gradient-based search (LCGGS) is not required to start as the gap is less than the aforementioned threshold level 5%.

The CPU times spent on the second network are very similar to that in the first one. The majority of CPU times are spend on solving the Lagrangain relaxed problems, i.e., P9 and P10, and it only take less than 1 second to obtain the upper bound and the corresponding feasible solutions.

From the two sets of numerical experiments conducted above, we can find that the merits of APHA are that it has good solution quality, and is able to solve much larger problems (e.g. either larger *N* or larger shipping network) which SAA and PHA cannot. However, it may still require long running time due to the iteration of Lagrangian multipliers.

One idea to reduce the running time of APHA is to apply the parallel computing technique into APHA. Note that the logic of APHA is to repetitively solve a number of decomposed problems with smaller scale. This feature happens to fit the logic of parallel computing. For example, our case needs to solve a large number of problems like **P11** repetitively, which can be run on multiple computers simultaneously.

5 Conclusion

The paper proposes a two-stage stochastic programming model for joint shipping service capacity planning and dynamic container routing in a shipping network with uncertain demands and delivery time constraints. The first stage focuses on minimising the costs of acquiring vessel capacity, and the second stage is to minimise the expected operational costs including transportation costs, lifting on/off costs, and unfulfilled job penalty costs. The second stage model can provide the operational performance of a given set of vessel capacities under uncertain demands and delivery time constraints.

Firstly, two relatively mature methods, Sample Average Approximation (SAA) and Progressive Hedging Algorithm (PHA), are used to solve the stochastic programming problem under consideration. Noting the computational limitation of SAA and PHA in solving large scale problems, we then designed a new solution method, Adapted Progressive Hedging Algorithm (APHA), which is able to solve larger scale problems (e.g. with more samples and more complex shipping networks). The idea of APHA is to further decompose scenario-based models into job (customer order) based problems using Lagrangian multipliers. Lower bound and upper bounds are provided to quantify the accuracy of the algorithm.

The involved three algorithms have been tested and compared on two datasets that have been used in Song and Dong (2012). According to the experiment results, we find that the merits of APHA include:

1) It is capable of solving large scale problems which cannot be solved by SAA and PHA;
2) APHA can provide the measurement of error bounds, which can quantify the accuracy of a feasible solution.
3) The solution generated from APHA has a good quality, and is close to the solution obtained from SAA and PHA for the smaller scale problem.

This paper has a few limitations. Firstly, we describe the demand using a known probability distribution. This might not be easy to obtain since forecasting demand is a big challenge in the shipping industry. In particular, the current shipping market is highly volatile. Secondly, we did not take into account the empty container repositioning issue. Since the world trade is severely imbalanced and empty container repositioning incurs a significant amount of cost to shipping lines, it would be desirable to incorporate it at the service capacity design stage. To extend our model to include empty container repositioning and investigate the computational complexity is a further research direction. Thirdly, from the experiments, it can be seen that although the APHA is able to solve the large-scale problems that cannot be solved by SAA and PHA, the computation time could be very long. Note that the APHA attempts to solve a large number of small-scale problems repetitively. This enables APHA to meet the requirements of parallel computing techniques such as Message Passing Interface (MPI) or Open Multi-Processing (OPENMP). These parallel computing techniques would allow us to use multiple CPUs or multi-core CPU to solve the multiple ILP problems in a single iteration in APHA simultaneously. Therefore, another further research direction is to implement the APHA using the parallel computing techniques and explore other ways to improve its computational efficiency.

Acknowledgement

This study was supported by a research grant from the Research Grants Council of Hong Kong, China (T32-620/11). The third author was partially supported by a Grant from the National Natural Science Foundation of China (71772071).

References

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

Agarwal, R., Ergun, O., 2008. Ship scheduling and network design for cargo routing in liner shipping. Transportation Science 42(2), 175–196.

Bell, M.G.H., Liu, X., Angeloudis, P., Fonzone, A., Hosseinloo, S.H., 2011. A frequency-based maritime container assignment model. Transportation Research B 45(1), 1152–1161.

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

Brouer, B.D., Pisinger, D., Spoorendonk, S., 2011. Liner shipping cargo allocation with repositioning of empty containers. INFOR 49(2), 109–124.

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

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

Dantzig , G. B., Thapa, M. N., 2003. Linear programming 2: theory and extensions. Springer-Verlag

Ermoliev, Y., 1983. Stochastic quasigradient methods and their applications to systems optimization. Stochastics 9, 1–36

Fisher, M. L., 2004. The Lagrangian Relaxation Method for Solving Integer Programming Problems. Management Science 50(12) Supplement, 1861–1871.

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

Gaivoronski, A.A., 1988. "Implementation of stochastic quasigradient methods" in: Y. Ermoliev and R. Wets, Eds., Numerical Techniques for Stochastic Optimization. Springer-Verlag, Berlin, 313–352.

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

Jaramillo, D. I., Perakis, A., 1991. Fleet deployment optimization for liner shipping Part 2. Implementation and results. Maritime Policy and Management 18(3), 235-262.

Liu, X., Ye, H. Q., Yuan, X. M., 2011. Tactical planning models for managing container flow and ship deployment. Maritime Policy Management 38(5),487–508.

Long, Y., Lee, L.H., Chew, E.P., 2012. The sample average approximation method for empty container repositioning with uncertainties. European Journal of Operational Research, 222(1), 65–75.

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, S., 2011a. Optimal operating strategy for a long-haul liner service route. European Journal of Operational Research 215(1), 105–114

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

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

Meng, Q, Wang, S., Henrik, A., Thun, K., 2014. Containership Routing and Scheduling in Liner Shipping: Overview and Future Research Directions. Transportation Science 48(2), 265–280.

Mourão M.C., Pato M.V., Paixão A.C., 2002. Ship assignment with hub and spoke constraints. Maritime Policy Management 29(2), 135–150.

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

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

Rockafellar, R., Wets, R., 1991. Scenarios and policy aggregation in optimization under uncertainty. Mathematics of Operations Research 16(1), 119–147

Ronen, D., 1983. Cargo ships routing and scheduling: Survey of models and problems. European Journal of Operational Research 12(2), 119-126.

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

Ronen, D., 2011. The effect of oil price on containership speed and fleet size. Journal of the Operational Research Society 62, 211–216.

Ruszczynski, A., Shapiro, A., 2003. Stochastic programming models. In: Ruszczynski, A., Shapiro, A. Eds., Handbooks in Operations Research and Management Science, Elsevier, 10, 1–64

Shintani, K., Imai, A., Nishimura, E., Papadimitriou, S., 2007. The container shipping network design problem with empty container repositioning. Transportation Research Part E 43(1), 39-59.

Slyke, R. V., Wets, R., 1969. L-shaped linear programs with applications to optimal control and stochastic programming. SIAM Journal on Applied Mathematics 17(4), 638–663.

Song, D.P., Dong, J.X., 2012. Cargo routing and empty container repositioning in multiple shipping service routes. Transportation Research Part B 46(10), 1556–1575.

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

Wang, S., Meng, Q., 2012a. Liner ship fleet deployment with container transhipment operations. Transportation Research Part E 48(2), 470-484.

Wang, S., Meng, Q., 2012b. Sailing speed optimization for container ships in a liner shipping network. Transportation Research Part E 48(3), 701-714

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

Wang, S., Meng, Q. Bell, M.G.H., 2013. Liner ship route capacity utilization estimation with a bounded polyhedral container shipment demand pattern. Transportation Research Part B 47, 57–76.

Wang, S., Meng, Q., Sun, Z., 2013. Container routing in liner shipping. Transportation Research Part E 49(1), 1–7.

Wang, S., 2014. A novel hybrid-link-based container routing model, Transportation Research Part E 61, 165-175

Yan, S., Chen, C.Y., Lin, S.C., 2009. Ship scheduling and container shipment planning for liners in short-term operations. Journal of Marine Science and Technology 144, 417-435

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 and Logistics 13(3), 278–297.