TWO ESSAYS ON LINER SHIPPING NETWORK DESIGN

JUN XIA

Ph.D

The Hong Kong Polytechnic University

2017

The Hong Kong Polytechnic University

Department of Logistics and Maritime Studies

Two Essays on Liner Shipping Network Design

Jun Xia

A thesis submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
June 2016

CERTIFICATE OF ORIGINALITY

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

___________(Singed)

__Jun Xia __ (Name of student)

ABSTRACT

Two Essays on Liner Shipping Network Design

by

Jun Xia

Liner shipping network design is a comprehensive decision problem for the carrier to plan their service operations. A series of related decisions from tactical and operational levels are coordinated to maximize the profits. In this thesis, we study two network design problems for carriers in liner shipping. The first work focuses on the joint decision makings for the network design, in which a general fuel consumption function is assumed. The second work studies the solution to a compact optimization model for the network design, in which the transshipment cost is taken into account. In this thesis, we have developed novel optimization techniques to solve both problems.

In the first study, we develop a mathematical programming model that addresses fleet deployment, speed optimization, and cargo allocation jointly, so as to maximize total profits for carriers at the strategic level. To capture fuel costs precisely, our model adopts a general fuel consumption function that depends on both vessel speed and vessel load. To overcome intractability caused by nonlinear terms in the model, we separate fuel consumption costs into two terms associated with ship speed and ship load, respectively, so as to obtain a mixed integer linear programming formulation for approximation. Based on column generation techniques, we develop an iterative search algorithm that adaptively reorganizes the approximated formulation. We conduct extensive experiments using generated data sets from actual liner shipping services in different regions of the world to show the effectiveness of our approach as well as the significant impact of speed-load factors on fuel consumptions. Managerial insights are obtained by testing the model under different scenarios, which may greatly assist decision makers in the liner shipping industry.

In the second study, we study a problem that aims at creating a set of regular services for a designated fleet of oceangoing ships to transport the containerized cargos among seaports. Containers can be transshipped from one ship to another at an intermediate port in order to improve the transportation efficiency. The objective of the problem is to maximize the revenues from the satisfied demands while minimizing the operating cost including the transshipment cost. For this issue, many solution methods known to be effective for problems assuming zero transshipment cost cannot directly apply, because the calculation of transshipment cost has significantly complicated the problem and its mathematical formulation. To tackle this challenge, we develop for this problem a new compact mixed-integer linear programming model. However, the new model may contain exponentially many variables and constraints, making its solution very challenging. Therefore, we propose a novel application of simultaneous column-and-row generation to solve the linear programming relaxation of the new model, so as to derive an upper bound for the profit of an optimal network design. Based on this, we have developed a branch-and-price to find optimal or near-optimal integer solutions for this problem. Results from experiments have shown the effectiveness and efficiency of our models and solution methods.

Publications arising from this thesis

The second chapter of this thesis is based on the following publication [78]:

- Joint Planning of Fleet Deployment, Speed Optimization and Cargo Allocation for Liner Shipping, co-authored with Kevin Li, Hong Ma and Zhou Xu, *Transportation Science*, 49(4), 922-938, 2015

ACKNOWLEDGEMENTS

I would like to thank the Hong Kong Polytechnic University (PolyU) to offer me the opportunity to pursue my PhD degree.

First of all, I would like to extend my sincere gratitude to my supervisor, Dr Zhou Xu, for his instructive guidance for my PhD study. His strict attitude and fantastic sense of doing researches have made a profound impression on me. He has walked me through all the research stages and provided me with a systematic academic training, which would be invaluable to my research career in the future.

Second, I would like to express my heartful thanks to Prof. Kevin Li and Dr Hong Ma, who have contributed to one chapter of this thesis. Their constructive suggestions have significantly improved the quality of this thesis. Moreover, I would like to sincerely thank Dr Yanzhi Li, Prof. Qiang Meng, and Dr Shuaian Wang for their insightful comments on this thesis.

I am grateful to the teachers, Prof. Xiuli Chao, Dr Xiaowen Fu, Dr Sudeep Ghosh, Dr Pengfei Guo, Dr Li Jiang, Prof. Eric Ngai, along with many others, who have shared their research expertise to broaden my knowledge and skills. I am also thankful to the administrative staff in LMS, FB and RO who have provided me with many kind services. Moreover, I am grateful to my dear friends, especially the classmates and colleagues at PolyU, who have accompanied me during the unforgettable campus time.

Finally, I would like to express my heartfelt gratitude to my family members for their selfless support and encouragements.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

Figure

LIST OF TABLES

Table

CHAPTER 1

Introduction

Liner Shipping Network Design (LSND) is regarded as an important decision problem for carriers to plan their service operations. In the past few decades, the service network hold by the global liner shipping carrier continued to expand [24]. New capacities were constantly ordered and deployed to satisfy the booming cargo demands [20]. Due to capricious nature of the shipping industry, both supply and demand had frequent fluctuations in the market. In response to the changing market, carriers required adjusting their service networks to keep the competitiveness.

Ever since the global financial crisis in 2008, demand for container shipping services has shrunk significantly [67]. As a result, carriers are faced with a huge surplus of tonnage. This overcapacity in the shipping industry is probably here to stay for some considerable time because the capacity coming up in the pipeline cannot be fully absorbed within a short while by restoration of the growth in demand. In addition to capacity surplus, the profit margins of carriers have come under pressure because of the increase in fuel prices and the cost of emission taxes. In order to save on fuel costs and to absorb excess capacity, as of January 2011, over 90 percent of the Far East-North Europe loops and over 70 percent of the Far East-US East Coast loops have adopted an *extra slow steaming* strategy in their daily operations [15]. Due to steady growth in both the U.S. and European economies, the industry saw an increase in trans-Pacific and Asia-Europe volumes during 2013, but huge capacity growth widens the supply-demand gap, and therefore, during the year 2014 the slow steaming strategy has still dominated container shipping [8].

However, speed reduction for cost saving is not sustainable. This is because, when ships operate at a low speed, saved fuel costs can barely offset the increased operating costs due to having to deploy additional ships. As well as adopting slow steaming, carriers are also systematically coordinating their service networks in order to make their fleets not only cost efficient but also market oriented. This is what has motivated this thesis to address an aggregate model for the planning of fleet operations with speed optimization. In Chapter 2, we have a particular attention on a joint optimization for LSND where the ship sailing speed is an important issue to be considered.

Transshipment operations also play an important role in liner shipping. According to [69], less than 20% of the country pairs are connected with each other through a direct service. As shown in Figure 1.1, from 1990 to 2012, the growth of total port throughput has accompanied with an increased share of transshipment cargos from 17.6% to 28%. Note that the transshipment operation and its cost will affect the decisions for network design. Some important transshipment operations can significantly reduce the transit times as well as the operating costs for the cargo demand fulfillment. Moreover, in the practice, since terminal operation costs vary from port to port, the transshipment cost cannot be overlooked in the planning as it will influence the decisions in LSND, such as which port to call and where to transship the cargos. In Chapter 3, we focus on the LSND that captures the transshipment cost in the optimization.

In the first study, a comprehensive mathematical programming model has been addressed to optimize fleet deployment, speed optimization, and cargo allocation jointly, so as to maximize total profits for carriers at the strategic level. Different

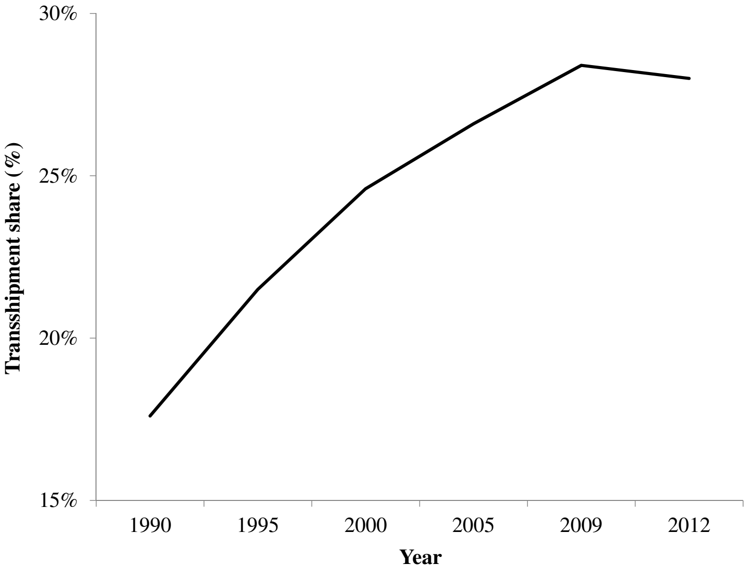


Figure 1.1: Growth of transshipment share over the total port throughput [52]

from the existing work, we consider a general fuel consumption function that depends on both vessel speed and vessel load. Due to the nonlinear fuel consumption function, the original model formulation is intractable. We thus introduce new techniques to handle the nonlinear terms so that a mixed integer linear programming formulation is then obtained for approximation.

Although a commercial solver, such as CPLEX [21], could directly tackle the approximate formulation, it is impossible to solve even for moderate sized instances since the number of both variables and constraints grows exponentially with the network size. By aggregating the route dependent constraints, we study a new linear programming relaxation which is solved by a column generation approach. Based on column generation techniques, we develop an iterative search algorithm that adaptively reorganizes the approximated formulation. We conduct extensive experiments using generated data sets from actual liner shipping services in different regions of the world. The effectiveness of our approach as well as the significant impact of speed-load factors on fuel consumptions can be observed from the experimental results. We test the model under different fleet deployment strategies, which may produce managerial insights to assist decision makers in the liner shipping industry.

In the second study, we investigate a general liner shipping network design problem. The objective of this decision problem is to maximize the revenues from the satisfied demands while minimizing the operating cost including the transshipment cost. We develop for this problem a new compact mixed-integer linear programming model. However, in the new model, the numbers of decision variables and constraints are both proportional to the number of rotations, and solving it becomes challenging. Therefore, we propose to applying an optimization technique, referred to as simultaneous column-and-row generation, to solving the linear programming relaxation of this model. Using this technique, we have developed several new solution methods to find optimal or near-optimal integer solutions for this problem. Numerical experiments are conducted to show the effectiveness and efficiency of our model and solution methods. Results from the numerical experiments show that our exact solution method solves more and larger instances and averagely consumes less running time than CPLEX's MIP-Solver. Our solutions also significantly outperform the existing benchmark solutions.

The remainder of this thesis is organized as follows. We first present the joint optimization problem of LSND with the consideration of a general fuel consumption function in Chapter 2. The network design problem with transshipment cost is then studied in Chapter 3. In Chapter 4, we summarize the above two studies together with a discussion on some future research directions.

CHAPTER 2

Joint Planning of Fleet Deployment, Speed Optimization, and Cargo Allocation for Liner Shipping

2.1 Introduction

The planning of liner shipping services has been studied for years, and a review of such literature was conducted by [63, 16, 17, 18, 48]. Liner shipping fleet deployment (LSFD) is an important decision-making issue that assigns containerships on a service network based on expected capacities and frequencies on loops. There is much literature focusing on LSFD. Two early representative studies by [55] and [31] discussed fleet deployment in liner shipping. The authors provided a comprehensive investigation of the relevant objectives, considerations and models for LSFD. Their work was then extended by Powell and Perkins [58], where the number and type of ships to serve each enumerated route were determined using an integer programming model. Christiansen et al. [17] later summarized a mixed integer linear programming (MILP) formulation to design voyages and lay-up times for each ship type in a given planning period.

LSFD is also studied along side other decisions made in liner shipping. Ship scheduling and cargo routing are typical decisions to be made when fleet deployment is studied along with network designs. Agarwal and Ergun [1] presented a MILP model to optimize the routes for both ships and cargos. They deployed ships on each cycle of a route, observing the frequency requirements. Three algorithms, including greedy heuristic, column generation and benders decomposition were investigated. Álvarez [3] consolidated ship routing, cargo routing and fleet deployment into one optimization problem. They developed an iterative framework that uses a meta-heuristic to control ship routing and fleet deployment at a master level, while solving cargo routing by mathematical programming at a secondary level. Gelareh and Pisinger [29] designed a hub and spoke network for liner service routes. An integrated model was proposed to determine hub locations, port connections, cargo flows and fleet deployment, for the purpose of profit maximization. They decomposed the model into a master problem for network design and fleet deployment, and a subproblem that optimizes cargo flow via linear programming. Liu et al. [37] considered both nonempty and empty container flows in fleet deployment decisions. In addition to a sequential model, they also developed a joint model solving the ship deployment and cargo flow simultaneously, in which the formulation of cost is independent from the ship speed. Wang and Meng [71] focused particularly on cargo transshipment operations, and they developed a MILP model to resolve fleet deployment directly using commercial solver.

Fuel consumption of a containership is significantly influenced by its operating speed, which for the most part follows an exponential function when sailing at a speed of above 14 knots [26, 41, 53, 64, 80]. Therefore, fuel cost is saved at the expense of increased traveling time. Among numerous studies focused on shipping speed optimization [62, 25, 51, 70], only a few have investigated LSFD by considering speed optimization. Meng and Wang [43] studied the optimal operating strategy including frequency, speed and number of ships deployed on a service route. They employed a piecewise linear function to approximate the relationship between fuel consumption and speed. A branch and bound method was proposed to obtain a near optimal solution with a guaranteed gap to the global optimum for a single liner service. Álvarez [3] defined a run as a combination of ship type, speed and service, and then assigned ships on a set of runs by enumeration. Gelareh and Meng [28] assumed the unit cargo cost on each individual link was correlated to the speed of the ship. Binary variables were set to decide the speed at which the ship will travel on different legs of service. In comparison, Álvarez [3] did not optimize the speed on each leg of a route. Gelareh and Meng [28] predetermined cargo allocations before developing the model, and therefore, their model can only be applied for a short term case. This work strives to overcome these limitations, that is, in addition to determining the optimal ship speed on each link of the route, we will also provide an integrated model that jointly optimizes ship routing, fleet deployment, cargo distribution and frequency for a complete liner shipping service network.

In addition to the operating speed, the fuel consumption of a containership is also affected by the amount of cargo loaded on board. “Load factor” has become even more important nowadays, because statistics show that, in practice, containerships are not making full use of available capacity, due to trade imbalance among regions and overcapacity in the market [56, 68, 39]. Kontovas and Psaraftis [34] showed that a realistic fuel consumption function consists of the load factor and transit time over the shipping legs. Wang et al. [77] linked the fuel consumption function with the ship displacement. Psaraftis [59] and Psaraftis and Kontovas [60] developed an explicit formulation for the general fuel consumption function that jointly depends on speed and load. Psaraftis and Kontovas [60] modeled the speed-load dependent fuel consumption function in their speed optimization model. The optimal speed was found to minimize the summation of fuel cost, inventory cost and charter cost. Among the other speed models reviewed by [60], very few works had investigated the speed-load dependent fuel cost. There is, therefore, still a gap in research as how to incorporate the speed-load fuel cost into the planning of a liner shipping service network, and our work seeks to fill this gap.

In our work, we aggregate globally distributed ports into a small number of regions according to their geographic locations, so as to decrease demand uncertainty and to facilitate planning. Network aggregation also appears both in practice and in the literature. For example, Hapag-Lloyd, a global liner shipping carrier, divides the entire market into nine regions. Their services and offices were also classified and searched by region. In the literature, Jepsen et al. [32] formulated a liner shipping network design model based on a novel view of demands on a trade basis instead of the origin-destination (o-d) pairs. The aggregation of demand can be further fine tuned to cater for both detailed networks for a smaller region and coarse network designs for a larger region. Mulder and Dekker [49] developed a specific way to cluster the ports based on a list of central ports; they first determined the network design based on port clusters, and then implemented cluster disaggregation in cargo routing. The service network aggregation approach that we take is to cluster ports into a smaller number of regions, and to rotate ships among these regions to fulfill the inter-regional demands. By alternating the aggregation scope, our proposed model is applicable to planning at different levels.

In summary, the contribution of this study is multi-fold. We implement aggregation to represent the carrier's service network, where the aggregation scope can be altered to make it compatible with different planning levels. Besides the fleet deployment decision, we strive to optimize ship routing, cargo allocation, speed and frequency simultaneously. Furthermore, based on the general fuel consumption function introduced in [60], this work is new within in extant literature in that it also incorporates the speed-load dependent fuel cost into the planning of liner shipping services. Effective models and solution methods are developed and examined to study the new problem, and managerial insights are acquired from extensive numerical studies for different liner shipping scenarios.

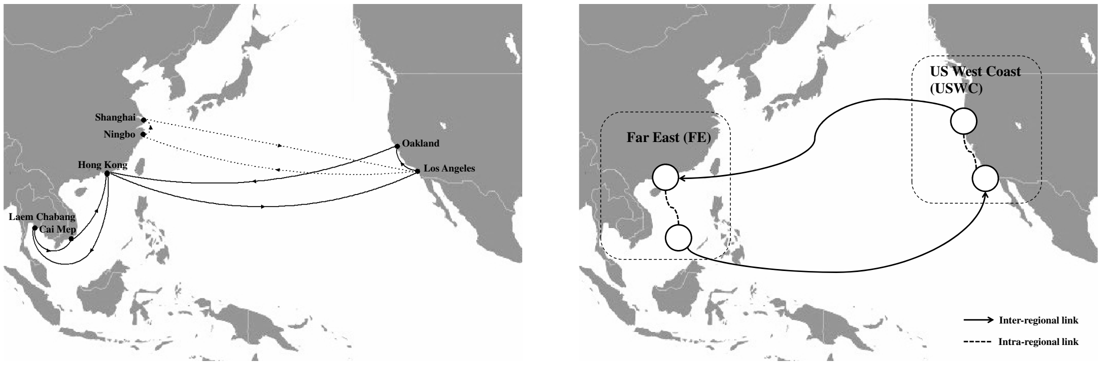
The rest of this work is organized as follows. In Section 2.2, we introduce an aggregate shipping network and the joint effect of speed and load on fuel consumption. An aggregate planning model is developed in Section 2.3. In Section 2.4, we study the solution techniques that can efficiently solve the proposed model. Computational results based on extensive data are presented in Section 2.5, followed by the summary in Section 2.6.

2.2 Problem Description

Consider a General Service Network (GSN) of two liner shipping routes, as shown in Figure 2.1(a), which consists of ports, links and services. From the perspective of a liner shipping carrier, nearby and competing ports within a specific maritime range could be grouped as regional clusters. Hence, for efficiency considerations, the carrier may opt to bypass one port so as to gain the advantage of another in the same regional cluster. An Aggregated Service Network (ASN) with two regions is then conceptualized as shown in Figure 2.1(b). Nodes in the ASN could be continental areas (e.g. Asia, Europe), oceanic areas (e.g. Far East, US West Coast, Mediterranean) or national areas (e.g. China, Australia), based on different division rules. In this work, we set the aggregate regional clusters based on the regional separations adopted in [30].

Based on the concept of regional clusters, the liner service routes in a GSN can be considered as a combination of inter-regional and intra-regional links in a ASN. In Figure 2.1, for example, FE-FE-USWC-USWC-FE is a feasible region rotation service route after the aggregation. The solid directed links between FE and USWC are inter-regional while the dashed undirected links FE-FE and USWC-USWC are intra-regional.

Without loss of generality, consider that an ASN has a set of regions Z indexed



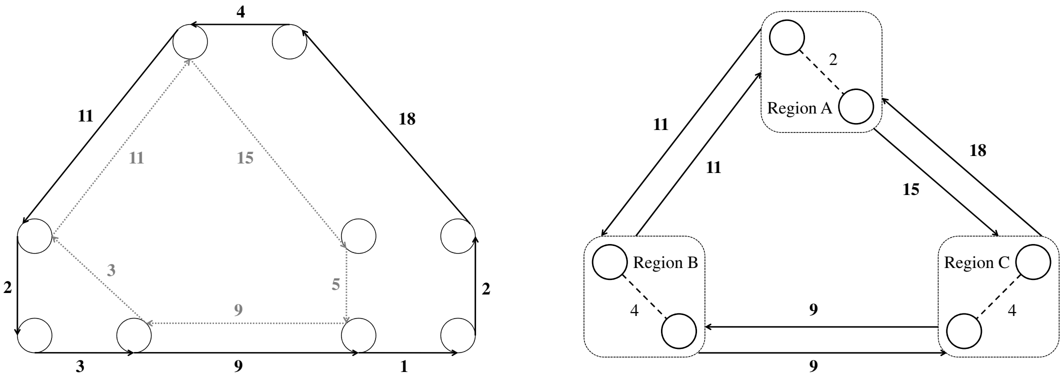
(a) General Service Network (GSN)

(b) Aggregated Service Network (ASN)

Figure 2.1: Aggregation of a network with two transpacific liner shipping services from [30]

by z, and a set of links A indexed by e. For each $e \in A$, let ℓ_e denote the average voyage length of the link e. Here ℓ_e is assigned by mean values from practical existing service routes. See Figure 2.2 for example, where a GSN of two shipping services and nine ports is illustrated in Figure 2.2(a) and its corresponding ASN of three regions is illustrated in Figure 2.2(b). In Figure 2.2(b), the intra-regional link B-B has a length of $(2+3+3)/2 = 4$ and the inter-regional link A-C has a length of $15/1 = 15$. Note that there exist errors for a region rotation route in an ASN so as to represent a realistic service route in a GSN. We ignore such errors because of their minor effects at an upstairs planning level.

It is worth noting that while cargo demand is usually identified by an origin port and a destination port, within the ASN cargo demand is aggregated accordingly. Let D denote a set of cargo demand generated on a weekly basis. Each cargo demand $(o, d) \in D$ is identified by its origin region o and destination region d, which generates a freight rate p_{od} per TEU of cargos. A maximum of $q_{od}^{\max}$ TEUs of cargos are generated for (o, d), among which a contracted amount of $q_{od}^{\min}$ TEUs must be fulfilled. We denote the route set by R, each route $r \in R$ bypasses the links $e \in A(r)$. Consider a fleet of containerships with ship type set S. The ship of type $s \in S$ has a capacity of m_s



(a) GSN

(b) ASN

Figure 2.2: An illustrative example from GSN to ASN

TEUs, a maximum fleet size of n_s, and a net weight w_s. Let a_s, b_s and τ_s denote the weekly operating cost, weekly idle cost and daily port stay cost, respectively for such a ship of type s.

Represent a certain liner service by a particular combination (s, r, k, v) that implies a total of k ships of type s operating on route r under a speed strategy v which maintains the service once a week. To guarantee a weekly frequency of k ships, the total voyage time of service (s, r, k, v) must be equal to k weeks. Define $v = [v_e : e \in A(r)]$ is a speed vector that contains the ship sailing speed on each link of the route. We assume the sailing speed of a ship is selected from a set of discrete values. In abbreviation, we use indicator i to differentiate and identify a particular liner service, while the associated combination is denoted by (s_i, r_i, k_i, v_i). Let Ω represent the total set of services, and define $\Omega(o, d)$ for candidate services reaching demand (o, d). Let $A(i, o, d)$ denote the set of links traveled by service i to serve (o, d). Let $D(i, e)$ denote the set of cargo demand shipped on link e of service i. We consider two types of decision variables in the model: x_{odi} denotes the TEUs of cargo demand (o, d) allocated to service i and y_i denotes the service frequency (number of sailing per week) of deploying service i.

In the aggregate planning, we hold the following assumptions: (i) The weekly demand is stabilized and deterministic; (ii) the fuel oil price is known; (iii) the daily port stay cost is identical in each region; and (iv) transshipments between two different services are not considered. We consider assumption (i) because the forecast of aggregate demand has relatively smaller uncertainties. In the planning, we settle the fuel oil price at an average value over the period so that assumption (ii) holds. Assumption (iii) is given to simplify the presentation. Assumption (iv) is considered to simplify the development of our model and solution method. To consider cargo transshipment operations, the model without transshipments can be extended such that the cargo flow on each link of the route is determined by the flow conservation constraints. Based on that, the solution techniques proposed in this chapter can still handle the extended model.

Based on the ASN, our aggregate planning is able to cope with a set of services with specific frequencies (speeds), and simultaneously allocates cargos to services in such a way that total profit is maximized. Considering an undetermined planning period, the objective becomes the amount of profit earned in a week, so that the total profits over a given period are equivalent to the weekly profits multiplied by the number of weeks involved in the planning period.

Speed and load are two important factors that determine the amount of fuel consumed during the transportation. Bektaş and Laporte [10] modeled the influence of load weight on vehicle fuel consumption in a classic vehicle routing problem. However, in extant fleet deployment and speed optimization literature, the fuel consumption of a ship is usually computed with respect to speed, while the displacement of loads is largely overlooked. As a particular concern of this work, we will study the fuel consumption of a ship by jointly considering speed and load. Define the fuel consumption rate γ by the tonnage of fuel oil used per sailing unit nautical mile. v is the operating speed of a ship, w is the weight of the ship without loads, and x is the amount of cargo loaded on the ship. Psaraftis and Kontovas [60] formulate the general fuel consumption function of a ship by $\gamma = (\beta_0 + \beta v^n)(w + x)^{\frac{2}{3}}$, where w and x are regarded as the total tonnages of the ship. Since the fuel consumption per unit time is mostly studied proportional to the cubic of sailing speed, for ease of presentation, we set $\beta_0 = 0$ and $n = 2$. The following exact formulation is used to calculate the speed-load dependent fuel consumption in our problem.

$$\gamma(v, x) = \beta v^2 (w + x)^{\frac{2}{3}}. \tag{2.1}$$

In the problem, w is counted in tons while x is counted in TEUs. To uniform the unit, we approximately measure the total tonnages of the ship in TEUs by assuming that the the average-loaded TEU container weighs about 8-13 tons.

2.3 Model Formulation

For the link e of a service i, let $\gamma(x_{ie})$ denote the associated fuel consumption rate, where $x_{ie} = \sum_{(o,d)\in D(i,e)} x_{odi}$ denotes the total TEUs of cargos to be allocated. We use $\phi_i(x_i)$ to represent a round trip cost to operate service i, and $x_i = [x_{ie} : e \in A(r_i)]$ denotes a cargo vector indicating the TEUs of cargos on each link of service i. We next formulate the round trip operating cost of service i as follows.

$$\phi_i(x_i) = a_{s_i} k_i + \tau_{s_i}(7k_i - \sum_{e\in A(r_i)} \frac{1}{24}\frac{\ell_e}{v_{ie}}) + \sum_{e\in A(r_i)} \alpha \ell_e \gamma(x_{ie}). \tag{2.2}$$

In Eq. (2.2), α denotes the fuel oil price. The summation of transit times on both inter-regional and intra-regional links should be no greater than the total voyage time, that is, $\sum_{e\in A(r_i)} \frac{1}{24}\frac{\ell_e}{v_{ie}} \leq 7k_i$ must be satisfied. While the first term denotes the fixed cost of a ship of type s_i for k_i weeks, the second term formulates the total port stay cost, calculated by multiplying the daily stay cost by the number of days of the stay. The third term denotes the total fuel cost during the round trip.

For each service i, the weekly operating cost is calculated by taking the round trip cost multiplied by the number of ships operating on it, and averaged by the number of weeks. Remember that each service is of weekly frequency, such that the number of weeks is equal to the number of ships operated on that service. This implies that the weekly operating cost is numerically equivalent to the round trip cost as formulated in Eq. (2.2). In a way similar to [3] and [37], we integrate all decision variables, constraints and objectives, and formulate the LSFD as a non-linear integer program.

$$\text{(LSFD)} \quad \max \quad \sum_{(o,d)\in D} \sum_{i\in\Omega(o,d)} p_{od} x_{odi} - \sum_{i\in\Omega} \phi_i(x_i) y_i - \sum_{s\in S} b_s (n_s - \sum_{i\in\Omega: s_i = s} k_i y_i) \tag{2.3a}$$

$$s.t. \quad q_{od}^{\min} \le \sum_{i\in\Omega(o,d)} x_{odi} \le q_{od}^{\max}, \quad \forall (o,d) \in D \tag{2.3b}$$

$$\sum_{(o,d)\in D(i,e)} x_{odi} \le m_{s_i} y_i, \quad \forall i \in \Omega, \forall e \in A(r_i) \tag{2.3c}$$

$$\sum_{i\in\Omega: s_i = s} k_i y_i \le n_s, \quad \forall s \in S \tag{2.3d}$$

$$x_{odi} \ge 0, y_i \in \mathbb{Z}^+ \cup \{0\}. \quad \forall (o,d) \in D, \forall i \in \Omega \tag{2.3e}$$

The objective function shows the maximization of weekly profits obtained by subtracting weekly operating costs and weekly idling costs from weekly revenues. Constraints (2.3b) restrict the fulfillment of cargo demand to no smaller than a mandatory volume and no greater than a maximum volume. Constraints (2.3c) provide capacity limits for cargo flows on each link of the operated service routes. Constraints (2.3d) restrict the number of ships of each type to be deployed under the given fleet sizes. The decision variables are given in Eq. (2.3e).

Note that the model is computationally difficult due to its non-linearity and non-convexity. Moreover, there are a large number of integer columns in the model which also increases the complexity. We thus need to carry out some preliminary work to properly transform the model so that the problem can be tackled by existing solution techniques in an efficient way. We first approximate the fuel consumption function γ by $\mu(v)+\omega x$, where $\mu(v)$ can be any function of v, and x is formulated in a linear form with coefficient ω. Substitute $\gamma(x_{ie})$ by $\mu(v_{ie})+\omega x_{ie}$ in Eq. (2.2) and we obtain the operating cost function in a separated form as follows.

$$\phi_i(x_i) = a_{s_i}k_i + \tau_{s_i}(7k_i - \sum_{e\in A(r_i)} \frac{1}{24}\frac{\ell_e}{v_{ie}}) + \sum_{e\in A(r_i)} \alpha\ell_e\mu(v_{ie}) + \sum_{e\in A(r_i)} \alpha\ell_e\omega x_{ie}.$$

Let $c_i(v_i)$, a function of speed vector v_i, denote the summation of the first three terms of $\phi_i(x_i)$.

$$c_i(v_i) = a_{s_i}k_i + \tau_{s_i}(7k_i - \sum_{e\in A(r_i)} \frac{1}{24}\frac{\ell_e}{v_{ie}}) + \sum_{e\in A(r_i)} \alpha\ell_e\mu(v_{ie}).$$

By replacing x_{ie} with $\sum_{(o,d)\in D(i,e)} x_{odi}$, the operating cost function is then represented as follows.

$$\begin{aligned}\phi_i(x_i) &= c_i(v_i) + \sum_{e\in A(r_i)}\sum_{(o,d)\in D(i,e)} \alpha\ell_e\omega x_{odi}\\ &= c_i(v_i) + \sum_{(o,d)\in D} \eta_{odi}x_{odi},\end{aligned}$$

where η_{odi} is a known constant that equals to $\sum_{e\in A(i,o,d)} \alpha\ell_e\omega$. Due to Constraints (2.3c), x_{odi} is equal to zero if $y_i = 0$. Therefore, in all cases, we have that $\phi_i(x_i)y_i = c_i(v_i)y_i + \sum_{(o,d)\in D}\eta_{odi}x_{odi}y_i = c_i(v_i)y_i + \sum_{(o,d)\in D}\eta_{odi}x_{odi}$. Substituting it into LSFD, we then obtain an approximate model denoted by LSFDA as follows.

$$\begin{aligned}\text{(LSFDA)}\quad \max\ & \sum_{(o,d)\in D}\sum_{i\in\Omega(o,d)} (p_{od}-\eta_{odi})x_{odi} - \sum_{i\in\Omega} c_i(v_i)y_i - \sum_{s\in S} b_s(n_s - \sum_{i\in\Omega:s_i=s} k_iy_i)\\ s.t.\ & (2.3b), (2.3c), (2.3d), (2.3e).\end{aligned}$$

We thus get the following proposition.

Proposition 2.1. *If γ is separated by two terms that depend on x and v and the term of x is linear, then LSFD can be approximated to an MILP.*

Proposition 2.1 provides an approximation condition of γ that approximates the original non-linear LSFD to an MILP formulation. It enables us to focus our work on LSFDA instead of LSFD. More importantly, as long as the approximation gap of γ is small, the optimum of LSFDA is able to provide a reliable benchmark which is intuitively close to the optimum of LSFD.

2.4 Solution Method

Our solution method for obtaining a heuristic solution of LSFD is structured as follows. In Section 2.4.1, we introduce an approximation of γ that allows the approximation model LSFDA to be an MILP. Since LSFDA consists of a huge number of decision variables and the constraints are associated with services, it is difficult to tackle it directly. In Section 2.4.2, we therefore propose a relaxation of LSFDA by relaxing integer variables and aggregating capacity constraints. In Sections 2.4.3 and 2.4.4, we discuss a column generation algorithm for solving the resulting linear programming model, with its pricing problem being solved by a path searching algorithm. This column generation algorithm generates a significantly smaller number of services, enabling us to select the best subset of services to construct a heuristic solution to LSFDA and LSFD. The solution can be further improved by a speed optimization procedure and an iterative search algorithm, which are discussed in Sections 2.4.5 and 2.4.6, respectively.

2.4.1 Approximation of γ

By conducting a series of sample fitting tests, we define $\mu(v) = \omega'' + \omega' v^2$ and approximate the fuel consumption rate function by Eq. (2.4), which, on average, has the R-square as no less than 0.96.

$$\gamma(v,x) = \omega'' + \omega' v^2 + \omega x. \tag{2.4}$$

Eq. (2.4) allows LSFDA to be an MILP, and we can further derive a valid upper bound on the optimal objective value of LSFD. We begin by establishing a lower bound for the approximation of γ. Let the coefficients ω' and ω be fixed from the regression, and then we can move the surface down by reducing the constant term of (2.4) to $\bar{\omega}''$, in such a way that the entire surface of (2.1) is always standing over the lower approximation surface among feasible x and v. Let $\bar{\gamma}(v,x) = \bar{\omega}'' + \omega' v^2 + \omega x$ be the lower bound approximation. Let $F(v,x) = v^2(w+x)^{\frac{2}{3}} - \omega' v^2 - \omega x$, and we then have the following proposition:

Proposition 2.2. *Given $[v_{\min}, v_{\max}]$ as the speed interval and m as the maximum capacity of a ship, $\bar{\gamma}(v,x) \le \gamma(v,x)$ for any $v \in [v_{\min}, v_{\max}]$ and $x \in [0,m]$, if the condition $\bar{\omega}'' = \min\{F(v_{\min},0), F(v_{\min},m), F(v_{\max},0), F(v_{\max},m)\}$ is satisfied.*

Proof. We prove this proposition by discussing the following two cases. (i) Suppose v is fixed by a value in $[v_{\min}, v_{\max}]$; then we have $F''(x) = -\frac{2}{9}v^2(w+x)^{-\frac{4}{3}}$ which is always negative. This thus shows that $F(x)$ is concave and that the minimum point should be located on the extreme points of x; (ii) Suppose x is fixed by a value in $[0,m]$, we then have $F'(v) = \Big(2(w+x)^{\frac{2}{3}} - 2\omega'\Big)\, v$. Since v only takes a positive value, the function $F(v)$ must be monotonic. Thus the minimum point of $F(v)$ must be located at the extreme points of v as well. In summary, the minimum point of $F(v,x)$ must be found from the four extreme points $(v_{\min},0)$, $(v_{\min},m)$, $(v_{\max},0)$ and $(v_{\max},m)$. □

Substituting $\bar{\gamma}(v,x)$ into Eq. (2.2), we get a lower bound $\phi_i(x)$ for the operating cost function. Further substituting $\bar{\phi}(x)$ into LSFD, we then obtain an MILP whose optimal objective value is an upper bound for LSFD.

2.4.2 Relaxation of LSFDA

Although LSFDA is an MILP after an approximation of γ, the number of decision variables and constraints it has are still proportional to the number of services, which grow exponentially with the number of regions in an ASN. We find it not feasible, using a commercial solver, to directly solve LSFDA or its linear programming relaxation, when the number of regions grows large. Moreover, in LSFDA, Constraints (2.3c) are associated with services, so it is also not feasible to directly apply the classical column generation approach to the linear programming relaxation. Therefore, we here relax LSFDA further by aggregating the capacity constraints with respect to the same ship type s and link e, so as to obtain a new relaxation model (P) as follows.

$$
\begin{aligned}
\text{(P)} \quad \max \quad & \sum_{(o,d)\in D} \sum_{i\in\Omega(o,d)} (p_{od} - \eta_{odi}) x_{odi} - \sum_{i\in\Omega} \left(c_i(v_i) - b_{s_i} k_i\right) y_i - \sum_{s\in S} b_s n_s && \text{(2.5a)} \\
s.t. \quad & \sum_{i\in\Omega(o,d)} x_{odi} \le q_{od}^{\max}, \quad \forall (o,d) \in D && \text{(2.5b)} \\
& \sum_{i\in\Omega(o,d)} x_{odi} \ge q_{od}^{\min}, \quad \forall (o,d) \in D && \text{(2.5c)} \\
& \sum_{i\in L(s,e)} \sum_{(o,d)\in D(i,e)} x_{odi} \le m_s \sum_{i\in L(s,e)} y_i, \quad \forall s \in S, \forall e \in A && \text{(2.5d)} \\
& \sum_{i\in\Omega: s_i = s} k_i y_i \le n_s, \quad \forall s \in S && \text{(2.5e)} \\
& x_{odi} \ge 0, y_i \ge 0. \quad \forall (o,d) \in D, \forall i \in \Omega && \text{(2.5f)}
\end{aligned}
$$

In the aggregated capacity constraints (Constraints (2.5d)), $L(s,e)$ is the subset of services which traverse link e and are operated by ships of type s. It is worth noting that the relaxation model (P) is a linear programming model, and that Constraints (2.5d) are not associated with services. The total number of constraints in the relaxation model (P) is reduced to $O(|D| + |S||A|)$.

2.4.3 Column Generation Based Heuristic

In this section, we propose a column generation based heuristic. It first employs a column generation procedure to solve the relaxation model (P), and then from the generated services (columns), it selects a subset of them to construct a solution to LSFDA and LSFD using an MILP solver. Similar approaches have been applied and shown in the literature to be competitive for solving various large-scaled MILP problems [1, 22].

Given the LP relaxation model of LSFDA, we initialize the Restricted Master Problem (RMP) that includes only a subset of columns Ω_a. Let θ_{od} and θ'_{od} be the dual variables of Constraints (2.5b) and (2.5c). Let π_{se} and ρ_s denote the dual variables of Constraints (2.5d) and (2.5e). Solving the RMP to optimality, we obtain a corresponding dual solution $(\theta, \theta', \pi, \rho)_a$ with respect to Ω_a. Note that if $(\theta, \theta', \pi, \rho)_a$ is feasible to the dual problem of (P), which is presented in Eq. (2.6a)-(2.6d), the optimal solution to the RMP associated with Ω_a is then optimal to (P). Otherwise, there exist columns in $\Omega \setminus \Omega_a$ such that Constraints (2.6b) or (2.6c) are violated, and we shall add these columns to the RMP to improve its optimal value.

$$
\begin{aligned}
\text{(D)}\quad \min\quad & \sum_{(o,d)\in D} q_{od}^{\max}\theta_{od} - \sum_{(o,d)\in D} q_{od}^{\min}{\theta'}_{od} + \sum_{s\in S} n_s\rho_s && \text{(2.6a)}\\
s.t.\quad & \theta_{od} - {\theta'}_{od} + \sum_{e\in A(i,o,d)} \pi_{s_i e} \geq p_{od} - \eta_{odi}, \forall (o,d)\in D, \forall i \in \Omega(o,d), && \text{(2.6b)}\\
& -\sum_{e\in A(r_i)} m_{s_i}\pi_{s_i e} + k_i\rho_{s_i} \geq -c_i(v_i) + b_{s_i}k_i, \forall i\in\Omega, && \text{(2.6c)}\\
& \theta_{od}, \theta'_{od}, \pi_{se}, \rho_s \geq 0. \forall (o,d)\in D, \forall s\in S, \forall e\in A. && \text{(2.6d)}
\end{aligned}
$$

We next formulate the pricing problem to identify the columns that violate the dual constraints. Define violation values $\Delta'_{od}(i)$ and $\Delta(i)$, respectively, as the amount by which Constraints (2.6b) (for each (o, d) pair and service i) and Constraint (2.6c) (for each service i) are violated. Based on the dual solution obtained from solving the RMP optimally, we compute $\Delta'_{od}(i)$ and $\Delta(i)$ in Eq. (2.7)-(2.8).

$$\Delta'_{od}(i) = p_{od} - \eta_{odi} - \theta_{od} + \theta'_{od} - \sum_{e \in A(i,o,d)} \pi_{s_i e}, \tag{2.7}$$

$$\Delta(i) = -c_i(v_i) + b_{s_i} k_i + \sum_{e \in A(r_i)} m_{s_i} \pi_{s_i e} - k_i \rho_{s_i}. \tag{2.8}$$

Note that for each column i (service i) and each (o, d) pair in $D(i)$, Constraints (2.6b) are violated if, and only if $\Delta'_{od}(i) > 0$, and that Constraints (2.6c) are violated if, and only if $\Delta(i) > 0$. Define the total violation value for column i as the weighted sum of violation values of those violated constraints, i.e., $\delta \max\{\Delta(i), 0\} + (1 - \delta) \sum_{(o,d) \in D(i)} \max\{0, \Delta'_{od}(i)\}$, where $\delta \in (0, 1)$. It can be seen that if this value is equal to 0, column i does not violate any dual constraints, and otherwise, column i must violate at least one of the dual constraints. The pricing problem is then to find a column of the maximal total violation value. If this value is positive, we find a column that violates Constraints (2.6b)-(2.6c). Otherwise the dual constraints are all satisfied and (P) reaches optimality.

When no constraints in (D) are violated, the relaxation model of LSFDA is solved to optimality, and the column generation procedure obtains a set of generated services. The total number of generated services is significantly smaller than that of all possible services, and therefore we can use an MILP solver to select a best subset of them to construct a heuristic solution to LSFDA, which is also a heuristic solution of LSFD.

2.4.4 Solving the Pricing Problem

It can be seen that the pricing problem defined in Section 2.4.3 is to determine a service combination (s, r, k, v) that achieves the maximal total violation value with regards to Constraints (2.6b) and (2.6c). Correspondingly, we can solve the problem for each ship type s. Given s, the problem can be formulated as a path search problem on a time-space network G_s (Figure 2.3), where the time horizon is discretized by T

time points, denoted by $0, 1, ..., T$, with the length between adjacent time points being one day. In the space dimension, each region z is duplicated to two copies z^- and z^+, and let Z^- and Z^+ indicate the set of these two types of copies. In G_s, a node is denoted by a region-time pair (z^-, t) or (z^+, t), indicating that a service may enter region z, or leave region z, respectively, at day t. Each edge $(z_u^-, t, z_u^-, t+1)$ from node (z_u^-, t) to node $(z_u^-, t+1)$ indicates a one-day port stay in region z_u during the period from t to $t+1$, while each edge (z_u^+, t, z_v^-, t') from node (z_u^+, t) to node (z_v^-, t') indicates an inter-regional voyage of a total $t'-t$ days, leaving region z_u at day t and entering region z_v at day t'. Similarly, each edge (z_u^-, t, z_u^+, t') from node (z_u^-, t) to node (z_u^+, t') indicates an intra-regional voyage of a total $t'-t$ days, within region z_u during the period from t to t'. It can be seen that for each edge indicating an inter-regional or intra-regional voyage, the speed of the voyage associated with the edge can be determined by the ratio between the distance and the transit time of the corresponding voyage.

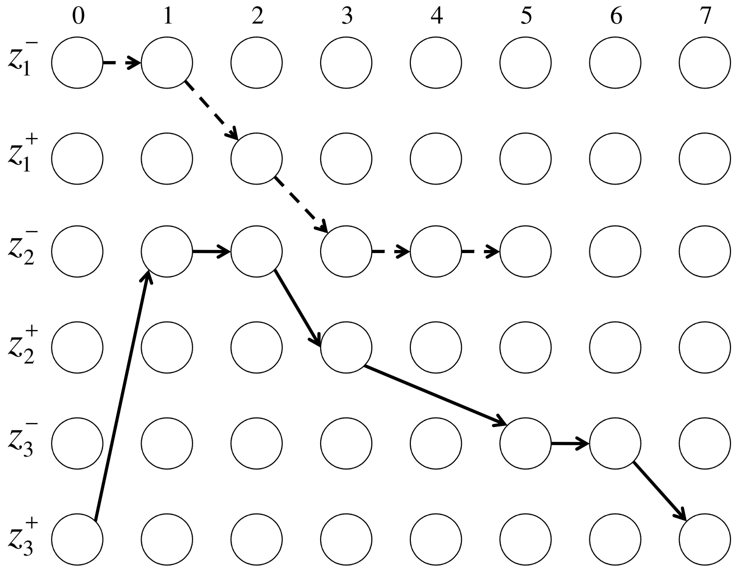


Figure 2.3: Time-space network representation of the pricing subproblem

Accordingly, a *feasible* path on the time-space network from $(z_u^-, 0)$ to $(z_u^-, 7k)$

for a particular k with $7k \leq T$ can represent a feasible service (s, r, k, v), with the route r following the same region sequence as the path, and with the vessel speed of each voyage equal to the speed associated with the corresponding voyage edge. For example, in Figure 2.3 the solid-line path starts at region z_3 and takes one day to sail to region z_2, then stays at port in region z_2 for one day, followed by a one day voyage in region z_2. The service returns back to region z_3 after traveling for two days, and afterwards spends one day at port and another day sailing in region z_3. The solid-line path forms a closed region sequence $r' = \{z_3 - z_2 - z_3\}$ with speed v' determined by corresponding edges, and ends at $(z_3^+, 7)$, implying $k = 1$. On the G_s, it thus corresponds to a feasible service $(s, r', 1, v')$. Similarly, we can see the dashed-line path does not represent a feasible service since it starts at $(z_1^-, 0)$ but does not end at any $(z_1^-, 7k)$ with $7k \leq T$.

Therefore, the pricing problem for a given ship type s is equivalent to finding an optimal feasible path on G_s that represents a service combination with the total violation value maximized. To solve the pricing problem for each given ship type s, we apply a path search algorithm (see Algorithm 1) that generates paths on G_s by adding new nodes to partial paths that have been obtained. Let $\text{Path}(z, t)$ indicate a collection of partial paths that have been obtained with the end node at (z, t), where $z \in Z^- \cup Z^+$. In each iteration of the algorithm, we choose an existing partial path in $\text{Path}(z', t')$ for some z' and t', and extend it by adding one more node to the tail. When the new partial path forms a feasible path that ends at a node (z, t) with $t = 7k$ for some $k \geq 1$, it can be transformed to a service combination (s, r, k, v). Among all the service combinations obtained, we choose the one with the maximal total violation value as a solution to the pricing problem.

However, due to the large number of possible paths on the time-space network G_s, it is not affordable to search among all feasible paths by enumeration. Therefore, in order to reduce the search space, we define the a dominance relationship between two partial paths, such that only those partial paths that are not dominated will be kept for further exploration. We define this dominance relationship using the violation value of Constraint (2.6c) associated with a service combination (s, r, k, v) in Eq. (2.9), which can be represented as follows.

$$\begin{aligned}-(a_s - b_s + \rho_s)k + (-\tau_s)(7k - \sum_{e \in A(r)} \frac{1}{24}\frac{\ell_e}{v_e}) \\ + \sum_{e \in A(r)} (-\alpha \ell_e(\omega'' + \omega' v_e^2) + m_s \pi_{se}). \qquad (2.9)\end{aligned}$$

Since $(7k - \sum_{e \in A(r)} \frac{1}{24}\frac{\ell_e}{v_e})$ indicates the total port stay time of the service, the second term in Eq. (2.9) can be represented by a total weight of port stay edges if we assign a weight of $-\tau_s$ to each port stay edge in G_s. The third term in Eq. (2.9), $\sum_{e \in A(r)}(-\alpha \ell_e(\omega''+\omega' v_e^2)+m_s \pi_{se})$, can be represented by a total weight of voyage edges if we assign a weight of $(-\alpha \ell_e(\omega'' + \omega' v_e^2) + m_s \pi_{se})$ to each voyage edge in G_s, while a voyage edge in G_s would correspond to a link e in the original ASN, on which the speed v_e can be determined by the endpoints of the voyage edge. Thus, for every partial path X, we can use its total edge weight, denoted by $\Delta(X)$, to represent a partial violation value (wrt. Constraints (2.6c)) associated with any service combination that can be obtained from X.

For any two partial paths X and Y that follow the same sequence of regions and end at the same node on G_s, we can define that X is dominated by Y if $\Delta(X) \leq \Delta(Y)$. Accordingly, if a partial path X is dominated by an existing partial path Y, then we can exclude X from further explorations. This is valid, because for any feasible path P on G_s that can be obtained from X, we can construct another feasible path P' by replacing X with Y. Since P and P' follow the same sequence of regions, the violation of Constraints (2.6b) for P and P' are the same, which, together with $\Delta(X) \leq \Delta(Y)$, implies that the total violation value for P is not greater than that of P'. Therefore, we can safely exclude X without losing the optimal path.

To further control the search space, we define $\Delta'_{od}(X)$, for each partial path X and for each (o, d) pair that appears in X, as the violation value (wrt. Constraints (2.6b)) associated with (o, d) as well as the path segment of X from o to d. Noting that $\Delta(X)$, defined earlier, denotes the partial violation value (wrt. Constraints (2.6c)) for X, we can assign each partial path X a total partial violation value, defined as the sum of $\delta \min\{0, \Delta(X)\}$ and $(1 - \delta) \min\{\Delta'_{od}(X)\}$ for all the (o, d) pairs appearing in X, and then use this value as another criteria to exclude X from further exploration. Specifically, for each node (z, t), among all the partial paths in $\text{Path}(z, t)$, we exclude those with relatively small total partial violation values, with the hope that only some of the partial paths of good qualities are explored.

Finally, in order to obtain service combinations that are consistent with patterns observed in practice, we keep a partial path for exploration only if it can generate services that satisfy the following rules: (i) No more than g regions are visited; (ii) no links are traveled more than g' times; (iii) the total duration does not exceed g'' weeks. In this study, we scale regions by continental areas and set $g = 3$, $g' = 1$, and $g'' = 15$, which identifies practical route patterns for liner shipping such as A-B-A(direct pattern), A-B-C-A(round-the-world pattern) and A-B-C-B-A (pendulum pattern). For more applications, as the region scope gets smaller and more regions are involved in the service network, we may adjust parameter g to be larger and allow rotating more region nodes in a service.

We can now summarize the above path search algorithm in Algorithm 1. For each ship type s, it returns a service combination (or column) (s, r^*, k^*, v^*) that corresponds to a feasible path in G_s with the largest total violation value obtained. Among all the columns returned for various ship types, we pick the one with the largest total violation value as a solution to the pricing problem.

Algorithm 1 Path Search Algorithm

```
Initialize the service combination $(s, r^*, k^*, v^*)$ that corresponds to the incumbent optimal path in $G_s$ by setting $(s, r^*, k^*, v^*) = NULL$;
for each node $(z, t)$ in the time-space network do
    Initialize Path$(z, t)$ = NULL;
end for
for each $z$ in $Z^- \cup Z^+$ do
    Initialize $X = \{(z, 0)\}$, and Path$(z, 0)$=Path$(z, 0) \cup \{X\}$;
end for
for each node $(z, t)$ that $z \in Z^- \cup Z^+$ and $t$ moves from 1 to $T$ sequentially do
    for each node $(z', t')$ that connects to $(z, t)$ with $t' < t$ do
        for each partial path $X'$ in Path$(z', t')$ do
            Construct a new partial path $X$ by adding node $(z, t)$ to the tail of $X$ and compute its total partial violation value;
            if X can generate services that satisfy the practical rules, and $X$ is not dominated by any other partial paths in Path$(z, t)$ then
                Update Path$(z, t)$ by adding $X$ to Path$(z, t)$ and removing partial paths dominated by $X$;
            end if
            if $X$ is a feasible path and its corresponding service combination $(s, r, k, v)$ is of a total violation value larger than that of $(s, r^*, k^*, v^*)$ then
                Update $(s, r^*, k^*, v^*)$ by setting $(s, r^*, k^*, v^*)$ to be $(s, r, k, v)$;
            end if
        end for
        Reduce Path$(z, t)$ by removing partial paths of relatively small total partial violation values;
    end for
end for
Return $(s, r^*, k^*, v^*)$.
```

2.4.5 Speed Optimization

In the column generation based heuristic, the speed strategy v_i of a service i is determined by the pricing problem, where the voyage cost with respect to speed v is related to the approximation of γ. Once a solution is obtained and the cargo flow decision variable $\bar{x}$ is calculated, we can then calculate the actual fuel cost associated with v after substituting $\bar{x}$ into Eq. (2.1). Based on the calculated actual fuel cost on each service i, we apply a speed optimization procedure which in essence solves a single-source shortest path problem on a time-space network.

Given (s_i, r_i, k_i) with respect to service i, we construct a time-space network (Figure 2.4) to optimize speed v_i. Similar to the time-space network described in Section 2.4.4, we expand, in the space dimension, the involved regions in route r_i with each region, except for the last one, being duplicated to two copies, resulting a total of $2[r_i] + 1$ regions, where $[r_i]$ indicates the number of inter-regional links that route r_i contains. For the time horizon, we discretize the k_i weeks voyage length by $7k_i + 1$ time points $0, 1, ..., 7k_i$ and define the length between adjacent time points as one day. We then define the network node as a region-time pair (z, t) which denotes a state in region of z at day t. An edge e connecting (z, t) to (z', t') denotes either a port stay or a voyage. If $z = z'$, the edge denotes a port stay at region z. On the other hand, if $z \neq z'$, the edge denotes a voyage from z to z' with a total time of $t' - t$ days. More specifically, depending on whether z and z' refer to the same region, the voyage can be further classified as an inter-regional voyage or an intra-regional voyage. The edge weight is calculated by $\alpha \ell_e \beta (w_{s_i} + \bar{x}_{ie})^{\frac{2}{3}} v_{ie}^2$, where $v_{ie} = \frac{\ell_e}{24(t'-t)}$. The port stay duration is also discretized into days, where the daily stay cost is given by τ_{s_i}.

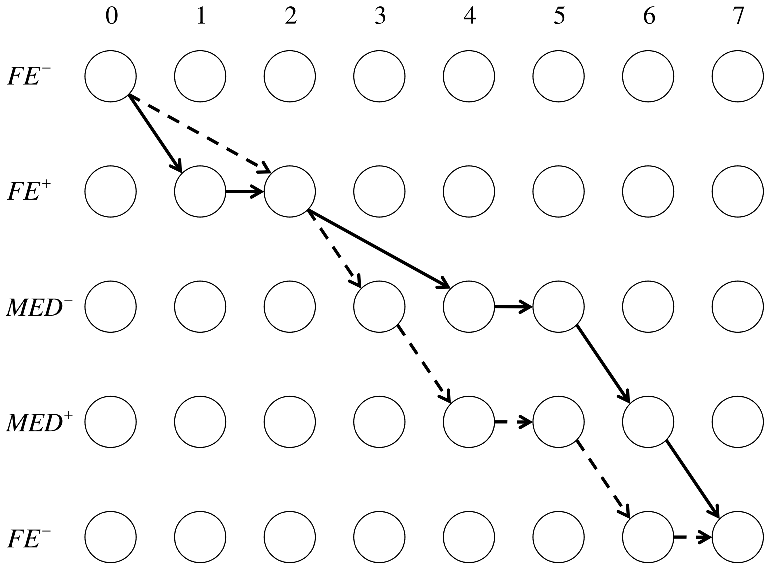


Figure 2.4: Time-space network representation of a one-week liner service

Figure 2.4 shows an illustrative example of a weekly service operated by one ship: $z_1(FE^-)-z_2(FE^+)-z_3(MED^-)-z_4(MED^+)-z_5(FE^-)$. The one week duration is discretized into seven days, so each path from the starting node $(z_1, 0)$ to the finishing node $(z_5, 7)$ represents a different speed strategy. Two speed strategy examples are illustrated by the dashed-line path and the solid-line path in Figure 2.4. The dashed-line path represents a schedule that starts from FE → sails inside FE for 2 days → sails to MED in 1 day → sails inside MED for 1 day → stops at MED for 1 day → sails back to FE in 1 day → stops at FE for 1 day before the voyage ends. Similarly, the solid-line path follows a schedule that starts from FE → sails inside FE in 1 day → stops at FE for 1 day → sails to MED in 2 days → stops at MED for 1 day → sails inside MED for 1 day → sails back to FE in 1 day and ends the voyage. It is easy to see that the optimal speed strategy on this time-space network corresponds to a shortest path from $(z_1, 0)$ to $(z_{2[r_i]+1}, 7k_i)$, which can be solved by the classic Dijkstra's algorithm [35].

Table 2.1: Approximation results of γ

Cargo (TEUs)	Speed (knots)	γ from Eq. (2.1)	γ' by from Eq. (2.4)	Errors (%)
2000	14	0.065	0.045	30.26
4000	14	0.077	0.067	13.68
6000	14	0.089	0.088	0.69
8000	14	0.100	0.110	10.07
10000	14	0.110	0.132	19.33
2000	18	0.107	0.103	3.38
4000	18	0.128	0.124	2.62
6000	18	0.147	0.146	0.69
8000	18	0.165	0.168	1.52
10000	18	0.182	0.189	3.83
2000	24	0.190	0.216	13.88
4000	24	0.227	0.238	4.79
6000	24	0.261	0.260	0.69
8000	24	0.294	0.281	4.21
10000	24	0.324	0.303	6.55

2.4.6 Iterative Search Algorithm

The approximation of γ inevitably produces errors in calculating the fuel consumption, which can be gauged in the following way. We first select a bundle of sample points from Eq. (2.1), where v and x are uniformly distributed over the feasible domain. ω'', ω' and ω are then fixed using least squares estimation. Eq. (2.4) has an approximation value for Eq. (2.1) over the sample space. Then, given v and x, we use Eq. (2.1) to compute the actual fuel consumption rate γ, and use Eq. (2.4) to compute the approximate fuel consumption rate γ'. Table 2.1 presents the approximation results of γ for a 10000 TEUs containership with loaded cargo from 2000 to 10000 TEUs and speed from 14 to 24 knots.

The first and second columns indicate the cargo and speed. The third and fourth columns show the actual value and approximated value of the fuel consumption rate as computed by Eq. (2.1) and Eq. (2.4), respectively. The gaps between the two are presented in the last column. We observe that the approximation does not perform consistently because the error is 7.78% on average, and can be as large 30% for a ship carrying 2000 TEUS cargo at a speed of 14 knots. To improve on the approximation results, we then resort to the classic tangent plane approximation method which is able to locally reduce approximation errors to a given point for a two variable function. In our case, speed and cargo are given by $\bar{v}$ and $\bar{x}$. If we regard v^2 as a variable, the tangent plane approximation of γ is obtained by the following equation:

$$\gamma(v^2, x) \approx \gamma(\bar{v}^2, \bar{x}) + \frac{\partial \gamma}{\partial v^2}(\bar{v}^2, \bar{x})(v^2 - \bar{v}^2) + \frac{\partial \gamma}{\partial x}(\bar{v}^2, \bar{x})(x - \bar{x})$$

In LSFD, decisions on speed and cargo flow are differentiated by edges and services. We define γ_{ie} as the fuel consumption rate at edge e of service i. The approximation parameters are initialized by ω_0'', ω_0' and ω_0 respectively. After a solution of LSFD is obtained, we apply the tangent plane approximation to update the approximation parameter γ_{ie} according to the known speed $\bar{v}_{ie}$ and cargo flow $\sum_{(o,d)\in D(r_i,e)} \bar{x}_{odi}$ on link e of service i. For each specific link e on service i, we set ω_{ie}'', ω_{ie}' and ω_{ie} adaptively as follows.

$$
\begin{aligned}
\omega''_{ie} &= -\frac{2}{3}\beta\bar{v}^2_{ie}(w_{s_i} + \sum_{(o,d)\in D(r_i,e)} \bar{x}_{odi})^{-\frac{1}{3}} \sum_{(o,d)\in D(r_i,e)} \bar{x}_{odi}, \\
\omega'_{ie} &= \beta(w_{s_i} + \sum_{(o,d)\in D(r_i,e)} \bar{x}_{odi})^{\frac{2}{3}}, \\
\omega_{ie} &= \frac{2}{3}\beta\bar{v}^2_{ie}(w_{s_i} + \sum_{(o,d)\in D(r_i,e)} \bar{x}_{odi})^{-\frac{1}{3}}.
\end{aligned}
$$

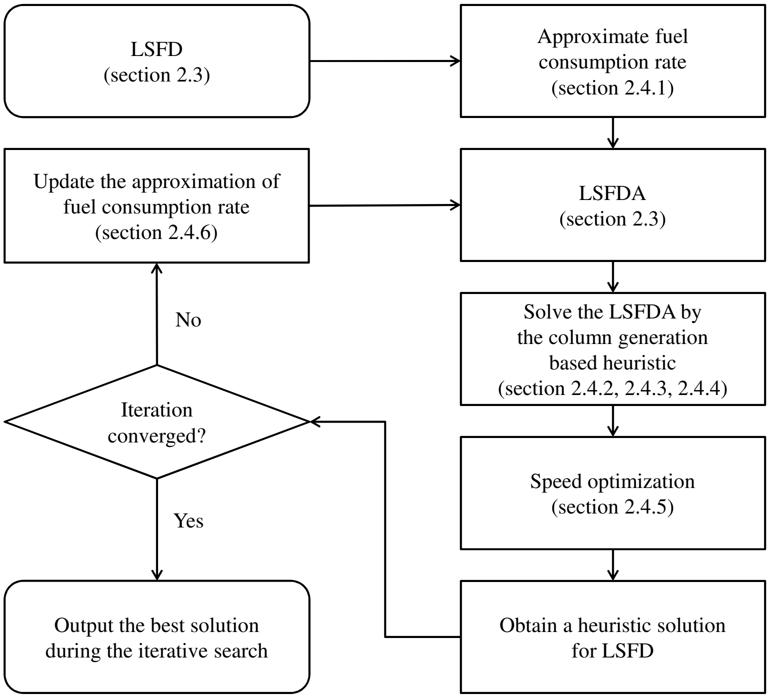

LSFD
(section 2.3)
Approximate fuel consumption rate
(section 2.4.1)
Update the approximation of fuel consumption rate
(section 2.4.6)
LSFDA
(section 2.3)
No
Solve the LSFDA by the column generation based heuristic
(section 2.4.2, 2.4.3, 2.4.4)
Iteration converged?
Speed optimization
(section 2.4.5)
Yes
Output the best solution during the iterative search
Obtain a heuristic solution for LSFD

Figure 2.5: Flow chart of the iterative search algorithm

As the approximation coefficients ω''_{ie}, ω'_{ie} and ω_{ie} change with the current solution, the model parameters of LSFDA, which approximates LSFD, change as well, and therefore a new solution may be obtained by re-optimizing the updated problem. We here propose an iterative search algorithm (ISA) that embeds the column generation based heuristic. Figure 2.5 illustrates the flow chart of ISA, where each procedure is referenced properly. It is also worth noting that we only update the approximation of γ_{ie} for services in operation (service frequency $y_i > 0$), so the pricing problem still uses the initial approximation to generate new services in the column generation heuristic. Finally, the termination of ISA occurs when convergence is achieved, i.e., the newly obtained solution is exactly the same as the previous solution.

2.5 Computational Experiments

In this section, extensive experiments are conducted to show the accuracy and efficiency of our model and solution methods. Different test cases are generated based on data in existing literature as well as from actual practice. We report on the performance of solution approaches discussed in Section 2.4, and show how significant the considered speed-load dependent fuel cost function is by comparing our solutions to a benchmark solution where the load factor is not considered. Next, we conduct a sensitivity analysis of various practical factors, such as demand variation and change in fuel cost. Finally, we evaluate and discuss a few deployment strategies used in actual practice when carriers establish a fleet deployment plan. All algorithms are implemented in C++, using IBM ILOG CPLEX 12.6 Callable Library interface. Experiments are carried out on an Intel Core i5 (3.4 GHz) desktop PC with 4 GB RAM.

2.5.1 Data

The experiment is based mainly on a global aggregate network of up to 9 regions, including Far East (FE), US West Coast (USWC), US East Coast (USEC), North Europe (NE), Mediterranean (MED), South Africa (SAF), West Africa (WAF), Central America (CA) and South America (SA). These nine market regions are represented by nine network nodes. Distances of inter-regional links and intra-regional links are estimated by average values in nautical miles among existing services, and are detailed in Table 2.2. The maximum weekly cargo demand of each region pair is normally distributed with the a mean of 5000 TEUs and standard deviation of 1000 TEUs. The minimum weekly demand for each region pair is set at 0. The length of the planning horizon, i.e., the maximum duration for planning a service, is set at 15 weeks (105 days). The freight rate for each cargo demand (o, d) is uniformly distributed in a range whose interval boundaries are proportional to the distance from region o to d. Profits and costs are measured in one thousand US dollars, and the fuel oil price is set at 0.7 thousand USD/TON. The distance matrix of the service network with nine regions is shown in Table 2. This test case is named REG-9. We next generate an additional eight test cases by trimming and expanding REG-9. Altogether, the nine test cases are categorized into three groups: Small-sized cases containing only 3-5 regions (REG-3, REG-4, REG-5); medium-sized cases containing 7-9 regions (REG-7, REG-8, REG-9); and large-sized cases (REG-12, REG-15, REG-18) that are generated by expanding the number of network nodes (regions) in REG-9 with random inter-regional and intra-regional distances. Finally, for each test case we generate ten test instances having random demand and freight rates.

Table 2.2: Distance matrix of a service network with nine regions in nautical miles (REG-9)

	FE	USWC	USEC	NE	MED	SAF	WAF	CA	SA
FE	3134	5564	10658	8164	5241	8418	12256	15846	13256
USWC	4832	644	5321	8486	7260	12414	8790	3427	8718
USEC	9752	5321	1461	3165	4187	9929	6305	2312	6264
NE	8183	8505	3184	959	2322	8156	4533	6822	6749
MED	5431	7260	3848	2215	1649	6585	4001	6290	6217
SAF	8494	12410	9902	8157	6588	832	3838	8987	4838
WAF	12284	8786	6193	4533	4001	3838	3773	5363	4300
CA	15828	3423	2307	6803	6272	8987	5363	4179	5291
SA	13284	8714	6169	6746	6217	4838	4300	5291	3092

We consulted a few managers in a large shipping company and noticed nowadays the average-loaded TEU container weighs about 8-13 tons. Without loss of generality, we then measure the ship net weight by TEUs as shown in Table 2.3. We consider four ship types with capacities of 10000, 8000, 5000 and 3000 TEUs. Ship specific parameters refer to MARIT MAERSK (10000 TEUs), TENO MAERSK (8004 TEUs), MAERSK DUBROVNIK (5041 TEUs) and MAERSK PENANG (2902 TEUs). The weekly fixed operating cost for ship types I, III, and IV is extracted from [71], while the cost for ship type II is set between the cost values of type I and III. We set the weekly fixed idle cost to 1/3 of the operating cost after some consultation. Likewise, following [71], we set the daily port stay cost as the hourly cost multiplied by 24 hours. Parameter β is set to $9.46 \cdot 10^{-7}$ after regression on fuel consumption data extracted from [53]. In general, we assume there are 80 ships in total ready for deployment, there being 20 of each type.

Table 2.3: Characteristics of different ship types

	Type I	Type II	Type III	Type IV
m_s (TEUs)	10000	8000	5000	3000
w_s (TEUs)	4513	3816	3212	1731
a_s (USD per week)	173076	150000	115384	76923
b_s (USD per week)	57692	50000	38461	25641
τ_s (USD per day)	80000	64000	40000	24000
n_s	20	20	20	20

Moreover, we introduce a benchmark model for LSFDA in which the fuel consumption rate is dependent only on speed, where m denotes the maximum capacity of the ship.

$$\gamma(v) \quad = \quad \beta v^2 (w+m)^{\frac{2}{3}} \tag{2.10}$$

Since the number of all possible services is exponential to the length of planning horizon, it is not affordable to include all possible combinations of (s, r, k, v) in the benchmark model, even for small-sized cases with three regions only. It is also worth noting that many possible routes, such as those visiting one region repetitively for many times, are not feasible for practical deployment. We therefore, in the benchmark model for LSFDA, only include the decision variables and constraints that are associated with routes satisfying those practical patterns proposed in Section 2.4.4. By doing so, we are able to reduce the number of feasible services significantly. The benchmark model for LSFDA can then be constructed by enumerating (s, r, k, v) with v taking the optimal speeds for each (s, r, k). For small and medium-sized cases, we can apply the speed optimization procedure introduced in Section 2.4.5 to obtain optimal speed v, and then use CPLEX's mixed integer programming solver on the benchmark model. For large-sized cases we can only use the column generation based heuristic to get approximated benchmarks, because we find that CPLEX is unable to build the model when the magnitude of the problem exceeds the computer memory. These benchmark results enable us to evaluate the significance of the speed-load dependent fuel cost function presented in Eq. (2.1).

2.5.2 Performance of the Solution Methods

We use the following notations in presenting the experimental results in Table 2.4. CPX and CG denote the two approaches that directly use CPLEX's mixed integer programming solver and the column generation based heuristic on the benchmark model, respectively. SR denotes re-optimization on given solutions using the speed optimization procedure, and CPX-SR and CG-SR denote the approaches that apply SR to CPX and to CG. ISA denotes the iterative search algorithm that solves the LSFD with the speed-load dependent fuel cost function. We also use UB to represent a bounding method that solves the LSFDA with $\bar{\gamma}$ defined in Section 2.4.1 using CPLEX's mixed integer programming solver directly. Results of all test cases are the averages of ten instances solved by different approaches with a runtime limit of one hour.

In Table 2.4, the performances of CPX, CG, CPX-SR, CG-SR and ISA are compared in percentage gaps with regard to objective profits. The total number of columns produced by CPLEX and CG when solving the benchmark model are listed. Results of ISA are compared with UB in addition to both CPX and CG. From Table 2.4, we observe that CPX solves two small-sized test cases (REG-3 and REG-4)

Table 2.4: Overview of algorithm performances

REG	CPX		CG		CPX-SR	CG-SR	ISA		
	COL	OPT(%)	COL	vs. CPX(%)	vs. CPX(%)	vs. CG(%)	vs. CPX(%)	vs. CG(%)	vs. UB(%)
REG-3	168	0.00	50.4	-0.11	0.32	0.33	0.37	0.48	-23.50
REG-4	580	0.00	108.8	-0.90	0.18	0.14	0.23	1.14	-19.14
REG-5	1284	0.05	190.5	-0.30	0.24	0.18	-0.07	0.23	-15.64
REG-7	4788	6.15	505.6	-3.85	0.42	0.36	1.18	5.23	-22.83
REG-8	7816	6.38	610.5	-3.56	0.56	0.38	2.34	6.11	-22.64
REG-9	11820	7.03	737.3	-1.00	0.43	0.40	4.06	5.11	-22.98
REG-12	31344	n/a	1369.7	n/a	n/a	0.69	n/a	7.58	n/a
REG-15	58708	n/a	2033.6	n/a	n/a	0.72	n/a	8.76	n/a
REG-18	98093	n/a	3011.7	n/a	n/a	0.46	n/a	8.32	n/a

to optimality. The number of generated columns increases exponentially as the test case size increases, and therefore for large-sized test cases (REG-12, REG-15 and REG-18) having a tremendous number of model variables and constraints, CPX results are not available ("n/a"). In comparison, CG produces good solution quality while significantly reducing the number of columns. SR is confirmed to be useful, as improvements are observed over both CPX and CG. Lastly, ISA performs very competitively with CPX in small-sized test cases while outperforming both CPX and CG in medium and large-sized instances, thus showing the effectiveness of speed re-optimization and iterative search algorithms. The performance gap of ISA to UB is also consistent with the increase in test case size.

The performance of ISA is further studied under various supply-demand conditions. Three new test cases, each consisting of ten instances, are then generated to simulate market conditions: REG-6 with 60 ships of Type IV, REG-12 with 80 ships of Type IV and REG-18 with 100 ships of Type III. Figure 2.6 shows the percentage improvements of ISA over CG, where the horizontal axes denote a decreasing trend in demand, from 100% down to 40% of the initial amount. We observe that ISA performs steadily under demand fluctuations in general, the improvement being more apparent with small demands. Given that the benchmark solutions assume that fuel

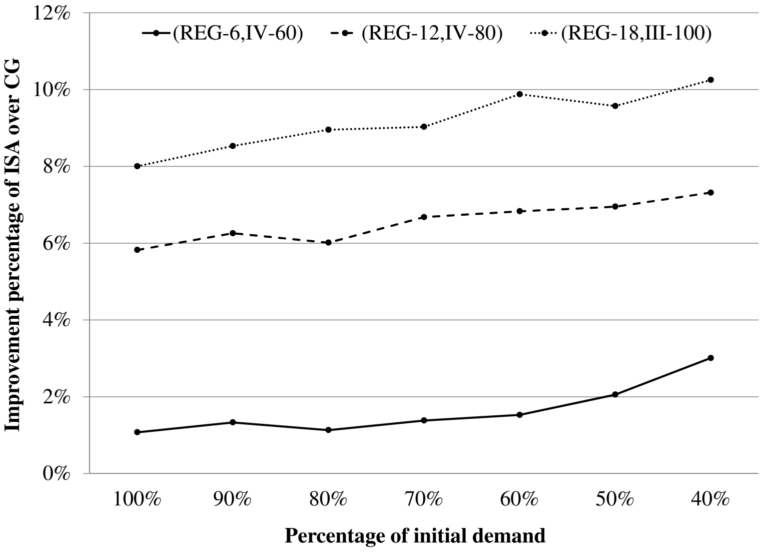


Figure 2.6: ISA performance under different capacity-demand market conditions for instances of three different combinations of region numbers, ship numbers and ship types (6 regions with 60 ships of Type IV, 12 regions with 80 ships of Type IV and 18 regions with 100 ships of Type III)

cost function are irrelevant to the actual loads of cargo, this observation suggests that when ships are more likely to be under-loaded in an over-capacity condition, a speed-load dependent fuel cost function has a big advantage in decision-making.

2.5.3 Sensitivity Analysis

In LSFD, parameters such as demand, fuel oil price and minimum speed are set as constants. In practice, however, fuel oil price and demand fluctuate considerably. Customers may also make particular requests with regard to the transit time of cargo shipments which in turn may restrict the minimum steaming speed. In this section, we show how fleet deployment decisions would respond accordingly when faced with these changes.

Let n'_s denote the number of ships of type s to be deployed. $\frac{\sum_{s \in S} m_s n'_s}{\sum_{s \in S} m_s n_s}$ is then the total capacity utilization rate and $\frac{n'_s}{n_s}$ is the fleet utilization rate for each ship of type s.

We obtain fleet deployment solutions using ISA on test case REG-7 with variations in fuel oil prices, demands and minimum speed requirements. The resulting total capacity utilization, fleet utilization rate and total profits are recorded.

We first study the impact of an increase in fuel oil price on deployment decisions where the fuel oil price ranges from 300 USD/TON to 2400 USD/TON. The results for deployment of 80 and 120 ships are reported in Figure 2.7.

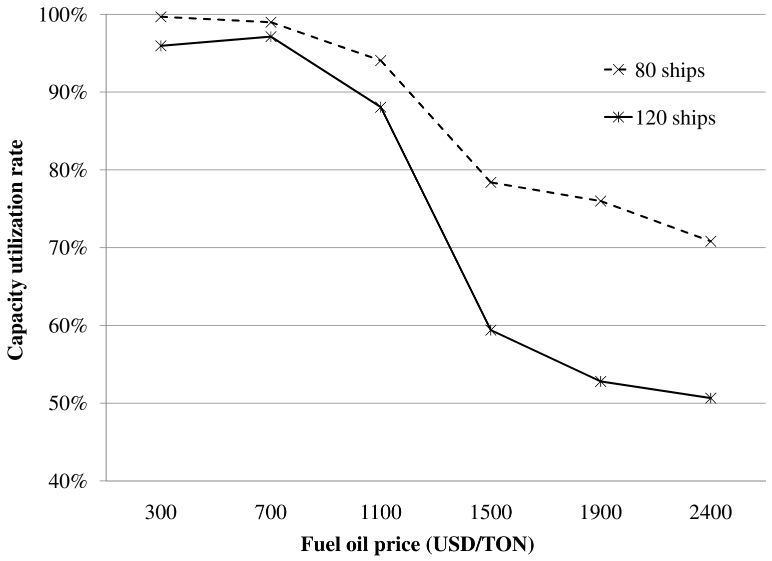
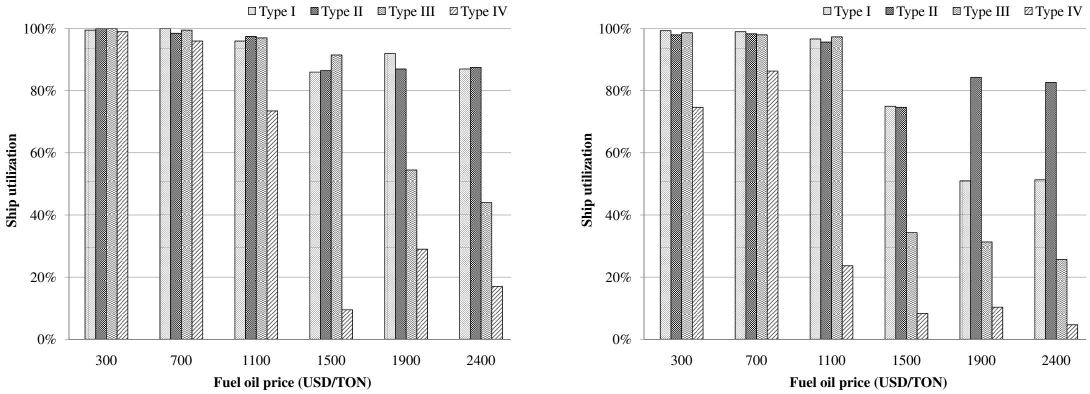


Figure 2.7: Impact of fuel oil price on total capacity utilization rate (80 and 120 ships)

The trend in Figure 2.7 shows that in general the capacity utilization rate decreases with an increase in fuel oil price. This result is as expected, because when revenues cannot compensate for an increase in operating costs, it is more profitable to operate fewer ships, which then results in more idle capacity. However, it is worth noting that there is an exception in the 120-ship case when the fuel oil price changes from 300 USD/TON to 700 USD/TON, which is likely a result of a slow steaming operation. In this case, idle capacity is added to services, and ships adopt slow steaming to counteract the increasing fuel cost. On the other hand, when fuel oil price drops from 700 USD/TON to 300 USD/TON, a larger carrier (the 120-ship case) is more



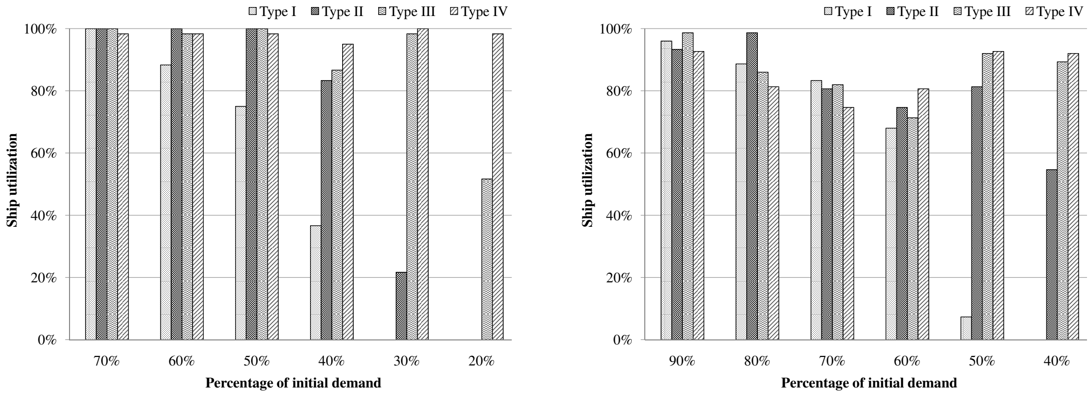
(a) 80 ships (b) 120 ships

Figure 2.8: Impact of fuel oil price on ship utilization (80 and 120 ships)

encouraged to speed up their services due to decreasing benefit from operating extra ships in slow steaming. It is then easy to see that capacity utilization rate is influenced by two conflicting factors, the fluctuating operational costs and the bonus in cost saving from slow steaming.

In Figure 2.8, we show the utilization rates for all ship types. It is easy to observe a sharp decrease in the utilization of ships of Type III and Type IV, whereas utilization of ships of Type I and Type II remains relatively stable. This suggests that larger ships (Type I and Type II) that possess an advantage in economies of scale are to be preferred when the fuel oil price increases. Ships with a large capacity are then encouraged to operate longer service routes so as to cover more markets, allowing more cargos among different origins and destinations to be serviced in a one-way trip.

Figure 2.9 shows the impact of demand variation (from 70% down to 20% of the initial amount for the 80 ship case, and from 90% down to 40% for the 120 ship case) on ship utilization. We observe that the utilized capacity ratio is initially maintained at a level close to 100% (at the 70% point for the 80 ship case, and at the 90% point for the 120 ship case), and then afterwards exhibits a series of drops. We notice that, with a reduction in demand, smaller ships are to be preferred, which is the opposite of what happens when fuel prices rise.



(a) 80 ships

(b) 120 ships

Figure 2.9: Impact of demand variation on ship utilization (80 and 120 ships)

The effect of a minimum speed requirement in fleet deployment decisions is shown in Figure 2.10. From a maximum speed of 24 knots, we create progressive minimum speed levels at which ships are allowed to operate. Carrier profits with regard to different minimum speed levels are shown in Figure 2.10. The profit curve is seen to be concave and monotonically increasing, suggesting that a slower steaming strategy does save on cost efficiently. At the same time, it is observed that the room for such savings diminishes after the minimum speed requirement reaches a level of 14 knots or lower.

2.5.4 Fleet Deployment Strategy

In this section, we analyze different strategies for carriers in fleet deployment when a fleet of new ships is available for use. If an adjustment in ship deployment leads to changes in service routes in the network, carriers would have to carefully consider the benefits, as well as the various costs, associated with these changes. Therefore, in most cases, new ships are simply pooled into the original fleet and are deployed to operate on existing routes. We briefly define this strategy as NF-OR. In addition, new ships can be planned to operate jointly with the old fleet on new routes in order to satisfy additional demands. We define this strategy as NF-NR. Furthermore, we

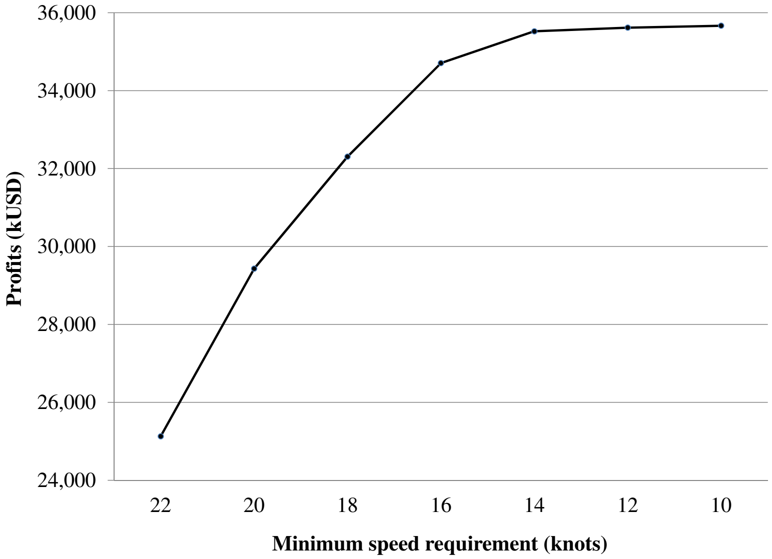


Figure 2.10: Effect of different minimum speed requirement

define OF-OR as a benchmark strategy that maintains the original services with all new ships remaining idle.

We apply our LSFD model to evaluate the above three fleet deployment strategies. The results are demonstrated by conducting experiments on an instance from test case REG-5. An original fleet of 60 ships, with 15 ships of each type, is considered. We then consider that 20 more ships are put into use, each ship type having 5 more. The result of OF-OR is computed by solving the LSFD with the original fleet and a full generated route set. The service routes operated in the solution of OF-OR constitute the original route set. The results of all three fleet deployment strategies are computed using ISA and presented in Table 2.5.

As shown in Table 2.5, OF-OR generates a weekly profit of 29,853 kUSD, while for NF-OR and NF-NR, weekly profit is 30,392 kUSD (1.8% higher than OF-OR) and 30,840 kUSD (3.3% higher than OF-OR) respectively. NF-NR is expected to perform the best among the three fleet deployment strategies, since it has the largest feasible solution space. We observe that the adoption of NF-OR sees changes in half of the

Table 2.5: Results under different fleet deployment strategies

OF-OR (29853 kUSD)				NF-OR (30392 kUSD)				NF-NR (30840 kUSD)			
route	s	k	freq	route	s	k	freq	route	s	k	freq
1-5-3-5-1	I	10	1	1-5-3-5-1	I	11	1	1-2-1	I	6	1
2-3-2	I	5	1	2-3-2	I	5	1	1-4-3-4-1	I	12	1
1-4-3-4-1	II	12	1	1-4-3-4-1	II	12	1	1-5-3-5-1	II	11	1
4-2-5-2-4	III	15	1	1-2-1	II	6	1	2-3-2	II	5	1
1-2-1	IV	6	2	4-2-5-2-4	III	15	1	2-4-2	III	8	1
4-5-4	IV	3	1	4-5-4	III	3	1	2-5-2	III	7	1
								4-5-4	III	3	1
								1-3-1	IV	11	1

original services. An extra ship of type I is put on route 1-5-3-5-1. The ship type is alternated on route 1-2-1 from IV to II and on route 4-5-4 from IV to III. Service frequency on route 1-2-1 is also changed from twice a week to once a week. On the other hand, if NF-NR is adopted, the original route 4-2-5-2-4 will be dropped, and replaced by three newly established service routes 2-4-2, 2-5-2, and 1-3-1.

Although the NF-NR strategy produces the best result for LSFD, it may not be adopted by carriers in practice. The main reason for this is that there is a considerable setup cost incurred in opening a new service route. NF-NR is only 1.5% better than NF-OR in terms of weekly profit, which is possibly not enough to offset additional setup costs involved in opening up a set of new routes.

2.6 Summary

In this work, we propose a comprehensive liner shipping service model for multi-decision making on ship routes, fleet deployment, speed, cargo allocation and frequency. Based on a general fuel consumption function, we incorporate a new speed-load dependent fuel consumption function into the comprehensive liner shipping service model to address the slow steaming strategy widely adopted at present in the industry. The resulting mathematical model is non-linear and very difficult to solve. An approximation method is implemented to transform the non-linear model to a mixed integer linear programming (MILP) model that can be solved by a column generation based heuristic. In addition, we propose an iterative search algorithm that adaptively changes the approximation to further enhance the heuristic. Extensive computational experiments are carried out, using different test cases adapted from actual practice, showing both the significance of the speed-load dependent fuel consumption function and the effectiveness of our approach. We have then investigated the impact of different scenarios on the solutions, such as fluctuations in fuel oil price, demand market, minimum speed requirements and various fleet deployment strategies.

In the problem, although the speed variables are assumed to take discrete values, our solution method is still applicable when the speed variables are continuous. Moreover, our aggregate planning model is versatile because it can be used at different planning levels according to the scope of aggregation. In applications that optimize global liner shipping services, carriers can move from a higher level to a lower level when making plans. By aggregating the service network, a carrier can first use the proposed model to obtain a high level solution, which consists of the design of region rotations, the associated cargo flows and ship assignment. For each region rotation in the higher level solution, by reducing the aggregation scope, i.e., by properly defining the port clusters, the model then becomes suitable for lower and operational level planning.

CHAPTER 3

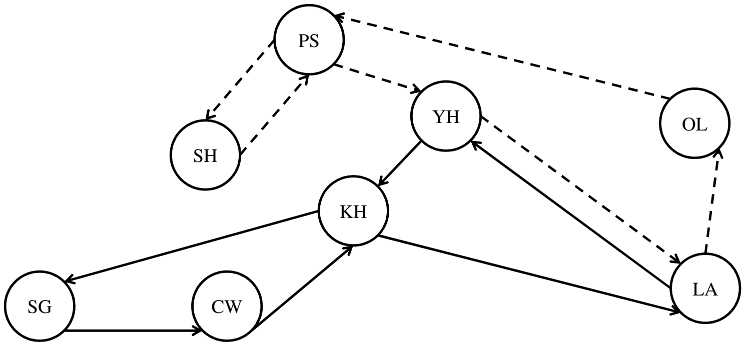
Liner Shipping Network Design with Transshipment Cost

3.1 Introduction

In liner shipping, containerships rotate among seaports to transport cargos with a regular service frequency [54, 40]. The sequences of port calls constitute a service network on which carriers are providing their transportation services. Due to capricious nature of the shipping industry, both supply and demand have frequent fluctuations in the market. In response to the changing market, carriers require adjusting their service networks to keep the competitiveness.

Liner Shipping Network Design (LSND) is to determine the sequences of port calls together with a series of related decisions from tactical and operational levels to maximize the profits. A fleet of similar sized ships is deployed to maintain a designated service frequency, known as *Fleet Deployment*, which decides the type and number of the used ships [47, 45, 46]. The transportation of cargos follows some paths from their origin ports to destination ports, and the determination of such paths is known as *Cargo Routing* [11, 12, 75, 74]. In the LSND, decisions on port call sequences, fleet deployment and cargo routing are often jointly investigated [3, 13, 72, 66].

In liner shipping, cargos can be delivered not only via a direct service but also via a collaboration of multiple services using *transshipment* which allows the transfer of cargos from one ship to another. Transshipment provides carriers with additional options on the cargo routing so as to improve the transportation efficiency. Nowadays, transshipment is at the core of liner shipping. From 1990 to 2012, the growth of total port throughput has accompanied with an increased share of transshipment cargos from 17.6% to 28% [52]. Moreover, more than 80% of the country pairs are connected with each other through transshipment operations [69]. In the practice, since terminal operation costs vary from port to port, the transshipment cost cannot be overlooked in the planning as it will influence the decisions in LSND, such as which port to call and where to transship the cargos.



Rotation A: SG – CW – KH – LA – YH – KH – SG

Rotation B: SH – PS – YH – LA – OL – PS – SH

SG	Singapore	LA	Los Angeles	KH	Kaohsiung	PS	Pusan
CW	Chiwan	YH	Yokohama	SH	Shanghai	OL	Oakland

Figure 3.1: Illustration of intra-transshipment and inter-transshipment

This chapter defines the liner shipping service as a *rotation* that not only involves the sequence of port calls but also a fleet of ships operating on it to maintain a designated frequency. The rotation is *non-simple* if it visits a port more than once, and such port is regarded as a *butterfly port*. In the practice, there are two scenarios for the cargo transshipment. Transshipments can occur on a rotation, in which cargos transfer at a butterfly port. We define it by *intra-transshipment*. Besides, carriers can also tranship the cargos between two different rotations at a port intersected by both of them, which is defined by *inter-transshipment*. We have illustrated both transshipment types in Figure 3.1. In the figure, rotation (A) and rotation (B) are represented by the solid and dashed links, respectively, and KH and PS are their butterfly ports. A path for the cargos from CW to SG starts at CW in the rotation (A). Having arrived at the butterfly port KH, cargos are transshipped to another ship of rotation (A) via intra-transshipment and carried to SG in the next stop. A cargo path from PS to SG starts at PS in the rotation (B). At the intersection port YH of the two rotations, cargos are transshipped from (B) to (A) through inter-transshipment. The transshipped cargos will finally move from YH to SG along the rotation (A).

Solving the LSND even with zero transshipment cost and single ship type is computational challenging, as it can be obtained by reducing a set-covering problem which is known NP-hard [13]. Without considering transshipment cost, LSND can be seen as the *service network design with asset management* (SNDAM) with multiple types of assets being deployed on cycles to maintain the designated service frequency. Many existing models for the SNDAM [see, e.g., 5, 4, 6, 22] have been derived from a Multi-Commodity Flow (MCF) network, in which the arc capacity is aggregated among the passing cycles to capacitate their carried flows [2]. Due to the decomposable structure of the formulation, column generation (CG) [23, 38], can be applied to solve its linear programming (LP) relaxation effectively, which promotes the development of exact and heuristic methods to find good integer solutions. It is worth noting that, although transshipment is addressed in their models, the cost of transshipment is ignored due to the hardness of capturing the amount of transshipped cargos in their MCF based formulations. As discussed in [1], to account for transshipment cost correctly on the MCF network, each arc needs to distinguish the capacity from different cycles to capacitate their carried cargos individually. As a result, the arcs should be duplicated for all feasible cycles, and the MCF network will expand to be tremendously huge. To formulate the problem as a mixed-integer linear program (MILP) based on the expanded MCF network, we find that both its variables and constraints are associated with the rotations that can be exponentially many. In that case, given a subset of rotations, the dual solution to a restricted master problem is not complete, hence conventional CG cannot be applied to solving the LP relaxation of this MILP.

Quite a few model formulations have also been developed for the LSND considering transshipment costs [3, 13, 57]. However, the calculation of transshipment cost has significantly complicated their mathematical formulations. When there are exponentially many rotations involved in the model, most existing works focus on finding heuristic solutions while seldom report the optimality gaps to their obtained solutions. Because the capture of transshipped cargos has destroyed the decomposable structure of the formulation, and it is difficult to solve the LP relaxations of their models using an efficient solution method such as CG to obtain an upper bounds for the problem. As a contribution of this work, we propose for the LSND a new compact MILP formulation whose partial variables (or columns) and constraints (or rows) are interdependent. Since a large number of rotations can be involved in the formulation, the model may contain an exponential number of columns and rows, making it very challenging to solve. Therefore, we develop a new optimization technique, referred to as simultaneous column-and-row generation, to solve the LP relaxation of the MILP, so as to obtain a valid upper bound for the optimal solution to the MILP. The idea behind the SCRG is to work on a restricted master problem that includes subset of variables and constraints. New rotations are generated to induce a simultaneous generation of the columns and rows to the restricted formulation. The SCRG continues to create columns and rows until no new rotations can be found to improve the solution objective of the currently updated restricted master problem. Compared with conventional CG which has been extensively applied, the implementation of SCRG is more sophisticated, and its applications are relatively rare. Based on the newly established model, the major contribution of this work is the development of SCRG for solving the LP relaxation of the model, so that a valid upper bound on the optimal integer solution can be obtained. Novel ideas and techniques have also been investigated to promote the SCRG to work effectively and efficiently. This work embeds the SCRG within a branch-and-price framework to find optimal or near-optimal solutions for the LSND. Furthermore, we have also applied the SCRG to a more complicated LSND with new decisions on frequency and speed. The obtained solutions are compared with the existing benchmark solutions from the literature, which have shown significant superiorities.

3.1.1 Literature Review

As aforementioned, the models for SNDAM are mostly derived from an MCF based formulation which adopts Danzig-Wolfe decomposition and CG to tackle their LP relaxation problems [see, e.g., 5, 6, 22]. However, in these models, arc capacity is shared on the network, transshipment cost cannot be captured. Besides, many decisions for liner shipping, such as the deployment of assets, the determination of frequency and speed for services, and etc, are neglected. It hence requires new models to characterize the LSND appropriately.

Optimizations relevant to liner shipping have been extensively studied and reviewed [16, 19, 48], in which LSND has attracted many attentions. Agarwal and Ergun [1] propose an MILP formulation for the LSND that allows cargo transshipment. But the cost of transshipment is ignored. Fixing the rotations, they present a computational study to discuss the effects of transshipment cost on the cargo routing decisions. To account for transshipment cost, Wang and Meng [71] and Meng and Wang [44] directly apply an integer programming (IP) solver to tackle their MILP models. However, both models reply on a given set of predetermined candidate rotations, which is often too large to enumerate. Reinhardt and Pisinger [61] propose a cost minimization model and develop for it a branch-and-cut method which is examined on small instances with at most only six vessels. Plum et al. [57] exploit a new network design model based on service flows and directly solve it via an IP solver. However, in both models, the number of services allowed to operate is limited. This restriction will significantly reduce the feasible solution space and inevitably deteriorate the solution quality.

Among other work on LSND addressing transshipment cost, quite a few models have been developed, which for the most part is not able to produce optimal solutions or even their upper bounds. Álvarez [3] proposes an MILP model that captures the cost of inter-transshipment only when the capacity constraints of the formulation are dependent of the rotations. The author exploits a heuristic CG to produce feasible rotations, but cannot obtain the optimality of the LP relaxation. In the implementation, additional capacity constraints on dummy arcs are employed to collect the dual information that is missing in the restricted master problem. Borrowing the idea of CG, the author estimates the reduced cost of a new variable using the collected values. Brouer et al. [13] develop another formulation that additionally captures the cost of intra-transshipment. But upper bound on the optimal solution cannot be produced. An MILP is proposed to generate new rotations while cargo routing is solved after fixing rotations. The rotation generation and cargo routing are carried out separately and iteratively for finding better integer solutions. Moreover, Mulder and Dekker [49] and Brouer et al. [14] also separate the rotation construction and the cargo routing when solving the LSND. Their work also provides no upper bound to gauge the solution quality. To conclude, for the LSND that captures transshipment cost, there lacks an appropriate formulation that promotes the development of an exact solution approach for solving its LP relaxation to obtain a valid upper bound. Moreover, although a few solution methods have been developed for solving the LSND, comparisons on their performances under identical problem settings are absent. In this work, we aim to fill these research vacancies.

The Simultaneous Column-and-Row Generation (SCRG) is known to be effective in solving a large-scale LP whose variables and constraints are interdependent. Namely, in the formulation, the appearance of rows is induced by the appearance of columns, and vice versa. Unlike conventional CG that adds variables only, SCRG involves a simultaneous generation of variables and constraints. It is worth noting that *column-and-row generation* has been used with a confusion on the concept of *branch-price-and-cut* in the literature [see, e.g., 27, 33, 79]. The SCRG studied in this work is fundamentally different from the branch-price-and-cut whose generated rows, known as valid constraints, have nothing to do with the optimality of original formulation but strengthen the LP relaxation. Therefore, row generation and column generation can be separated. However, in the SCRG, the generated rows are structural constraints that are necessary to define the original formulation. As new variables are introduced, one must add the structural constraints to validate the feasible region of a solution. Thus, the generations of constraints and variables cannot be separately handled.

SCRG has been employed to solve a few applied IP problems, such as the multiple stock cutting problem in [81] and the time-constrained routing problem in [7]. However, the SCRG may stop prematurely at a suboptimal solution to their LP relaxations since their stopping conditions fail to seize the optimality. Maher [42] investigates an integrated airline recovery problem and solve its LP relaxation by extending a conventional CG approach to the SCRG. The author removes partial constraints at the beginning and add them gradually and tactically via CG. Dual variables for the missing constraints are assigned with proper values to perceive the reduced cost of a missing variable in the restricted master problem. However, in our problem, constraints are associated with a set of rotations which can be too large to enumerate. It is thus not possible to apply the proposed approach to assigning values to the dual variables associated to all those absent constraints. Muter et al. [50] present a generic SCRG framework to solve a large-scale LP with column-dependent rows. They define a row generating pricing problem that requires assigning look-ahead values to those missing dual variables to calculate the reduced cost for a candidate variable. The row generating pricing problem is defined as a bi-level optimization problem, known as an optimization nested by another, which is not always easy to solve especially when constraints are too many to enumerate. If their proposed solution framework is directly applied to our problem, the row generating pricing problem requests the rotations to be explicitly given. Hence, as the candidate rotations are exponentially many, solving the pricing problem will be rather computational expensive. Feillet et al. [27] employ the SCRG to solve an LP relaxation formulation of a bus routing problem. Having obtained the optimal dual solution for a restricted master problem with limited columns and rows, they develop an LP model to assign the values of dual variables for those missing rows, which minimizes the violations to dual constraints. Optimality is achieved when the LP model returns its objective by zero. However, the optimality stopping condition is hard to reach in our case, since each operated rotation in our problem associates with a non-zero cost which can be different from each other. As a result, SCRG will work inefficiently because a large number of rotations are likely to be generated when it stops at the optimality.

3.1.2 Main Results and Contributions

Primary results and contributions of this chapter are summarized as follows.

First, we have developed for LSND a new compact MILP model where costs of intra-transshipment and inter-transshipment are both captured. The formulation can be applied to solve many other practical problems, such as telecommunication network design and railway line planning, in which relevant issues tp transshipment are necessary to consider. Different from existing models for LSND that address transshipment cost, our model formulation is appropriate for applying SCRG to solve the LP relaxation problem so that a valid upper bound on the optimal solution value can be obtained.

Second, in the application of SCRG to resolve the LP relaxation, we introduce a novel two-phase scheme to construct the dual solution to the master problem, which simplifies the pricing problem to finding minimum cost cycles on a weighted network. The proposed solution framework can be generalized and contributed to tackling many other applications whose models have interdependent variables and constraints. We further embed the SCRG within a branch-and-price to obtain optimal or near-optimal solutions for the LSND. Our model and solution methods are extended for tackling more complicated problems with new decisions on service frequency and ship speed.

Third, numerical experiments are conducted to show the effectiveness and efficiency of our model and solution methods. Compared with the MIP-Solver of CPLEX 12.6, our branch-and-price solves more and larger instances and averagely consumes less running time. Our heuristic solutions for the extended problem of LSND are also compared with the existing benchmark solutions, some of which have not been compared under identical settings. As shown in the experiments, our solutions significantly outperform the existing ones, and the average improvement reaches at least 5.3%.

3.1.3 Outline

The rest of this work is organized as follows. The problem and its model formulation is introduced in Section 3.2. The SCRG approach is presented in Section 3.3. We study an extended problem in Section 3.4. Experimental results are reported in Section 3.5, followed by a summary in Section 3.6.

3.2 Model Formulation

We initiate this section with a brief problem description on the LSND. We consider all generated rotations are with weekly frequency, and ship speed on each rotation is given and fixed. To capture the amount of transshipped cargos, we introduce a planning network together with appropriate notations to formulate the problem as an MILP. Since the formulation is of large dimensions both in the number of integer variables and the number of constraints, solving this model will be computational challenging as the problem size increases. To tackle this MILP, in this section, we investigate its LP relaxations and briefly discuss the solution methodologies for solving the relaxation problems, so as to obtain valid upper bounds for the MILP.

3.2.1 Problem Description

Let P denote the set of ports. A transportation demand $k \in D$, identified by its origin port $o(k)$ and destination port $d(k)$, generates a volume q_k each week. For demand k, a unit revenue p'_k is rewarded for transporting a unit cargo. A penalty p''_k is charged for rejecting a unit cargo. Ships are of different types, and we use S to denote the set of ship types. A ship of type $s \in S$ has a maximum capacity m_s, a designed speed v_s (knots), a fuel consumption rate F_s (tons/hour) at the designed speed, and a fleet size h_s. Given a complete set of rotations R, we assume each rotation $r \in R$ has a round-trip operating cost a_r and a total of b_r ships of the same type are deployed on it to maintain the weekly frequency. All ships are assumed to sail at the designed speed which is predetermined. The objective of the problem is to deploy the fleet of ships on a subset of rotations in R that transport the cargo demands of D to maximize the profits obtained per week.

We now represent the problem on a planning network defined as follows. A set of voyage nodes N and transshipment nodes T are involved in the planning network.

Each voyage node represents a port call, and each transshipment node is seen as the quayside of a port. Since the butterfly port may be called twice in a non-simple rotation, we duplicate the voyage nodes for each candidate butterfly port so that the repeated port calls in a non-simple rotation can be differentiated. Let $z(i)$ indicate the physical port for node i. A voyage arc connects a voyage node i to another voyage node j if $z(i) \neq z(j)$. All voyage arcs are collected in set A. Let $A^{-}(i)$ and $A^{+}(i)$ denote a subset of voyage arcs, respectively, that start and end at node i. Each voyage arc $e \in A$ has a length ℓ_e in nautical miles. Each transshipment arc links a voyage node i and a transshipment node j if $z(i) = z(j)$. Let L denote the transshipment arc set. We notate $L^{-}(i)$ and $L^{+}(i)$ as the subset of transshipment arcs that start and end at node i. Note that if a transshipment arc connects a voyage node to a transshipment node, it represents an unloading operation for the cargo movements from the ship to the quayside of a port. On the contrary direction, a loading operation is represented. We divide L into a subset L' that collects the transshipment arcs indicating loading operations and the other subset L'' with the included transshipment arcs showing unloading operations. For each arc $e \in A \cup L$, let $head(e)$ represent its head node and $tail(e)$ represent its tail node. Let c_e^k denote the cost for moving a unit cargo of demand k through arc e. Suppose each rotation $r \in R$ contains a fixed sequence of voyage nodes in $N(r)$ which are linked by a set of voyage arcs in $A(r)$. Many ships of type $s(r)$ are operating on it. Let $L(r)$ denote the subset of transshipment arcs that cross with r. Figure 3.2 presents an example planning network for the two rotations in Figure 3.1. In the planning network, a non-simple rotation can be represented as a simple cycle by splitting the butterfly port calls into separated voyage nodes. In that way, the transshipment arcs connected to a butterfly port can be differentiated to facilitate the representation of an intra-transshipment operation.

We assume that each port call takes τ days for handling containers at the terminal. Since the ship of type s sails at the designed speed v_s, to maintain a weekly frequency,

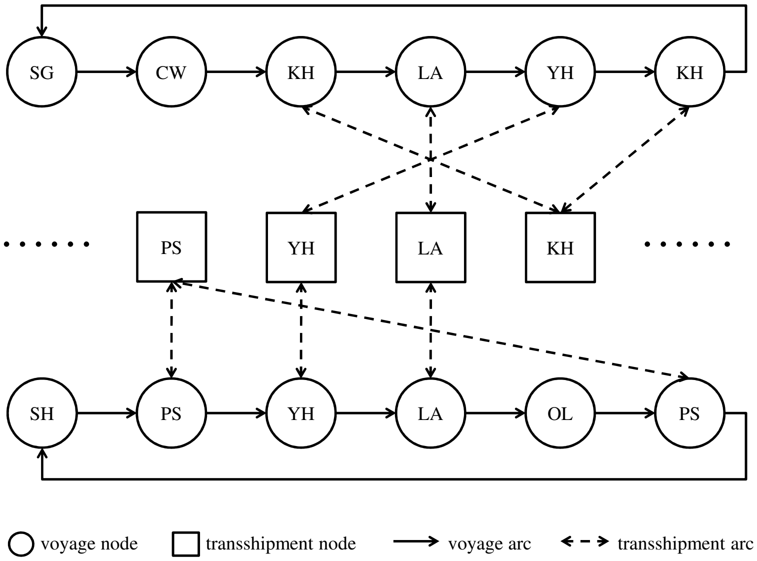


Figure 3.2: An example of the planning network with two rotations

the number of deployed ships on rotation r is calculated by

$$b_r \quad = \quad \left\lceil \frac{1}{7} \sum_{e \in A(r)} \left(\frac{\ell_e}{24 v_s} + \tau \right) \right\rceil . \tag{3.1}$$

For a ship of type s, fuel cost $w^F_{s,e}(v)$ is taken to sail through a voyage arc e using speed v. We calculate the fuel cost by $w^F_{s,e}(v) = (\ell_e/v) \cdot F_s(v) \cdot O$, where O denotes the unit ton fuel oil price and $F_s(v) = (v/v_s)^3 \cdot F_s$ indicates the fuel consumption rate of a ship of type s operating at speed v. A fixed port call cost $w^P_{s,z}$ is required to berth a ship of type s at port z and the port stay cost $w^S_{s,z}$ is charged per day. The weekly cost for running a ship of type s is denoted by w^O_s. The round-trip cost to operate r is thus computed by

$$a_r \;=\; w^O_{s(r)} b_r + \sum_{e \in A(r)} w^F_{s(r),e}(v_s) + \sum_{i \in N(r)} \left(w^P_{s(r),z(i)} + w^S_{s(r),z(i)} \right) \tag{3.2}$$

For each rotation, its weekly operating cost is calculated by the round-trip cost multiplied by the number of ships operating on it, and averaged by the number of weeks. For each rotation, to guarantee a weekly frequency, the number of ships deployed must be equal to the number of weeks used for a round-trip travel. Hence a_r is also seen as the weekly operating cost on r. For ease of representation, we set $w_{s,e} = w^F_{s,e}(v_s) + \frac{1}{2}(w^P_{s,z(head(e))} + w^S_{s,z(head(e))} + w^P_{s,z(tail(e))} + w^S_{s,z(tail(e))})$ to reformulate the cost calculation of Eq. (3.2) based on arcs. The weekly operating cost on r is then presented by

$$a_r \;=\; w^O_{s(r)} b_r + \sum_{e \in A(r)} w_{s(r),e}. \tag{3.3}$$

Three groups of decision variables are included in the model. The continuous variable $f^k_{r,e}$ decides the quantity of demand k to be moved on arc e of rotation r. The continuous variable x_k decides the amount of cargos that are satisfied for demand k. The binary variable y_r decides whether rotation r is operated or not. In abbreviation, let vectors $\mathbf{f} = [f^k_{r,e} : k \in D, r \in R, e \in A(r) \cup L(r)]$, $\mathbf{x} = [x_k : k \in D]$, and $\mathbf{y} = [y_r, r \in R]$ denote their decision sets subjecting to a feasible solution space Φ. The decision problem of LSND could be described as follows:

$$\max_{(\mathbf{f},\mathbf{x},\mathbf{y}) \in \Phi} \left\{ \sum_{k \in D} p'_k x_k - \sum_{k \in D} p''_k (q_k - x_k) - \sum_{k \in D} \sum_{r \in R} \sum_{e \in A(r) \cup L(r)} c^k_e f^k_{r,e} - \sum_{r \in R} a_r y_r \right\}. \tag{3.4}$$

The first two terms in (3.4) calculate the profits by subtracting the penalties of the unsatisfied demands from the revenues of the satisfied demand. The third term computes the cost of cargo movements on the network arcs including the transsshipment cost. The last term calculates the weekly cost for the operated rotations.

3.2.2 An MILP Model

Combine the first two terms of (3.4), the coefficient of variable x_k is replaced by $p_k = p'_k + p''_k$ and there is a constant term $\{-\sum_{k\in D} p''_k q_k\}$ in the objective value. Without affecting the optimality, we can remove the constant term from the objective formulation, but this term will still be included when we calculate the objective value in the computational experiments. The LSND is formulated as an MILP as follows:

$$
\begin{aligned}
\text{(LSND)} \quad \max \quad & \sum_{k\in D} p_k x_k - \sum_{k\in D}\sum_{r\in R}\sum_{e\in A(r)\cup L(r)} c_e^k f_{r,e}^k - \sum_{r\in R} a_r y_r, && (3.5a)\\
\text{s.t.} \quad & \sum_{e\in A^+(i):e\in A(r)} f_{r,e}^k + \sum_{e\in L^+(i)} f_{r,e}^k - \sum_{e\in A^-(i):e\in A(r)} f_{r,e}^k - \sum_{e\in L^-(i)} f_{r,e}^k = 0, \\
& \qquad \forall k\in D, \forall r\in R, \forall i\in N(r), && (3.5b)\\
& \sum_{e\in L^+(i)} \sum_{r\in R:tail(e)\in N(r)} f_{r,e}^k \\
& - \sum_{e\in L^-(i)} \sum_{r\in R:head(e)\in N(r)} f_{r,e}^k = \begin{cases} -x_k, & o(k) = z(i) \\ 0, & otherwise \\ x_k, & d(k) = z(i) \end{cases} \\
& \qquad \forall k\in D, \forall i\in T, && (3.5c)\\
& \sum_{r\in R:s(r)=s} b_r y_r \le h_s, \quad \forall s\in S, && (3.5d)\\
& \sum_{k\in D} f_{r,e}^k - m_{s(r)} y_r \le 0, \quad \forall r\in R, \forall e\in A(r), && (3.5e)\\
& f_{r,e}^k - q_k y_r \le 0, \quad \forall k\in D, \forall r\in R, \forall e\in A(r), && (3.5f)\\
& x_k \le q_k, \quad \forall k\in D, && (3.5g)\\
& x_k \ge 0, f_{r,e}^k \ge 0, \quad \forall k\in D, \forall r\in R, \forall e\in A(r)\cup L(r), && (3.5h)\\
& y_r \in \{0,1\}, \quad \forall r\in R. && (3.5i)
\end{aligned}
$$

The objective function (3.5a) is derived from (3.4). Constraints (3.5b) balance the cargo flows at the voyage nodes on all rotations. Constraints (3.5c) balance the cargo flows at all transshipment nodes. These two groups of flow balance constraints can guarantee that the flows on transshipment arcs represent the volume of transshipped cargos. Constraints (3.5d) restrict that the number of utilized ships cannot exceed the fleet size associated to each ship type. Constraints (3.5e) enforce the cargo flows on each rotation to be no greater than the maximum capacity of the deployed ship. Constraints (3.5f) enforce the cargo flows on each arc to be no greater than their maximum demand volumes. Constraints (3.5g) request the satisfied cargo demands to be no greater than their maximum volumes. Finally, the nonnegativity and integrality of the decision variables are imposed by (3.5h) and (3.5i).

As is observed from the MILP, both its variables and constraints grow with the number of rotations. It is hence infeasible to directly solve the model using an IP solver when a potentially large number of rotations are involved in the formulation. We next investigate its LP relaxations and briefly discuss some solution techniques for solving the relaxation problems.

3.2.3 LP Relaxations

Relaxing the integer condition, we obtain an LP relaxation of the MILP, which is denoted by LPR. Since R grows quickly with the problem size, solving LPR is computational difficult even for moderate sized instances. Recall that both variables and constraints in the formulation are dependent of the rotations, CG hence cannot work effectively. In this work, to overcome the computational difficulty, we intend to employ an SCRG approach, which is extended from CG, to solve the LPR.

Before presenting the SCRG approach, we adopt the aggregation technique mentioned in Chapter 2 to make further relaxations on LPR. Constraints (3.5b) are aggregated among the rotations corresponding to the same demand k, ship type s, and voyage node i:

$$\sum_{r\in R:s(r)=s}\left(\sum_{e\in A^{+}(i):e\in A(r)} f_{r,e}^{k}+\sum_{e\in L^{+}(i)} f_{r,e}^{k}-\sum_{e\in A^{-}(i):e\in A(r)} f_{r,e}^{k}-\sum_{e\in L^{-}(i)} f_{r,e}^{k}\right)=0, \quad \forall k\in D,\forall s\in S,\forall i\in N. \tag{3.6}$$

Constraints (3.5e) are aggregated among the rotations corresponding to the same ship type s and arc e:

$$\sum_{r\in R:s(r)=s}\left(\sum_{k\in D} f_{r,e}^{k}-m_s y_r\right)\leq 0, \quad \forall s\in S,\forall e\in A. \tag{3.7}$$

Constraints (3.5f) are aggregated among the rotations corresponding to the same demand k, ship type s, and arc e:

$$\sum_{r\in R:s(r)=s}\left(f_{r,e}^{k}-q_k y_r\right)\leq 0, \quad \forall k\in D,\forall s\in S,\forall e\in A. \tag{3.8}$$

By aggregating the route dependent constraints, we have another version of LP relaxation, ALPR, which are constrained by (3.5a), (3.5c), (3.5d), (3.5g), (3.5h), (3.5i), (3.6), (3.7), (3.8). Different from LPR, a limited number of constraints are involved in the ALPR, and they are all independent of the rotations. As a result, CG is available to work on the restricted master problem of ALPR with an initial subset of columns.

Since ALPR is a relaxation of LPR, its optimal objective provides an upper bound that can be looser than LPR. Moreover, from numerical experiments, we find that CG converges more slowly to reach the optimality of ALPR, because many rotations are generated without improving the solution objective of the restricted master problem (known as the degeneracy). Therefore, the investigation of an efficient solution approach for solving the LPR will be necessary.

3.3 The Solution Method via Simultaneous Column-and-Row Generation

In this section, the SCRG approach is formally presented to solve the LPR. We start off the section by providing a solution framework in Section 3.3.1. The solution framework describes the theoretical foundation behind the SCRG method, in which strong duality is applied to identify the optimality condition. Following the framework, it is necessary to find a dual solution of LPR to check the optimality at each iteration of the SCRG. Since the dual solution to LPR is partially derived from solving the restricted master problem, we propose a two-phase scheme, detailed in Section 3.3.2, that constructs the values for the unknown dual variables in the first phase, and then solve a pricing problem in the second phase to find violated dual constraints based on the obtained values of dual variables. For the first phase, two approaches have been proposed to construct the values of dual variables including a sequential approach and an LP-based approach. To solve the pricing problem for the second phase, we develop solution methods for solving the pricing problem in Section 3.3.3. Since the objective of a dual feasible solution provides an upper bound to LPR, we investigate a method in Section 3.3.4 that can compute an upper bound at any iteration of the SCRG. Moreover, in Section 3.3.5, we discuss the integration of the SCRG into a branch-and-price that seeks for optimal or near-optimal integer solutions for the LSND.

3.3.1 Framework

Given a subset of rotations in $\tilde{R}$, we define RMP as the restricted master problem of LPR on $\tilde{R}$. The dual problems for LPR and RMP are denoted by DLPR and DRMP, respectively. We define the solution of LPR by Θ and the solution of DLPR by Π. Let $\tilde{\Theta}$ and $\tilde{\Pi}$ represent the optimal solutions for RMP and DRMP, respectively.

Each solution corresponds to a objective value $Z(\cdot)$.

Since $\tilde{\Theta}$ and $\tilde{\Pi}$ are optimal to RMP and DRMP, we have $Z(\tilde{\Theta}) = Z(\tilde{\Pi})$ by strong duality. Based on $\tilde{\Theta}$, a primal feasible solution Θ could be constructed by letting the rest variables related to $R \setminus \tilde{R}$ take zeros, and there is $Z(\Theta) = Z(\tilde{\Theta})$. Given $\tilde{\Pi}$, if we could find a dual feasible solution Π which has an equal objective to $\tilde{\Pi}$, then the strong duality condition, $Z(\Pi) = Z(\Theta)$, will be obtained to guarantee Π is an optimal solution to LPR.

Associate the dual variables $\lambda^k_{r,i}$, μ^k_i, δ_s, $\pi_{r,e}$, $\gamma^k_{r,e}$, θ_k to Constraints (3.5b), (3.5c), (3.5d), (3.5e), (3.5f), (3.5g), respectively. The DLPR is obtained by (3.9a - 3.9i) as follows.

$$\text{(DLPR)} \quad \min \quad \sum_{s\in S} h_s\delta_s + \sum_{k\in D} q_k\theta_k, \tag{3.9a}$$

$$s.t. \quad \mu^k_{o(k)} - \mu^k_{d(k)} + \theta_k \geq p_k, \quad \forall k \in D, \tag{3.9b}$$

$$\lambda^k_{r,head(e)} - \lambda^k_{r,tail(e)} + \pi_{r,e} + \gamma^k_{r,e} \geq -c^k_e, \quad \forall k \in D, \forall r \in R, \forall e \in A(r), \tag{3.9c}$$

$$\lambda^k_{r,head(e)} - \mu^k_{tail(e)} \geq -c^k_e, \quad \forall k \in D, \forall r \in R, \forall e \in L' \cap L(r), \tag{3.9d}$$

$$-\lambda^k_{r,tail(e)} + \mu^k_{head(e)} \geq -c^k_e, \quad \forall k \in D, \forall r \in R, \forall e \in L'' \cap L(r), \tag{3.9e}$$

$$b_r\delta_{s(r)} - \sum_{e\in A(r)} m_{s(r)}\pi_{r,e} - \sum_{k\in D}\sum_{e\in A(r)} q_k\gamma^k_{r,e} \geq -a_r, \quad \forall r \in R, \tag{3.9f}$$

$$\lambda^k_{r,i} \text{ unrestricted}, \quad \forall k \in D, \forall r \in R, \forall i \in N(r), \tag{3.9g}$$

$$\mu^k_i \text{ unrestricted}, \quad \forall k \in D, \forall i \in T, \tag{3.9h}$$

$$\delta_s \geq 0, \pi_{r,e} \geq 0, \theta_k \geq 0, \gamma^k_{r,e} \geq 0, \quad s \in S, \forall k \in D, \forall r \in R, \forall e \in A(r). \tag{3.9i}$$

Let Π_0 represent a subset of dual variables $[\lambda^k_{r,i} : k \in D, r \in R \setminus \tilde{R}, i \in N(r)]$, $[\pi_{r,e} : r \in R \setminus \tilde{R}, e \in A(r)]$ and $[\gamma^k_{r,e} : k \in D, r \in R \setminus \tilde{R}, e \in A(r)]$. Given $\tilde{\Pi}$ as the optimal solution to DRMP, we can develop for DLPR a complete solution Π by combining $\tilde{\Pi}$ with Π_0. After the combination we have $Z(\Pi) = Z(\tilde{\Pi})$ due to dual objective (3.9a) does not depend on Π_0. Remember that if Π is a feasible solution to DLPR, the strong duality condition implies Π is also optimal to DLPR and optimality of LPR is guaranteed. Knowing that $\tilde{\Pi}$ is feasible to DRMP, partial constraints of DLPR, (3.9b) and (3.9c - 3.9f) for $r \in \tilde{R}$, must be satisfied by Π. Therefore, Π is a feasible solution to DLPR if a Π_0 can be found to satisfy the dual constraints (3.9c - 3.9f) for $r \in R \setminus \tilde{R}$.

Given $\tilde{R}$, the SCRG work on a RMP with a limited number of variables and constraints. Due to dual constraints (3.9c - 3.9f) for $r \in R \setminus \tilde{R}$ are excluded in the DRMP, we construct Π_0 and check whether it satisfies the dual constraints (3.9c - 3.9f) for $r \in R \setminus \tilde{R}$. If violations are found, the violated dual constraints will be added to the DRMP, namely, some new primal variables of $[f^k_{r,e} : k \in D, r \in R \setminus \tilde{R}, e \in A(r) \cup L(r)]$ and $[y_r : r \in R \setminus \tilde{R}]$ having positive reduced costs are added to the RMP. Recall that variables and constraints are interdependent, the generation of these new primal variables requires the generation of their structural constraints. Therefore, we update the RMP by simultaneously generating the columns and rows at each iteration of the SCRG. The SCRG terminates at the optimality of LPR when the constructed Π_0 satisfies the dual constraints (3.9c - 3.9f) for $r \in R \setminus \tilde{R}$.

3.3.2 Construction of Π_0

As aforementioned, to check the optimality of LPR, we need to construct Π_0 to form Π as a (not necessarily feasible) solution to DLPR. However, if $R \setminus \tilde{R}$ still has lots of rotations, there will also have many dual variables in Π_0 that are constrained by many constraints of DLPR. It is hence not possible to generate all these variables and constraints to check if Π_0 is feasible.

To construct and check Π_0 in a more efficient way, we define $\hat{\Pi}_0$ to have a set of auxiliary values $[\hat{\lambda}^k_{s,i} : k \in D, s \in S, i \in N]$, $[\hat{\pi}_{s,e} : s \in S, e \in A]$ and $[\hat{\gamma}^k_{s,e} :$

$k \in D, s \in S, e \in A]$. Based on the auxiliary values, we construct Π_0 by setting $\lambda^k_{r,i} = \hat{\lambda}^k_{s(r),i}$, $\pi_{r,e} = \hat{\pi}_{s(r),e}$, $\gamma^k_{s(r),e} = \hat{\gamma}^k_{s,e}$, for each rotation $r \in R \setminus \tilde{R}$. Thus, we only need to determine a limited number of auxiliary values in $\hat{\Pi}_0$, which facilitates the construction of Π_0. We next develop a two-phase scheme to determine the auxiliary values in $\hat{\Pi}_0$ and check if the constructed Π_0 is feasible.

Phase 1: Construct $\hat{\Pi}_0$ subjecting to the following constraints:

$$\hat{\lambda}^k_{s,head(e)} - \hat{\lambda}^k_{s,tail(e)} + \hat{\pi}_{s,e} + \hat{\gamma}^k_{s,e} \geq -c^k_e, \quad \forall k \in D, \forall s \in S, \forall e \in A, \tag{3.10a}$$

$$\hat{\lambda}^k_{s,head(e)} - \mu^k_{tail(e)} \geq -c^k_e, \quad \forall k \in D, \forall s \in S, \forall e \in L', \tag{3.10b}$$

$$-\hat{\lambda}^k_{s,tail(e)} + \mu^k_{head(e)} \geq -c^k_e, \quad \forall k \in D, \forall s \in S, \forall e \in L'', \tag{3.10c}$$

$$\hat{\lambda}^k_{s,i} \text{ unrestricted}, \quad \forall k \in D, \forall s \in S, \forall i \in N, \tag{3.10d}$$

$$\hat{\pi}_{s,e} \geq 0, \hat{\gamma}^k_{s,e} \geq 0, \quad \forall k \in D, \forall s \in S, \forall e \in A. \tag{3.10e}$$

Phase 2: Given $\hat{\Pi}_0$, next find a rotation in $R \setminus \tilde{R}$ that corresponds to the largest violation among Constraints (3.9f). The largest violation value, denoted by Δ, is obtained by solving the following pricing problem:

$$\Delta = \max_{r \in R \setminus \tilde{R}} \left\{ -a_r - b_r \delta_{s(r)} + \sum_{e \in A(r)} m_{s(r)} \hat{\pi}_{s(r),e} + \sum_{k \in D} \sum_{e \in A(r)} q_k \hat{\gamma}^k_{s(r),e} \right\}. \tag{3.11}$$

In Phase 1, the assigned auxiliary values in $\hat{\Pi}_0$ can produce such a Π_0 that is feasible to Constraints (3.9c - 3.9e) for $r \in R \setminus \tilde{R}$. The violation check is left as a pricing problem in Phase 2 which aims to find a rotation in $R \setminus \tilde{R}$ being dedicated to the largest violation among (3.9f). If $\Delta \leq 0$ in Phase 2, it implies the constructed Π_0 is feasible to dual constraints (3.9c - 3.9f) for $r \in R \setminus \tilde{R}$, and the optimality of LPR is achieved. Otherwise, the most violated rotation will be identified and added to the RMP. In this subsection, we focus on Phase 1, in which a sequential approach and an LP-based approach are developed to determine the auxiliary values in $\hat{\Pi}_0$. As $\hat{\Pi}_0$

is determined, in Section 3.3.3, we solve the pricing problem in Phase 2 to introduce new rotations.

To determine $\hat{\Pi}_0$, we present a sequential approach that first decides the values for $\hat{\lambda}^k_{s,i}$, and next decides the values for $\hat{\pi}_{s,e}$ and $\hat{\gamma}^k_{s,e}$. Given $[i]$ to indicate the corresponding transshipment node for voyage node i, we assign the values to $\hat{\lambda}^k_{s,i}$ as follows:

$$\hat{\lambda}^k_{s,i} = \mu^k_{[i]}, \quad \forall k \in D, \forall s \in S, \forall i \in N. \tag{3.12}$$

The values assigned to $\hat{\lambda}^k_{s,i}$ ensure Constraints (3.10b) and (3.10c) are always satisfied since $c^k_e \geq 0$. Based on (3.12), we next assign nonnegative values to $\hat{\pi}_{s,e}$ and $\hat{\gamma}^k_{s,e}$ so that Constraints (3.10a) are satisfied as well.

$$\hat{\pi}_{s,e} = 0, \quad \forall s \in S, \forall e \in A, \tag{3.13}$$

$$\hat{\gamma}^k_{s,e} = \max\{0, -\mu^k_{[head(e)]} + \mu^k_{[tail(e)]} - c^k_e\}, \quad \forall k \in D, \forall s \in S, \forall e \in A. \tag{3.14}$$

The auxiliary values in $\hat{\Pi}_0$ are thus determined by (3.12 - 3.14) in a sequential manner. Note that, in addition to (3.13) and (3.14), $\hat{\pi}_{s,e}$ and $\hat{\gamma}^k_{s,e}$ can also take other values which are sufficiently large to satisfy (3.10a). Thus, we obtain Proposition 3.1, which show that many feasible $\hat{\Pi}_0$ may exist.

Proposition 3.1. *There exist many $\hat{\Pi}_0$ that are feasible to (3.10a - 3.10e).*

It is worth noting that the auxiliary values assigned in Phase 1 have an influence on the violation check in Phase 2, and it is important to find a "good" $\hat{\Pi}_0$. Intuitively, smaller $\hat{\pi}_{s,e}$ and $\hat{\gamma}^k_{s,e}$ are preferred as they lead to smaller violation values in Phase 2 and the optimality condition $\Delta \leq 0$ is closer to reach. Suppose Phase 1 returns two different groups of auxiliary values $\hat{\Pi}^{(1)}_0$ and $\hat{\Pi}^{(2)}_0$ that are both feasible to (3.10a - 3.10e). Based on each group, the pricing problem (3.11) returns $\Delta^{(1)}$ and $\Delta^{(2)}$

respectively. If condition $m_s\hat{\pi}^{(1)}_{s,e} + \sum_{k\in D} q_k\hat{\gamma}^{k,(1)}_{s,e} \leq m_s\hat{\pi}^{(2)}_{s,e} + \sum_{k\in D} q_k\hat{\gamma}^{k,(2)}_{s,e}$ holds for all $s \in S$ and $e \in A$, we have the following relationship:

$$\begin{aligned}
\Delta^{(1)} &= \max_{r\in R\backslash\tilde{R}} \left\{ -a_r - b_r\delta_{s(r)} + \sum_{e\in A(r)} \left(m_{s(r)}\hat{\pi}^{(1)}_{s(r),e} + \sum_{k\in D} q_k\hat{\gamma}^{k,(1)}_{s(r),e} \right) \right\} \\
&\leq \max_{r\in R\backslash\tilde{R}} \left\{ -a_r - b_r\delta_{s(r)} + \sum_{e\in A(r)} \left(m_{s(r)}\hat{\pi}^{(2)}_{s(r),e} + \sum_{k\in D} q_k\hat{\gamma}^{k,(2)}_{s(r),e} \right) \right\} \\
&\leq \Delta^{(2)}.
\end{aligned}$$

We say $\hat{\Pi}^{(1)}_0$ is "better" than $\hat{\Pi}^{(2)}_0$ since a smaller violation value is returned from Phase 2. Based on this, we have Proposition 3.2 to show the dominance relationship of two groups of auxiliary values, which further promotes the development of an LP-based approach to determine $\hat{\Pi}_0$.

Proposition 3.2. *Given two groups of auxiliary values $\hat{\Pi}^{(1)}_0$ and $\hat{\Pi}^{(2)}_0$ from Phase 1, $\hat{\Pi}^{(1)}_0$ dominates $\hat{\Pi}^{(2)}_0$ if $m_s\hat{\pi}^{(1)}_{s,e} + \sum_{k\in D} q_k\hat{\gamma}^{k,(1)}_{s,e} \leq m_s\hat{\pi}^{(2)}_{s,e} + \sum_{k\in D} q_k\hat{\gamma}^{k,(2)}_{s,e}$ holds for each $s \in S$ and $e \in A$.*

Based on Proposition 3.2, $\hat{\pi}_{s,e}$ and $\hat{\gamma}^k_{s,e}$ are suggested to take appropriate values to minimize the value of $m_s\hat{\pi}_{s,e} + \sum_{k\in D} q_k\hat{\gamma}^k_{s,e}$. To incorporate this property to construct $\hat{\Pi}_0$ purposely, we develop an LP that minimizes the total values of $m_s\hat{\pi}_{s,e} + \sum_{k\in D} q_k\hat{\gamma}^k_{s,e}$ for all $s \in S$ and $e \in A$. The LP is given as follows:

$$\begin{aligned}
\min \quad & \sum_{s\in S}\sum_{e\in A} \left\{ m_s\hat{\pi}_{s,e} + \sum_{k\in D} q_k\hat{\gamma}^k_{s,e} \right\}, \\
s.t. \quad & (3.10a - 3.10e).
\end{aligned}$$

The LP-based approach is expected to provide better values for $\hat{\Pi}_0$ since it integrates the perception to set smaller values for $\hat{\pi}_{s,e}$ and $\hat{\gamma}^k_{s,e}$ so as to relief the violation value overestimation. In the SCRG, a terrible overestimation may lead to the generation of many rotations with invalid violations. As a consequence, many rotations added to the RMP will not improve the solution objective, and it is likely to generate too many rotations when the SCRG stops.

3.3.3 Solving the Pricing Problem

After $\hat{\Pi}_0$ is determined to construct Π_0 in Phase 1 that satisfies the constraints (3.9c - 3.9e) for $r \in R \setminus \tilde{R}$, we now move to Phase 2 to check whether the constructed Π_0 is feasible to (3.9f) for $r \in R \setminus \tilde{R}$. As mentioned in Section 3.3.2, the pricing problem in Phase 2 is to find a rotation in $R \setminus \tilde{R}$ that has the largest violation value. Given ship type s, the pricing problem is equivalent to finding a minimum cost cycle on a weighted physical network $G_s = (V_s, E_s)$, where V_s consists of a set of available ports and E_s includes a set of directed edges that connect the ports. We next define some practical rules to enforce the generated cycles are consistent with the patterns observed from the practice. (i) Each port is visited no more than twice and each edge is travelled no more than once in a cycle. (ii) At most one butterfly port is involved in a cycle. (iii) At most α port calls are included in a cycle. (iv) The round-trip time does not exceed β weeks. In our study, we choose α from $[5, 11]$ and β from $[10, 15]$. Substituting (3.1) and (3.3) into (3.11), for each s, the pricing problem defined on G_s is given as follows:

$$\begin{aligned} \Delta_s \quad = \quad & \max_{r \in R \setminus \tilde{R}: s(r) = s} -(w_s^O + \delta_s) \left\lceil \frac{1}{7} \sum_{e \in A(r)} (t_e + \tau) \right\rceil \\ & - \sum_{e \in A(r)} \left(w_{s,e} - m_s \hat{\pi}_{s,e} - \sum_{k \in D} q_k \hat{\gamma}_{s,e}^k \right). \end{aligned} \tag{3.15}$$

Among the constraints (3.9f) for $r \in R \setminus \tilde{R}$, the maximum violation value is obtained as $\Delta = \max_{s \in S} \{\Delta_s\}$, and the constructed Π_0 is feasible if $\Delta \leq 0$ is obtained.

Notice that, based on $\hat{\Pi}_0$, the pricing problem may output an existing rotation in $\tilde{R}$ which has a violation on (3.9f). To prevent the pricing problem from generating the existing rotations, we define a forbidden rotation set $R^* = \tilde{R}$ and any rotation in R^* is not allowed to output.

To solve this pricing problem, we develop a cycle search algorithm which is convenient to output multiple promising cycles at a time and utilize accelerating strategies to find heuristic solutions quickly. Also, we investigate an MILP-based method which is more advantageous to get the optimality of the pricing problem or its upper bound while returning only single rotation at a time. At each iteration of the SCRG, the cycle search algorithm and the MILP-based method will be sequentially applied to generate new rotations.

3.3.3.1 A cycle search algorithm

For each ship type s, we apply a cycle search algorithm that generates cycles on G_s by adding new vertices to paths that have been obtained. Let $l(e)$ denote the corresponding physical edge for voyage arc e. According to (3.15), we assign edge l with cost $w(l(e)) = w_{s,e} - m_s\hat{\pi}_{s,e} - \sum_{k\in D} q_k\hat{\gamma}^k_{s,e}$. Since butterfly port calls are represented by separated voyage nodes, there exist multiple voyage arcs that correspond to the same physical edge on G_s. In that case, we assign the edge cost by $w(l) = \min_{e\in A:l(e)=l}\left\{w_{s,e} - m_s\hat{\pi}_{s,e} - \sum_{k\in D} q_k\hat{\gamma}^k_{s,e}\right\}$, which guarantees all potential rotations are with nonnegative costs when the pricing problem outputs a nonnegative cost cycle as an optimal solution. As long as a cycle is obtained that corresponds to rotation r, we additionally assign a cost $w'(r) = (w^O_s + \delta_s)\left\lceil \frac{1}{7}\sum_{e\in A(r)}(t_e + \tau)\right\rceil$ to this cycle. Given the weighted physical network, solving the pricing problem (3.15) is equivalent to finding a minimum cost cycle on G_s. Let W_s denote the minimum cycle cost on G_s. The optimal solution of the pricing problem is known as $\Delta_s = -W_s$ and the SCRG stops if $\max_{s\in S}\{\Delta_s\} \le 0$.

Algorithm 2 (Cycle Search Algorithm)

```
Initialize $p^*(s)$ that corresponds to the incumbent optimal cycle on $G_s$, and set $p^*(s) = \emptyset$;
for each vertex $z$ in $V_s$ do
    Initialize $p = \{z\}$ and $Path(z) = \{p\}$;
end for
while $\cup_{z \in V_s} Path(z) \neq \emptyset$ do
    for a path $p'$ in $\cup_{z \in V_s} Path(z)$ do
        for a vertex $z$ that is connected by $Head(p')$ do
            Construct a path $p$ by adding $z$ to the tail of $p'$;
            if $p$ can generate feasible cycles then
                Set $Path(z) = Path(z) \cup \{p\}$.
            end if
            if $p$ forms a cycle that corresponds to a rotation $r$ not in $R^*$, and the cycle cost is less than that of $p^*(s)$ then
                Set $p^*(s) = p$;
            end if
        end for
        Removing $p'$ from $Path(Head(p'))$.
    end for
    Update $\cup_{z \in V_s} Path(z)$ if applicable.
end while
```

Let $Head(p)$ and $Tail(p)$ represent the last vertex and first vertex on path p. Let $Path(z)$ indicate a collection of obtained paths with the end vertex at z. In each iteration of the algorithm, we choose an existing path in $Path(z')$ for some z', and extend it by adding one more vertex to the tail. Let $E(p)$ denote a subset of edges passed by path p. When the new path forms a cycle with regards to a rotation r, and $w'(r) + \sum_{l \in E(p)} w(l) < 0$, we regard it as a candidate cycle to output. If the path forms a cycle in the forbidden set R^*, we abandon to generate this cycle and continue to search another. The cycle search algorithm is summarized in Algorithm 1.

In the cycle search algorithm, there are too many possible paths to enumerate even for a moderate sized G_s. It is thus unaffordable to search among all feasible paths on G_s by enumeration. Therefore, it is important to reduce the search space by defining some path elimination rules to restrict that only qualified paths will be kept for exploration.

In the algorithm, we first define a pruning rule to speed up the solution algorithm. As is shown in [36], a negative cost cycle always has a segment of its partial path with negative cost. It is therefore only necessary to explore the paths having negative costs so as to detect the potential negative cost cycles. During the search, since the part of cycle cost $w'(\cdot)$ is nonnegative, we can directly prune a path p if it has $\sum_{l\in E(p)} w(l) \geq 0$. To enhance the pruning, we rewrite (3.15) as follows:

$$\Delta_s = \max_{r\in R_a\setminus\tilde{R}:s(r)=s} \left\{ \begin{array}{l} -(w_s^O+\delta_s)\left(\left\lceil \frac{1}{7}\sum\limits_{e\in A(r)}(t_e+\tau)\right\rceil - \frac{1}{7}\sum\limits_{e\in A(r)}(t_e+\tau)\right) \\ -\sum\limits_{e\in A(r)}\left(\frac{1}{7}(w_s^O+\delta_s)(t_e+\tau)+w_{s,e}-m_s\hat{\pi}_{s,e}-\sum\limits_{k\in D}q_k\hat{\gamma}_{s,e}^k\right) \end{array} \right\}. \tag{3.16}$$

Based on (3.16), $w(l(e)) = \frac{1}{7}(w_s^O+\delta_s)(t_e+\tau)+w_{s,e}-m_s\hat{\pi}_{s,e}-\sum_{k\in D}q_k\hat{\gamma}_{s,e}^k$ is assigned as a larger cost on each arc of the network, and the pruning condition will be easier to reach during the search.

In addition to the pruning rule, we also implement accelerating strategies to speed up the solution, which can quickly output heuristic solutions for the pricing problem. For path p, let $V(p)$ denote the subset of ports visited by p and let $Time(p)$ denote the time length on p. Given two paths p_1 and p_2, we say p_1 dominates p_2 if the following conditions are simultaneously fulfilled: $Head(p_1) = Head(p_2)$, $Tail(p_1) = Tail(p_2)$, $V(p_1) \subseteq V(p_2)$, $Time(p_1) \leq Time(p_2)$, and $\sum_{l\in E(p_1)} w(l) \leq \sum_{l\in E(p_2)} w(l)$. This is a heuristic dominance rule for the path elimination, because it is likely to give up some promising paths that are dominated by the partial paths of those forbidden cycles. To accelerate the algorithm, we also define a threshold ϕ to capacitate the number of available paths to be explored at each path set $Path(z)$. We sort the paths in $Path(z)$ according to their cost in ascending order, and select the top ϕ paths for exploration.

3.3.3.2 An MILP-based method

Based on G_s, we next investigate an MILP-based method to solve the pricing problem (3.15). In the model, binary variable K_z equals one if port z is visited and equals zero otherwise. Binary variable B_z equals one if port z is a butterfly node in a rotation and equals zero otherwise. Binary variable M_z equals one if port z is the starting port of the rotation. Continuous variable I_z reflects the sequence of port z in the rotation by its value. Binary variable X_l decides whether edge l is involved in the cycle. Integer variable Y decides the number of operated ships. Denote by $o'(l)$ and $d'(l)$ the rear port and head port for edge l. Let $E_s^+(z)$ and $E_s^-(z)$, respectively, represent the subset of incoming and outgoing edges for port z. The voyage time on edge l is denoted by t_l'. An MILP formulation (3.17a - 3.17q) is provided to solve the pricing problem (3.15).

The objective function (3.17a) is derived from (3.15). Constraints (3.17b) balance the number of edges entering and leaving any port in the cycle. Constraints (3.17c) allow an edge is included by a cycle if its rear port and head port are both visited. Constraint (3.17d) follows the practical rule (ii) that allows at most one butterfly port in a cycle. The number of edges that enter or leave a port is restricted by Constraints (3.17e) and (3.17f), respectively. Constraint (3.17g) allow exactly one port identified as the starting port of the cycle. Constraints (3.17h) enforce the butterfly port to be the starting port in the cycle. Constraints (3.17i) restrict I_z to be greater than zero if and only if port z is visited in the cycle. Constraints (3.17j) assign I_z by increasing values along the cycle before reaching the starting port. Note that Constraints (3.17g-3.17j) are employed to eliminate subtours. Constraints (3.17k) require an enough number of ships to be operated on the rotation to maintain the service frequency. Constraint (3.17l) follows practical rule (iii) that allows at most α port calls in a cycle. Constraint (3.17m) follows practical rule (iv) that restricts the round-trip duration no more than β weeks. The forbidden rotations in R^* are avoided to output by Constraints (3.17n). Integrality and nonnegativity of the variables are presented in (3.17o-3.17q).

$$
\begin{aligned}
\Delta_s = \quad \max \quad & (-w_s^O - \delta_s)Y - \sum_{l \in E_s} w_l X_l, && \text{(3.17a)}\\
s.t. \quad & \sum_{l \in E_s^+(z)} X_l - \sum_{l \in E_s^-(z)} X_l = 0, \quad \forall z \in V_s, && \text{(3.17b)}\\
& X_l \leq \frac{1}{2}(K_{o'(l)} + K_{d'(l)}), \quad \forall l \in E_s, && \text{(3.17c)}\\
& \sum_{z \in V_s} B_z \leq 1, && \text{(3.17d)}\\
& K_z \leq \sum_{l \in E_s^+(z)} X_l \leq K_z + B_z, \quad \forall z \in V_s, && \text{(3.17e)}\\
& K_z \leq \sum_{l \in E_s^-(z)} X_l \leq K_z + B_z, \quad \forall z \in V_s, && \text{(3.17f)}\\
& \sum_{z \in V_s} M_z = 1, && \text{(3.17g)}\\
& B_z \leq M_z \leq K_z, \quad \forall z \in V_s, && \text{(3.17h)}\\
& K_z \leq I_z \leq |P| \, K_z, \quad \forall z \in V_s, && \text{(3.17i)}\\
& 1 + I_{o'(l)} - |P| \, M_{d'(l)} - |P| \, (1 - X_l) \leq I_{d'(l)}, \quad \forall l \in E_s, && \text{(3.17j)}\\
& \sum_{l \in E_s} (t'_l + \tau) X_l \leq 7Y, && \text{(3.17k)}\\
& \sum_{l \in E_s} X_l \leq \alpha, && \text{(3.17l)}\\
& \sum_{l \in E_s} (t'_l + \tau) X_l \leq 7\beta, && \text{(3.17m)}\\
& \sum_{l \in E(r)} (1 - X_l) + \sum_{l \notin E(r)} X_l \geq 1, \quad \forall r \in R^* : s(r) = s, && \text{(3.17n)}\\
& K_z, B_z, M_z \in \{0,1\}, I_z \geq 0, \quad z \in V_s, && \text{(3.17o)}\\
& X_l \in \{0,1\}, \quad l \in E_s, && \text{(3.17p)}\\
& Y \in \mathbf{N}^+. && \text{(3.17q)}
\end{aligned}
$$

As Δ_s is obtained by solving the MILP for each s, $\Delta = \max_{s \in S}\{\Delta_s\}$ is returned and SCRG stops if $\Delta \leq 0$. Note that the MILP-based method can automatically avoid generating the forbidden rotations due to the restriction of Constraints (3.17n). Our computational experience of employing an IP solver to tackle this MILP is that the instances with no more than 20 ports could be solved quickly in most cases.

3.3.3.3 Violation verification

In this subsection, we propose a violation verification process to prevent the pricing problem from outputting the cycles with invalid violations. It is worth noting that not all rotations output from the pricing problem lead to true violations. Assume rotation r is returned from solving the pricing problem, we implement an auxiliary LP described in (3.18) to decide the values for $\lambda^k_{r,i}$, $\pi_{r,e}$ and $\gamma^k_{r,e}$ in Π_0.

$$\begin{aligned} \Delta(r) = \quad \min \quad & -a_r - b_r\delta_{s(r)} + \sum_{e\in A(r)} m_{s(r)}\pi_{r,e} + \sum_{k\in D}\sum_{e\in A(r)} q_k\gamma^k_{r,e}, \\ \text{s.t.} \quad & \lambda^k_{r,head(e)} - \lambda^k_{r,tail(e)} + \pi_{r,e} + \gamma^k_{r,e} \geq -c^k_e, \quad \forall k \in D, \forall e \in A(r), \\ & \lambda^k_{r,head(e)} - \mu^k_{tail(e)} \geq -c^k_e, \quad \forall k \in D, \forall e \in L' \cap L(r), \\ & -\lambda^k_{r,tail(e)} + \mu^k_{head(e)} \geq -c^k_e, \quad \forall k \in D, \forall e \in L'' \cap L(r), \\ & \lambda^k_{r,i} \text{ unrestricted}, \quad \forall k \in D, \forall i \in N(r), \\ & \pi_{r,e} \geq 0, \quad \gamma^k_{r,e} \geq 0, \forall k \in D, \forall e \in A(r). \end{aligned} \tag{3.18}$$

Suppose $(\hat{\lambda}^k_{s,i}, \hat{\pi}_{s,e}, \hat{\gamma}^k_{s,e})$ is obtained in Phase 1. Given r, we define $\hat{\Delta}(r)$ as the objective value of (3.18) by letting $\pi_{r,e} = \hat{\pi}_{s(r),e}$ and $\gamma^k_{r,e} = \hat{\gamma}^k_{s(r),e}$. Obviously, $(\hat{\lambda}^k_{s,i}, \hat{\pi}_{s,e}, \hat{\gamma}^k_{s,e})$ can form a feasible solution to (3.18) because the the auxiliary values constrained by (3.10a - 3.10e) also satisfy the constraints of (3.18). Suppose r' is returned from Phase 2 that has a violation. Given r', by solving (3.18) optimally, if we have $\Delta(r') \leq 0 < \hat{\Delta}(r')$, it implies that the detected violation on r' can be avoided when the dual variables $\lambda^k_{r',i}$, $\pi_{r',e}$, $\gamma^k_{r',e}$ in Π_0 are assigned with the optimal solution of (3.18), so the detected violation is invalid. It is hence not necessary to generate r'

as it will surely not improve the objective of RMP.

To avoid generating the rotations with invalid violations, we implement (3.18) as a verification process in the pricing problem to see whether the output cycle leads to a valid violation. In the cycle search algorithm, anytime as a cycle is verified with $\Delta(r) \leq 0$, it can be abandoned immediately. For the rest candidate cycles with $\Delta(r) > 0$, we sort them on $\Delta(r)$ in descending order and select the top cycles as the output solutions for the pricing problem. In the MILP-based method, if $\Delta(r) \leq 0$, we not only reject to output this cycle but also add the rejected cycle to the forbidden set R^*. We then repeat solving the MILP (3.17a - 3.17q) with an updated R^* to get another one. The repeated solution procedure stops till a cycle r is obtained with $\hat{\Delta}(r) \leq 0$ or $\Delta(r) > 0$.

3.3.4 Computing an Upper Bound

At any iteration of the SCRG, according to the weak duality, the objective of a feasible solution to DLPR can be an upper bound to the LPR. We construct Π_0 and combine it with $\tilde{\Pi}$ to form a complete dual solution Π. Note that Π may be infeasible to DLPR since the constructed Π_0 can have violations on (3.9f). In order to avoid such violations, for each s we update δ_s to δ'_s which is large enough to guarantee (3.9f) are all satisfied. After the updating, Π will be feasible to DLPR and $Z(\Pi)$ provides an upper bound to LPR which is also known as an upper bound to LSND.

Since $Z(\Pi)$ is increasing in δ_s, to obtain a tighter upper bound, the value of δ'_s should be set as small as possible while keeping that (3.9f) are satisfied by Π_0. To find such a small δ'_s, we use $\Delta_s(\omega)$ to denote the optimal solution value of the pricing problem defined in (3.17a - 3.17q) with $Y = \omega$. Assume Ω to include all possible numbers of the deployed ships on a rotation. Given $s \in S$ and $\omega \in \Omega$, we have the following MILP to obtain $\Delta_s(\omega)$:

$$
\begin{aligned}
\Delta_s(\omega) = \quad \max \quad & (-w_s^O - \delta_s)Y - \sum_{l \in E_s} w_l X_l, \\
s.t. \quad & (3.17b - 3.17p), \qquad (3.19) \\
& Y = \omega.
\end{aligned}
$$

As the value of $\Delta_s(\omega)$ is obtained for all possible s and ω, the following proposition is given to compute a compact upper bound to LPR.

Proposition 3.3. *Suppose $\{\delta_s : s \in S\}$ and $\{\theta_k : k \in D\}$ are obtained from solving the RMP. Replace δ_s by δ'_s for each s following $\delta'_s = \delta_s + \max_{\omega \in \Omega} \left\{ \frac{\Delta_s(\omega)}{\omega} \right\}^+$. Then $\sum_{s \in S} h_s \delta'_s + \sum_{k \in D} q_k \theta_k$ yields an upper bound to LPR.*

Proof. Suppose we have obtained the auxiliary values in $\hat{\Pi}_0$ to construct Π_0. To guarantee Π_0 is feasible, we adjust value of δ_s to a larger value δ'_s that ensures the partial Constraints (3.9f) for $r \in R \setminus \tilde{R}$ are all satisfied. Hence, for each r not in $\tilde{R}$, we have

$$
\begin{aligned}
b_r \delta'_{s(r)} \quad & \geq \quad -a_r + \sum_{e \in A(r)} m_{s(r)} \hat{\pi}_{s(r),e} + \sum_{k \in D} \sum_{e \in A(r)} q_k \hat{\gamma}^k_{s(r),e} \\
& \geq \quad -\left(\delta_{s(r)} + w^O_{s(r)}\right) b_r - \sum_{e \in A(r)} \left(w_{s(r),e} - m_{s(r)} \hat{\pi}_{s(r),e} - \sum_{k \in D} q_k \hat{\gamma}^k_{s(r),e} \right) + b_r \delta_{s(r)}.
\end{aligned}
\tag{3.20}
$$

Given $\Delta_s(\omega)$ as the optimal solution value of (3.17a - 3.17q) with $Y = \omega$, for any $r \in R \setminus \tilde{R}$ with $b_r = \omega$, there is

$$
\Delta_s(\omega) \geq -\left(\delta_{s(r)} + w^O_{s(r)}\right) \omega - \sum_{e \in A(r)} \left(w_{s(r),e} - m_{s(r)} \hat{\pi}_{s(r),e} - \sum_{k \in D} q_k \hat{\gamma}^k_{s(r),e} \right). \tag{3.21}
$$

In order to ensure (3.20) is fulfilled for all $r \in R \setminus \tilde{R}$, based on (3.21) we have the following sufficient condition to set the values for δ'_s:

$$\omega\delta'_s \geq \Delta_s(\omega) + \omega\delta_s, \quad \forall s \in S, \forall\omega \in \Omega. \tag{3.22}$$

Based on (3.22), the smallest δ'_s will be determined by $\delta_s + \max_{\omega\in\Omega}\left\{\frac{\Delta_s(\omega)}{\omega}\right\}$ for each s. Note that if $\max_{\omega\in\Omega}\left\{\frac{\Delta_s(\omega)}{\omega}\right\} \leq 0$, it implies $\Delta_s \leq 0$ and Constraints (3.9f) with respect to the rotations using ship type s are already satisfied, and we set $\delta'_s = \delta_s$ in that case. After the updating, the dual solution Π is feasible to DLPR and $Z(\Pi) = \sum_{s\in S} h_s\delta'_s + \sum_{k\in D} q_k\theta_k$ is an upper bound to LPR by weak duality. □

Note that the efficiency of the upper bound computation largely replies on the time spent on solving (3.19). Fortunately, even if the MILP does not reach the optimality, its upper bound $\Delta'_s(\omega)$ can also be used to determine the value of δ'_s, and the obtained Π still provides a valid upper bound to LPR. In the SCRG, we restrict the time for computing the upper bounds by imposing a solution time limit. Moreover, to avoid occupying too much computation time, we only compute the upper bound at some selected iterations.

3.3.5 A Branch-and-Price Algorithm

We now embed the SCRG into a branch-and-price framework [65, 9], to find an optimal integer solution for this problem. On the explored nodes of the search tree, we apply SCRG to solve their relaxation problems which provide upper bounds for the problem. At a tree node, branching occurs when no new rotations are found to improve the restricted master problem. As a solution to the relaxation problem is returned, we branch on y_r with fractional value closest to 0.5, and fix $y_r = 0$ in one branch and $y_r = 1$ in the other branch. A feasible solution to LSND is obtained when all y_r take integers, whose objective provides a lower bound. The incumbent best lower bound of the problem is updated if any better feasible solution appears.

We prune a tree node if its upper bound is no greater than the incumbent best lower bound. The branch-and-price stops at an optimal solution when the problem's upper bound is no larger than the incumbent best lower bound.

We consider two node selection strategies. One is the depth-first strategy where branching is done on the current node. The other is the best-bound strategy where the node with the largest upper bound is branched. The depth-first strategy is applied at the beginning of the branch-and-price. We hope that some quickly feasible integer solutions can activate the pruning as early as possible to improve the solution efficiency. The best-bound strategy is applied in latter stages and expected to improve the upper bound to tighten the optimality gap.

However, not all relaxation problems at the tree nodes can be solved to optimality quickly because the convergence of SCRG will get slower when quite a number of rotations have been generated. Moreover, for larger instances, solving the pricing problem to optimality also takes much time. To avoid spending too much time on solving the relaxation problem at each node, we allow stopping the SCRG prematurely at a suboptimal solution. A consequence of the premature termination is that a looser upper bound might be obtained on that node. As compensation, more time will be spent on exploring new nodes that motivate the improvement of the incumbent lower bound solution. Therefore, in the branch-and-price, the premature termination can appropriately balance the improvements of upper bounds and lower bounds to enhance the solution efficiency.

3.4 Extensions

In this section, we study an extended problem of LSND in which each rotation is additionally allowed with a biweekly frequency, and moreover, the ship speed on each operated rotation is to be decided. To address these new decisions, in Section 3.4.1 we develop for the extended problem a new MILP model. Due to the huge number of potential candidate rotations, the new MILP is even more difficult to solve. To efficiently tackle an extended problem that is scaled to a practical size, in Section 3.4.2 we adopt a speed updating strategy in a heuristic solution procedure to find near-optimal solutions.

3.4.1 Model Extension

To decide the service frequency and ship speed, we define each rotation r is associated with a frequency $g(r)$, which implies the time interval between two regular port calls is $g(r)$ weeks. We set the values $g(r) = 1$ and $g(r) = 2$ to illustrate the weekly frequency and biweekly frequency. Besides, we associate a speed $v(r)$ to rotation r which means all ships on that rotation are operating at the given speed. It is reasonable to assume a constant speed for each rotation because the optimal speed is proven uniform over all legs so as to minimize the fuel cost since the fuel consumption function $F_s(v)$ is convex on v [73]. Discrete speed levels are often assumed to characterize the decision of speed in the network design [3, 13]. Let $\{v_{\min}, v_{\min}+\epsilon, ..., v_{\max}\}$ denote a set of discrete speed levels with ϵ indicating the interval size. When all possible ship types, service frequencies and speed levels are included, we obtain a new rotation set $\mathit{\Xi}$ and its size is about $\frac{2(v_{\max}-v_{\min})}{\epsilon}$ times larger than that of R.

For a rotation $r \in \mathit{\Xi}$, the transit time on each voyage arc $e \in A(r)$ is measured by $t_e = \frac{\ell_e}{24v(r)}$. To satisfy the frequency requirement, the number of ships required on each rotation r is calculated by

$$b'_r \ = \ \left\lceil \frac{1}{7g(r)} \sum_{e\in A(r)} \left(\frac{\ell_e}{24v(r)} + \tau \right) \right\rceil . \qquad (3.23)$$

Recall that the weekly operating cost of rotation can be calculated by the round-trip cost multiplied by the number of ships operating on it, and averaged by the number of weeks. Hence, given a'_r as the round-trip cost of a rotation $r \in \Xi$, we can calculate its weekly cost by $\frac{1}{g(r)}a'_r$. Similar with (3.2), the round-trip cost is calculated as follows:

$$a'_r \quad = \quad w^O_{s(r)}b'_r + \frac{1}{g(r)}\sum_{e\in A(r)} w^F_{s(r),e}(v(r)) + \frac{1}{g(r)}\sum_{i\in N(r)}\left(w^P_{s(r),z(i)} + w^S_{s(r),z(i)}\right). \tag{3.24}$$

Under the required frequency, every week the available capacity expected to serve the cargo demands is measured by $\frac{1}{g(r)}m_{s(r)}$. With similar objective function, constraints and variables as the LSND, the extended problem is formulated as follows:

$$\begin{aligned}
\max\quad & \sum_{k\in D} p_k x_k - \sum_{k\in D}\sum_{r\in \varXi}\sum_{e\in A(r)\cup L(r)} c^k_e f^k_{r,e} - \sum_{r\in\varXi}\frac{1}{g(r)}a'_r y_r,\\
\text{s.t.}\quad & \sum_{e\in A^+(i):e\in A(r)} f^k_{r,e} + \sum_{e\in L^+(i)} f^k_{r,e} - \sum_{e\in A^-(i):e\in A(r)} f^k_{r,e} - \sum_{e\in L^-(i)} f^k_{r,e} = 0,\\
& \qquad \forall k\in D, \forall r\in\varXi, \forall i\in N(r),\\
& \sum_{e\in L^+(i)}\sum_{r\in\varXi:tail(e)\in N(r)} f^k_{r,e} - \sum_{e\in L^-(i)}\sum_{r\in\varXi:head(e)\in N(r)} f^k_{r,e} = \begin{cases} -x_k, & o(k)=z(i)\\ 0, & otherwise\\ x_k, & d(k)=z(i)\end{cases}\\
& \qquad \forall k\in D, \forall i\in T\\
& \sum_{r\in\varXi:s(r)=s} b'_r y_r \le h_s, \quad \forall s\in S,\\
& \sum_{k\in D} f^k_{r,e} - \frac{1}{g(r)}m_{s(r)}y_r \le 0, \quad \forall r\in\varXi, \forall e\in A(r),\\
& f^k_{r,e} - q_k y_r \le 0, \quad \forall k\in D, \forall r\in\varXi, \forall e\in A(r),\\
& x_k \le q_k, \quad \forall k\in D,\\
& x_k \ge 0, f^k_{r,e}\ge 0, \quad \forall k\in D, \forall r\in\varXi, \forall e\in A(r)\cup L(r),\\
& y_r\in\{0,1\}, \quad \forall r\in\varXi.
\end{aligned}$$

Notice that the new MILP formulation also has both its variables and constraints associated with Ξ that involves more rotations than R. When we apply the SCRG to solve the LP relaxation, for any possible combination of (ship type, service frequency, speed level) there is a pricing problem to solve in Phase 2. Therefore, many more pricing problems are required to be handled at each iteration of SCRG. If each pricing problem cannot be solved very quickly, the implementation of SCRG would be computationally expensive. In this work, to efficiently tackle the extended problem scaled to a practical size, we introduce a heuristic to find its near-optimal solutions.

3.4.2 Heuristic

Instead of enumerating all possible speed levels, we adopt an updating strategy to determine the ship speed on each operated rotation. In the application of SCRG, given $s \in S$ and $g \in \{1, 2\}$, we define the pricing problem on a physical network $G_{s,g}(\bar{v})$, where $\bar{v}$ is randomly assigned within the interval $[v_{\min}, v_{\max}]$ and the edge cost is assigned regarding speed $\bar{v}$. The pricing problem is solved to return the rotation with a premier speed $\bar{v}$. Suppose r is returned, by substituting $\bar{v}$ into (3.23), a number of b'_r ships are obtained to maintain the required frequency $g(r)$. Since the fuel cost function $w^F_{s,e}(v)$ is increasing in v, smaller speed is thus preferred to save the fuel cost while keeping an equal service frequency. Given $\ell(r)$ and $\tau(r)$ respectively denoting the distance and port stay time on rotation r, to fulfill the frequency requirement, there is

$$7g(r)b'_r \;\; \geq \;\; \frac{\ell(r)}{24v(r)} + \tau(r) \tag{3.26}$$

A smallest operating speed of r is then updated by

$$v(r) \;\; = \;\; \max\left\{\frac{\ell(r)}{24(7g(r)b'_r - \tau(r))}, v_{\min}\right\}. \tag{3.27}$$

Using the speed updating strategy, we only need the information of ship type and frequency to define a pricing problem, while the arc cost of the physical network is based on a randomly produced speed. As a result, a smaller number of pricing problems are to be handled at each iteration of SCRG. Based on this, computation will be much faster but optimality of the relaxation problem cannot be guaranteed. We next develop a heuristic solution method to obtain near-optimal solutions to the extended problem. In the heuristic, SCRG is applied to solve the LP relaxation of the extended problem using the speed updating strategy. Many rotations with diversified speeds will be generated when the SCRG stops, and a subset of them will be selected to construct a solution to the MILP using branch-and-bound. Similar approaches have been applied and shown to be competitive for solving various large-scale MILP problems using column generation [1, 22].

3.5 Numerical Experiments

We conduct a computational study based on the test cases (Baltic and WAF) from the benchmark suite LINER-LIB 12 proposed in [13]. Both randomly generated instances and benchmark instances are utilized to examine the proposed methods in this work. All algorithms are implemented in C++ using the CPLEX of version 12.6. Experiments are conducted on an Intel Core i7 (3.4 GHz) Desktop PC with 8 GB RAM. We first introduce the test instances in Section 3.5.1. The solution performances on the LP relaxations are discussed in Section 3.5.2. The performance of exact and heuristic solutions are compared in Section 3.5.3 and Section 3.5.4, respectively.

3.5.1 Test Instances

Our test instances are with the number of ports ranging from 6 to 20, which are consistent with small and moderate sized network design cases from the practice. We classify the test instances into two groups. In the first group, based on the data information of Baltic (12 ports and 22 demands) and WAF (20 ports and 37 demands), we randomly generate the instances by differentiating their involved number of ports and number of demands. Each random instance is expressed by "pX-dY-Z" that implies "X" ports and "Y" demands from test case "Z" are involved, in which Baltic and WAF are notated by "Z=b" and "Z=w", respectively. In the second group, the original benchmark instances of Baltic and WAF in LINER-LIB 12 are included. As introduced in [13], each benchmark instance is associated with a scenario from {*low*, *base*, *high*} to reflect the supply and demand conditions of the market. Given the base scenario as a standard, freight rate is multiplied by 1.4 and fleet size is multiplied by 0.8 in a low scenario. In a high scenario, freight rate is multiplied by 0.8 and fleet size is multiplied by 1.2. The number of ships for each scenario is rounded to the nearest integer. We represent each benchmark instance by "pX-dY-Z-S" and "S" is additionally selected from $\{l, b, h\}$ to indicate the scenario {*low*, *base*, *high*} respectively.

The fuel oil price is set by $O = 600$ and the port stay time for each port call is set by $\tau = 1$, which are consistent with the benchmark instance settings in [57], [13] and [14]. Furthermore, we select *weekly* or *(bi)weekly* as the service frequency requirement. The decision of ship speed is allowed (*yes*) or not allowed (*no*). The penalty cost for rejecting a cargo is set by 0 or 1000. In the experiment, we attach the test instances with different requirements on the service frequency, ship speed and penalty cost. Solution methods to solving the LSND and solving the extended problem of LSND are examined in the experiments. For LSND, we enumerate all possible rotations for some smaller instances and then apply CPLEX's MIP-Solver and LP-Solver to tackle the MILP formulation and the LPR to obtain benchmark results. Moreover, CG is implemented to solve the ALPR while SCRG is applied to solve the LPR. Branch-and-price is implemented to find optimal or near-optimal solutions to LSND. For the extended problem of LSND, the SCRG based heuristic with speed updating strategy is implemented to find near-optimal solutions. The obtained solutions are compared with the benchmark solutions from [57] (denoted by Plum14), [13] (denoted by Brouer14a) and [14] (denoted by Brouer14b), respectively, under identical problem settings. Note that due to the solution objectives reported in Brouer14a and Brouer14b are measured by the profits within 180 days, we report their objective values by timing (7/180) so that they are comparable with our weekly profits.

Table 3.1: Overview of the test instances

INST	FREQ	SPEED	PENALTY	LENGTH	DURATION	SOL	vs.SOL
Group I							
p6-d6-b	weekly	no	0	5	10	SCRG, B&P	CG, CPLEX
p6-d6-w	weekly	no	0	5	10	SCRG, B&P	CG, CPLEX
p6-d10-b	weekly	no	0	5	10	SCRG, B&P	CG, CPLEX
p6-d10-w	weekly	no	0	5	10	SCRG, B&P	CG, CPLEX
p9-d12-b	weekly	no	0	5	10	SCRG, B&P	CG, CPLEX
p9-d12-w	weekly	no	0	5	10	SCRG, B&P	CG, CPLEX
p9-d15-b	weekly	no	0	5	10	SCRG, B&P	CG, CPLEX
p9-d15-w	weekly	no	0	5	10	SCRG, B&P	CG, CPLEX
p12-d20-w	weekly	no	0	7	10	B&P	-
p12-d22-w	weekly	no	0	7	10	B&P	-
p15-d26-w	weekly	no	0	7	15	B&P	-
p15-d28-w	weekly	no	0	7	15	B&P	-
p18-d32-w	weekly	no	0	11	15	B&P	-
p18-d34-w	weekly	no	0	11	15	B&P	-
Group II							
p12-d22-b-l	weekly	no	0	7	10	B&P	Plum14
p12-d22-b-b	weekly	no	0	7	10	B&P	Plum14
p12-d22-b-h	weekly	no	0	7	10	B&P	Plum14
p20-d37-w-l	weekly	no	0	11	15	B&P	Plum14
p20-d37-w-b	weekly	no	0	11	15	B&P	Plum14
p20-d37-w-h	weekly	no	0	11	15	B&P	Plum14
p12-d22-b-l	(bi)weekly	no	1000	7	10	Heuristic	Plum14
p12-d22-b-b	(bi)weekly	no	1000	7	10	Heuristic	Plum14
p12-d22-b-h	(bi)weekly	no	1000	7	10	Heuristic	Plum14
p20-d37-w-l	(bi)weekly	no	1000	11	15	Heuristic	Plum14
p20-d37-w-b	(bi)weekly	no	1000	11	15	Heuristic	Plum14
p20-d37-w-h	(bi)weekly	no	1000	11	15	Heuristic	Plum14
p12-d22-b-l	weekly	yes	1000	7	10	Heuristic	Brouer14b
p12-d22-b-b	weekly	yes	1000	7	10	Heuristic	Brouer14b
p12-d22-b-h	weekly	yes	1000	7	10	Heuristic	Brouer14b
p20-d37-w-l	weekly	yes	1000	11	15	Heuristic	Brouer14b
p20-d37-w-b	weekly	Yes	1000	11	15	Heuristic	Brouer14b
p20-d37-w-h	weekly	yes	1000	11	15	Heuristic	Brouer14b
p12-d22-b-l	(bi)weekly	yes	1000	7	10	Heuristic	Brouer14a
p12-d22-b-b	(bi)weekly	yes	1000	7	10	Heuristic	Brouer14a
p12-d22-b-h	(bi)weekly	yes	1000	7	10	Heuristic	Brouer14a
p20-d37-w-l	(bi)weekly	yes	1000	11	15	Heuristic	Brouer14a
p20-d37-w-b	(bi)weekly	yes	1000	11	15	Heuristic	Brouer14a
p20-d37-w-h	(bi)weekly	yes	1000	11	15	Heuristic	Brouer14a

We present in Table 3.1 the details of all test instances. Column ”INST” denotes the name of the instance. Columns ”FREQ”, ”SPEED” and ”PENALTY” show the problem setting imposed on each instance, including the frequency requirement, ship speed and penalty cost. Column ”LENGTH” and ”DURATION” are the maximum number of port calls and maximum round-trip duration for defining a possible rotation. Column ”SOL” and ”vs.SOL” respectively involves our solutions and the benchmark solutions.

3.5.2 Comparing the Relaxations

We now compare the performances of the proposed solution approaches on solving the LP relaxations for the LSND. The solution of ALPR is obtained via adopting CG. By enumerating all possible rotations, we tackle the LPR via the CPLEX's LP-Solver and report the obtained solution under (LPR + CPLEX). Applying the SCRG approach, we report the solution of LPR under (LPR + SEQ) if construction of Π_0 follows the sequential approach, and report the solution under (LPR + LPB) if it follows the LP-based approach. As the violation verification process in Section 3.3.3.3 is additionally applied to solving the pricing problem, we report the obtained solution under (LPR + LPB + VV). All solution approaches are implemented with a runtime limit of 300 seconds. All obtained solutions are reported on their number of generated rotations under column ”COL” and the computation time under the column ”TIME”. Column ”OBJ(%)” below (ALPR) shows the percentage gap between the optimal objectives of ALPR and LPR.

As is shown in Table 3.2, LPR provides tighter upper bounds, which is as expected since ALPR is known a relaxation to LPR. To solve the LPR, CPLEX's LP-Solver can reach optimal solutions for those instances with 6 ports, and the optimality is not achieved within 300 seconds for the instances with 9 ports. Clearly, the SCRG using sequential approach performs better than the CPLEX's LP-Solver on the solution efficiency, but the optimality is still not guaranteed for those instances with 9 ports.

In comparison, the SCRG using the LP-based approach for the dual solution construction can achieve optimal solutions within 300 seconds for all reported instances. Moreover, under (LPR + LPB), obviously fewer rotations are generated and less computation time is consumed when SCRG reaches optimality. That is consistent with our initial intuition that the LP-based dual solution construction of the two-phase scheme is advantageous to provide suitable values for the absent dual variables in Phase 1 that helps avoid the detections of invalid violations in Phase 2. Experimental evidence also shows that SCRG will have a fast convergence if the values of Π_0 are assigned appropriately. Moreover, by comparing the results under (LPR + LPB) and those under (LPR + LPB + VV), it is evident to see that the adoption of violation verification in solving the pricing problems can effectively reduce the number of generated rotations without increasing the computational time. Because the proposed verification process effectively avoids the generation of rotations with invalid violations. For the rest of experiments implementing SCRG, we uniformly adopt the approach under (LPR + LPB + VV) to solve the LP relaxation.

Table 3.2: Solution performance on the relaxations

	ALPR			LPR + CPLEX		LPR + SEQ		LPR + LPB		LPR + LPB + VV	
INST	OBJ(%)	COL	TIME	COL	TIME	COL	TIME	COL	TIME	COL	TIME
p6-d6-b	0.0	35	1	732	5	161	5	59	1	35	1
p6-d6-w	0.2	34	1	395	5	177	6	42	1	21	1
p6-d10-b	0.6	81	1	511	10	209	7	56	1	32	1
p6-d10-w	4.1	133	2	467	34	168	6	45	1	26	1
p9-d12-b	1.6	443	102	4190	300†	787	300†	84	1	80	2
p9-d12-w	5.5	234	29	4702	300†	492	300†	175	21	80	21
p9-d15-b	2.2	388	72	4393	300†	589	300†	83	1	71	1
p9-d15-w	0.8	337	56	4702	300†	335	300†	162	18	66	12

Notes: INST denotes the name of the instance. OBJ(%) reports the percentage gap between the optimal objectives of ALPR and LPR, which is measured by $\frac{\text{ALPR-LPR}}{\text{LPR}}$. COL reports the number of generated rotations. TIME reports the computation time in seconds. †: Optimality not proved.

3.5.3 Comparing the Exact Solutions

We next compare the solution performance of branch-and-price with that of the CPLEX's MIP-Solver and that of Plum14 for solving the LSND. All reported test instances are with weekly service frequency, fixed speed, and zero penalty cost. The tested solution method is with a runtime limit of 2 hours. The column ”COL” records the number of all possible rotations by enumeration. If the number exceeds one billion, we state ”$> 10^9$” on the table. The solution performance is reported regarding the objective value (”OBJ”), the upper bound value (”UB”), the optimality gap (”GAP”) and the computation time (”TIME”).

Table 3.3: Exact solutions for the LSND

		CPLEX				Branch-and-price			
INST	COL	OBJ	UB	GAP(%)	TIME	OBJ	UB	GAP(%)	TIME
p6-d6-b	732	102797	102797	0.0	13	102797	102797	0.0	1
p6-d6-w	395	684748	684748	0.0	55	684748	684748	0.0	3
p6-d10-b	511	277019	277019	0.0	27	277019	277019	0.0	3
p6-d10-w	467	1074382	1074382	0.0	2277	1074382	1074382	0.0	2889
p9-d12-b	4190	368314	476473	22.7	7200	374040	374040	0.0	3
p9-d12-w	4702	1172185	1619487	27.6	7200	1281405	1281405	0.0	2631
p9-d15-b	4393	612629	943813	35.1	7200	712921	712921	0.0	61
p9-d15-w	4702	2996155	3590359	16.6	7200	3202709	3247785	1.4	7200
p12-d20-w	901184	n/a	n/a	n/a	n/a	3632189	3878197	6.3	7200
p12-d22-w	893776	n/a	n/a	n/a	n/a	2710615	2906484	6.7	7200
p15-d26-w	957809	n/a	n/a	n/a	n/a	3734023	4127781	9.5	7200
p15-d28-w	909432	n/a	n/a	n/a	n/a	5121447	5553856	7.8	7200
p18-d32-w	$> 10^9$	n/a	n/a	n/a	n/a	5579441	6240528	10.6	7200
p18-d34-w	$> 10^9$	n/a	n/a	n/a	n/a	5405048	6175337	12.5	7200
		Plum14				Branch-and-price			
INST	COL	OBJ	UB	GAP(%)	TIME	OBJ	UB	GAP(%)	TIME
p12-d22-b-l	385323	427485	611015	30.0	3600	662766	687349	3.6	7200
p12-d22-b-b	932654	408771	669774	39.0	3600	748618	763933	2.0	7200
p12-d22-b-h	955985	636152	657021	3.2	3600	785773	802831	2.1	7200
p20-d37-w-l	$> 10^9$	1940817	6051697	67.9	10800	5106296	6262000	18.5	7200
p20-d37-w-b	$> 10^9$	3372618	6399041	47.3	10800	5523499	6764104	18.3	7200
p20-d37-w-h	$> 10^9$	3899767	6614613	41.0	10800	6009782	6973469	13.8	7200

Notes: INST denotes the name of the instance. COL records the number of all possible rotations. OBJ reports the solution objective. UB reports the upper bound on the optimal solution objective. GAP(%) reports the optimality gap computed by $\frac{\text{UB-OBJ}}{|\text{UB}|}$. TIME reports the computation time in seconds.

From Table 3.3, we see that when all possible rotations are generated a prior, CPLEX's MIP-Solver can solve the instances with 6 ports to optimality, and achieve an optimality gap from 16.6% to 35.1% for the instances with 9 ports. Since the total number of rotations increases exponentially by the network size, CPLEX fails to output a feasible solution for larger instances in which "n/a" is remarked. In comparison, the branch-and-price solves all the instances with 6 ports and most of the instances with 9 ports to optimality within 1 hour, which outperforms CPLEX's MIP-Solver regarding both solution quality and time efficiency. For the other instances, with an increase of the network size, the branch-and-price can achieve solutions with optimality gaps varying from 1.4% to 18.5%.

We also compare our solutions with the benchmark solutions of Plum14, which are to our knowledge the only results reporting the upper bounds on the benchmark suite instances. It can be seen that our solutions completely outperform theirs, and the obtained improvements are substantial. In addition, significantly smaller optimality gaps are also achieved by our branch-and-price. Note that some of our solution objectives even exceed the reported upper bounds of Plum14, mainly because the allowed service number for Plum14 is set too small to output a better solution under the limited solution space. Hence, the upper bounds reported in Plum14 are not available to evaluate our solutions.

3.5.4 Comparing the Heuristic Solutions

Our heuristic is examined to benchmark against that of Plum14, Brouer14a and Brouer14b for the extended problem of LSND. Solutions are compared under the same setting of the frequency requirement, the speed decision and the penalty cost. The heuristic is carried out with a runtime limit of 2 hours. We report the solution performances in Table 3.4 concerning the solution objectives ("OBJ") and computation times ("TIME"). The improvements of our solutions over the benchmark solutions are reported under the column "IPM".

We first benchmark our solution against Plum14 where both weekly and biweekly frequency are allowed, but speed is not to be decided. The results show that our heuristic solutions averagely outperform Plum14 by almost 500% while significantly consumes less computation time. Compared with the results of Brouer14b where a weekly frequency is required and speed is to be decided, our solutions achieve 5.3% on average better than theirs. When both weekly and bi-weekly frequency are allowed and speed is to be decided, our solutions averagely improve the solutions of Brouer14a by 17.4%. The overall results in Table 3.4 demonstrate that our SCRG approach can effectively generate high-quality rotations for acquiring superior solutions.

Table 3.4: Heuristic solutions for the extended problem of LSND

				Plum14		Heuristic		
INST	FREQ	SPEED	PENALTY	BEST	TIME	OBJ	IPM(%)	TIME
p12-d22-b-l	(bi)weekly	no	1000	-265117	3600	121240	145.7	515
p12-d22-b-b	(bi)weekly	no	1000	-134687	3600	455360	438.1	647
p12-d22-b-h	(bi)weekly	no	1000	-183348	3600	642827	450.6	1797
p20-d37-w-l	(bi)weekly	no	1000	411317	10800	5090495	1137.6	7200
p20-d37-w-b	(bi)weekly	no	1000	1059352	10800	5638555	432.3	7200
p20-d37-w-h	(bi)weekly	no	1000	1281583	10800	6081794	374.6	7200
Average							**496.5**	

				Brouer14b		Heuristic		
INST	FREQ	SPEED	PENALTY	BEST	TIME	OBJ	IPM(%)	TIME
p12-d22-b-l	weekly	yes	1000	-133389	27	-131455	1.4	389
p12-d22-b-b	weekly	yes	1000	239556	105	263224	9.9	166
p12-d22-b-h	weekly	yes	1000	420000	33	430593	2.5	198
p20-d37-w-l	weekly	yes	1000	4433333	95	4659476	5.1	7200
p20-d37-w-b	weekly	yes	1000	5444444	93	5965974	9.6	7200
p20-d37-w-h	weekly	yes	1000	6261111	98	6457341	3.1	7200
Average							**5.3**	

				Brouer14a		Heuristic		
INST	FREQ	SPEED	PENALTY	BEST	TIME	OBJ	IPM(%)	TIME
p12-d22-b-l	(bi)weekly	yes	1000	235044	300	234172	-0.4	812
p12-d22-b-b	(bi)weekly	yes	1000	325306	300	495600	52.3	2422
p12-d22-b-h	(bi)weekly	yes	1000	609700	300	673681	10.5	1337
p20-d37-w-l	(bi)weekly	yes	1000	4486261	900	5339942	19.0	7200
p20-d37-w-b	(bi)weekly	yes	1000	5565389	900	6305260	13.3	7200
p20-d37-w-h	(bi)weekly	yes	1000	6220044	900	6831991	9.8	7200
Average							**17.4**	

Notes: INST denotes the name of the instance. FREQ is the frequency requirement allowed for each service. SPEED shows whether speed is to be decided. PENALTY is the penalty cost. BEST shows the objectives from the best existing solution. OBJ reports the objective of our solution. IPM(%) reports the percentage of improvement computed by $\frac{\text{OBJ-BEST}}{|\text{BEST}|}$. TIME reports the computation time in seconds.

3.6 Summary

In this work, we have developed a compact model to capture the transshipment cost in a liner shipping network design problem. The proposed formulation is appropriate for applying the simultaneous column-and-row generation approach to solving its linear programming relaxation so as to obtain a valid upper bound. In the approach, we have proposed a novel two-phase scheme to construct values for the dual variables that are missing in the restricted problem. Following the construction, a pricing problem is solved to seek violated dual constraints by searching cycles of negative costs on a weighted network. Based on the simultaneous column-and-row generation, we have developed a branch-and-price to find the optimal or near-optimal network design. Moreover, we have extended our new models and solution approaches to a more general problem where biweekly frequency is allowed and ship speed on each operated rotation is also to be decided. By adopting a speed updating strategy, a heuristic based on the simultaneous column-and-row generation has been proposed to obtain the near-optimal solution for the extended version of the decision problem. Numerical experiments have been conducted to examine the performance of the proposed solution methods on some small and moderate sized instances from the benchmark suite LINER-LIB 12. The results have shown that our simultaneous column-and-row generation approach can solve the LP relaxation of the problem with an appropriate construction of the values of dual variables. Also, the obtained solutions significantly outperform the existing benchmark solutions, and the improvements are significant.

CHAPTER 4

Conclusions

This thesis comprises of two essays on liner shipping network design. In the first essay, we have developed a joint optimization model for multi-decision making on ship routes, fleet deployment, speed, cargo allocation and frequency. A new speed-load dependent fuel consumption function is assumed in the model to address the speed optimization together with the network design. Since the proposed model is non-linear and very difficult to solve. We approximate the non-linear model to a mixed integer linear programming (MILP) model that can be solved by a column generation based heuristic. In addition, we propose an iterative search algorithm that adaptively changes the approximation to further enhance the heuristic. Extensive experiments have been conducted to show both the significance of the speed-load dependent fuel consumption function and the effectiveness of our approach. We have then provided a sensitivity analysis that investigates the impact of different scenarios on the solutions, such as fluctuations in fuel oil price, demand market, minimum speed requirements and various fleet deployment strategies.

For future research of the first essay, disaggregation techniques in [49] can be developed to restore the Aggregate Service Network (ASN) to the General Service Network (GSN) when decision makers plan on route design and cargo allocation. The other direction for future work is to take ballast water into consideration. In many cases, containerships sail with ballast water so as to maintain stable sailing if not enough cargos are loaded. This implies that the impact of loads on fuel consumption could actually be a piecewise function, rather than the continuous one under current consideration. Therefore, this work can also be extended by investigating an adjusted or more generalized fuel consumption function.

In the second essay, we have presented a compact MILP model for the liner shipping network design problem with transshipment cost. The proposed formulation is appropriate for applying the simultaneous column-and-row generation approach to solving its linear programming relaxation so as to obtain a valid upper bound. In the simultaneously column-and-row generation, we propose a novel two-phase scheme to link the generation of new rotations to solving a familiar pricing problem. Based on that, simultaneously column-and-row generation can work similarly as a column generation. Results from extensive numerical experiments have shown that the proposed approach can efficiently stop at the optimality of the relaxation problem.

Based on the simultaneous column-and-row generation, we have developed a branch-and-price to find its optimal or near-optimal solution. The basic model is further extended to involve new decisions on service frequency and ship speed. A heuristic is proposed to find near-optimal solutions for the extended problem. Numerical experiments have been conducted to examine the performance of the proposed methods to benchmark against the existing solutions from the literature. Our solutions averagely have significant improvements over the benchmark solutions.

Future work of the second essay may focus on solving other larger instances in the benchmark suite. To solve larger instances, one could follow the solution procedure of [13] and [14] while the generation of new rotations relies on the simultaneous column-and-row generation approach. Furthermore, the proposed simultaneous column-and-row generation is relatively new, one can also apply it to many other problems that are modeled with interdependent columns and rows in the formulation.

Moreover, both essays assume static cargo demands which are independent of the service quality in liner shipping. However, customers are sometimes sensitive to the transit times that are used to fulfill their cargo demands [76]. In response to the service level dependent demand, the operated liner shipping services should also be configured accordingly, especially for the decisions on cargo routing and speed optimization. Therefore, considering the impact of service level on demands can be another important future direction for the liner shipping network design.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] R. Agarwal and Ö. Ergun. Ship scheduling and network design for cargo routing in liner shipping. *Transportation Science*, 42(2):175–196, 2008.

[2] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, 1993.

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

[4] J. Andersen, T. G. Crainic, and M. Christiansen. Service network design with management and coordination of multiple fleets. *European Journal of Operational Research*, 193:377–389, 2009.

[5] J. Andersen, T. G. Crainic, and M. Christiansen. Service network design with asset management: Formulations and comparative analyses. *Transportation Research Part C: Emerging Technologies*, 17:197–207, 2009.

[6] J. Andersen, M. Christiansen, T. G. Crainic, and R. Grohaug. Branch and price for service network design with asset management constraints. *Transportation Science*, 41(1):33–49, 2011.

[7] P. Avella, B. D'Auria, and S. Salerno. A lp-based heuristic for a time-constrained routing problem. *European Journal of Operational Research*, 173:120–124, 2006.

[8] S. Bailey. Supply-demand gap in container shipping market to widen in 2014. Hellenic Shipping News, 2014. Feb 13, 2014.

[9] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P.H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998.

[10] T. Bektaş and G. Laporte. The pollution-routing problem. *Transportation Research Part B: Methodological*, 45(8):1232–1250, 2011.

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

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

[13] B. D. Brouer, J. F. Álvarez, C. E. M. Plum, D. Pisinger, and M. M. Sigurd. A base integer programming model and benchmark suite for liner-shipping network design. *Transportation Science*, 48(2):281–312, 2014.

[14] B. D. Brouer, G. Desaulniers, and D. Pisinger. A matheuristic for the liner shipping network design problem. *Transportation Research Part E: Logistics and Transportation Review*, 72:42–59, 2014.

[15] BRS. Shipping and shipbuilding markets - 2011 annual review. *Barry Rogliano Salles*, pages 85–86, 2011.

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

[17] M. Christiansen, K. Fagerholt, B. Nygreen, and D. Ronen. Maritime transportation. In C. Barnhart and G. Laporte, editors, *Handbooks in operations research and management science (volume 14): Transportation*, pages 189–284. Elsevier, 2007.

[18] M. Christiansen, K. Fagerholt, B. Nygreen, and D. Ronen. Ship routing and scheduling in the new millennium. *European Journal of Operational Research*, 228(3):467–483, 2012.

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

[20] World Shipping Council. Container supply review, May 2011.

[21] IBM CPLEX. http:// www-01.ibm.com/software/commerce/optimization/cplex-optimizer, 2015.

[22] T. G. Crainic, M. Hewitt, M. Toulouse, and D. M. Vu. Service network design with resource constraints. *Transportation Science*, 2014.

[23] J. Desrosiers and M.E. Lübbecke. *A primer in column generation*. Springer, New York, spring 2005.

[24] Drewry. Container market review and forecast : Annual report 2014/2015, 2015.

[25] K. Fagerholt, G. Laporte, and I. Norstad. Reducing fuel emissions by optimizing speed on shipping routes. *Journal of the Operational Research Society*, 61(3): 523–529, 2010.

[26] K. Fagerholt, N. T. Gausel, J. G. Rakke, and H. N. Psaraftis. Maritime routing and speed optimization with emission control areas. *Transportation Research Part C: Emerging Technologies*, 52:57–73, 2015.

[27] D. Feillet, M. Gendreau, A. L. Medaglia, and J. L. Walteros. A note on branch-and-cut-and-price. *Operations Research Letters*, 38:346–353, 2010.

[28] S. Gelareh and Q. Meng. 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, 2010.

[29] S. Gelareh and D. Pisinger. Fleet deployment, network design and hub location of liner shipping companies. *Transportation Research Part E: Logistics and Transportation Review*, 47(6):947–964, 2011.

[30] Hapag-Lloyd. urlwww.hapag-lloyd.com, 2015. Accessed March, 2015.

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

[32] M. K. Jepsen, B. Løfstedt, C. E. M. Plum, D. Pisinger, and M. M. Sigurd. A path based model for a green liner shipping network design problem. *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 2, 2011.

[33] N. Katayama, M. Chen, and M. Kubo. A capacity scaling heuristic for the multicommodity capacitated network design problem. *Journal of Computational and Applied Mathematics*, 232(1):90–101, 2009.

[34] C. A. Kontovas and H. N. Psaraftis. Reduction of emissions along the maritime intermodal container chain. *Maritime Policy & Management*, 38(4):455–473, 2011.

[35] C. E. Leiserson, R. L. Rivest, C. Stein, and T. H. Cormen. *Introduction to algorithms*. MIT press, 2001.

[36] S. Lin and W. Kernighan. An effective heuristic algorithm for the traveling salesman problem. *Operations Research*, 21(2):498–516, 1973.

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

[38] M.E. Lübbecke and J. Desrosiers. Selected topics in column generation. *Operations Research*, 53(6):1007–1023, 2005.

[39] E. MacArthur. Towards the circular economy: Accelerating the scale-up across global supply chains. Technical report, World Economic Forum, January 2014.

[40] Maersk-Line. www.maerskline.com, 2016. Accessed June 7, 2016.

[41] E. F. Magirou, H. N. Psaraftis, and T. Bouritas. The economic speed of an oceangoing vessel in a dynamic setting. *Transportation Research Part B: Methodological*, 76:48–67, 2015.

[42] S. J. Maher. Solving the integrated airline recovery problem using column-and-row generation. *Transportation Science*, 50(1):216–239, 2016.

[43] Q. Meng and S. Wang. Optimal operating strategy for a long-haul liner service route. *European Journal of Operational Research*, 215(1):105–114, 2011.

[44] Q. Meng and S. Wang. Liner shipping service network design with empty container repositioning. *Transportation Research Part E: Logistics and Transportation Review*, 47(5):695–708, 2011.

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

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

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

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

[49] J. Mulder and R. Dekker. Methods for strategic liner shipping network design. *European Journal of Operational Research*, 235(2, SI):367–377, 2014.

[50] İ. Muter, Ş. İ. Birbil, and K. Bülbül. Simultaneous column-and-row generation for large-scale linear programs with column-dependent-rows. *Mathmatical Programming*, 142:47–82, 2013.

[51] I. Norstad, K. Fagerholt, and G. Laporte. Tramp ship routing and scheduling with speed optimization. *Transportation Research Part C: Emerging Technologies*, 19(5):853–865, 2011.

[52] T. Notteboom, F. Parola, and G. Satta. Partim transshipment volumes. Technical report, 2014.

[53] T. E. Notteboom and B. Vernimmen. The effect of high fuel costs on liner service configuration in container shipping. *Journal of Transport Geography*, 17 (5):325–337, 2009.

[54] OOCL. www.oocl.com, 2016. Accessed June 7, 2016.

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

[56] PIERS. Capacity utilization report 2011. Technical report, PIERS, August 2012.

[57] C. E. M. Plum, D. Pisinger, and M. M. Sigurd. A service flow model for the liner shipping network design problem. *European Journal of Operational Research*, 235 (2):378–386, 2014.

[58] B. J. Powell and A. N. Perkins. Fleet deployment optimization for liner shipping: an integer programming model. *Maritime Policy & Management*, 24(2):183–192, 1997.

[59] H. N. Psaraftis. A ship pickup and delivery model with multiple commodities, variable speeds, cargo inventory costs and freight rates. In *Proceedings of ODYSSEUS 2012*, 2012.

[60] H. N. Psaraftis and C. A. Kontovas. Speed models for energy-efficient maritime transportation: A taxonomy and survey. *Transportation Research Part C: Emerging Technologies*, 26:331–351, 2013.

[61] L. B. Reinhardt and D. Pisinger. A branch and cut algorithm for the container shipping network design problem. *Flexible Services and Manufacturing Journal*, 24(3):349–374, 2012.

[62] D. Ronen. The effect of oil price on the optimal speed of ships. *Journal of the Operational Research Society*, 33:1035–1040, 1982.

[63] D. Ronen. Ship scheduling: The last decade. *European Journal of Operational Research*, 71(2):325–333, 1993.

[64] D. Ronen. The effect of oil price on containership speed and fleet size. *Journal of the Operational Research Society*, 62(1):211–216, 2011.

[65] M.W.P Savelsbergh. A branch-and-price algorithm for the generalized assignment problem. *Operations Research*, 45(6):831–841, 1997.

[66] D. P. Song and J. X. Dong. Long-haul liner service route design with ship deployment and empty container repositioning. *Transportation Research Part B: Methodological*, 55:188–211, 2013.

[67] UNCTAD. Review of maritime transport 2009. Annual report, 2009.

[68] UNCTAD. Review of maritime transport 2014. Annual report, 2014.

[69] UNCTAD. Review of maritime transport 2015. Annual report, 2015.

[70] S. Wang and Q. Meng. Sailing speed optimization for container ships in a liner shipping network. *Transportation Research Part E: Logistics and Transportation Review*, 48(3):701–714, 2012.

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

[72] S. Wang and Q. Meng. Liner shipping network design with deadlines. *Computers & Operations Research*, 41:140–149, 2014.

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

[74] S. Wang, Z. Liu, and M. G. H. Bell. Profit-based maritime container assignment models for liner shipping networks. *Transportation Research Part B: Methodological*, 72:59–76, 2015.

[75] S. Wang, Q. Meng, and C. Y. Lee. Liner container assignment model with transit-time-sensitive shipment demand and its applications. *Transportation Research Part B: Methodological*, 90:135–155, 2016.

[76] S. Wang, Q. Meng, and Z. Liu. Containership scheduling with transit-time-sensitive container shipment demand. *Transportation Research Part B: Methodological*, 54:68–83, 2013.

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

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

[79] C. Yan and J. Kung. Robust aircraft routing. *Transportation Science*, Article in Advance, 2016.

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

[81] E. J. Zak. Row and column generation technique for a multistage cutting stock problem. *Computers & Operations Research*, 29:1143–1156, 2002.