Network Design, Fleet Deployment and Empty Repositioning in Liner Shipping

Rahimeh Neamatian Monemi[a,*], Shahin Gelareh[b]

[a] Centre for Operational Research, Management Science and Information Systems (CORMSIS), University of Southampton, Southampton, SO17 1BJ, United Kingdom
[b] Portsmouth Business School, Portsmouth, United Kingdom

Abstract

We present an integrated modelling framework for the joint problems of network design, fleet deployment and empty repositioning in liner shipping. In our problem the number of service routes and their design are an endogenous part of the problem. The cost of a route is a set function mapping a subset of edges, vessel types and quantities to deploy to the set of non-negative real numbers. Since such cost structures cannot be accommodated in a compact formulation, our modelling framework, which is based on the paradigm of the Benders reformulation, integrates separate problems aiming to obtain a solution to the integrated problem. In this work we look at the Benders approach as a tool for integrating separate optimization problems rather than decomposing an integrated holistic optimization problem. Our numerical experiments show that the method is very efficient in solving instances of this problem with respect to both the problem size and the computational time.

Keywords: Location allocation, liner shipping, network design, fleet deployment, empty repositioning, valid inequalities, branch and cut, Benders decomposition.

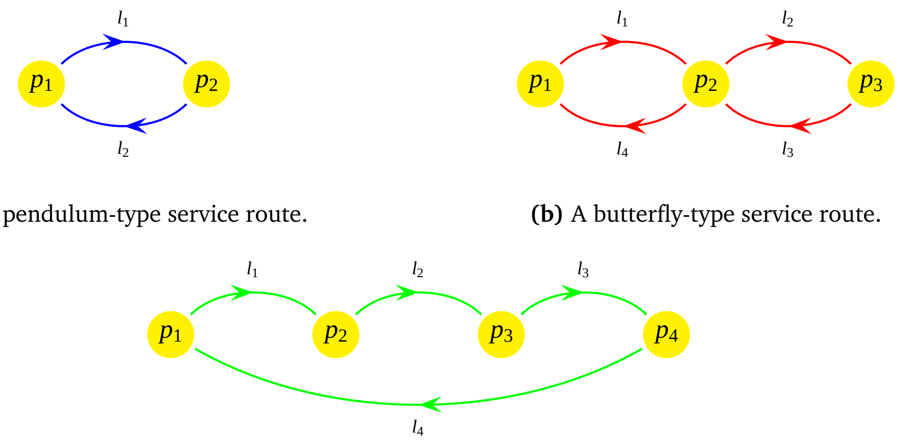
1. Introduction

The container trade imbalance on a container flow direction refers to the situation in which the number of export and import containers (even with respect to the type of containers) differs significantly. This occurs for several reasons. While in developing countries the volume and even sometimes the value of import containerized freight is significantly more than the volume and value of export containerized freight, in some developed countries the imbalance is of a different nature. In the second group of economies, often the imbalance occurs because the import containers and the export containers are not of the same type, that is, one is composed of mainly 1-TEU (twenty-foot equivalent unit) containers and the other contains more 2-TEU (1-FEU, forty-foot equivalent unit) containers.

A major part of this imbalance is due to the spatial distribution of production and consumption centres and the nature of imports and exports in different economies around the globe (Wang, 2007). Maritime shippers spend on average $100 billion per year on operating their container assets, of which around $16 billion is spent on repositioning empty containers (Rodrigue, 2016). However, according to the UNCTAD (2015), in 2015 the global ports' throughput was almost 2.2 times the whole containerized trade volume, implying a significant load of repositioning activities.

The East-West trade imbalance is continuing to expand, and the operation is becoming increasingly costly. Among the three major East-West trade routes (i.e., Transpacific, Transatlantic and Europe-Asia), in 2015 carriers operating between Asia and North America had to reposition 1.2 million TEUs more than they did

*Corresponding author, R.Neamatian-Monemi@soton.ac.uk



(a) A pendulum-type service route.

(b) A butterfly-type service route.

(c) A cyclic service route.

Figure 1: The simple service route building blocks.

in 2014. This is an indication of the increasing imbalance on the corresponding trade routes. On the other hand, the amount of empty repositioning has decreased by around 600,000 TEUs for the trade between Asia and Europe (including the Mediterranean region). However, still on the corresponding route(s), the total number of empty containers repositioned is as high as around 7.8 million TEUs (UNCTAD, 2015).

Drewry concludes that *'The East-West trades were a bad place to be for ocean carriers in 2015. Carrier profitability/losses in 2016 will be heavily influenced by the exposure to hot or cold trades, while carriers should also expect additional associated empty container repositioning costs from the widening East-West services trade gap'* (see Waters (2016) and Container Insight Weekly[1]).

In the literature the problems of network design and empty repositioning in liner shipping are often studied separately. What we refer to as a 'network design problem' is the problem of designing from scratch a network (characterized by a set of nodes and a set of arcs incident to those nodes) and the corresponding service routes composed of those arcs (legs of calls) constituting the ship round trips. Our problem is not to choose routes only from among set of *a priori* existing and known ones.

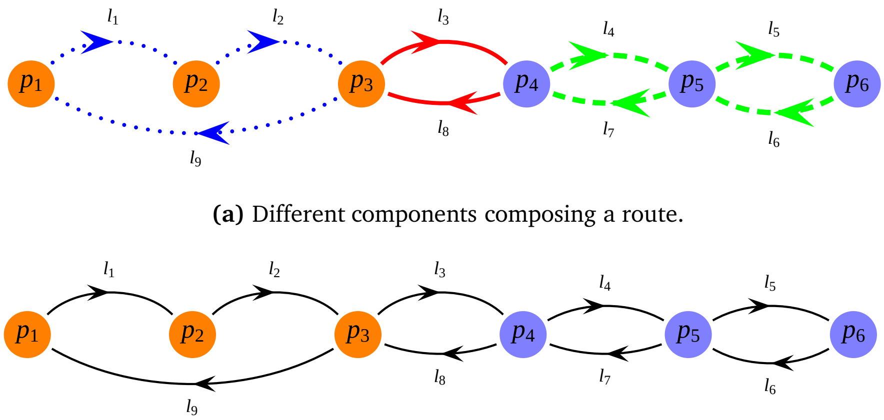
In liner shipping service networks, three main patterns may be observed on a service route: 1) *pendulum* segments (Figure 1a), which are in the form of direct calls between two major ports, 2) *butterfly* segments (Figure 1b), in which certain ports are called at twice, and 3) *cyclic* segments (Figure 1c), in which every port is visited exactly once on a route and which are composed of more than two ports. In practice, a route can be of a general form, as one can observe in Figure 2.

Designing a liner shipping network from scratch is a complex process. Here, neither the total number of routes in an optimal design nor the combination and number of segments of each category on every route are known in advance. However, in reality one can exploit the existing knowledge of real practice to preprocess and identify: 1) the ports that must/must not be served by a main route, 2) the ones that must not be served unless by feeder services, 3) the direct links that are unlikely to be established (e.g. a call of a medium/mega-sized vessel to Greenland or Iceland, a normal call to a port in Libya before heading towards a U.S. port or a port call to an Israeli port before calling at Beirut in Lebanon or Tartous in Syria) and 4) the ports that cannot be called at unless using certain vessel classes (e.g. due to the draught limit, port efficiency issues, etc.).

1.1. Context and Contribution

This work has been motivated by an industrial project that we carried out for a major (above the mid-size) liner shipping company to design a decision support system for analysing the operation. The results

[1] www.ciw.drewry.co.uk



(a) Different components composing a route.

(b) A general service route structure.

Figure 2: A general route structure and fundamental cycles.

were validated through several months (over two years of CPU time) of computational experiments.

This work contributes in the following ways. It proposes the first integrated modelling framework and scalable algorithmic approach that can fully (with a practically accepted level of details) characterize the structure of an optimal operational network for a liner service provider. It allows liners to analyse their network in different scenarios (entire redesign or incremental and zonal redesign), such as fixing part of their network and analysing the remaining part. From the methodological point of view, it proposes a solution-method-driven modelling framework that can integrate separately designed models into a single optimization problem. More importantly, it extends the Benders decomposition to an exact method that can accommodate experts' knowledge.

1.2. Literature Review

Over the past few years, this research area has become very active. However, the connection with the computational operations research is still rather weak. In the following we review some of the closely related work in the literature. We first review the articles dealing purely with the network design and deployment decision followed by those that also take into account empty repositioning feature.

Huang et al. (2015)) addressed the problem of fleet deployment and empty repositioning. Although it is said to be a liner service network design problem, they assumed that the routes are precisely known in advance and one only decides on whether or not to deploy a vessel on a given route, thereby identifying whether or not it becomes part of the operational network. Here, no decision is made on the structure of a route. They proposed an MIP model and used CPLEX to solve it. Gelareh and Pisinger (2011) studied the problems of fleet deployment, network design and hub location. They proposed a mathematical model and a Benders decomposition algorithm. Their network structure is based on hub-and-spoke structures. However, no empty repositioning is taken into account, and the network structure is not of a general form. They solved rather small instances to optimality.

Reinhardt and Pisinger (2012) proposed another model for the liner shipping network design, taking into account the cost of transhipment, a heterogeneous fleet, route-dependent capacities and butterfly routes. Again no empty repositioning is taken into account. They proposed a branch-and-cut approach to solve instances of the problem. Shintani et al. (2007) addressed a network design and empty repositioning problems. They dealt with a single route, and the route is identified by solving a knapsack problem in a two-stage modelling framework. Fleet deployment and empty repositioning are considered in this modelling framework, and a genetic algorithm is applied to solve the problem.

Agarwal and Ergun (2008) studied a space-time network model and routing. Other features of this work include the determination of service frequency, capacity allocation and transhipment. All possible routes are known in advance, and empty containers are not taken into account. Meng and Wang (2011) dealt with the liner shipping service network design problem with combined hub-and-spoke and multi-port-calling operations as well as empty container repositioning. Their network design problem again assumed that the shipping routes –composed of *segments*– are all known in advance. They used CPLEX for solving instances of the problem. Wang and Meng (2012) studied liner ship fleet deployment with container transhipment operations. They proposed a mixed-integer linear programming model for container transhipment operations without explicitly defining any container transhipment variables. Plum et al. (2014) proposed a service network design model. The empty repositioning has not been taken into account, the computational experiments are rather limited and they used CPLEX to solve the model.

Imai et al. (2009) dealt with two typical service networks with different ship sizes: multi-port calling by conventional-sized ships and hub-and-spoke operation by mega-ships. The solution process is composed of two phases: a service network design and container distribution. Fagerholt et al. (2009) presented a new model for a fleet deployment problem in liner shipping as well as a multi-start local search heuristic to solve the problem. The proposed heuristic is able to produce, within a few minutes, high-quality solutions to a real planning problem with more than 55 vessels and 150 voyages over a planning horizon of 4-6 months. Gelareh and Meng (2010) proposed a mixed-integer non-linear programming model for a short-term fleet deployment problem. They linearized their model and solved it using CPLEX.

Chang et al. (2008) studied substitution between empty containers of different types in an attempt to reduce the cost of empty container interchange. They proposed a heuristic method to yield an integer solution relatively fast. Chuang et al. (2010) assumed that the demand in container shipping is not crisp, rather uncertain, and can be modelled in a fuzzy environment. They used a genetic algorithm to solve instances of the problem. Dang et al. (2012) dealt with the problem of positioning empty containers in a port area with multiple depots. They sought optimal policies corresponding to different methods of inland positioning to minimize the expected total costs. They used a genetic-like method to solve this problem.

Shen and Khoong (1995) used network optimization techniques for multi-period planning of empty container distribution. This work presented a decision support system, which is also capable of recommending efficient leasing strategies. Dong and Song (2009) considered a joint container fleet sizing and empty container repositioning problem in multi-vessel, multi-port and multi-voyage shipping systems with dynamic, uncertain and imbalanced customer demand environments. They minimized the expected total costs, including inventory holding, loading and unloading, transportation, repositioning and lost-sale penalty costs. They were all then integrated into a simulation-based optimization tool equipped with genetic and evolutionary algorithms.

There are also other related contributions, such as that by Brouer et al. (2013) who presented a general overview of the problem and a rather simplified mathematical model. Song and Dong (2013) dealt with a single-liner long-haul service route design problem including route structure design, ship deployment and empty container repositioning. Although they proposed a symbolic mathematical model, they resorted to a three-stage heuristic method as a solution method.

Huang et al. (2015) proposed a model for liner service network design and fleet deployment with empty container repositioning. However, as far as the network design part is concerned, it only chooses the routes from among a set of existing ones, and no particular solution algorithm was proposed. The model is an especially aggregated one and contains fewer details. Wang et al. (2015) proposed an optimal container liner shipping network alteration problem based on the idea of a 'segment', which is a sequence of legs from a head port to a tail port that are visited by the same type of ship more than once in the existing shipping network. This was again an aggregation approach and a type of network coarsening to reduce the computational intractability and did not explore the entire solution space. The authors managed to solve an instance with 46 port and 11 vessels.

Among the work dealing with empty repositioning, one can refer to the following. Song and Dong (2012) presented a model with operational-level details. They proposed a seven-index integer programming model that is practically impossible to solve using any of the existing MIP solvers and given the current hardware. They suggested two MIP-based heuristics. Brouer et al. (2011) proposed a cargo allocation model considering the empty repositioning of containers and used LP relaxation with delayed column generation. The feasible solutions are obtained by rounding. Braekers et al. (2011) dealt with the challenges involved in managing empty container movements at multiple planning levels. They described the problem in detail and elaborated on the opportunities for reducing the costs of empty container movements.

Interested readers are also encouraged to refer to the recent survey on network optimization in liner shipping by Tran and Haasis (2015), the collection of interesting contributions made by Lee and Meng (2015) and the survey on the routing and scheduling aspects by Meng et al. (2013).

Table 1: Some very closely related and comparable contributions in the literature.

Reference	Decision			Solution approach	Feature
	network design	fleet deployment	empty repositioning		
This work	✓	✓	✓	Benders decomposition	transhipment
Shintani et al. (2007)	×	✓	✓	-	single route
Agarwal and Ergun (2008)	✓	×	×	CPLEX	service frequency, capacity allocation and transshipment
Chang et al. (2008)	×	×	✓	heuristic	×
Dong and Song (2009)	×	×	✓	simulation-based optimization	inventory, uncertainty, multi-vessel, multi-port and multi-voyage
Imai et al. (2009)	✓	×	×	×	multi-port calling by conventional ship size and hub-and-spoke by mega-ship
Fagerholt et al. (2009)	×	✓	×	multi-start local search	×
Gelareh and Meng (2009)	×	✓	×	CPLEX	×
Meng and Wang (2011)	✓	×	✓	CPLEX	hub-and-spoke and multi-port-calling
Gelareh and Pisinger (2011)	✓	✓	×	Benders Decomposition	transshipment
Reinhardt and Pisinger (2012)	✓	✓	×	branch and cut	transshipment
Dang et al. (2012)	×	✓	✓	genetic-like method	fuzziness
Wang and Meng (2012)	×	✓	×	×	transshipment
Huang et al. (2015)	existing	✓	✓	CPLEX	×
Song and Dong (2013)	single service	✓	✓	heuristic	×

Table 1 represents a synthesis of the very closely related works in the literature. To the best of our knowledge and as shown in Table 1, the literature is unaware of a holistic mathematical modelling approach of which the solution represents a global network composed of a number of service routes and their design (sequence of legs of calls) in addition to the vessel allocation, transhipment, fleet deployment and flow (laden and empty) routing decision.

The rest of this paper is organized as follows. In section 2 the problem and the assumptions are described. In section 3 the mathematical modelling framework as well as several classes of valid inequalities are presented, and in section 4 the solution method, separation algorithms and so on are elaborated. In section 5 we report the parameters and results of our extensive computational experiments. Finally, in section 6 a summary and conclusion together with some propositions for further research directions are provided.

2. Problem Statement

Let $G = (\mathcal{P}, \mathcal{L})$ be a graph where $\mathcal{P}$ represents the set of nodes corresponding to the ports to be served and $\mathcal{L}$ be the set of potential arcs corresponding to a set of *legs of calls* on the existing service network.

A *(service) route* is composed of an ordered set (multiplet) of port calls $(i_0, i_1, \ldots, i_p, i_{p+1}, \ldots, i_{m_r}, i_0)$ starting from and ending at port i_0 on the route. Leg j, say l_j, refers to an arc from port i_j (tail) to port i_{j+1} (head).

A port that lies on at least one service route is referred to as a *route port*. Every port either lies on at least one service route or it is an *isolated port* that is allocated to another port via which the origin-destination demands are transported (e.g. ports #12, #15, #17, #23 and #27 in Figure 3). Such an allocation is represented by a dotted arc from the isolated port towards one single route port (allocation arc). The transportation service between an isolated port and a route port is considered as being outsourced and is not subject to any fleet allocation.

We define a *risk factor* that is a (monetary equivalent) value associated with every potential arc $(i, j) \in \mathcal{L}$, indicating the degree of reluctance (or emphasis) of experts (management) to include certain legs of calls in the service network —regardless of the service route. This reluctance may stem from several sources, including political, strategic and business-related ones or any other confidential and sensitive context. Moreover,

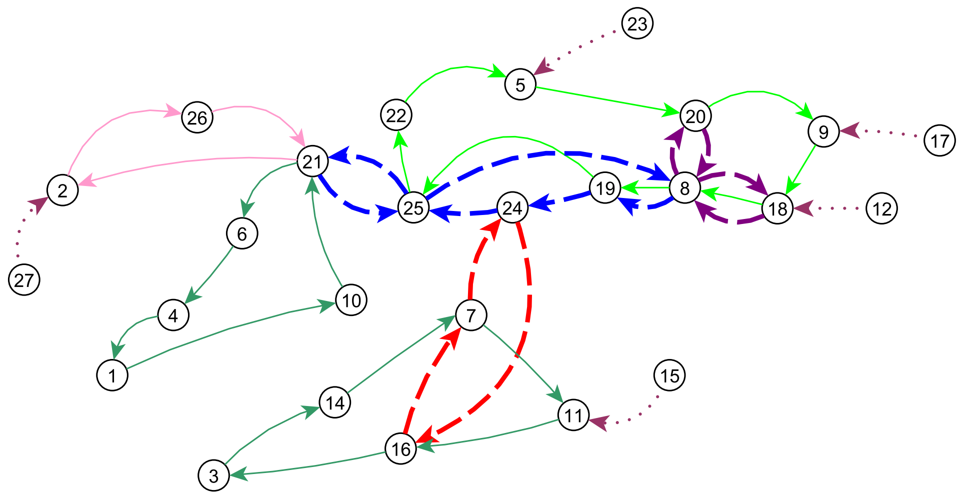


Figure 3: An artificial example of a feasible solution network design. Here, the tick dashed arcs belong to the main routes (distinguished by colours) and the thin solid arcs belong to the feeder routes. The main route-level network is a connected network. Every port along such a route is a main port where transhipment takes place; therefore, every such port must be on at least two routes (whether a feeder or the main route). The dotted arcs correspond to the allocation arcs for the isolated ports.

a penalty cost is associated with the allocation of an isolated port to every other port. This cost structure is a quantification of qualitative measures associated with the legs and is extracted from experts' knowledge. A transit time is associated with every arc (leg or allocation), which represents the expected transit time on that leg of calls in the current market.

There are three kinds of service route. 1) The main service routes can be no longer than α_1 weeks. Each port on such a route is a main port and must be served by at least two service routes (at least one of them must be a main route). 2) The feeder services are no longer than $\alpha_2 < \alpha_1$ weeks' transit time.[2] 3) The services connect *isolated* ports to route ports. As mentioned previously, the demands and supplies of such outsourced services are delivered by a second company without having to know, ourselves, the details of that operation.

Often in practice an isolated port is a port that does not send/receive a significant quantity of containers and thus it is less profitable (economically or even politically) for a liner to call at it (e.g. because the turnaround time and the idle time costs exceed the profit).

We also use the term *string*, $s = ((i_0, i_1, \ldots, i_p, i_{p+1}, \ldots, i_{m_r}, i_0), v, f)$, a triplet referring to a service route, the vessel class and the service frequency on that service route. Given that the current exercise in liner shipping is based on weekly departures, the length of a route is the key factor that determines the number of vessels operating on a given service route.

An LSP operates a heterogeneous fleet of vessels from $|\mathcal{V}|$ distinct containership classes. The cost of operating a route is a cost associated with the string as a set (of legs and their political/economic characteristics and vessels with their cost structures) rather than individual elements. This cost is a function of the transit time of the legs in the set, the port handling cost per unit of container, varying insurance regimes on different legs of calls, security considerations, various regulations of operation in different territorial waters (e.g. speeds limits) that may impose some route-specific bunker consumption patterns and port call, bridge and

[2] In practice, α_1 is often considered as 12 weeks while α_2 is known to be often around 3 weeks.

canal fees, to name a few. Moreover, some experts' knowledge (political, strategic and economic factors) interferes when the operational cost of a route is being calculated. However, in the latter case, often no closed-form formulation is known for most of these factors.

Hence, the operational cost of a route $r \in 2^{\mathcal{L}}$ operating a set of vessels of class $v \in \mathcal{V}$ (to maintain a weekly frequency) can be defined as:

$$C_r^v : \quad (2^{\mathcal{L}}, v) \to \mathbb{R} \tag{1}$$

where $\mathcal{L}$ is the set of potential legs and $2^{\mathcal{L}}$ represents its power set, $\wp(\mathcal{L})$. Moreover, our empirical studies show that for every $X \in \wp(\mathcal{L}), x_1, x_2 \in \mathcal{L}/\ X$ we have $C_r^v(X \cup \{x_1\}) + C_r^v(X \cup \{x_2\}) \geq C_r^v(X \cup \{x_1, x_2\}) + C_r^v(X)$. Therefore, it behaves akin to a submodular function and as a consequence it has all the properties of a convex function.

In addition to the operational cost of route r served by vessel $v \in \mathcal{V}$, C_r^v, the container handling cost (including loading, discharging and transhipment for both empty and laden containers) is another component of the string operating cost. While the origin and destination of demands are inputs (the exogenous part) of the problem and do not include any flexibility, the choice of transhipment port and the number of transhipments along every origin-destination (O-D) path are an endogenous part of the problem. As a result, one may expect that the operators may always find a port p where the transhipment cost is cheaper than the loading plus discharging cost.

In this study we assume that the liner company owns the entire fleet operating on its service routes and that charter costs and so on do not apply here. We further assume that the LSP is allowed to fulfil any portion of an O-D demand. However, the liner currently has enough capacity to transport its current demand volume.

As it can be observed in Figure 3, every service route (distinguished by colour) is composed of a combination of the building blocks shown in Figure 1, namely pendulum, butterfly and cyclic (sub-)routes. The service routes are constructed by chaining together the building blocks based on some qualitative criteria. This relies chiefly on the experts' knowledge, taking into account some political/economic considerations.

In the following we list the assumptions used throughout this work. For some political reasons, the liner company is not keen for more than two non-butterfly feeder routes to pass through a feeder port. It also does not want a non-transhipment port to be the middle port on a butterfly route. If $i- > j- > k- > l$ is a segment shared between route 1 and route 2, then two vessels spend time at i, j, k and l. This is not an ideal situation, and two feeder routes sharing more than two consecutive legs of calls in the same direction must be avoided. While the idea of a dedicated container terminal at a main port is very tempting, the company does not want a main port to be at the intersection of more than three main routes and prefers to have some diversification. In addition, there are at least two routes (feeder or main) that pass by a hub port (i.e. a main port must be on the intersection of at least two main services or at least one feeder service route plus one main one), otherwise no transhipment takes place at a main port, which is not acceptable for the company (if it calls at a main port, it is expected that at least one of its feeder services operates there and that it should be benefiting from the local feeder market).

The problem can be described formally as in the following: *Given all the aforementioned assumptions and information, one seeks to determine strings (routes described by the legs of calls, vessel class and quantity) by respecting the round-trip transit times, identifying isolated ports, the routing and transhipment of laden containers and the repositioning of empty containers. The objective is to maximize the profit that accounts for the difference between the revenue generated by the volume of transported demands and the cost of operating strings (as well as the cost of loading/unloading/transhipment operations for laden and empty containers at different ports) as well as the cost of the risks associated with each arc. The market transit time of routes must be multiples of weeks to maintain a weekly frequency.*

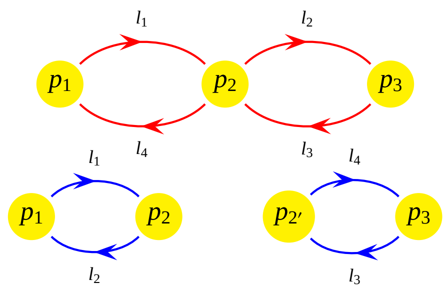


Figure 4: Decomposition of a butterfly route into two feasible pendulum routes.

To this end we use *port* and *node*; *hub port* and *main port*; and *spoke port*, *regional port* and *feeder port* interchangeably, and by *main route* we refer to a route composed of only hub/main ports; any other route is referred to as a *feeder route*.

3. Mixed-Integer Linear Programming Formulations

Given that the cost structure for routes is a set function, a polynomial formulation of this problem seems to be non-trivial. Therefore, instead of proposing a compact formulation, our modelling framework relies on a Benders reformulation of the problem. The master problem (MP) is a *route generation problem (RGP)* with the aim of proposing network structures (in the form of the components shown in Figure 1), while the subproblem (SP) solves a *fleet deployment and flow routing problem (FDFRP)*.

The Benders decomposition offers great flexibility in modelling such kinds of integrated problems with separate structures by allowing an efficient exchange of information between the MP and the SP(s) through parameterizing the SP with realized variables of the MP.

Here, in this problem the solution to the MP, rather than being a complete description of routes, is a set of different feasible components (butterfly, pendulum and cycles). Such components can be separate routes or chained together to create feasible routes (by exploiting experts' knowledge). However, we make sure that such a mapping –resulted from experts' knowledge– from the set of components to the set of routes is a one-to-one mapping. This guarantees that no two sets of distinct components result in the same set of feasible routes.

The expert knowledge is a database of exactly 816 rules, which are mostly confidential, indicating several different instance-based and generic rules, such as the infeasibility of some sequential calls, the implications of certain sequences of calls due to strategic, business, cargo type and several other criteria. Among such rules, we also have ones that try to combine components and obtain better approximation of multiple-week sailing times. Some such rules, in fact, even help to preprocess and fix quite a substantial part of the network or other elements of the service.

As mentioned earlier, we are seeking routes with transit times of a multiple of weeks. As the essential components are basically the *pendulum* and *cyclic* components, it suffices to generate pendulum and cyclic routes, while butterfly components can be obtained by chaining two neighbouring pendulum components.

3.1. Route Generation Problem (RGP)

The parameters are listed in Table 2:

The decision variables are defined as follows: y^f_{ij} is 1 if the arc (i, j) belongs to a feeder service route and 0 otherwise y^h_{ij} is 1 if arc (i, j) belongs to at least one main service route and 0 otherwise. We define h^j_i to be 1 if node i is a main port and is served by at least j main service routes and 0 otherwise.

τ_{ij}:	the usual market transit time associated with the arc (i, j), which includes an approximation of the turnaround time at both end-points,
R_f^l:	the feeder route minimum transit time,
R_f^u:	the feeder route maximum transit time,
R_h^l:	the main route minimum voyage time,
R_h^u:	the main route maximum voyage time,
ζ_{ij}^f: :	the estimated monetary risk factor associated with the arc (i, j),
ψ_{ij}^f: :	the penalty cost associated with the allocation of an isolated port i to a route port j.

Table 2: Model parameters for RGP.

$t_{ij}, j \neq i$ is 1, if node i is an isolated port that is not on any service route (whether feeder or main) but will be served by (outsourced) direct service(s) via route port j and 0 otherwise. In addition, t_{ii}^j is 1 if i is a port that is on at least j service routes (feeder, main or both).

We define For $S \subset \mathcal{P}$, let define $\delta(S)^+ = \{(i, j)| i \in S, j \in V/\ S, \}$, $\delta(S)^- = \{(i, j)| j \in S, i \in V/\ S\}$ and $\gamma(S) = \{(i, j)| i \in S, j \in S, j \neq i\}$.

The master problem (MP), which is a Route Generation Problem (RGP) only generates feasible pendulum and cyclic route components. $\eta \geq 0$ represents the maximum profit that can be generated from FDFRP as subproblem.

$$(RGP) \quad \max \eta - \sum_{i,j \neq i} \zeta(i, j)\, y_{ij}^f - \sum_{i,j \neq i} \zeta(i, j)\, y_{ij}^h - \sum_{i,j \neq i} \psi(i, j)\, t_{ij} \tag{2}$$

$s.\ t.$

$$\sum_{j \neq i} y_{ij}^h = h_i^1 + h_i^2 + h_i^3, \qquad \forall i, \tag{3}$$

$$\sum_{j \neq i} y_{ji}^h = h_i^1 + h_i^2 + h_i^3, \qquad \forall i, \tag{4}$$

$$\sum_{j \neq i} y_{ji}^h = \sum_{j \neq i} y_{ji}^h, \qquad \forall i, \tag{5}$$

$$h_k^2 \leq h_k^1, \qquad \forall k, \tag{6}$$

$$h_k^3 \leq h_k^2, \qquad \forall k, \tag{7}$$

$$t_{ii}^1 \leq \sum_{j \neq i} y_{ij}^f \leq t_{ii}^1 + t_{ii}^2, \qquad \forall i, \tag{8}$$

$$\sum_{j \neq i} y_{ij}^f = \sum_{j \neq i} y_{ji}^f, \qquad \forall i, \tag{9}$$

$$t_{ii}^1 \leq \sum_{j \neq i} y_{ji}^f \leq t_{ii}^1 + t_{ii}^2, \qquad \forall i, \tag{10}$$

$$t_{ii}^2 \leq t_{ii}^1, \qquad \forall i, \tag{11}$$

$$y_{ij}^f \leq t_{ii}^1, \qquad \forall i, j \neq i, \tag{12}$$

$$y_{ij}^f \leq t_{jj}^1, \qquad \forall i, j \neq i, \tag{13}$$

$$y_{ij}^h \leq h_i^1, \qquad \forall i, j \neq i, \tag{14}$$

$$y^h_{ij} \leq h^1_j, \qquad \forall i, j \neq i, \qquad (15)$$
$$t_{ij} \leq t^1_{jj}, \qquad \forall i, j \neq i, \qquad (16)$$
$$h^1_i \leq t^1_{ii}, \qquad \forall i, \qquad (17)$$
$$\sum_{j \neq i} t_{ij} + t^1_{ii} = 1, \qquad \forall i, \qquad (18)$$
$$\eta \leq \Phi(y^f_{ij}, y^h_{ij}), \qquad \text{extreme points}, \qquad (19)$$
$$y^h, t^1, t^2, y^f \in \mathbb{B}^{|\mathcal{P}|^2}, \qquad (20)$$
$$h^1, h^2, h^3 \in \mathbb{B}^{|\mathcal{P}|}, \eta \geq 0. \qquad (21)$$

Constraints (3)-(4) ensure that a main port can be at the intersection of one, two or three main routes. Constraints (5) ensure that at every main port, the number of arriving and departing arcs are equal.

Constraints (6)(equiv. (7)) make sure that the second (third) main route may pass through a node only if the first (second) one already exists. Constraints (8)-(10) ensure that at least one and at most two feeder routes pass through every non-isolated port. Constraints (11) ensure that a port that is called at on exactly two feeder routes is a non-isolated port. Constraints (12)- (13) make sure that both end-points of a feeder leg are non-isolated feeder ports. Constraints (14)-(15) ensure that both end-points of a main route arc must be main ports. An isolated port can only be allocated to a route port. Constraints (16) guarantee this. A port can be a main port if it is not an isolated feeder port as in constraints (17). Every port is either a route port or an isolated port allocated to a route port as stated in constraints (18).
Constraints (19) are the Benders cuts where $\Phi(y^f_{ij}, y^h_{ij})$ is a convex function representing the maximum objective function form FDFRP.

We still need to make sure that the total length of a route remains an 'almost' integer multiple of weeks within $[R^l_f, R^u_f]$ for a feeder service and within $[R^l_h, R^u_h]$ for a main service.

When working with real data, normally the condition of '*being a multiple of weeks*' can be fulfilled automatically given the structure of the transit time matrix. In general cases, we use '*almost* a multiple of weeks' because some minor adjustments can be msde by stretching (slow steaming) or shrinking the transit time at the operational level and maintaining the weekly services.

The model is completed once the *route transit time* upper and lower bound constraints and the connectivity constraints are added to (3)-(21).

3.1.1. Route bound inequalities

Let Δ be a *realistic* amount of time by which one can stretch or shrink the transit time of a component to make a multiple of weeks. We also refer to such components as *feasible components*, as they can be managed at the operational level. We establish a support graph $G = (\mathcal{P}, \mathcal{L})$ corresponding to a (fractional) solution to the RGP and identify all the cycles with longer and shorter transit times than the expected lower and upper bounds for each component type. Let us define $\overline{G}^h_r$ and $\underline{G}^h_r$ ($\overline{G}^f_r$ and $\underline{G}^f_r$) as the sets of upper-violated and lower-violated components for main (feeder) routes with a violation more than Δ. Two types of cuts can be added:

1. *Knapsack constraints*: a sufficient number of arcs must be removed from an infeasible component;

$$\sum_{(i,j) \in \mathcal{L} \cap \omega} t_{ij} y^h_{ij} \leq R^u_h, \qquad \forall \omega \in \overline{G}^h_r \qquad (22)$$
$$\sum_{(i,j) \in \mathcal{L} \cap \omega} t_{ij} y^f_{ij} \leq R^u_f, \qquad \forall \omega \in \overline{G}^f_r \qquad (23)$$

2. *Combinatorial cuts*: at least one arc from among the arcs on every such route must be removed:

$$\sum_{(i,j)\in\mathcal{L}\cap\omega} y^h_{ij} \le |\omega| - 1, \qquad \forall \omega \in \overline{G}^h_r \tag{24}$$

$$\sum_{(i,j)\in\mathcal{L}\cap\omega} y^f_{ij} \le |\omega| - 1, \qquad \forall \omega \in \overline{G}^f_r \tag{25}$$

On the other hand, when a component is infeasible with respect to the lower bound, only combinatorial cuts can be added.

$$\sum_{(i,j)\in\mathcal{L}\cap\omega} y^h_{ij} \le |\omega| - 1, \qquad \forall \omega \in \underline{G}^h_r \tag{26}$$

$$\sum_{(i,j)\in\mathcal{L}\cap\omega} y^f_{ij} \le |\omega| - 1, \qquad \forall \omega \in \underline{G}^f_r \tag{27}$$

3.1.2. Subtour elimination constraints (SECs) on the main routes

Every cut size in a feasible directed main sub-network has the capacity at least equal to 1 as the main route subnetwork is a connected one. Let $S \subset \mathcal{P}$ and $S' = \mathcal{P} / S$:

$$\sum_{k\in S, l\in S'} y^h_{kl} \ge (h^1_{ii} + h^1_{jj} - 1), \qquad \forall i \in S, j \in S' \tag{28}$$

$$\sum_{k\in S, l\in S'} y^h_{lk} \ge (h^1_{ii} + h^1_{jj} - 1), \qquad \forall i \in S, j \in S' \tag{29}$$

3.1.3. Connectivity of the whole network

Given that the whole network (including the allocation arcs representing an outsourced service between an isolated port and a route port) is a connected directed graph composed of a single components, any cut size has a capacity of at least one (at least one outgoing and one incoming arc).

$$\sum_{i\in S, j\in\mathcal{P}/S} \left(y^f_{ij} + y^h_{ij} + t_{ij} + t_{ji}\right) \ge 1, \qquad \forall S \subset \mathcal{P} \tag{30}$$

$$\sum_{i\in S, j\in\mathcal{P}/S} \left(y^f_{ji} + y^h_{ji} + t_{ji} + t_{ij}\right) \ge 1, \qquad \forall S \subset \mathcal{P} \tag{31}$$

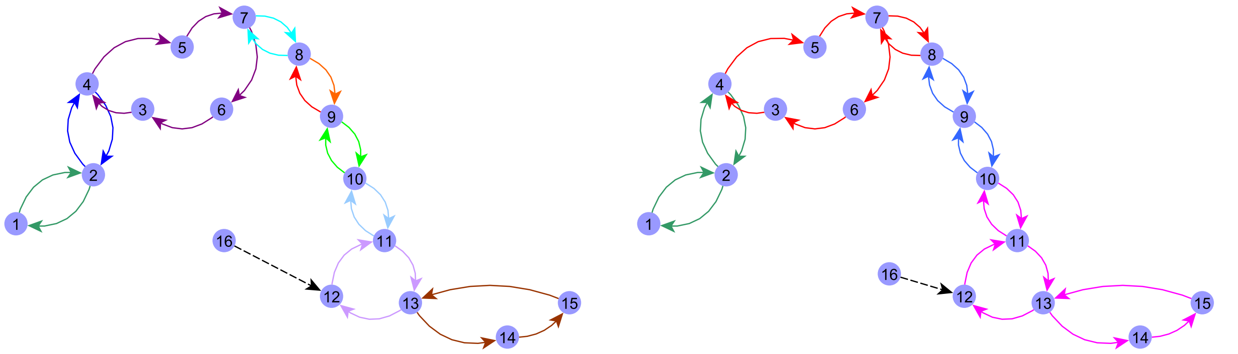
Constraints (22)-(31) complete the RGP model.

3.2. Fleet deployment and flow routing problem (FDFRP)

Once route components, C, are known, we then construct the routes themselves with respect to those components. Experts' knowledge intervene at this stage and is represented as a *bijection* $f : C \longmapsto \mathcal{R}$, mapping a set of components to a set of routes. Such a mapping needs to be a bijection because we expect that no two distinct sets of components being mapped to the same set of routes and two different sets of routes should not be un-chained to the same set of components.

Figure 5a depicts a set of components covering all the nodes plus one isolated port allocated to a route port. These components are distinguished by different colours. In Figure 5b one can observe that the experts' knowledge mapping function has resulted in chaining the components to construct the four routes in this figure.

Once the structure of the routes is known, the total transit time on every route can be calculated. Since we must ensure a weekly frequency, the number of deployed containerships can be deduced from the total



(a) The set of components covering all the nodes —except isolated node #16, which is allocated to node #12.

(b) The set of routes covering all the nodes —except isolated node #16, which is allocated to node #12.

Figure 5: An example of transforming (chaining) a set of components to generate a set of routes.

transit time of the route. Hence, one can calculate a portion of the total cost that accounts for the flow-independent costs, that is, the route-specific operating costs composed of the cost of deployed vessels of each class.

The subproblem is the fleet deployment and flow routing problem, which takes as input the routes proposed by the aforementioned mapping function and decides on the type of vessel as well as the routing of the flow. In the FDFRP one decides on the fraction of O-D flows to transport, the path that flows traverse between every O-D and the transhipment pattern, the type of vessel (capacity) to be allocated to every service route and the way in which empty containers are repositioned/distributed across the network. The goal is to find the optimal pattern with the objective of maximizing the profit after deducting the costs incurred. Wang and Meng (2012) and Wang et al. (2014) proposed a mathematical model for similar problems. The parameters are listed in Table 3.

$\mathcal{R}$:	the set of service routes corresponding to the current solution to the RGP,
$\mathcal{V}$:	the set of vessel classes,
L_p^{exp}:	the loading cost of laden containers at port p,
L_p^{imp}:	the unloading cost of laden containers at port p,
L_p^{tr}:	the transhipment cost of laden containers at port p,
E_p^{exp}:	the loading cost of empty containers at port p,
E_p^{imp}:	the unloading cost of empty containers at port p,
E_p^{tr}:	the transhipment cost of empty containers at port p,
C_v^r:	the operational cost of the number of vessels of class v deployed on route r,
S_p:	the supply of port p, $\sum_{j \in \mathcal{P}} d_{pj}$,
D_p:	the demand of port p, $\sum_{j \in \mathcal{P}} d_{jp}$,
Γ_v:	the capacity of vessel class v,
$\mathcal{M}$:	a sufficiently big positive value,
$\rho = [\rho_{ij}]$:	the O-D revenue per unit of flow,
$\mathcal{D} = [d_{ij}]$:	the O-D demand matrix.

Table 3: Model parameters for the FDFRP .

We assume that we are committed to offering a service on every route resulting from chaining the components proposed by the RGP. This means that there will be vessels deployed on every route proposed by the experts' knowledge. We further assume that the operator has sufficient capacity for its current O-D market demand.

The following decision variables are used in the FDFRP: x_r^v is equal to 1 if a set of vessels of class v is deployed on route r and 0 otherwise; $f_{orl} \geq 0$ represents the total number of laden containers with origin o traversing leg l of route r; $0 \leq w_{ij} \leq 1$ stands for the fraction of fulfilled O-D demand; $z_{orl}^{+} \geq 0$ stands for the total number of laden containers with origin port o loaded onto a ship visiting port of call l on route r; $z_{orl}^{-} \geq 0$ represents the total number of laden containers with origin port o unloaded from a ship visiting port of call l on route r; $z_p^{tr} \geq 0$ determines the total number of laden containers that are transhipped at port p; $e_{rl} \geq 0$ stands for the total number of empty containers transported on leg l of route r; $z_{rl}^{e\,+} \geq 0$ represents the total number of empty containers loaded onto a ship visiting port of call l on ship route r; $z_{rl}^{e\,-} \geq 0$ stands for the total number of empty containers unloaded from a ship visiting port of call l on route r; $z_p^{e\,+} \geq 0$ represents the total number of empty containers that are repositioned from port p; $z_p^{e\,-} \geq 0$ stands for the total number of empty containers that are repositioned to port p; and $e_p^{tr} \geq 0$ represents the total number of empty containers that are transhipped at port p.

We define $\overline{\mathbf{y}} = \max\{\mathbf{y}^f, \mathbf{y}^h\}$, where y_l is the characteristic function indicating whether leg l appears in any component from the RGP-proposed components, where $l = (p, q)$. N^r represents the number of vessels required to maintain a weekly frequency on the service route, $r \in \mathcal{R}$, of the current incumbent solution to the RGP. The FDFRP is modelled as follows:

$$(FDFRP) \quad \max \sum_{(i,j)\in\mathcal{W}} \rho_{ij} w_{ij} d_{ij} - \sum_{r\in\mathcal{R}} \sum_{v\in\mathcal{V}} C_v^r x_r^v$$
$$- \sum_{p\in\mathcal{P}} \left(L_p^{+} \sum_{(p,d)\in\mathcal{W}} w^{pd} d_{pd} + L_p^{tr} z_p^{tr} + L_p^{-} \sum_{(o,p)\in\mathcal{W}} w^{op} d_{op} \right)$$
$$- \sum_{p\in\mathcal{P}} \left(E_p^{+} z_p^{e\,+} + E_p^{tr} e_p^{tr} + E_p^{-} z_p^{e\,-} \right) \tag{32}$$

$s.\,t.$

$$f_{r\ l-1}^{o} + z_{orl}^{+} = f_{r\ l}^{o} + z_{orl}^{-} \qquad \forall r \in \mathcal{R}, l \in \mathcal{L}_r, o \in \mathcal{P} \tag{33}$$

$$\sum_{r\in\mathcal{R}} \sum_{l\in\mathcal{L}_r, p_{rl}=p} \sum_{o\in P} S_o\, z_{orl}^{+} = z_p^{tr} + \sum_{d} w^{pd} d_{pd} \qquad \forall p \in \mathcal{P} \tag{34}$$

$$\sum_{r\in\mathcal{R}} \sum_{l\in\mathcal{L}_r, p_{rl}=p} \sum_{o\in P} S_o\, z_{orl}^{-} = z_p^{tr} + \sum_{o} w^{op} d_{op} \qquad \forall p \in \mathcal{P} \tag{35}$$

$$\sum_{r\in\mathcal{R}} \sum_{l\in\mathcal{L}_r, p_{rl}=p} S_p \left(z_{prl}^{+} - z_{prl}^{-} \right) = \sum_{d} w_{pd} d_{pd}, \qquad \forall p \in \mathcal{P} \tag{36}$$

$$\sum_{r\in\mathcal{R}} \sum_{l\in\mathcal{L}_r, p_{rl}=p} S_o \left(z_{orl}^{+} - z_{orl}^{-} \right) = -w_{op} d_{op} \qquad \forall o, p \in \mathcal{P} : p \neq o \tag{37}$$

$$e_{r\ l-1} + z_{rl}^{e\,+} = e_{r\ l} + z_{rl}^{e\,-} \qquad \forall r \in \mathcal{R}, l \in \mathcal{L}_r \tag{38}$$

$$\sum_{r\in\mathcal{R}} \sum_{l\in\mathcal{L}_r, p_{rl}=p} \left(z_{rl}^{e\,+} - z_{rl}^{e\,-} \right) = \sum_{j} d_{jp} w^{jp} - \sum_{j} d_{pj} w^{pj} \qquad \forall p \in \mathcal{P} \tag{39}$$

$$\sum_{r\in\mathcal{R}} \sum_{l\in\mathcal{L}_r, p_{rl}=p} z_{rl}^{e\,+} = e_p^{tr} + z_p^{e\,+} \qquad \forall p \in \mathcal{P} \tag{40}$$

$$\sum_{r\in\mathcal{R}} \sum_{l\in\mathcal{L}_r, p_{rl}=p} z_{rl}^{e\,-} = e_p^{tr} + z_p^{e\,-} \qquad p \in \mathcal{P} \tag{41}$$

$$\sum_{o\in\mathcal{P}} f_{orl} + e_{rl} \leq \sum_{v\in V} N^r \Gamma_v x_r^v \qquad \forall r \in \mathcal{R}, l \in \mathcal{L}_r \tag{42}$$

$$\sum_{v\in\mathcal{V}} x_r^v = 1 \qquad \forall r \in \mathcal{R} \qquad (43)$$

$$f_{r\ l}^o \leq \Delta_{i\in\mathcal{L}_r}\ \bar{y}_l \qquad \forall r \in \mathcal{R}, l \in \mathcal{L}_r, o \in \mathcal{P} \qquad (44)$$

$$e_{r\ l-1} \leq \mathcal{M}\Delta_{l\in\mathcal{L}_r}\ \bar{y}_l \qquad \forall r \in \mathcal{R}, l \in \mathcal{L}_r, o \in \mathcal{P} \qquad (45)$$

$$z_{orl}^+ \leq \Delta_{l\in\mathcal{L}_r}\ \bar{y}_l \qquad \forall r \in \mathcal{R}, l \in \mathcal{L}_r, o \in \mathcal{P} \qquad (46)$$

$$z_{orl}^- \leq \Delta_{l\in\mathcal{L}_r}\ \bar{y}_l \qquad \forall r \in \mathcal{R}, l \in \mathcal{L}_r, o \in \mathcal{P} \qquad (47)$$

$$z_{rl}^{e\ +} \leq \mathcal{M}\Delta_{l\in\mathcal{L}_r}\ \bar{y}_l \qquad \forall r \in \mathcal{R}, l \in \mathcal{L}_r, o \in \mathcal{P} \qquad (48)$$

$$z_{rl}^{e\ -} \leq \mathcal{M}\Delta_{l\in\mathcal{L}_r}\ \bar{y}_l \qquad \forall r \in \mathcal{R}, l \in \mathcal{L}_r, o \in \mathcal{P} \qquad (49)$$

$$z_{orl}^{tr} \leq \mathcal{M}\Delta_{l\in\mathcal{L}_r}\ \bar{y}_l \qquad \forall r \in \mathcal{R}, l \in \mathcal{L}_r, o \in \mathcal{P} \qquad (50)$$

$$e_{rl}^{tr} \leq \mathcal{M}\Delta_{l\in\mathcal{L}_r}\ \bar{y}_l \qquad \forall r \in \mathcal{R}, l \in \mathcal{L}_r, o \in \mathcal{P} \qquad (51)$$

$$x_r^v \in \{0, 1\} \qquad \forall r \in \mathcal{R}, v \in \mathcal{V}i \qquad (52)$$

$$z_{orl}^+, z_{orl}^-, f_{orl}, w_{od} \in [0, 1] \qquad \forall r \in \mathcal{R}, l \in \mathcal{L}_r, o, d \in \mathcal{P} \qquad (53)$$

$$z_{rl}^{e\ +}, z_{rl}^{e\ -}, e_{rl}^{tr}, e_{rl} \in \mathbb{R}^+ \qquad \forall r \in \mathcal{R}, l \in \mathcal{L}_r \qquad (54)$$

$$z_p^{e\ +}, e_p^{tr}, z_p^{e\ -}, z_p^{tr} \in \mathbb{R}^+ \qquad \forall p \in \mathcal{P}. \qquad (55)$$

The objective function (32) is a maximization one accounting for the profit calculated as the difference between the revenue generated by the volume of flow transported and the costs incurred by this operation including fleet deployment, loading, unloading and transhipment costs.

Constraints (33) are the classical flow conservation constraints for the laden containers. For every origin p and every port of call l on every route r, the volume of flow originating from port p at leg of call l is equal to the volume it had from the same origin on call $l-1$ of the same route r plus the volume (from the same origin) that it loaded (arrived via other routes) at call l minus the volume unloaded (transhipped via other routes) at port of call l.

Constraints (34) ensure that the total volume of flow loaded at port p is equal to the volume of transhipment at p plus its volume of supply. Constraints (35) state that the total volume of flow unloaded at port p is equal to the volume of transhipment at p plus the demand of p, which is fulfilled.

Constraints (36) guarantee that the total number of laden containers from port p loaded at this port is equal to the supply of port p while constraints (37) make sure that for any origin o and any other port, p, the flow balance of loaded and unloaded volumes equals to the volume of demand of p fulfilled by o.

Constraints (38) are analogous to constraints (33) for empty containers. However, as the empties are generic commodities, they do not need to be distinguished by their suppliers.

Constraints (39) guarantee that the difference between the loaded and unloaded quantity of containers at port p equals the imbalance of that port.

Constraints (40)-(41) are analogous to constraints (34)-(35) for empty containers.

On every route r, the total flows on leg l must respect the capacity of the vessels deployed on l. Constraints (42) guarantee this. Moreover, there must be one unique vessel class deployed on every generated route as per constraint (43).

Constraints (44)-(55) play a very important role in connecting our RGP to the FDFRP by verifying the compatibility between the flow variable and the existing arcs.

It must be noted that the two almost independent problems RGP and FDFRP are linked via constraints (44)-(51).

4. Branch, Cut and Benders Algorithm (BCB)

As mentioned earlier, our solution approach is based on the paradigm of the Benders decomposition. The Benders decomposition (Benders, 2005, 1962) is a primal decomposition method, which has proven to be an efficient method of dealing with large-scale MIP models in facility location/network design types of problems. The idea of the Benders decomposition relies on relaxing the complicating variables (e.g. those variables that are integers). It then exploits the primal/dual relationship to generate cuts, separate the solutions of the master problem and tighten the outer approximation, iteratively, until it achieves optimality.

If a solution to the Benders master problem is not a feasible (partial) solution to the problem as a whole, the generated cut(s) in the Benders approach is (are) a feasibility cut(s), otherwise we separate optimality Benders cut(s).

The Benders decomposition has been applied to similar problems for example by Gelareh and Nickel (2011) and Gelareh and Pisinger (2011). In particular, Gelareh and Nickel (2011) proposed replacing the natural choice of the master problem with an auxiliary model introduced earlier by Maculan et al. (2003) that guarantees connected subgraphs of a given set of nodes. On the other hand, Codato and Fischetti (2006) and Fischetti et al. (2009) proposed other approaches based on identifying the *minimal infeasible subsystem* to deal with infeasible solutions.

Several techniques have been proposed in the literature to improve the performance of the Benders algorithm: Magnanti and Wong (1981) dealt with degenerate subproblems and choosing the best solution among many dual solutions; Magnanti et al. (1986) addressed network design problems, in particular. Among many other contributions, we refer to Poojari and Beasley (2009), McDaniel and Devine (1977) and Sherali and Fraticelli (2002) for further reading.

Here, our master problem is the RGP and the subproblem is the FDFRP. However, there is a step that differentiates our method from the classical Benders decomposition scheme. In the classical Benders decomposition, a solution to the MP is directly used in the SP and there are no intermediate steps between the proposal by the MP and the solution and cut separation by the SP. A rather abstract representation can be seen as an MP that gives some solutions (with no repetition) and an *Oracle* that solves a convex SP and returns some primal/ dual information or cuts and cuts off the MP's optimal solution. We emphasize that, once a solution from the master problem has been cut off, it will no longer be proposed by the RGP as a candidate - hence, there is no repetition.

While, in the classical Benders decomposition, an exchange of information takes place during iterations of Benders (separation phases) between an MP (of any form as long as a solver is known for it) and one or more SPs (which only need to be convex or locally convex), our approach includes one intermediate step that incorporates experts' knowledge into the solution process: *inference rules based on experts' knowledge*. We introduce a new element, which is a one-to-one mapping. Such a function is an automated process supported by a database of rules and a learning mechanism allowing us to establish the previously mentioned *bijection*. This additional step (which can be seen as verifying some hard-to-express constraints on the fly and filter the solutions without using a closed-form inequality) transforms a set of components of the RGP (the MP) into a set of routes, which are the input to the FDFRP (the SP).

To solve the master problem efficiently, we apply a branch and cut by separating the exponential number of constraints (22)-(31) as well as other tightening valid inequalities alongside the Benders cuts. Therefore, the overall solution process can be seen as a *branch, cut and Benders (BCB)* algorithm. When solving the master problem, our approach is composed of exploring a decision tree, cuttings and a preprocessing phase for possibly fixing some variables and identifying instance-dependent valid inequalities in addition to the cutting plane technique.

It must be noted that two particular features make our approach different from the classical Benders approach. The first one is that the structure of routes resulting from chaining the RGP components is identified in the intermediate step between the solutions to the RGP and the FDFRP. The second point is that the subproblem is an MIP and we are deprived of a primal/dual relationship.

4.1. Preprocessing and Variable Fixing

When dealing with industrial applications, in certain data instances some particular patterns are present. Such information can often help to preprocess the problem instances and possibly to remove some variables and constraints. Among such information one can refer to the following examples, to name a few: 1) for an LSP$_\alpha$, a call at Singapore immediately after departing from Xiamen might be –for some business-related reason– inevitable westbound while calling at Hong Kong just before Xiamen on the eastbound route becomes very important, 2) a leg of call connecting Hamburg to Bombay might never be viable economically for an operator α, 3) a call to country β before calling at country γ might not be possible at all due to tensions and so on, 4) traversing certain piracy-ridden geographical areas might be too expensive and include a significant risk and as a result increases the insurance costs, 5) the draught limit and the turnaround time (resulting from the performance of the port) can make certain ports less attractive (with respect to different vessel classes) to be considered as a main port to be called at by mega-vessels and 6) a call to port θ in a given region to replenish the bunker (as part of the itineraries) might be inevitable.

4.2. Initial Relaxation

The initial relaxation of the model is a composed of constraints (3)-(21) with the exception of constraints (14)-(15). Constraints (14)-(15) and (22)-(31) are separated in the course of the branch-and-cut.

4.3. Valid Inequalities and Local Cuts

Some classes of globally and locally valid inequalities for this problem are introduced in this section.

Proposition 1. *Let i, j and k be three nodes each of which has an in-degree and an out-degree 1 in $\mathcal{L}$. Let (i, j) and (j, k) belong to the set of potential arcs $\mathcal{L}$. In any feasible solution to the RGP, if $t^1_{ii} = 0$ then both t^1_{jj} and t^1_{kk} will be 0, while if at least one of them takes 1, the other two will also have to be 1. Moreover, none of them can become a main node, that is, $h^1_i = h^1_j = h^1_k = 0$.*

Proof. The first and second parts can easily be verified. The latter is supported by one of our aforementioned practice-based assumptions. We cannot have a main port that is visited by one single route and no other main or feeder routes calling at it. □

Proposition 2. *Let (i, j) and (j, k) belong to the set of potential arcs $\mathcal{L}$ each of which has an in-degree and an out-degree 1 in $\mathcal{L}$. If any of these three nodes becomes a non-main route port, the other two will also have to become non-main route nodes and therefore:*

$$y^f_{ij} + y^f_{jk} + t^1_{jj} + t^1_{kk} \geq 4t^1_{ii}, \qquad \forall i, j, k : distinct, \tag{56}$$

$$y^f_{ij} + y^f_{jk} + t^1_{ii} + t^1_{kk} \geq 4t^1_{jj}, \qquad \forall i, j, k : distinct, \tag{57}$$

$$y^f_{ij} + y^f_{jk} + t^1_{ii} + t^1_{jj} \geq 4t^1_{kk}, \qquad \forall i, j, k : distinct. \tag{58}$$

Proposition 3. *Let i and j be two consecutive and adjacent nodes each of which has an in-degree and an out-degree 1, in the initial network structure, $\mathcal{L}$. Let us also assume that i' and j' are the unique arcs adjacent to i and j, respectively. In any feasible solution to the RGP, we have:*

$$\frac{1}{2}(t_{i'i} + t_{j'j}) + \frac{1}{3}(y^f_{i'i} + y^f_{ij} + y^f_{jj'}) \geq 1. \tag{59}$$

Proposition 4. *The following 2-matching constraints for the main and feeder routes are valid for the RGP polytope, $\mathcal{P}(RGP)$:*

$$\sum_{a\in\gamma(H)} y^f_a + \sum_{a\in(\delta^+(T)\cup\delta^-(T))} y^f_a \leq \sum_{i\in H}(t^1_{ii} + t^2_{ii}) + \frac{|\mathcal{T}| - 1}{2}, \tag{60}$$

$$\sum_{a\in\gamma(H)} y^h_a + \sum_{a\in(\delta^+(T)\cup\delta^-(T))} y^h_a \leq \sum_{i\in H}(h^1_i + h^2_i + h^3_i) + \frac{|\mathcal{T}| - 1}{2}, \tag{61}$$

where for all $H \subset \mathcal{P}$ and all $\mathcal{T} \subset (\delta^+(H) \cup \delta^-(H))$ satisfying,
i) $|\{i,j\} \cap H| = 1, \qquad \forall (i,j) \in \mathcal{T},$
ii) $\{i,j\} \cap \{k,l\} = \emptyset, \qquad \forall (i,j) \neq (k,l) \in \mathcal{T}$, *and*
iii) $|\mathcal{T}| \geq 3$ *and odd.*

$\mathcal{T} \subset \mathcal{P}$ is called teeth (with respect to the main route subgraph or the feeder route subgraph, that is, $\mathcal{T}^r, H^r, \mathcal{T}^y$ and H^y) and $H \subset \mathcal{P}$ is known as a handle.

Proof. Concerning inequalities (60), one knows that in every feasible solution of RGP we have,

$$2y^f(\gamma(H)) + y^f(\delta^+(H)) + y^f(\delta^-(H)) = \sum_{i \in H} (y^f(\delta^+(i)) + y^f(\delta^-(i)))$$

and from constraints (8) and (10) one obtains:

$$\sum_{k \in H} \sum_{l \neq k} (y^f_{kl} + y^f_{lk}) \leq 2 \sum_{k \in H} (t^1_{kk} + t^2_{kk}).$$

Hence,

$$\begin{aligned} 2 \sum_{k \in H} (t^1_{kk} + t^2_{kk}) &\geq 2y^f(\gamma(H)) + y^f(\delta^+(H)) + y^f(\delta^-(H)) \\ &= 2y^f(\gamma(H)) + y^f(\delta^+(H/\mathcal{T}) + y^f(\delta^+(\mathcal{T})) + y^f(\delta^-(H/\mathcal{T})) + r(\delta^-(\mathcal{T}))). \end{aligned} \tag{62}$$

Given that $y^f(\delta^+(\mathcal{T}) \cup \delta^-(\mathcal{T})) \leq |\mathcal{T}|$ deduced from the bound constraints $y^f_e \leq 1$, we add this set of constraints to (62) and we obtain:

$$\begin{aligned} 2 \sum_{k \in H} (t^1_{kk} + t^2_{kk}) + |\mathcal{T}| &\geq 2y^f(\gamma(H)) + y^f(\delta^+(H/\mathcal{T}) + y^f(\delta^-(H/\mathcal{T})) + 2y^f(\delta^+(\mathcal{T})) + 2y^f(\delta^-(\mathcal{T}))) \\ &\geq 2y^f(\gamma(H)) + 2y^f(\delta^+(\mathcal{T})) + 2y^f(\delta^-(\mathcal{T})). \end{aligned} \tag{63}$$

Again, given that $|\mathcal{T}|$ is an odd number, by multiplying both sides by $\frac{1}{2}$, we conclude,

$$y^f(\gamma(H)) + y^f(\delta^+(\mathcal{T}) + y^f(\delta^-(\mathcal{T}))) \leq \sum_{k \in H} (t^1_{kk} + t^2_{kk}) + \frac{|\mathcal{T}| - 1}{2} \tag{64}$$

and the proof is complete.
The proof of validity of (61) follows a similar way knowing that:

$$\sum_{k \in H} \sum_{l \neq k} (y^h_{kl} + y^h_{lk}) \leq 2 \sum_{k \in H} (h^1_i + h^2_i + h^3_i).$$

□

Proposition 5. *For $|\mathcal{T}| = 1$, valid inequalities (60) ((61)) reduce to constraints (12) and (13) ((14)-(15)).*

Proof. Consider (60) where $|\mathcal{T}| = 1$. Hence, $\gamma(H) = \emptyset$ and therefore $|\gamma(H)| = 0$. The proof is then complete given that no more than two arcs can be adjacent to a node $i \in H$ in any feasible solution. □

Proposition 6. *The following comb inequalities for both main and feeder networks are valid for $\mathcal{P}(RGP)$:*

$$\sum_{a \in A(H)} y^f(a) + \sum_{j=1}^{t} y^f(A(T_j)) \leq \sum_{i \in H} (t^1_{ii} + t^2_{jj}) + \sum_{j=1}^{t} |T_j| - \frac{3t+1}{2}, \tag{65}$$

$$\sum_{a\in A(H)} y^h(a) + \sum_{j=1}^{t} y^h(A(T_j)) \leq \sum_{i\in H}(h^1_{ii} + h^2_{ii} + h^3_{ii}) + \sum_{j=1}^{t} |T_j| - \frac{3t+1}{2}, \tag{66}$$

where for all $S \subset \mathcal{P}$ and all $T_i \subset \mathcal{P}$, $\forall i \in \{1,\dots,t\}$ satisfying,
i) $H, T_1, T_2, \dots, T_t \subseteq \mathcal{P}$,
ii) $T_j \setminus H \neq \emptyset$, $\quad \forall j \in \{1,\dots,t\}$,
ii) $T_j \cap H \neq \emptyset$, $\quad \forall j \in \{1,\dots,t\}$,
ii) $T_i \cap T_j = \emptyset$, $\quad \forall j \in \{1,\dots,t\}$, and
iii) $t \geq 3$ and odd.

$T_1, T_2, \dots, T_t$ are called teeth (with respect to the main subgraph or the feeder route subgraph, i.e., T^r_i, H^r_i, T^y_i and H^y, $\forall i \in \{1,\dots,t\}$) and $H \subset \mathcal{P}$ is a handle.

Proof. For $S \subset \mathcal{P}$, in every feasible solution of RGP we have,

$$y^f(\gamma(H)) = \sum_{i\in H}(t^1_{ii} + t^2_{ii}) - \frac{\delta^+(S) + \delta^-(S)}{2}, \tag{67}$$

and for every $T_j \subset \mathcal{P}$, $j \in \{1,\dots,t\}$ we have,

$$y^f(\gamma(T_j)) = \sum_{i\in T_j}(t^1_{ii} + t^2_{ii}) - \frac{\delta^+(T_j) + \delta^-(T_j)}{2}. \tag{68}$$

Furthermore, from (67) and (68), one yields:

$$y^f(\gamma(H)) + \sum_i y^f(\gamma(T_i)) = \sum_{i\in H}(t^1_{ii} + t^2_{ii}) + \sum_j \sum_{i\in T_j}(t^1_{ii} + t^2_{ii}) - \frac{1}{2}\left((\delta^+(H) + \delta^-(H)) + \sum_j(\delta^+(T_j) + \delta^-(T_j))\right). \tag{69}$$

Let $\delta^+_i(H) \cup \delta^-_i(H)$ denote the cut set associated with arcs having one end-point in $H \cap T_i$ and another end-point outside H. We know that $\delta^+(H) + \delta^-(H) \geq \sum_i(\delta^+_i(H) + \delta^-_i(H))$.

It can be easily shown that $(\delta^+_i(H) + \delta^-_i(H)) + (\delta^+(T_i) + \delta^-(T_i)) \geq 3$. We also know that $\delta^+(H) + \delta^-(H)$ and $\delta^+(T_i) + \delta^-(T_i)$ are even numbers and therefore $\delta^+(H) + \delta^-(H) + \sum_i(\delta^+(T_i) + \delta^-(T_i)) \geq 3t + 1$. By substituting in (69) the proof is complete.

The proof of validity of (66) follows a similar way. □

Proposition 7. *The following odd-hole inequalities are valid for the $\mathcal{P}(RGP)$.*

$$t_{ij} + t_{jk} + t_{ki} \leq 1, \qquad \forall i, j, k : \textit{distinct}. \tag{70}$$

Proposition 8. *The following inequalities are valid for the RGP polytope.*

1. *Neither a feeder route arc nor an main route arc can overlap with an arc connecting an isolated port to another port (in any directions).*

$$y^h_{ij} + t_{ij} + t_{ji} \leq 1, \qquad \forall i, j,, \tag{71}$$
$$y^f_{ij} + t_{ij} + t_{ji} \leq 1, \qquad \forall i, j. \tag{72}$$

2. *Both end-points of a feeder (main) route arc are adjacent to other feeder (main) route arcs.*

$$y^f_{ij} \leq \sum_{k\notin\{i\}} y^f_{ki}, \qquad \forall i, j \neq i, \tag{73}$$

$$y^h_{ij} \leq \sum_{k \notin \{j\}} y^h_{jk}, \qquad \forall i, j \neq i, \tag{74}$$

3. *Either one or more entering arcs and one or more leaving arcs are adjacent to port i or an allocation arc connects it to a non-isolated port.*

$$\sum_{j \neq i} y^h_{ij} + \sum_{j \neq i} t_{ij} \leq 1, \qquad \forall i, \tag{75}$$

$$\sum_{j \neq i} y^f_{ij} + \sum_{j \neq i} t_{ij} \leq 1, \qquad \forall i, \tag{76}$$

4. *No isolated triangular route (whether a feeder route or a main route) can exist in any feasible solution.*

$$y^f_{ij} + y^f_{jk} + y^f_{ki} - \frac{1}{2}\sum_{l \neq i}(y^h_{il} + y^h_{li}) - \frac{1}{2}\sum_{l \neq j}(y^h_{jl} + y^h_{lj}) - \frac{1}{2}\sum_{l \neq k}(y^h_{kl} + y^h_{lk}) \leq 2, \qquad \forall i, j, k \ \ distinct, \tag{77}$$

$$y^h_{ij} + y^h_{jk} + y^h_{ki} - \frac{1}{2}\sum_{l \neq i}(y^f_{il} + y^f_{li}) - \frac{1}{2}\sum_{l \neq j}(y^f_{jl} + y^f_{lj}) - \frac{1}{2}\sum_{l \neq k}(y^f_{kl} + y^f_{lk}) \leq 2, \qquad \forall i, j, k \ \ distinct, \tag{78}$$

5. *The following valid inequalities are valid for the polytope of the RGP because they state that both end-points of a route arc must be a route port and such arcs must not coincide with the allocation arcs.*

$$y^f_{ij} \leq t_{jj} - t_{ij}, \qquad \forall i, j \neq i, \tag{79}$$

$$y^f_{ij} \leq t_{ii} - t_{ji}, \qquad \forall i, j \neq i, \tag{80}$$

$$y^h_{ij} \leq t_{jj} - t_{ij}, \qquad \forall i, j \neq i, \tag{81}$$

$$y^h_{ij} \leq t_{ii} - t_{ji}, \qquad \forall i, j \neq i, \tag{82}$$

Moreover, (79)-(80) *dominate* (12)-(13) *and* (81)-(82) *dominate* (14)-(15). *In addition,* (71)-(72) *are also dominated by* (79)-(82).

4.4. Cutting Plane

Separation of valid inequalities (30)-(31). We establish a support graph $G^\dagger = (\mathcal{P}^\dagger, A^\dagger)$ where $\mathcal{P}^\dagger = \mathcal{P}$ and all $a = (i, j) \in A$ for which $y^{f*}_a + y^{h*}_e + t^*_{ij} + t^*_{ji} > 0$ weighted by $\max\{y^{f*}_a, y^{h*}_a, (t^*_{ij} + t^*_{ji})\}$. We then use the max-flow algorithm of Edmonds and Karp between every pair of nodes, say i and j and identify S and $\mathcal{P}\dagger / S$. Wherever the max-flow is less than ϵ we add the cut for the given S and $\mathcal{P}^\dagger / S$.

Separation of valid inequalities (29) . For a given solution $(y^{h*}, t^{1*}, t^{2*}, y^{f*}, h^{1*}, h^{2*}, h^{3*})$, we establish a support graph $G' = (\mathcal{P}', A')$ where $\mathcal{P}'$ is composed of all nodes in the solution for which $h^{1*}_i > 0$ and all $e \in E$ for which $y^f_a > 0 (y^h_a > 0)$ weighted by $y^{f*}_a > 0 (y^{h*}_a > 0)$. We then use the max-flow algorithm of Edmonds and Karp between every pair of nodes, say i and j. If the cut size is less than $2(y^{f*}_i + y^{h*}_j - 1)$ and is bigger than a given ϵ, we separate a cut.

Separation of valid inequalities (22)-(27). For all solution $(y^{h*}, t^{1*}, t^{2*}, y^{f*}, h^{1*}, h^{2*}, h^{3*})$, we establish a support graph $G^\ddagger = (\mathcal{P}^\ddagger, A^\ddagger)$ where $\mathcal{P}^\ddagger$ is composed of all $a \in A$ for which $y^{h*}_a > 0$ or $y^{f*}_a > 0$. We then enumerate all the cycles in such a graph using Tiernan's method (see Tiernan (1970)) and examine every cycle for identifying violated cuts.

Separation of valid inequalities (60) *and* (61). Our computational experiments have shown that the approach by Padberg and Rao (1982) performs slightly better than that of Fischetti et al. (1998) which is a heuristic mechanism.

Separation of valid inequalities (65) *and* (66). Separation based on block decomposition (as implemented for TSP in Concorde) is shown to be more efficient than that of Letchford and Lodi (2002) which is a polynomial-time algorithm for separating such inequalities.

4.4.1. Benders cuts

The master and subproblem possess some particular properties that need to be taken into account when dealing with the Benders decomposition. 1) For any integer feasible solution to the RGP, the subproblem is feasible and optimality is guaranteed. Hence, we are only concerned with the *optimality Benders cuts*. 2) The subproblem is a mixed-integer linear programme with binary variables. Although it is a mixed-integer programme, it shows some integrality-like behaviours. In other words, the optimal LP solution is often an integer and therefore the duals can still be extracted by solving the LP relaxation. 3) If an optimality cut could not be separated, at least combinatorial cuts can be separated.

Let $\mathbf{u^1}, \mathbf{u^2}, \ldots, \mathbf{u}^{19}$ correspond to the dual multipliers of constraints (33)-(51).

Constraints (33) - (42) are equality constraints with 0 on the right-hand side. Therefore, only constraints (43)-(51) contribute to generating Benders cuts.

If the dual values corresponding to an optimal solution to the FDFRP are available, an optimality Benders cut has the following form:

$$\eta \leq \sum_{r\in\mathcal{R}} u^{11} + \sum_{\substack{r\in\mathcal{R},l\in\mathcal{L}_r,\\ o\in\mathcal{P}}} (u_l^{12} + u_l^{13} + u_l^{14} + u_l^{15})(y_l^f + y_l^h) + \sum_{\substack{r\in\mathcal{R},l\in\mathcal{L}_r,\\ o\in\mathcal{P}}} \mathcal{M}(u_l^{16} + u_l^{17} + u_l^{18} + u_l^{19})(y_l^f + y_l^h) \tag{83}$$

It is very likely that the dual variables corresponding to constraints (48)-(51) will be zero, and in reality such a Benders cut is only composed of the following part:

$$\eta \leq \sum_{r\in\mathcal{R}} u^{11} + \sum_{r\in\mathcal{R},l\in\mathcal{L}_r,o\in\mathcal{P}} (u_l^{12} + u_l^{13} + u_l^{14} + u_l^{15})(y_l^f + y_l^h) \tag{84}$$

Even if those duals are zero, it is theoretically possible to generate dual solutions with non-zero coefficients for at least some of the dual coefficient components of (48)-(51). However, our extensive computational experiments do not recommend this.

Retrieving dual values. To generate such a Benders cut, we require the set of dual values. There are two ways to retrieve this set: 1) if the LP relaxation is solved with an optimal integer solution, the dual values can be used to generate (84); 2) if such an LP solution is not an integer solution, one can resort to a Lagrangian relaxation of the FDFRP in which constraints (42)-(43) are relaxed in a Lagrangian fashion. We use multipliers $\mathbf{u^{10}}$ and $\mathbf{u^{11}}$ initialized by the LP relaxation dual values. Such a relaxation is then subject to a few iterations of the subgradient or bundle method to obtain a good approximation of dual values. Given that the number of dual multipliers is rather limited, that is, $\sum_{r\in\mathcal{R}} |\mathcal{L}_r| + |\mathcal{R}|$, one can expect very quick convergence of either of these methods in particular the bundle method.

It must be noted that there are situations in which the expert system is not able to propose any set of feasible routes from the components generated by the RGP. In such a case, only combinatorial cuts are added to cut off the RGP solution.

4.5. Algorithm

We use our `BENMIP`[3] platform to carry out this computational experiment. `BENMIP` is built on top of the COIN-OR platform. `BENMIP` includes utilities allowing interfacing with different solvers. We use CPLEX

[3]BENMIP: A generic Bender decomposition-based (mixed-) integer programming solver, a project funded by the PGMO.

as an LP/MIP solver when required to solve an MP or SP. When using CPLEX, we disable almost all kinds of procedures, including probing, primal reduction, dual reduction, cut generation and any sort of heuristic that can be disabled.

Tree management. Our effort is put into quickly improving the upper bound (maximization problem), and accordingly we process the nodes following a breadth-first scheme.

Branching rules. Our extensive computational experiments reveal that the solution method is rather robust to the branching decisions both in the MP and in the SP. Therefore, no particularly promising branching scheme is identified.

Cut management. We need to distinguish between two types of feasibility. When a solution to the master problem satisfies all constraints (3)-(31) we call it *MP-feasible*. An MP-feasible solution is then supplied to an inferencing tool for proposing route structures based on the criteria and assumptions mentioned in section 2 as well as automated expert system rule verification. If the set of proposed routes respects the aforementioned criteria in chaining components, such a solution is referred to as a *real feasible* or *feasible* solution.

We add cuts mainly during the solution to the MP. Some cuts are added to almost all the nodes of the branch-and-bound tree, while others are added only at certain nodes. All such cuts are added if they meet a threshold of violation, ϵ. All the globally valid cuts are added as global cuts. This means that once added they remain in the LP until the end of the branch-and-bound process. No cut pool is maintained.

Constraints (14)-(15) and (22)-(31) are part of the RGP polytope description and are separated at the root node as well as all the integer nodes.

In addition, the 2-matching constraints (61) are separated at every node. Valid inequalities (60), (61), (65), and (66) are separated at every 200 and 500 fractional nodes.

The Benders cuts are only separated when no other valid inequality is identified and the MP solution is a *real feasible* solution. In the case of an MP-feasible solution, which is not a *real feasible* solution, one adds combinatorial Benders cuts.

5. Computational Results

The `BENMIP` framework is coded in C++ and can be compiled both in Windows (Microsoft C++ compiler) and in Linux (Ubuntu). For this project we compile `BENMIP` using the VC++ compiler in Windows 10 and run on a personal computer with an Intel Core i7 CPU, 3.4 GHz and 16 GB of RAM. CPLEX 12.6.3 is used as an MIP solver. The max-flow algorithms are solved using the boost implementation of the Edmonds-Karp algorithm for directed graphs.

To carry out our computational experiments, we use the instances from a real data set for a worldwide-operating mid-size liner shipping company. However, due to confidentiality, the data have been perturbed in such a way that, while preserving the general structure of the data, the important information remains masked.

Our main instance is a network composed of around 830 ports (see Figure 6) that are directly or indirectly served by our liner shipping company. While the demand matrix for such an instance is very sparse, we have serious difficulties in solving the MP for such a huge number of variables. Therefore, we resort to some kinds of aggregation (akin to the one proposed by Ernst and Krishnamoorthy (1996)) to establish grids on the map and to introduce new virtual nodes as the centre of gravity of every cell in the grid with a demand equal to the demand of all the nodes within the grid cell. Examples of such smaller instances resulting from such aggregations are depicted in Figure 7.

Our flow matrices have varying levels of density. We have instances from almost 30% (mainly for larger instances) to almost 90% density (for smaller ones).

We deactivated several internal cut generation procedures of CPLEX. These include FlowCovers, FlowPaths, MIRCuts, FracCuts, LiftProjCuts, Cliques, Covers, FlowCovers, GUBCovers, FracCuts, MIRCuts, DisjCuts, ZeroHalfCuts and MCFCuts. We turned off all the presolved phases to avoid premature convergence

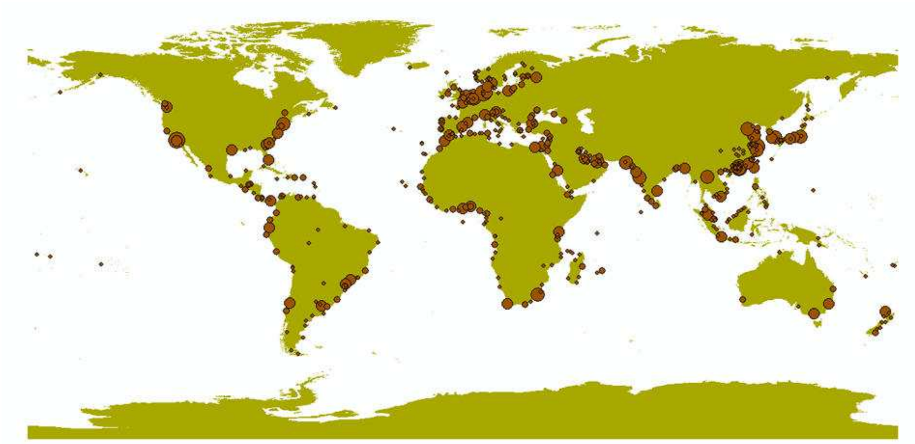
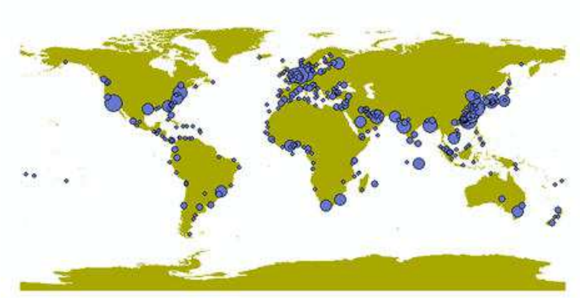


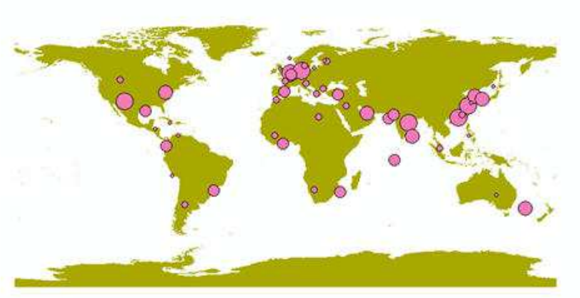
Figure 6: The main instance composed of 830 ports.

(a) 500-port aggregated instance.



(b) 400-port aggregated instance.

(c) 200-port aggregated instance.



(d) 100-port aggregated instance.

Figure 7: Some of the smaller instances generated by aggregation of the 830-port instance.

in Benders.

A time limit of 30 hours, equivalent to 108,000 seconds, and an optimality tolerance of 5 per cent are used as termination criteria —whichever occurs first. The tolerance $\epsilon = 10^{-4}$ was set as the threshold of violation of separated valid cuts. Our extensive computational experiments confirmed that the approach itself is sufficiently efficient and that further efforts to generate sharper cuts from the SP only add extra overheads without making any significant improvements.

In Table 4 and Table 5, we report our computational experiments. The first column, 'Instance', represents the instance name in the format 'MX', where X refers to the number of ports in the instance. In the second column, 'TimeRoot', we report the time spent on solving the LP relaxation. The third and fourth columns, 'LPobj' and 'LPStatus', respectively, stand for the LP objective value and the LP status in the column. In the column 'TimeIP', we report the elapsed time for solving the IP. The column 'Nnodes' represents the number of processed nodes. The columns 'Gap' and 'IPStatus' stand for the integrality gap and the CPLEX status, respectively. The IP objective function upon termination is reported in column 'IPobj'. The best upper bound (maximization problem) is presented in the column 'bestObj'. In the column 'NUserCuts', we report the number of user cuts added in the course of the BCB approach.

Table 4: Branch, Benders and Cut for the small to mid-size instances.

Instance	TimeRoot	LPobj	LPStatus	TimeIP	Nnodes	Gap	IPStatus	IPobj	bestObj	NUserCuts
M10	3.19	12176.61	Optimal	5.61	7	0.00	Optimal	6678.92	6678.92	370
M15	7.28	64592.40	Optimal	74.89	13	0.00	Optimal	15048.36	15048.36	266
M20	25.74	59074.29	Optimal	39.56	11	0.00	Optimal	1833.72	1833.72	786
M25	33.90	14772.11	Optimal	45.07	14	0.00	Optimal	3976.61	3976.61	492
M30	11.08	132388.52	Optimal	692.07	26	0.00	Optimal	2391.22	2391.22	760
M35	39.96	243778.12	Optimal	1014.34	64	0.00	Optimal	94539.11	94539.11	2874
M40	64.93	292087.09	Optimal	1226.21	44	0.00	Optimal	131191.52	131191.52	1176
M45	53.43	36137.57	Optimal	255.63	78	0.00	Optimal	28856.06	28856.06	3392
M50	86.34	203722.04	Optimal	798.41	34	0.00	Optimal	138424.59	138424.59	3411
M55	60.70	202262.80	Optimal	3821.08	93	0.00	Optimal	99368.49	99368.49	4810
M60	89.37	378770.82	Optimal	3090.57	113	0.00	Optimal	101048.34	101048.34	1463
M65	91.85	479150.94	Optimal	548.96	130	0.00	Optimal	11439.33	11439.33	2532
M70	87.60	319593.61	Optimal	1173.76	114	0.00	Optimal	175278.27	175278.27	5499
M75	130.41	468854.80	Optimal	4168.85	133	0.00	Optimal	232686.54	232686.54	6888
M80	110.66	619600.92	Optimal	32353.64	127	0.00	Optimal	523128.34	523128.34	4830
M85	131.17	589037.44	Optimal	19337.78	167	3.35	OptimalTol	322306.73	541489.14	4496
M90	100.91	714656.70	Optimal	1934.11	170	5.53	OptimalTol	340359.14	506445.35	7837
M95	136.06	83444.27	Optimal	8480.81	162	1.41	OptimalTol	12917.76	16400.38	1800
M100	141.77	381266.15	Optimal	39834.36	198	0.75	OptimalTol	12647.83	23577.82	6213

Table 5: Branch, Benders and Cut for the large size instances.

Instance	TimeRoot	LPobj	LPStatus	TimeIP	Nnodes	Gap	IPStatus	IPobj	bestObj	NUserCuts
M105	145.93	442736.50	Optimal	41260.48	161	2.81	OptimalTol	258576.97	441144.09	2800
M110	163.64	671116.41	Optimal	16833.34	201	5.28	OptimalTol	416329.19	610284.23	7119
M115	192.06	886531.34	Optimal	4877.03	223	3.40	OptimalTol	61357.35	92137.27	8030
M120	177.23	197328.92	Optimal	1961.07	282	3.94	OptimalTol	98611.90	173404.81	7636
M125	193.84	73520.37	Optimal	13966.89	255	4.55	OptimalTol	24170.98	34890.74	5448
M130	166.63	377962.80	Optimal	108129.11	238	4.51	TiLim	201147.70	374774.64	2750
M135	198.79	845495.54	Optimal	108169.93	225	5.26	TiLim	289945.56	298442.14	9464
M140	199.42	379400.24	Optimal	108175.57	287	7.80	TiLim	218326.25	270895.41	11961
M145	169.45	670458.77	Optimal	108125.31	316	1.69	TiLim	516068.54	600890.10	9156
M150	209.13	650761.38	Optimal	108168.09	325	1.01	TiLim	311527.16	325854.65	9889
M155	219.02	103883.21	Optimal	108112.30	316	4.86	TiLim	48993.53	82280.96	3030
M160	289.11	1381324.08	Optimal	108022.77	362	4.21	TiLim	829620.16	1163435.07	8773
M165	272.82	927177.09	Optimal	108022.67	339	9.08	TiLim	445190.79	854416.20	9824
M170	309.91	720745.23	Optimal	108022.65	367	7.88	TiLim	23206.72	43329.80	14553
M175	250.85	20906.16	Optimal	108103.48	330	1.71	TiLim	298.56	480.33	4590
M180	244.50	881669.19	Optimal	108159.80	382	5.11	TiLim	561787.25	778706.08	13895
M185	323.41	1001263.76	Optimal	108142.65	392	4.95	TiLim	291062.34	299385.44	12600
M190	362.39	221168.84	Optimal	108069.45	400	5.06	TiLim	85123.91	121152.57	15688
M195	414.13	2409422.64	Optimal	108095.66	387	4.10	TiLim	1128657.91	2051331.28	15390
M200	460.35	1803414.29	Optimal	108137.83	392	9.35	TiLim	832853.15	1509349.44	10140

In Table 4 one can observe that the proposed approach solves instances with up to 80 ports to optimality and within a very reasonable time. The maximum CPU time elapsed in solving the MIP belongs to M80, which is around 9 hours of computation. The other instances of this table are solved to optimal tolerance, which is set to a gap of 5 per cent. We tried to let the algorithm continue even further beyond this time, but the additional time spent did not pay off with any better solution quality. The numbers of separated valid inequalities and Benders cuts are rather reasonable and increase as the size of the instance increases.

An interesting observation is that the number of processed branch-and-bound nodes is very limited and never reaches 200 nodes. It indicates that most of the time is spent either on separating valid inequalities or on solving the node LPs, while the initial LPs are rather easy to solve.

In Table 5 we face some difficulties in larger instances. Up to M125 the termination criterion to be met is always optimal tolerance of 5 per cent. Again the number of processed nodes is reasonable and always fewer than 300 nodes, but the number of separated cuts is increasing. This makes the LP increasingly difficult to resolve.

For larger instances from M130 up to M200, the process always terminates by reaching the time limit of 30 hours. Even within this time limit, the reported gaps are always below 10 per cent. A slight deviation from this time span of 30 hours is due to the fact that, when the time limit has been reached, the node LP is still being solved and needs some more seconds to terminate. Again the number of processed nodes is reasonable and never exceeds 400. The number of separated cuts is very significant, and this explains why such a small number of nodes are processed within 30 hours, firstly because the separation routines are time consuming and secondly due to the node LPs, which become increasingly more difficult to solve.

In general we can solve relatively large instances of up to 200 nodes with high-quality solutions within a very reasonable time limit. The main issue here seems to be the time-consuming separation routines and the quality of the LP relaxation, which can be concluded from the difference between the optimal solution to LPs and the best-found IP solutions.

6. Conclusions

In this paper we have proposed the first primal (Benders) decomposition-based modelling approach for the problem of a simultaneous network design, flow and fleet deployment problem, taking into account the repositioning of empty containers. We highlighted the potential of Benders decomposition in integrating independent yet related problems and acting as a modelling tool. We showed how this approach facilitates the integration of experts' knowledge into the modelling and solution method (e.g. where a closed-form constraint is not at hand). We studied the polytope of the master problem and identified several classes of valid inequalities and the corresponding efficient separation routines to tighten the formulation of the master problem. We then exploited particular properties of the subproblem to extract primal/dual information and separate the Benders cuts. We also presented some techniques for fixing variables and some instance-dependent preprocessing. The computational results using a set of real instances showed that the proposed approach can efficiently address industrial-scale instances of the problem.

Further research directions include polyhedral analysis, identifying some valid and tightening inequalities and improving the performance of separation routines. Moreover, the hybridization of metaheuristic and exact methods, which can deliver solutions with known quality, deserves more attention.

Acknowledgment

The BENMIP solver development has been supported by the PGMO, the Gaspard Monge Programme for Optimization and Operational Research, in the framework of the BENMIP project.

References

Agarwal, R. and Ergun, O. (2008). Ship schedulinig and network design for cargo routing in lienr shipping. *Transportation Science*, pages 1–22.

Benders, J. (2005). Partitioning procedures for solving mixed-variables programming problems. *Computational Management Science*, 2(1):3–19.

Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische mathematik*, 4(1):238–252.

Braekers, K., Janssens, G. K., and Caris, A. (2011). Challenges in managing empty container movements at multiple planning levels. *Transport Reviews*, 31(6):681–708.

Brouer, B. D., Alvarez, J. F., Plum, C. E., Pisinger, D., and Sigurd, M. M. (2013). A base integer programming model and benchmark suite for liner-shipping network design. *Transportation Science*, 48(2):281–312.

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

Chang, H., Jula, H., Chassiakos, A., and Ioannou, P. (2008). A heuristic solution for the empty container substitution problem. *Transportation Research Part E: Logistics and Transportation Review*, 44(2):203–216.

Chuang, T.-N., Lin, C.-T., Kung, J.-Y., and Lin, M.-D. (2010). Planning the route of container ships: A fuzzy genetic approach. *Expert Systems with Applications*, 37(4):2948–2956.

Codato, G. and Fischetti, M. (2006). Combinatorial Benders' cuts for mixed-integer linear programming. *OPERATIONS RESEARCH-BALTIMORE THEN LINTHICUM-*, 54(4):756.

Dang, Q.-V., Yun, W.-Y., and Kopfer, H. (2012). Positioning empty containers under dependent demand process. *Computers & Industrial Engineering*, 62(3):708–715.

Dong, J.-X. and Song, D.-P. (2009). Container fleet sizing and empty repositioning in liner shipping systems. *Transportation Research Part E: Logistics and Transportation Review*, 45(6):860–877.

Ernst, A. T. and Krishnamoorthy, M. (1996). Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location Science*, 4:139–154.

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

Fischetti, M., Gonzalez, J. J. S., and Toth, P. (1998). Solving the orienteering problem through branch-and-cut. *INFORMS J. on Computing*, 10(2):133–148.

Fischetti, M., Salvagnin, D., and Zanette, A. (2009). Minimal infeasible subsystems and Benders cuts. *Mathematical Programming to appear*.

Gelareh, S. and Meng, Q. (2009). A novel modeling approach for the fleet deployment problem within a short-term planning horizon. *Transportation Research Part E: Logistics and Transportation Review*.

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

Gelareh, S. and Nickel, S. (2011). Hub location problems in transportation networks. *Transportation Research Part E*, 47:1092–1111.

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

Huang, Y.-F., Hu, J.-K., and Yang, B. (2015). Liner services network design and fleet deployment with empty container repositioning. *Computers & Industrial Engineering*, 89:116–124.

Imai, A., Shintani, K., and Papadimitriou, S. (2009). Multi-port vs. hub-and-spoke port calls by containerships. *Transportation Research Part E: Logistics and Transportation Review*, 45(5):740–757.

Lee, C.-Y. and Meng, Q. (2015). *Handbook of Ocean Container Transport Logistics*. Springer.

Letchford, A. N. and Lodi, A. (2002). *Integer Programming and Combinatorial Optimization: 9th International IPCO Conference Cambridge, MA, USA, May 27–29, 2002 Proceedings*, chapter Polynomial-Time Separation of Simple Comb Inequalities, pages 93–108. Springer Berlin Heidelberg, Berlin, Heidelberg.

Maculan, N., Plateau, G., and Lisser, A. (2003). Integer linear models with a polynomial number of variables and constraints for some classical combinatorial optimization problems. *Pesquisa Operacional*, 23(1):161 – 168.

Magnanti, T., Mireault, P., and Wong, R. (1986). Tailoring Benders decomposition for uncapacitated network design. *Netflow at Pisa*, pages 112–154.

Magnanti, T. and Wong, R. (1981). Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, 29(3):464–484.

McDaniel, D. and Devine, M. (1977). A modified Benders' partitioning algorithm for mixed integer programming. *Management Science*, 24(3):312–319.

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

Meng, Q., Wang, S., Andersson, H., and Thun, K. (2013). Containership routing and scheduling in liner shipping: overview and future research directions. *Transportation Science*, 48(2):265–280.

Padberg, M. W. and Rao, M. R. (1982). Odd minimum cut-sets and b-matchings. *Mathematics of Operations Research*, 7(1):67–80.

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

Poojari, C. and Beasley, J. (2009). Improving benders decomposition using a genetic algorithm. *European Journal of Operational Research*, 199(1):89–97.

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

Rodrigue, D. J.-P. (2016). The geography of transport systems. website.

Shen, W. and Khoong, C. (1995). A dss for empty container distribution planning. *Decision Support Systems*, 15(1):75–82.

Sherali, H. and Fraticelli, B. (2002). A modification of Benders' decomposition algorithm for discrete subproblems: An approach for stochastic programs with integer recourse. *Journal of Global Optimization*, 22(1):319–342.

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

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

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

Tiernan, J. C. (1970). An efficient search algorithm to find the elementary circuits of a graph. *Commun. ACM*, 13(12):722–726.

Tran, N. K. and Haasis, H.-D. (2015). Literature survey of network optimization in container liner shipping. *Flexible Services and Manufacturing Journal*, 27(2-3):139–179.

UNCTAD (2015). Unctad transport newsletter. Technical report.

Wang, J. J. (2007). *Ports, cities, and global supply chains*. Ashgate Publishing, Ltd.

Wang, S., Liu, Z., and Meng, Q. (2015). Segment-based alteration for container liner shipping network design. *Transportation Research Part B: Methodological*, 72:128–145.

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

Wang, S., Wang, T., Qu, X., Liu, Z., and Jin, S. (2014). Liner ship fleet deployment with uncertain demand. *Transportation Research Record Journal of the Transportation Research Board*, 2409(1):49 – 53.

Waters, W. (2016). East-west trade imbalance widens. http://www.lloydsloadinglist.com/freight-directory/news/East-West-trade [Online; accessed December 2016].