Operational Planning of Routes and Schedules for a Fleet of Fuel Supply Vessels

Abstract. This paper considers a real operational problem of routing and scheduling a fleet of fuel supply vessels used to service customer ships anchored outside a major port. The problem can be formulated as a rich multi-trip vehicle routing problem, including constraints related to stowage and time-dependent sailing times. An arc-flow and a path-flow model are developed and compared. A computational study shows that the path-flow model is superior and can be used in real planning situations. We also discuss how the model can be used in a real-time setting when new orders arrive and deviations from the plan occur.

Keywords: Maritime transport, routing, scheduling, stowage, time-dependent sailing times

1 Introduction

Ocean shipping is the major transportation mode of the world trade today, and the volume carried by seaborne trade is growing (UNCTAD, 2014). Ships operate between ports for loading and unloading passengers and cargo, as well as for loading fresh water, supplies, and discharging waste. Another important task for ships in certain ports is fuel refilling. The problem studied in this paper regards the fuel supply business, where incoming customer ships are supplied with fuels by a given fleet of specialized fuel supply vessels. Even though fuel refilling is an important task for ships entering ports, the planning problem considered in this paper has, to the authors' knowledge, only been studied in one previous paper (Christiansen et al., 2015). As in that case study, we consider the problem of a Hellenic oil company operating in the broader area of Pireaus Port delivering fuel to customer ships, as illustrated in Figure 1. The incoming customer ships anchor in a specified area outside the port waiting to be supplied by the company's fuel supply vessels. The supply vessels load at refineries in the inner part of the port area before supplying the customer ships. The refineries offer different types of fuel, and a given customer ship may require more than one type. Fuel transported to the customer ships must be allocated to compartments on board the supply vessels, and different fuel types cannot be mixed in the same compartment. Each customer ship needs to be serviced within a given agreed time window. The planning problem, which we denote as the Fuel Supply Vessel Routing Problem (FSVRP), consists of determining routes and schedules for the fleet of supply vessels such that costs are minimized and all customer ships are serviced within their time windows. The vessels can perform more than one voyage during the planning horizon. The problem also includes allocating the different types of fuel to separate compartments within the supply vessels, which adds substantial complexity. The FSVRP can be considered as a rich version of the multi-trip vehicle routing problem with time windows, see for example Nguyen et al. (2013) and Cattaruzza et al. (2014).

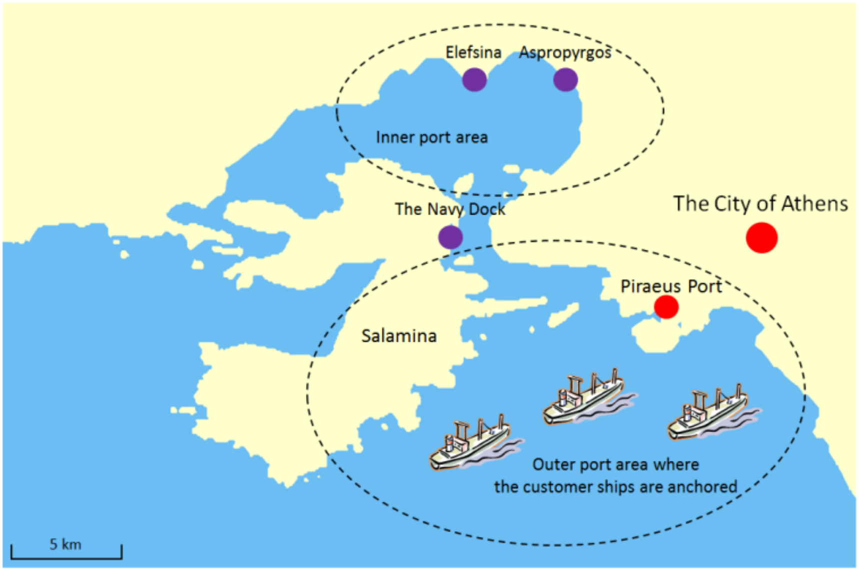

Figure 1: Map of Piraeus port area

The fuel supply business in Piraeus Port, as probably in most other ports, has long traditions, and the business is to a large extent characterized by manual efforts in determining routes and schedules for the fuel supply vessels. However, many complicating factors and the large amount of money involved increase the demand for good decision support systems in the fuel supply business.

Christiansen et al. (2015) presented an arc-flow model for the FSVRP with some additional elements related to the customer ships. It was assumed that all different orders for the same customer ships could be serviced by more than one fuel supply vessel and the orders were optional and not contracted. They also proposed some alternatives of how to simplify the model to make it easier to solve using a commercial solver. It was shown that the simplified version where one ensures that all orders for any customer ship are serviced by the same fuel supply vessel (i.e. without customer splitting) provided similar results as the one without this simplification (i.e. with customer splitting). This is also in correspondence with what is experienced in practice where customer splitting is rarely performed. Unfortunately, Christiansen et al. (2015) experienced that even the simplified model was extremely hard to solve for realistic instances, and large optimality gaps were reported even after 10,000 seconds of running time.

The objective of this paper is to describe the operational planning problem of designing routes and schedules for a fleet of fuel supply vessels providing fuel to customer ships. Furthermore, the contributions of this paper are an enhanced arc-flow model for the FSVRP where no customer splitting is allowed as well as a new path-flow model. We also show that the proposed path-flow model and corresponding solution method is superior to the arc-flow model in Christiansen et al. (2015) with regards to computational performance, and realistic instances are solved to optimality within reasonable solution times. The planners need to use the model when new orders appear during the day or other unforeseen deviations from the plan occur. Therefore, we provide a discussion on how the model can support decision-making in a real-time setting.

The outline of the remaining of the paper is as follows: Section 2 presents the FSVRP in more detail, while Section 3 surveys relevant literature. Section 4 presents the arc-flow and path-flow models for the problem, while the algorithm for generating the paths (i.e. feasible vessel voyages) is described in Section 5. A computational study is conducted in Section 6, while Section 7 discusses how the proposed models can be used as decision support in a real-time setting. Finally, concluding remarks are provided in Section 8.

2 Problem description

Here, we distinguish the fuel supply vessels from the cargo and passenger ships that enter the port area to receive fuel, by using the words *vessel* or *supply vessel* to denote the fuel supply vessels, and *ship* or *customer ship* to denote the ships that are serviced by the fuel supply company.

A given heterogeneous fleet of supply vessels is used to supply customer ships anchored in a port area. In the start of a planning horizon, some supply vessels may not be available for loading until some specified time since they may still be occupied delivering fuel from a previous planning period. The customer ships place orders of different fuel types. The supply vessels load all fuel types at refineries, denoted as *depots*, which are located in the *inner port area*. After finishing loading at the depot, the supply vessels start sailing to the customer ships, which are located in the outer *port area*. During nights, the vessels must sail around the Salamina Island since they are for security reasons not allowed to sail in the area of the navy dock, which is located in between the inner and outer port area (Figure 1). Since this takes longer time, the sailing time between the depots and the customer ships is dependent on the hour of the day, resulting in time-dependent sailing times from the depots to the customer ships. Since all customer ships anchor in the outer port area the sailing times between them are not dependent of time. Furthermore, since the sailing times between the customer ships are usually small compared their service times, they are assumed to be similar between all pairs of customer ships.

A vessel's voyage starts with loading at the depot, continues with sailing to and servicing one or several customer ships before returning empty to the depot. Within the planning horizon, a vessel may perform more than one voyage. Hence, every time a vessel loads at the depot, it starts a new voyage. A vessel may wait at a customer ship or at the depot before operation starts.

In this case study, there exist two refineries and both of them produce all fuel types. The quays at the refineries are also used by vessels from other companies. It is therefore not known long time in advance which refinery and which quay to visit before a particular voyage. We have adapted the case company's planning practice where they, based on experience, only use an approximated fixed time, independent of vessel and loading quantity, for the loading operation at the refineries. Furthermore, the distances between the refineries are almost negligible for this particular case study, so we assume that the refineries can be modeled as a single depot. The depot has a berth capacity, which implies that a maximum number of vessels may load simultaneously at a time. For this particular problem we assume berth capacity of one. This means that at most one of the company's vessels can visit the depot at the same time, which will reduce the probability of having to wait for a quay.

The customer ships may place orders of different fuel types to be delivered at the same time. Each customer ship states a time window in which all its orders must be serviced and a given quantity is specified for each order. All orders at a customer ship are serviced by the same supply vessel, and the service of the orders must happen continuously. The supply vessels are obliged to service all customers during the planning horizon.

The fleet of supply vessels is heterogeneous, where the vessels have different capacities, costs and starting times when they become available. Each vessel has several compartments with given capacities where the fuel types are loaded. A compartment may carry several fuel types, but it may only contain one fuel type at a time. The same fuel type may be carried in several compartments on board the same supply vessel, and large orders may be split between compartments. Moreover, if different customer ships order the same fuel type, the orders may be allocated to the same compartment.

There exist two types of compartments in the supply vessels, i.e. one type dedicated for marine gas oil and one dedicated for up to four alternative types of fuel oil. The various types of fuel oil are very similar and no cleaning of these compartments is necessary between voyages when changing fuel oil type. This means that cleaning of compartments between voyages can be disregarded.

The planning problem consists of determining routes and schedules for the fleet of supply vessels such that the transportation costs are minimized and all orders are serviced within their time windows. The total costs consist of fixed daily costs for using the vessels and variable sailing costs. The problem also includes allocating the different types of fuel to separate compartments within the supply vessels.

To summarize: The planning problem can be considered as a rich vehicle routing problem, including multiple trips, time windows, tank allocation or stowage constraints, and time-dependent travel times.

3 Literature review

Maritime transportation planning problems have attracted considerable attention in the literature during the last decades; see for example the surveys by Christiansen and Fagerholt (2014) and Christiansen et al. (2013). The FSVRP studied in this paper has however received very limited attention so far except for Christiansen et al. (2015), which proposed and tested an arc-flow formulation for the problem. Some studies integrate refueling decisions when planning shipping routes. Besbes and Savin (2009) and Kim et al. (2012) study the handling of refueling decisions for a single vessel when determining its route, while Vilhelmsen et al. (2014) and Meng et al. (2015) deal with a similar problem when routing a fleet of ships. There, varying fuel prices between ports are taken into account. While the FSVRP focuses on the routing of the supply vessels, the latter studies look at the problem from the customer ships' perspective.

Path-flow models, like the one we propose in this paper, have proven to be very efficient for solving many routing problems, see for example Poggi and Uchoa (2015). Several studies have demonstrated the usefulness of such models also for maritime versions of the problem. As an example, Andersson et al. (2011) propose two alternative path-flow models for a maritime pickup and delivery problem. Similar to our experience for the FSVRP, they demonstrate that the path-flow models perform significantly better than the corresponding arc-flow model. The models in Andersson et al. (2011) have continuous time in contrast to the model we propose in this paper, which uses discrete time to better handle the time-dependent sailing times.

The allocation of products to compartments (i.e. stowage) is an important aspect of the FSVRP. Hvattum et al. (2009) describe a *tank allocation problem (TAP)* motivated from chemical shipping, and deals with the allocation of liquid bulk cargoes to the tanks on board a given vessel. They present several constraints that are similar to our problem, such as tank capacity and regarding no mixing of product types in the same tank. In contrast to the FSVRP, they do not consider routing and scheduling decisions. Only few studies in maritime transportation combine routing with allocation decisions like in the FSVRP. Kobayashi and Kubo (2010) deal with routing a fleet of oil tankers, where each tanker has several fixed compartments and different cargoes cannot be in the same compartment. Al-Khayyal and Hwang (2007) and Li et al. (2010) assume that each compartment is dedicated to specific products. Agra, Christiansen and Delgado (2013) consider both the case without any allocation of different fuel products into different cargo tanks as well as the case where there are dedicated tanks for families of products. The ship routing and scheduling problem studied by Fagerholt and Christiansen (2000) is also a combined routing and allocating problem, where different dry bulk products cannot be stored together. However, in contrast to our paper, the tanks are not given as the cargo hold can be divided into a number of different configurations.

Another special characteristic with the FSVRP considered in this paper is the time-dependent sailing times between the depot and the customer ships. To the authors' knowledge there exists no studies in maritime routing where this has been considered. However, in land-based routing time-dependent travel times are more common to capture, as the traffic, and hence the travel times, vary with time. See for example the recent review paper on time-dependent routing problems by Gendreau et al. (2015). Most models with time-dependent travel times ensure that a vehicle will never arrive at its destination earlier by postponing its departure, which is a reasonable assumption in most land-based routing problems. This is in contrast to the situation in the FSVRP studied in this paper where a supply vessel sometimes can avoid sailing the longer route around the Salamina Island by waiting.

4 Mathematical models

In this section, we propose two different mathematical models for the FSVRP. The first model is an arc-flow model and is a mixed integer programming model with binary variables on the arcs between nodes, while the other model is a path-flow model where the paths represents feasible voyages for each ship. Section 4.1 introduces some assumptions and definitions that are used in the mathematical models. Section 4.2 describes the notation used for the arc-flow model. Furthermore, the objective function and the constraints of the arc-flow model are described in Section 4.3. Finally, Section 4.4 presents the path-flow model including the necessary additional notation.

4.1 Modeling approach and assumptions

We have chosen to develop a discrete time model due to the time dependent sailing time between the inner and outer port area. With discrete time representation, the planning horizon is divided into time periods of equal lengths.

We define one node for each customer ship since we assume that one vessel services all orders of a customer ship. This deviates from Christiansen et al. (2015) where a node represented an order placed by a customer ship. In addition to the nodes representing the customer ships, we include a depot node and a dummy end node. The depot node represents both refineries, while the dummy end node represents returning to the depot, without starting a new voyage. This is a fictive node representing where the vessels end up after servicing all customer nodes.

Each vessel may execute multiple voyages during the planning horizon. In the mathematical model the numbering of voyages is related to each supply vessel. The time window of a customer ship represents the earliest and latest start of servicing a customer ship. The end time of the service is the important requirement for the customer ships, but the start of service can easily be calculated based on this since we assume a continuous unloading of the different fuel types.

4.2 Notation

Indices

v	supply vessel
i, j	customer ship node
0	the depot node
d	the dummy end node
f	fuel type
c	Compartment
m	Voyage number
t	time period

Sets

$\mathcal{V}$	supply vessels
$\mathcal{N}$	customer ship nodes
$\mathcal{N}^T$	all nodes, i.e. $\mathcal{N}^T \cup \{0\} \cup \{d\}$
$\mathcal{F}$	fuel types
$\mathcal{F}_c \subseteq \mathcal{F}$	fuel types allowed in compartment c
$\mathcal{C}_v$	compartments in supply vessel v
$\mathcal{M}_v$	voyages for vessel v
$\mathcal{T}$	time periods

$\mathcal{T}^{DAY} \subseteq \mathcal{T}$	time periods that represent a day's first time period. For example, when the planning horizon starts with time period 0 and one time period represent one hour, time periods 0, 24, 48 etc. are time periods in the set.

Parameters

T^{SA}_{vijt}	sailing time when vessel v sails directly between nodes i and j when arriving at node j in time period t
T^{SD}_{vijt}	sailing time when vessel v sails directly between node i and j when departing node j in time period t
T^{O}_{vi}	vessel v's operating time at node i
$\underline{T}_i$	start of time window for start of service at customer ship node i
$\overline{T}_i$	end of time window for start of service at customer ship node i
T^{M}_{v}	the minimum time a vessel may use on any voyage
T^{E}_{v}	the earliest time vessel v is available for operation
H	number of time periods within 24 hours
B	berth capacity of the depot
D_{if}	demanded quantity of fuel type f for customer ship node i
Q_{cv}	load capacity of compartment c on vessel v
C^{F}_{v}	fixed daily cost of using vessel v
C^{S}_{v}	sailing cost per time period for vessel v
R_f	revenue per quantity delivered of fuel type f

Variables

x_{vijmt}	1, if vessel v starts sailing in time period t from node i directly to node j on voyage m/ 0, otherwise
y_{vimt}	1, if vessel v starts operating node i in time period t on voyage m / 0, otherwise
w_{vimt}	1, if vessel v is waiting in time period t at node i on voyage m/ 0, otherwise
δ_{vt}	1, if vessel v is utilized the day that start with time period t/ 0, otherwise
k_{vfcm}	1, if compartment c of vessel v is allocated to fuel type f on voyage m/ 0, otherwise
l_{vijfcm}	quantity of fuel type f in compartment c of vessel v when sailing directly from node i to j on voyage m

Before presenting the model, we would like to add some comments to the notation and the variables. The sailing and operating variables, $\boldsymbol{x_{vijmt}}$ and $\boldsymbol{y_{vimt}}$, equal 1 if a vessel *starts* sailing or operating the given time period. The sailing or the operation itself may take more than one time period. The durations of these activities are given by the sailing time parameters, $\boldsymbol{T^{SA}_{vijt}}$ and $\boldsymbol{T^{SD}_{vijt}}$, and the operating time parameters, $\boldsymbol{T^{O}_{vi}}$. The waiting variables, $\boldsymbol{w_{vimt}}$, equal 1 for *each* time period a vessel waits at a node. All these types of variables are illustrated in Figure 2, which is an example of a vessel's flow in a time-space network.

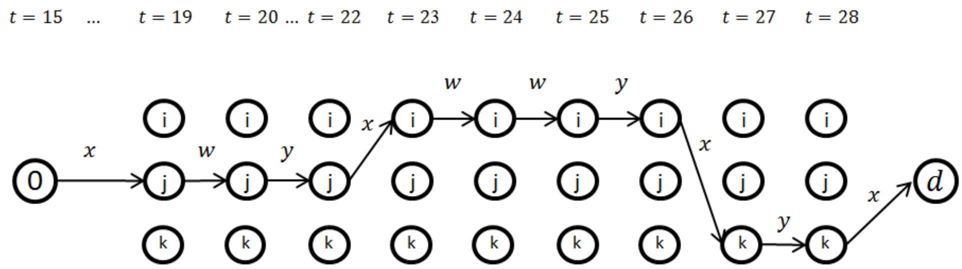

Figure 2: Example of a vessel's flow in a time-space network. The arc labels are $\boldsymbol{y}$ for operating, $\boldsymbol{x}$ for sailing and $\boldsymbol{w}$ for waiting. In this example the vessel starts by sailing from the depot, then it operates at nodes $\boldsymbol{j}$, $\boldsymbol{i}$ and $\boldsymbol{k}$ before it sails to the dummy end node. Note that the operating time of node $\boldsymbol{j}$, $\boldsymbol{T^{O}_{vj}}$, is 2 time periods, while the operating times of the two other nodes are 1 time period. The sailing from the depot to node $\boldsymbol{j}$, $\boldsymbol{T^{SA}_{v0j(t=19)}}$, is 4 time periods, while the other sailing times in this example are only 1 time period.

4.3 Arc-flow model

The mathematical formulation of the arc-flow model consists of the objective function and constraints related to routing, scheduling, loading and unloading.

Objective function

The objective function (1) represents the company's transportation costs. It comprises the variable sailing costs and daily fixed costs of using the vessels. By including daily fixed costs in this way, the model will strive towards solutions where the vessels are busy some days, and are idle other days. This is assumed to be practical in the real case problem, as longer breaks of whole days allow for necessary repairs and time off for the crew.

$$min\ \text{AF} = \sum_{t\in\mathcal{T}} \sum_{m\in\mathcal{M}_v} \sum_{i\in\mathcal{N}^T} \sum_{j\in\mathcal{N}^T} \sum_{v\in\mathcal{V}} C^S_v T^{SD}_{vijt} x_{vijmt} + \sum_{t\in\mathcal{T}^{DAY}} \sum_{v\in\mathcal{V}} C^F_v \delta_{vt} \tag{1}$$

Routing and scheduling constraints

The flow or routing constraints are given as follows:

$$\sum_{t=\underline{T}_i}^{\bar{T}_i} \sum_{m \in \mathcal{M}_v} \sum_{v \in \mathcal{V}} y_{vimt} = 1 \qquad i \in \mathcal{N} \tag{2}$$

$$\sum_{t \in \mathcal{T}} y_{v0mt} \leq 1 \qquad v \in \mathcal{V}, m \in \mathcal{M}_v \tag{3}$$

$$\sum_{\tau=0}^{t-T_v^M} y_{v0(m-1)\tau} - y_{v0mt} \geq 0 \qquad v \in \mathcal{V}, m \in \mathcal{M}_v, t \in \mathcal{T} | m > 1 \tag{4}$$

$$y_{vim(t-T_{vi}^O)} = \sum_{j \in \mathcal{N}^T} x_{vijmt} \qquad v \in \mathcal{V}, i \in \mathcal{N} \cup \{0\}, m \in \mathcal{M}_v, t \in \mathcal{T} \tag{5}$$

$$\sum_{j \in \mathcal{N} \cup \{0\}} x_{vjim(t-T_{vjit}^{SA})} + w_{vim(t-1)} = y_{vimt} + w_{vimt} \qquad v \in \mathcal{V}, i \in \mathcal{N}, m \in \mathcal{M}_v, t \in \mathcal{T} | t > 1 \tag{6}$$

$$\sum_{j \in \mathcal{N}} x_{vj0m(t-T_{vj0t}^{SA})} + w_{v0m(t-1)} = y_{v0(m+1)t} + w_{v0mt} \qquad v \in \mathcal{V}, m \in \mathcal{M}_v, t \in \mathcal{T} \tag{7}$$

$$\sum_{t \in \mathcal{T}} y_{v01t} - \sum_{t \in \mathcal{T}} \sum_{m \in \mathcal{M}_v} \sum_{j \in \mathcal{N}} x_{vjdmt} = 0 \qquad v \in \mathcal{V} \tag{8}$$

$$\sum_{\tau=t}^{t+(H-1)} \sum_{m \in \mathcal{M}_v} \sum_{i \in \mathcal{N} \cup \{0\}} (y_{vim\tau} + \sum_{j \in \mathcal{N} \cup \{0\}} x_{vijm\tau}) - H\delta_{vt} \leq 0 \qquad v \in \mathcal{V}, t \in \mathcal{T}^{DAY} \tag{9}$$

$$\sum_{\tau=\max\{0, t-T_{v0}^O+1\}}^{t} \sum_{m \in \mathcal{M}_v} \sum_{v \in \mathcal{V}} y_{v0m\tau} \leq B \qquad t \in \mathcal{T} \tag{10}$$

$$x_{vijmt} \in \{0,1\} \qquad v \in \mathcal{V}, i, j \in \mathcal{N}^T, m \in \mathcal{M}_v, t \in \mathcal{T} \tag{11}$$

$$y_{vimt} \in \{0,1\} \qquad v \in \mathcal{V}, i \in \mathcal{N} \cup \{0\}, m \in \mathcal{M}_v, t \in \mathcal{T} \tag{12}$$

$$w_{vimt} \in \{0,1\} \qquad v \in \mathcal{V}, i \in \mathcal{N} \cup \{0\}, m \in \mathcal{M}_v, t \in \mathcal{T} \tag{13}$$

$$\delta_{vt} \in \{0,1\} \qquad v \in \mathcal{V}, t \in \mathcal{T}^{DAY} \tag{14}$$

Constraints (2) ensure that every customer ship node is serviced exactly once, by one vessel on one voyage. The constraints also control that the customer nodes are serviced within their time windows. Furthermore, constraints (3) make sure that the vessels operate at the depot at most once on each voyage. Constraints (4) control that a vessel cannot start a new voyage if it has not ended the previous voyage. The constraints also ensure that the previous voyage takes at least time T_v^M, which is the minimum time any vessel may use on a voyage. In constraints (5), it is described that when a vessel has finished servicing a node, it must start sailing to a customer node, the depot node or the dummy end node. Constraints (6) make sure that a vessel either starts waiting or operating at a customer ship node when the vessel arrives at the node. Moreover, if a vessel waits at a node in a time period, it is restricted to either operate or wait at the node in the following time period. Constraints (7) are equivalent to the previous constraints, but concern the depot node. They make sure that when a vessel arrives at the depot, it must either start loading at the depot for a new voyage or wait at the depot on the current voyage. If a vessel waits at the depot in a time period, it may start operating on a new voyage or keep waiting on the current voyage in the next time period. Constraints (8) control that every vessel, if it is used at all, executes the fictive sailing to the dummy end node once during the planning horizon. Constraints (9) ensure that the variable δ_{vt} equals 1 if a given vessel is utilized the day which starts with time period t. Waiting is not included, since it is possible to wait at the depot which in practice corresponds to not utilizing the vessel. Constraints (10) ensure that in any time period, the company cannot have more than B vessels loading at the depot. Finally, the binary restrictions for the variables are given in (11)-(14).

Loading and unloading constraints

The load management or tank allocation on board the vessels is taken into account by the following constraints:

$$\sum_{c \in \mathcal{C}_v} \sum_{j \in \mathcal{N} \cup \{0\}} l_{vjifcm} - \sum_{t \in \mathcal{T}} D_{if} y_{vimt} - \sum_{c \in \mathcal{C}_v} \sum_{j \in \mathcal{N}^T} l_{vijfcm} = 0 \qquad v \in \mathcal{V}, i \in \mathcal{N}, f \in \mathcal{F}, m \in \mathcal{M}_v \tag{15}$$

$$\sum_{c \in \mathcal{C}_v} \sum_{f \in \mathcal{F}_c} l_{vijfcm} - \sum_{t \in \mathcal{T}} \sum_{c \in \mathcal{C}_v} Q_{vc} x_{vijmt} \leq 0 \qquad v \in \mathcal{V}, i \in \{0\}, j \in \mathcal{N}, m \in \mathcal{M}_v \tag{16}$$

$$\sum_{c \in \mathcal{C}_v} \sum_{f \in \mathcal{F}_c} l_{vijfcm} - \sum_{t \in \mathcal{T}} \sum_{c \in \mathcal{C}_v} (Q_{cv} - \sum_{f \in \mathcal{F}_c} D_{if}) x_{vijmt} \leq 0 \qquad v \in \mathcal{V}, i \in \mathcal{N}, j \in \mathcal{N}^T, m \in \mathcal{M}_v \qquad (17)$$

$$\sum_{f \in \mathcal{F}_c} k_{vfcm} \leq 1 \qquad v \in \mathcal{V}, c \in \mathcal{C}_v, m \in \mathcal{M}_v \qquad (18)$$

$$l_{vijfcm} - \min\{Q_{vc}, \sum_{k \in \mathcal{N}} D_{kf}\} k_{vfcm} \leq 0 \qquad v \in \mathcal{V}, i, j \in \mathcal{N} \cup \{0\}, f \in \mathcal{F}_c, c \in \mathcal{C}_v, m \in \mathcal{M}_v \qquad (19)$$

$$\sum_{c \in \mathcal{C}_v} \sum_{f \in \mathcal{F}_c} \sum_{j \in \mathcal{N}} l_{vjifcm} = 0 \qquad i \in \{0\} \cup \{d\}, v \in \mathcal{V}, m \in \mathcal{M}_v \qquad (20)$$

$$k_{vfcm} \in \{0,1\} \qquad v \in \mathcal{V}, f \in \mathcal{F}_c, c \in \mathcal{C}_v, m \in \mathcal{M}_v \qquad (21)$$

$$l_{vijfcm} \geq 0 \qquad v \in \mathcal{V}, i, j \in \mathcal{N} \cup \{0\}, f \in \mathcal{F}_c, c \in \mathcal{C}_v, m \in \mathcal{M}_v \qquad (22)$$

The difference in load within a supply vessel's compartments before and after operating a customer ship node equals the demanded fuel quantity of the node. This is ensured by constraints (15) for each fuel type. The load variables, l_{vijfcm}, can be denoted as arc-load flow variables. Agra et al. (2013) describe the strengthening advantages of having arc-load flow variables instead of the more common load variables not including a destination node j. Constraints (16)-(17) control that the l_{vijfcm} variables are assigned non-zero values only if the given vessel, v, sails directly between nodes i and j, and that the compartments' capacity limits are not exceeded. Constraints (18) ensure that only one fuel type is allocated to a compartment on each voyage. The constraints also make sure that a compartment is only loaded with a fuel type that it is allowed to carry. Constraints (19) control that the arc-flow load variables only take values for combinations of fuel type and compartment if the fuel type is actually allocated to that compartment. To facilitate the reading, we introduce constraints (20) to ensure that the vessels do not carry any load when returning to the depot or the dummy end node. Finally, the binary and non-negativity requirements for the variables related to loading are given in (21)-(22).

4.4 Path-flow model

As an alternative to the arc-flow model presented in Section 4.3, we suggest a path-flow formulation with variables that correspond to feasible and non-dominated vessel voyages. A voyage starts at the depot and then sails to one or several customer ships before returning to the depot. The same notation as in the previous section is used. In addition, the following notation is needed:

Index

r	a supply vessel voyage

Set

$\mathcal{R}_v$	feasible voyages for supply vessel v

Parameters

C_{vr}	cost of sailing voyage r using supply vessel v
A_{vir}	1 if supply vessel ship v visits node i on voyage r / 0 otherwise
O_{vtr}	1 if supply vessel ship v is occupied (sailing, loading/unloading or waiting) with voyage r in time period t / 0 otherwise
D_{vtr}	1 if supply vessel ship v is at the depot in time period t on voyage r / 0 otherwise

Variable

λ_{vr}	1, if supply vessel v operates voyage r/ 0, otherwise

The path-flow model of the FSVRP can be formulated as follows:

$$min\,\mathrm{PF} = \sum_{v \in \mathcal{V}} \sum_{r \in R_v} C_{vr}\lambda_{vr} + \sum_{t \in \mathcal{T}^{DAY}} \sum_{v \in \mathcal{V}} C_v^F \delta_{vt} \tag{23}$$

$$\sum_{v \in \mathcal{V}} \sum_{r \in R_v} A_{vir}\lambda_{vr} = 1 \qquad i \in \mathcal{N} \tag{24}$$

$$\sum_{r \in R_v} O_{vtr}\lambda_{vr} \leq 1 \qquad v \in \mathcal{V}, t \in \mathcal{T} \tag{25}$$

$$\sum_{\tau = t}^{t+(H-1)} O_{vtr}\lambda_{vr} - H\delta_{vt} \leq 0 \qquad v \in \mathcal{V}, t \in \mathcal{T}^{DAY} \tag{26}$$

$$\sum_{v \in \mathcal{V}} \sum_{r \in R_v} D_{vtr}\lambda_{vr} \leq B \qquad t \in \mathcal{T} \tag{27}$$

$$\lambda_{vr} \in \{0,1\} \qquad v \in \mathcal{V}, r \in \mathcal{R}_v \tag{28}$$

$$\delta_{vt} \in \{0,1\} \qquad v \in \mathcal{V}, t \in \mathcal{T}^{DAY} \qquad (29)$$

The objective function (23) minimizes the variable sailing costs and daily fixed vessel costs and corresponds to the objective function (1). Constraints (24) ensure that each customer ship is serviced exactly once by one vessel on one voyage and substitute constraints (2) in the arc-flow model. A supply vessel might sail several voyages during the planning period, but in any time period each vessel can at most be occupied on one voyage. This is taken care of by constraints (25), corresponding to constraints (4) in the arc-flow model. Furthermore, constraints (26) and (27) correspond to constraints (9) and (10), respectively. Finally, the binary requirements for the variables are given by constraints (28) and (29). The routing constraints (3) and (5) – (8) of the arc-flow model are satisfied through the generation of the voyages or paths, as described in Section 5.

5 Path generation algorithm

Before solving the path-flow model presented in Section 4.4, all feasible voyages for each supply vessel must be generated. To do this, we have used a labeling algorithm, following the approach described by Irnich and Desaulniers (2005). They present labeling algorithms as a dynamic programming approach that may be applied to find the set of Pareto-optimal paths to a shortest path problem with resource constraints (SPPRC). In their approach, a label is used to represent a (partial) path from the depot to a node i, together with information regarding the accumulation of resources along the path. Each resource is required to stay within given limits, referred to as resource windows, at each node along the path. Labels are extended along arcs in the problem-defining network creating new labels where the resources are updated according to resource extension functions and checked for feasibility with respect to the resource windows. Any label extended to the end node of the graph represents a complete feasible path through the network. To avoid generating sub-optimal paths, a dominance step is introduced to remove labels whose extension cannot become Pareto-optimal.

Applied to the problem studied in this paper, the SPPRC is defined on a graph where the set of nodes consists of the depot node 0, the set of customer ships $i \in \mathcal{N}$, and the artificial end node d. The set of arcs contains arcs from the depot to every customer ship, arcs between all pairs of customer ships, and arcs from all customer ships to the end node. A path is considered feasible if it does not violate constraints (3)-(9), nor constraints (15)-(22). In the following we describe in detail what information is stored in a label, and how a label is extended. Further, we present how pairs of labels can be compared with respect to dominance, and how we extend a label to the depot node and create (possibly more than one) voyage for a given label. Finally, an overview of the labeling algorithm is given, together with a pseudo-code.

5.1 Label data

Each label represents a partial path from the depot node 0 to node i. For each label we store the following information:

- i – the current node
- p – the predecessor label
- c – the accumulated cost when arriving at node i
- t – the accumulated time when arriving at node i
- $\mathcal{N}^V$ – set of nodes visited

The current node is stored to know the end node of the partial path, while the predecessor label is stored to be able to recursively re-trace the path. In addition, cost and time are the resources accumulated along the path, while the set of nodes visited helps to check that the path is elementary and stowage feasible. In the following we use $i(L)$ to denote the current node of label L, and similarly use $p(L)$, $c(L)$, $t(L)$, and $\mathcal{N}^V(L)$ for the predecessor label, accumulated cost, accumulated time, and set of nodes visited, respectively.

5.2 Extending labels

Initially, the algorithm starts with only one label, representing a path visiting only the depot node. After this, all new labels are created by adding customer ships to existing labels. When extending an existing label L along arc $(i(L), j)$, we create a new label L', where the resource extension functions can be stated as follows:

- $i(L') = j$ (30)
- $p(L') = L$ (31)
- $c(L') = c(L) + C_v^S T_{vi(L)jt(L)}^{SD}$ (32)
- $t(L') = \max\{\underline{T}_j, t(L) + T_{vi(L)}^O + T_{vi(L)jt(L)}^{SD}\}$ (34)
- $\mathcal{N}^V(L') = \mathcal{N}^V(L) \cup \{j\}$ (35)

Equations (30), (31) and (32) set the current node of the label, the predecessor label, and the accumulated cost of the label, respectively. Further, equations (34) and (35) update the accumulated time and the set of visited nodes, respectively.

The new label L' is considered feasible if the following holds:

- $t(L') \leq \overline{T}_j$
- $j \notin \mathcal{N}^V(L)$
- *stowfeas*($\mathcal{N}^V(L')$)

The method *stowfeas*($\mathcal{N}^V(L')$) checks whether there exists a feasible stowage plan on board the supply vessel for servicing all the customer ships visited on the voyage represented by label L'. The method is implemented as a simple depth-first dynamic programming algorithm which assigns the orders of the different fuel types from the relevant customers to cargo holds on board the vessel. Once a feasible stowage plan has been found or it has been proven that no such plan exists, the method terminates. Since the labeling algorithm is run for each time period, the *stowfeas()* method will be executed several times for the same set of customer ships ($\mathcal{N}^V(L')$). To avoid solving the same problem multiple times, the solution for a given set of customer ship nodes is stored the first time the method is executed.

5.3 Dominating labels

The dominance criteria used to remove dominated labels are as follows:

Proposition 1
A label L_1 dominates label L_2 if:

$c(L_1) \leq c(L_2)$

$t(L_1) \leq t(L_2)$

$\mathcal{N}^V(L_1) = \mathcal{N}^V(L_2)$

This dominance criterion is almost the same as the one stated in Irnich and Desaulniers (2005) for elementary SPPRC, with two notable exceptions. Since we want the best voyage that visits each subset of customer ships, we require all Pareto-optimal paths also with respect to the nodes visited. Thus, we have replaced $\mathcal{N}^V(L_1) \subseteq \mathcal{N}^V(L_2)$ with $\mathcal{N}^V(L_1) = \mathcal{N}^V(L_2)$, requiring that that the paths must have visited the same set of nodes. In addition, we can exploit the fact that the travel times between all customer ships are assumed to be equal, thus allowing us to omit the term $i(L_1) = i(L_2)$. Given that the above dominance criterion holds, any feasible extension of the label L_2 will also be feasible for L_1, even if they depart from different customer ships.

5.4 Extending a label to the depot node

Extending a label L to the depot node is not straight forward in the cases where the supply vessel is finished servicing customer ships and is ready to return to the depot during the night. The supply vessel then has, in many cases, two choices: Return immediately to the depot sailing around Salamina Island and thus returning earlier but at a higher cost. The second choice is to wait until the vessel is allowed to sail past the navy dock, and return to the depot later, but at a lower cost. Since both options can be optimal in given situations, we create two paths for each label where this is the case. In all other cases, we create one path.

5.5 An overview of the labeling algorithm

A pseudo-code for the generation of voyages for vessel v in time period t is given in Algorithm 1. The algorithm starts with one initial label, which is added to a set of unprocessed labels U. This initial label L_0 is created for a given vessel and start time combination and has the following values: $i(L_0) = 0$, $p(L_0) = null$, $c(L_0) = 0$, $t(L_0) = t$, and $\mathcal{N}^V(L_0) = \emptyset$. Until the set of unprocessed labels is empty, one label is removed from U, using the function *remove*. This function returns the label L representing the path with the least accumulated time. For each customer ship, a new label L' is created by extending label L to the corresponding node. If L' is resource feasible and not dominated by any other labels in U, it is added to the set of unprocessed labels. We also check whether the new label L' dominates any existing labels, and if so, remove the dominated labels from U. Finally, we extend label L to the depot node, and save the path(s) corresponding to label L in the set of paths $\mathcal{R}_{vt}$. The set $\mathcal{R}_{vt}$ is then returned by the algorithm. Algorithm 1 is run once for each $v \in \mathcal{V}$ and $t \in \{T_v^E, \ldots, |T|\}$, and the set $\mathcal{R}_v = \bigcup_{t=T_v^E}^{|T|} \mathcal{R}_{vt}$ is used by the path-flow model.

Algorithm 1: Pseudo-code for the labeling algorithm for vessel v in period t

Create initial label L_0
$U = \{L_0\}$
While $U \neq \emptyset$ **do**
    $L = remove(U)$
    For each customer ship $j \in \mathcal{N}$ **do**
        Create new label L' by adding node j to label L
        If L' is feasible **then**
            If no label in U dominates L' **then**
                Add L' to U
                Remove all labels in U that are dominated by L'
            End-if
        End-if
    End-for
    Extend L to d and add the corresponding paths to $\mathcal{R}_{vt}$
End-while
Return $\mathcal{R}_{vt}$

6 Computational study

This section presents a computational study performed on a number of test instances generated from real data from the case company. The mathematical models described in Section 4 have been implemented in Mosel and solved using the commercial optimization software Xpress v7.8 64-bit, while the path-generating algorithm described in Section 5 has been implemented in Java. All computational tests have been run on a HP DL 160 G5 computer with an Intel Xeon QuadCore E5472 3.0 GHz processor, 16 GB RAM, and running on a Linux operating system. Section 6.1 describes the test instances, while the computational experiments and results are presented in Section 6.2. There, we present both the results from solving the arc-flow model, as presented in Section 4.3, and the path-flow model, as presented in Section 4.4, as well as a comparison of the two.

6.1 Test instances

The test instances are generated based on data regarding customer ships and their fuel orders provided by the case company. All test instances include three supply vessels, as this was the number of vessels in the shipping company's fleet. Information regarding the vessels' compartments, load capacities, costs and pumping rates was also given. The vessels have from five to seven compartments and their total load capacities are in the range of approximately 1300 to 3000 m^3.

Since the sailing times are relatively small compared to the operating times at the customer ships and the depot (three to 12 hours), we have approximated the sailing times between customer ships and between the depot and the customer ships to one hour. Exceptions are the sailing time between the depot and the customer ships during night time, which is set to four hours because of the navy dock closure. Taking these sailing times into account, we have chosen to use a time discretization of 1 hour.

The customer ships typically have from one to three orders each (with two on average). Most customer ships have wide time windows specifying service within a given day (i.e. during a period of 24 time periods). However, some of the ships request morning deliveries where the deliveries must be made between 7 am and 2 pm on the given day.

The number of time periods to include in the planning horizon was set to the end time of the latest time window of the customer ships: $|\mathcal{T}| = \max_{(i \in \mathcal{N})} \bar{T}_u$. This varied between 48 and 96 hours (i.e. two to four days). The start of the planning horizon was set to $t = 0$. Since the vessels are already engaged in fuel deliveries when solving this problem (from the previous planning period), they are given different times for when they become available. Vessel 1 becomes available for loading at the depot from time period $t = 17$, meaning $T_1^E = 17$, while vessels 2 and 3 are available from $t = 7$ and $t = 0$, respectively.

Table 1. Test instances with varying number of customer ships on different days

Test instance	# Ships day 1	# Ships day 2	# Ships day 3	# Total ships	# Time periods
4_4_0	4	4	0	8	72
3_3_2	3	3	2	8	96
10_0_0	10	0	0	10	48
5_5_0	5	5	0	10	72
6_6_0	6	6	0	12	72
4_4_4	4	4	4	12	96

Table 1 summarizes the different test instances used in the computational study, where for example instance 4_4_0 denotes a test problem with four customer ships on days 1 and 2, and none on day 3. It should be noted that day 1 starts at $t = 24$, meaning that there is a day 0 previous to the service of the customer ships (due to the availability of some fuel supply vessels).

6.2 Computational experiments and results

Table 2 shows the best obtained solutions within a running time limit of 10,000 seconds for the arc-flow model. The table shows the best objective value found within the time limit together with the best lower bound and the gaps between these two values. We also present the value of the linear programming (LP) relaxation, i.e. the value of the root node in the branch-and-bound (B&B) tree, and the number of nodes in the B&B tree.

Table 2. Computational results from solving the arc-flow model.

Test instance	After 10,000 seconds			LP-relaxation	# B&B no-des	Solution time [s]
	Obj. value	Best bound	Gap [%]			
4_4_0	106	65.0	63.1	24.3	435,438	10,000
3_3_2	76	76.0	0	23.2	2373	188
10_0_0	73	73.0	0	37.9	19,682	896
5_5_0	75	73.0	2.7	29.1	373,549	10,000
6_6_0	105	88.2	19.1	34.7	198,185	10,000
4_4_4	113	88.0	28.4	33.8	102,163	10,000

As shown in Table 2, we were only able to find proven optimal solutions within the given time limit for two of the test instances, i.e. test instances 3_3_2 and 10_0_0. Three of the remaining instances still have large gaps even after 10,000 seconds running time. The reason for this is the weak LP relaxation which results in a large number of nodes to explore in the B&B tree.

Table 3 summarizes the results obtained when running the path-flow model. In addition to the information provided in Table 2 for the arc-flow model, we also report the number of paths (i.e. feasible vessel voyages) that were generated and the time spent on this. Since all test instances were solved to optimality within the time limit, we do not report the lower bounds and the gaps.

Table 3. Computational results from solving the path-flow model.

Test instance	Objective value [s]	# paths	LP-releax-tion	# B&B nodes	Time path generati-on [s]	Time path-flow model [s]
4_4_0	71	14,243	55.5	45	0.4	4.3
3_3_2	76	17,797	54.6	27	0.5	4.6
10_0_0	73	19,892	58.2	15	2.9	4.0
5_5_0	75	38,602	63.8	11	2.7	13.3
6_6_0	98	102,926	71.7	82	79.0	96.8
4_4_4	108	129,448	74.7	459	57.8	129.4

It can be noted from Table 3 that using the solution algorithm based on the path-flow model we were able to solve all instances to optimality within reasonable time. Even the two largest test instances were solved in about three minutes, including the time spent on generating the paths. Why the path-flow model performs much better than the arc-flow model can be explained by the improved LP relaxations, which result in much lower numbers of B&B nodes. For example, by comparing the results on test instance 10_0_0, which we were able to solve to optimality with both models, we see that the LP relaxation for the arc-flow model is 37.9, while it is 58.2 for the path-flow model. Similarly, while the number of B&B nodes is almost 20 thousand for the arc-flow model, the path-flow model only needs to explore 15 nodes to prove optimality.

Table 4. Comparison of the performance between the arc-flow and path-flow models.

Test instance	Arc-flow model: Gap from optimal solution [%]	Arc-flow model: Time [s]	Total time path-flow model [s]
4_4_0	49.3	10,000	4.7
3_3_2	0	188	5.1
10_0_0	0	896	6.9
5_5_0	0	10,000	16.0
6_6_0	7.1	10,000	175.8
4_4_4	4.6	10,000	187.2

To compare the results between the two models, we have summarized some of the information from Tables 2 and 3 in Table 4. It should be noted that the gaps reported in the second column now are the gaps from the optimal solutions (i.e. the solutions from the path-flow model). It can be noted that except for test instance 4_4_0, the solutions obtained by the arc-flow model are not very far away from the optimal ones as one might believe when studying the gaps from the lower bounds in Table 2. However, especially when taking the solution times into account, the results in Table 4 still show that the performance of the path-flow model is superior to the arc-flow model.

7 Practical use of the model in a real-time setting

The routing and scheduling of the fleet of supply vessels is performed following a rolling horizon planning principle, where the plan can be updated in specified time intervals or when new important information becomes available (e.g. new orders or information about delays). This means that the schedule made is quite likely to be changed after starting executing the plan. Thus, the planner needs to solve the problem over and over again with only minor changes to the current data, such as with the new orders and/or new times for when the vessels become available due to delays.

In practice customers request fuel orders about three days ahead of delivery. Some of these customers have long-term contracts and these orders are mandatory to service. When such orders arrive, the planner must solve the model again, and hopefully find a new feasible solution including the new orders. However, some of the requests may be spot orders that come from customers to which the company has no contractual obligations, and these orders can be considered as optional. In these situations, the planner may also not agree on the demanded quantities, if for example the fleet is short on capacity at the time of the delivery. The planner might instead propose a new offer, which is a fraction of the quantity originally demanded. Moreover, in this case the planner must solve the model updated with the new order, possibly with the quantity given as a decision variable; see the model proposed in Christiansen et al. (2015). When the spot orders are accepted, they become mandatory contract orders and the quantities are fixed. Therefore, the planner must be careful when accepting new spot orders, as it might restrict the possibilities for accepting (mandatory) future orders.

Since one frequently has to update the plan when new information arrives (e.g. new orders), it becomes important to find solutions quickly. The planners in the case company have suggested that solutions need to be obtained within a time frame of approximately ten minutes to be useful in the daily planning. Therefore, the results from using the path-flow model, summarized in Table 3, are very promising with respect to developing an optimization-based decision support system based on this research.

8 Concluding remarks

We have presented two alternative models for a combined fuel supply vessel routing and tank allocation problem, i.e. an arc-flow and a path-flow model, as well as an algorithm for generating the paths (i.e. feasible vessel voyages) as input to the path-flow model. We have also conducted a computational study on a number of test instances generated based on data from the case company.

The test results showed that the performance of the path-flow model was superior to the one by the arc-flow model. By using the arc-flow model we were only able to find proven optimal solutions to two of the six test instances with a running time limit of 10,000 seconds. However, using the path-flow model we were able to find optimal solutions to all test instances in less than about three minutes, which is less than what the case company has suggested as a reasonable time frame to provide practical decision support. In a real-time setting, the problem will be solved over and over again when new information arrives. This can for example be when the company receives a new request for a spot order and needs to decide whether to accept it or not, or when a supply vessel is delayed. Therefore, the results are very promising with respect to developing an optimization-based decision support system based on the proposed path-flow model.

Acknowledgement

We want to thank the two anonymous reviewers for their insightful comments and suggestions that have helped us improve the paper.

References

1. Agra, A., Christiansen, M. and Delgado, A. (2013). Mixed integer formulations for a short sea fuel oil distribution problem. Transportation Science 47(1), 108-124.
2. Ahuja, R., Magnanti, T., and Orlin, j. (1993). Network flows: Theory, Algorithms, and Applications. Prentice Hall, Upper Saddle River, New Jersey.
3. Al-Khayyal, F. and Hwang, S-J. (2007) Inventory constrained maritime routing and scheduling for multi-commodity liquid bulk, Part I: Applications and model. European Journal of Operational Research 176, 106-130.
4. Andersson, H., Christiansen, M. and Fagerholt, K. (2011). The maritime pickup and delivery problem with time windows and split loads. INFOR 49(2), 79-91.
5. Besbes, O. and Savin, S. (2009). Going bunkers: The joint route selection and refueling problem. Manufacturing & Service Operations Management 11, 694-711.
6. Cattaruzza, D., Absi, N., Feillet, D. and Vigo, D. (2014). An iterated local

search for the multi-commodity multi-trip vehicle routing problem with time windows. Computers & Operations Research 51, 257-267.
7. Christiansen, M. and Fagerholt, K. (2014). Ship Routing and Scheduling in Industrial and Tramp Shipping. In Vehicle Routing – Problems, Methods, and Applications, Second Edition, Toth P. and Vigo D. (eds.), MOS-SIAM Series on Optimization, 371-408.
8. Christiansen, M., Fagerholt, K., Nygreen, B. and Ronen, D. (2013). Ship routing and scheduling in the new millennium. European Journal of Operational Research 228(3), 467-483.
9. Christiansen, M., Fagerholt, K., Rachaniotis, N. R., Tveit, I. and Øverdal, M. V. (2015). A decision support model for routing and scheduling a fleet of fuel supply vessels. In Lecture Notes in Computer Science - Computational Logistics, Corman, F., Voss, S. and Negenborn, R. R. (eds.), Springer-Verlag Berlin Heidelberg, Vol. 9335, 46-60.
10. Fagerholt, K. and Christiansen, M. (2000). A combined ship scheduling and allocation problem. Journal of the Operational Research Society (15) 7, 834-842.
11. Gendreau, M., Ghiani, G. and Guerriero, E. (2015) Time-dependent routing problems: A review. Computers & Operations Research 64, 189-197.
12. Hvattum, L. M., Fagerholt, K. and Armentano, V. A. (2009). Tank allocation problems in maritime bulk shipping. Computers & Operations Research 36, 3051-3060.
13. Irnich, S. and Desaulniers, G. (2005). Shortest path problems with resource constraints. In: Desaulniers, G., Desrosiers, J., Solomon M. (Eds.), Column generation, Gerad 25th Annaversiary Series, Springer, 33-65.
14. Kim, H. J., Chang, T., Kim, T. and Kim, H. J. (2012) An epsilon-optimal algorithm considering greenhouse gas emissions for the management of a ship's bunker fuel. Transportation Research Part D: Transport and Environment 17(2), 97-103.
15. Kobayashi, K. and Kubo, M. (2010). Optimization of oil tanker schedules by decomposition column generation, and time-space network techniques. Japan Journal of Industrial and Applied Mathematics 27(1), 161-173.
16. Li, J, Karimi, I.A. and Srinivasan R. (2010). Efficient bulk maritime logistics for the supply and delivery of multiple chemicals. Computers & Chemical Engineering 34, 2118-2128.
17. Meng, Q., Wang, S. and Lee, C-Y. (2015). A tailored branch-and-price approach for a joint tramp ship routing and bunkering problem. Transportation Research Part B: Methodological 72, 1-19.
18. Nguyen, P.K., Crainic, T. G. and Tolouse, M. (2013). A tabu search for time-dependent multi-zone multi-trip vehicle routing problem with time windows. European Journal of Operational Research 231(1), 43-56.
19. Poggi, M. and Uchoa, E. (2014) New exact algorithms for the capacitated vehicle routing problem. In Vehicle Routing – Problems, Methods, and Applications, Second Edition, Toth P. and Vigo D. (eds.), MOS-SIAM Series on Opti-

mization, 59-86.
20. Vilhelmsen, C., Lusby, R. and Larsen, J. (2014). Tramp ship routing and scheduling with integrated bunker optimization. EURO Journal on Transportation and Logistics 3(2), 143-175.
21. UNCTAD (2014). Review of Maritime Transport 2012. United Nations, New York and Geneva.