DTU Library

Full-shipload tramp ship routing and scheduling with variable speeds

Wen, M.; Røpke, Stefan; Petersen, Hanne Løhmann; Larsen, Rune; Madsen, Oli B.G.

Published in:
Computers and Operations Research

Link to article, DOI:
10.1016/j.cor.2015.10.002

Publication date:
2016

Document Version
Peer reviewed version

Full-Shipload Tramp Ship Routing and Scheduling with Variable Speeds

M. Wen[a,*], S. Ropke[b], H. L. Petersen[c], R. Larsen[c], O. B.G. Madsen[c]

[a]*Xi'an Jiaotong–Liverpool University, 111 Ren Ai Road, Suzhou Industrial Park, Suzhou, Jiangsu, 215123, China*
[b]*Department of Management Engineering, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark*
[c]*Department of Transport, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark*

Abstract

This paper investigates the simultaneous optimization problem of routing and sailing speed in the context of full-shipload tramp shipping. In this problem, a set of cargoes can be transported from their load to discharge ports by a fleet of heterogeneous ships of different speed ranges and load-dependent fuel consumption. The objective is to determine which orders to serve and to find the optimal route for each ship and the optimal sailing speed on each leg of the route so that the total profit is maximized. The problem originated from a real-life challenge faced by a Danish tramp shipping company in the tanker business. To solve the problem, a three-index mixed integer linear programming formulation as well as a set packing formulation are presented. A novel Branch-and-Price algorithm with efficient data preprocessing and heuristic column generation is proposed. The computational results on the test instances generated from real-life data show that the heuristic provides optimal solutions for small test instances and near-optimal solutions for larger test instances in a short running time. The effects of speed optimization and the sensitivity of the solutions to the fuel price change are analyzed. It is shown that speed optimization can improve the total profit by 16% on average and the fuel price has a significant effect on the average sailing speed and total profit.

Keywords: tramp shipping, speed optimization, heuristic column generation

1. Introduction

Sea shipping is one of the most important transportation modes especially for large-volume goods between continents. It is estimated that sea cargo is responsible for around 80% of global trade by volume and over 70% by value, and these percentages are even higher in most developing countries (UNCTAD 2013). Among the various operational costs of sea shipping, fuel cost accounts for a large proportion. For example, in a liner shipping company, bunker cost can be around 50% (Notteboom 2006)

*Corresponding author
Email address: min.wen@xjtlu.edu.cn (M. Wen)

or even above 60% (Golias et al. 2009) of the total operational cost. In addition, the large consumption of fuel results in significant CO_2 and NO_x emissions, which have recently attracted media attention for their negative impacts on climate change and air pollution. According to the International Maritime Organization (Buhaug et al. 2009), CO_2 emissions from the maritime sector in 2007 has increased by 86% compared to 1990 and accounted for 3.3% of the world's total emissions. These emissions are expected to continue to increase by 150-250% in 2050 if no action is taken.

The fuel cost, and consequently CO_2 emissions, are strongly dependent on the sailing speed. Ryder and Chappell (1980) and Ronen (1982) showed that a cubic function can describe the relationship between fuel consumption and speed. The ocean conservation group OCEANA states: "Reducing commercial ship speeds, by only a few knots, yields salutary results for both shipping companies and the environment." (OCEANA 2008). Therefore optimizing sailing speed in order to reduce fuel cost and CO_2 emissions is an important issue to investigate.

Speed optimization for shipping routes has attracted some attention recently. For liner shipping, most of the previous work focuses on optimizing the speed for one or several fixed route(s) (Corbett et al. 2009, Ronen 2011, Lindstad et al. 2011 and Wang and Meng 2012). In these studies, the sailing speed is treated as a variable, which not only determines the fuel consumption but also determines the required ship and/or fleet size to maintain the given service frequency. The yearly profit is expressed as a function of sailing speed, and the optimal sailing speed is obtained when the marginal profit equals zero. Both Ronen (2011) and Corbett et al. (2009) concluded that when the fuel price is increased, the optimal speed is likely to be reduced when maximizing the profit. This reduction of speed also leads to a reduction of CO_2 emissions. Wang and Meng (2012) dealt with the problem of finding the optimal route for containers and the optimal speed for the ships given fixed shipping routes and origin-destination pair of each container. They proposed an outer-approximation method, which uses piecewise-linear functions to approximate the fuel consumption as a function of speed within a predetermined tolerance level.

In contrast to liner shipping, tramp shipping does not consider service frequency. A ship can sail at different speeds on different sailing legs of a route. The objective of tramp shipping is to minimize the total cost or maximize the total profit for transporting cargoes. Fagerholt et al. (2010) considered the problem of optimizing the speed on each leg of a single fixed route. Gatica and Miranda (2011) and Norstad et al. (2011) studied simultaneous routing and speed optimization for a full-shipload problem and a less-than-shipload problem respectively. In Fagerholt et al. (2010), the sequence of ports and the time window of visiting each port on a given route are fixed. Three models were presented for solving the problem. The first two models are non-linear models, where the speed is a continuous variable between a minimum and a maximum, and the fuel consumption per distance unit is expressed as a quadratic function of speed. The third model discretizes the time window and converts the problem into a shortest path problem on an acyclic graph. These models were tested on data with up to 16 ports and 10-day time window. It was shown that, compared to the continuous model, the discrete model is able to provide quality solution within short computational time. Gatica and Miranda (2011) applied the same idea of discretized time window to the solution of a full-shipload routing and speed optimization problem. They tested their method on generated data instances with up to 15 discretized time-window points, 50 cargoes and 9 ships. Norstad et al. (2011) formulated the less-than-shipload routing and speed optimization problem as a pickup and delivery problem with an extra continuous decision variable for the speed on each leg. They proposed a recursive smoothing algorithm for determining the optimal speed for a given route, which is shown to be superior to the discretization of arrival time presented in Fagerholt et al. (2010). To solve the routing and speed optimization simultaneously, they proposed a multi-start local search heuristic and tested it on data with up to 10-day time windows.

The problem of routing and scheduling for tramp shipping is very similar to the well-know vehicle routing problem (Eksioglu et al. (2009), Laporte et al. (2013)), which has been studied intensively in the literature. Christiansen et al. (2007) and Christiansen et al. (2013) gave a good overview of maritime transportation and present different models for routing and scheduling of both tramp and liner ships. A few variations of tramp shipping problems have also been addressed in the previous work. Gatica and Miranda (2011) studied a full-shipload problem, where each cargo corresponds to a full-shipload and a ship can only carry one cargo at a time as mentioned above. Brønmo et al. (2007a), Korsvik et al. (2010), Lin and Liu (2011) and Norstad et al. (2011) studied the less-than-shipload problem, in which several cargoes are allowed to be onboard at the same time. The cargo size is fixed in their studies. Brønmo et al. (2007b), Korsvik and Fagerholt (2010) and Brønmo et al. (2010) allowed the shipping company to choose the quantity of each cargo to be transported, and therefore the delivery quantity becomes a variable between given upper and lower bounds. In Henrik et al. (2011), Korsvik et al. (2011) and Stålhane et al. (2012), a single cargo can be split between multiple ships.

The work described in this paper originated from a routing and scheduling problem encountered by a large Danish product tanker shipping operator, who transports oil products in full-shipload mode. The shipping operator has a fleet of heterogeneous ships with different sizes and fuel consumptions. It receives cargo orders from the customers, each of which has a specific pickup port, a delivery port and a time window within which the cargo should be picked up. The operater needs to decide whether to accept the cargo, which ship should pickup the cargo at what time, and at which speed each ship should sail on each leg in order to maximize the overall profit. This problem is similar to the full-shipload routing and speed optimization problem considered in Gatica and Miranda (2011). The difference is that, in our problem, the fuel consumption of each ship depends individually on the sailing speed and the load of the ship, i.e., whether the ship is loaded (laden) or empty (ballast); whereas, in Gatica and Miranda (2011), the fuel consumption only depends on the speed. To the best of the authors' knowledge, the load dependent fuel consumption has not been considered in the literature. In addition, Gatica and Miranda (2011) did not propose any tailored solution method but relied on solving the models using an IP solver. We have formulated two mathematical models for the problem, developed a Branch-and-Price (B&P) algorithm with heuristic column generation, and tested the proposed algorithm on instances generated from real-life data. The computational results show that the branch-and-price heuristic produces optimal or near-optimal solutions in relatively short time.

The main contribution of this paper is the proposed heuristic. The heuristic is able to find high quality solutions in short computation time. It clearly outperforms a commercial IP solver when it comes to computation time and size of the instances that can be handled. In terms of modeling, the present paper is one of the first to consider variable speeds in a tramp shipping planning context and it is the first to consider a load dependent fuel consumption. Although the modeling is greatly simplified by the assumption of full-vessel loads, the assumption itself is realistic. The company used as a case in this project does not carry multiple orders simultaneously on a single ship.

The remainder of the paper is organized as follows. Two mathematical formulations are presented in Section 2 and the heuristic B&P algorithm is described in Section 3. The computational results are given in Section 4, followed by conclusions in Section 5.

2. Mathematical Model

Our problem can be defined on the graph $G = (N, A)$, where N is the set of all the nodes and A is the set of feasible arcs in the graph. Let S denote the set of ships. Each ship $s \in S$ starts from node $o(s)$ and ends at dummy node $d(s)$. Let O and D denote the set of all origins and destinations of the ships. Let N_0 denote the set of cargoes. As this is a full-shipload problem, each node $i \in N_0$ corresponds to a cargo, which is transported directly from its load port to discharge port, and is associated with a sailing distance g_i from the load port to the discharge port, a port service time t_i for loading and unloading the cargo and a time window $[a_i, b_i]$, within which the cargo should start being loaded. For $i \in O \cup D$, we set $g_i = 0$, $t_i = 0$, $a_i = 0$ and $b_i = \max_{j \in N_0}\{b_j\}$. The set of all the nodes is $N = N_0 \cup O \cup D$. Let $N_i^+ = \{j : (i, j) \in A\}$ and $N_i^- = \{j : (j, i) \in A\}$ be the set of nodes that can be reached from node i and can reach node i respectively. The distance between two nodes i and j is denoted as d_{ij}, which is the distance from the discharge port of cargo i, or the location of i if $i \in O$, to the load port of cargo j. For the dummy nodes $j \in D$, we set $d_{ij} = 0$, since we do not decide in advance where the ship should end its journey. The ships can sail at different speeds. Let V denote the set of speeds and l^v the time of sailing one distance unit at speed $v \in V$. The fuel consumption depends on the ship, the sailing speed and the load of the ship. Let e_s^v denote the ballast fuel cost of ship s sailing one distance unit at speed v. Let c_{is}^v be the cost of ship s sailing one distance unit at speed v with cargo i. It should be noted that c_{is}^v is determined by the light weight of the ship s, the weight of cargo i and the speed. Examples of approximation functions of c_{is}^v can be found in Psaraftis and Kontovas (2013). Due to the fact that the ships are different, not all the ships can serve all the cargoes. For example, a small ship cannot carry a large cargo, and a large ship with deep draft cannot serve a cargo with a shallow load or discharge port. Let binary variable p_{is} be 1 if is feasible for ship s to serve cargo i and 0 otherwise. Let r_i^v be the reward of cargo i when served at speed v.

2.1. A Three-Index Formulation

Let binary variable x_{ijs}^v be 1 iff ship s sails from cargo node i to cargo node j at speed v, and let binary variable y_{is}^v be 1 iff ship s serves cargo i at speed v. Let variable z_{is} denote the time when ship s starts loading cargo i, and auxiliary variable w_{ijs} denote the time span of ship s from arriving at the loading port of cargo i to arriving at the loading port of cargo j. The mathematical model can be presented as follows:

$$\min \sum_{i \in N_0} \sum_{s \in S} \sum_{v \in V} c_{is}^v g_i y_{is}^v + \sum_{(i,j) \in A} \sum_{s \in S} \sum_{v \in V} e_s^v d_{ij} x_{ijs}^v - \sum_{i \in N_0} \sum_{s \in S} \sum_{v \in V} r_i^v y_{is}^v \tag{1}$$

subject to:

$$\sum_{j \in N_i^+} \sum_{s \in S} \sum_{v \in V} x_{ijs}^v \leq 1 \qquad \forall i \in N_0 \tag{2}$$

$$\sum_{j \in N_{o(s)}^+} \sum_{v \in V} x_{o(s)js}^v = 1 \qquad \forall s \in S \tag{3}$$

$$\sum_{j \in N_i^+} \sum_{v \in V} x_{ijs}^v - \sum_{j \in N_i^-} \sum_{v \in V} x_{jis}^v = 0 \qquad \forall i \in N_0, s \in S \tag{4}$$

$$\sum_{j \in N_{d(s)}^-} \sum_{v \in V} x_{jd(s)s}^v = 1 \qquad \forall s \in S \tag{5}$$

$$t_i + \sum_{v \in V} g_i l^v y_{is}^v + \sum_{v \in V} d_{ij} l^v x_{ijs}^v = w_{ijs} \qquad \forall (i,j) \in A, s \in S \tag{6}$$

$$z_{is} + w_{ijs} - M(1 - \sum_{v \in V} x_{ijs}^v) \leq z_{js} \qquad \forall (i,j) \in A, s \in S \tag{7}$$

$$a_i \quad \leq \quad z_{is} \quad \leq b_i \qquad \forall i \in N, s \in S \tag{8}$$

$$\sum_{v \in V} y_{is}^v \leq p_{is} \qquad \forall i \in N_0, s \in S \tag{9}$$

$$\sum_{j\in N_i^+}\sum_{v\in V} x_{ijs}^v = \sum_{v\in V} y_{is}^v \qquad \forall i \in N_0, s \in S \tag{10}$$

$$x_{ijs}^v, y_{is}^v \in \{0,1\} \qquad \forall i,j \in N, v \in V, s \in S \tag{11}$$

$$z_{is} \geq 0 \qquad \forall i \in N, s \in S \tag{12}$$

$$w_{ijs} \geq 0 \qquad \forall i,j \in N, s \in S \tag{13}$$

The objective function (1) minimizes the sum of the transportation cost minus the total reward of the served cargoes. It can also be reformulated as maximization of the total profit. It should be noted that other ship costs, such as maintainance costs and crew salaries, are not considered in this work. Constraints (2) ensure that each cargo is served by at most one ship at one speed. If a subset of the cargoes is mandatory, we need to change the inequality in this constraint to an equality for those mandatory cargoes. Constraints (3) to (5) are flow conservation constraints. Constraints (6) calculate the time span w_{ijs} of ship $s \in S$ from the arrival at the loading port of cargo i to the arrival at the loading port of cargo j, which is the sum of the loading and unloading time of cargo i, the sailing time between the load and discharge ports of cargo i and the sailing time from cargo i to cargo j. Constraint (7) imposes that if ship s sails from i to j, then $z_{js} \geq z_{is} + w_{ijs}$. Parameter M is a sufficiently large number, and is set to $b_i + (t_i + \max_{v\in V} g_i l^v + \max_{v\in V} d_{ij} l^v) - a_i$. Constraints (8) are the time window constraints. Constraints (9) ensure that each cargo can only be served by a compatible ship. Constraints (10) enforce the relationship between x and y variables. Constraints (11), (12) and (13) define the variables.

In order to tighten and simplify the model, it is advantageous to only generate variables corresponding to arcs that are feasible. An arc (i,j) is infeasible, if $b_j < a_i + t_i + (g_i + d_{ij}) \cdot l^{v_{\max}}$, where $v_{\max}$ corresponds to the highest possible speed.

This model is solved by CPLEX 12.5. The tests show that removing infeasible arcs reduces computational time significantly. For an instance with 100 cargoes, 20 ships and a 60-day planning horizon, the number of arcs in the graph and the computational time is reduced by 78% and 66% respectively. This is because our problem is a full-shipload problem and the service time of a cargo, including the loading time, transportation time from pickup to delivery and unloading time, is relatively long compared to the pickup time window. Further computational results will be presented in Section 4.

2.2. *Set Packing Formulation*

This problem can also be formulated as a Set Packing problem. Let P^s be the set of feasible routes for ship $s \in S$. Let c_p^s denote the cost of route p for ship s, calculated as the transportation cost minus the total reward of the cargoes served on the route. Let a_{ip} equal 1 if route p covers cargo i, and 0 otherwise. Let the binary variable y_p^s be 1 if route p is taken by ship s, and 0 otherwise. The problem can be formulated as follows:

$$\min \sum_{s\in S} \sum_{p\in P^s} c_p^s y_p^s \tag{14}$$

subject to

$$\sum_{s\in S} \sum_{p\in P^s} a_{ip} y_p^s \leq 1 \qquad \forall i \in N_0 \tag{15}$$

$$\sum_{p\in P^s} y_p^s \leq 1 \qquad \forall s \in S \tag{16}$$

$$y_p^s \in \{0,1\} \qquad \forall p \in P^s, s \in S \tag{17}$$

The objective is to minimize the cost of the selected routes in such way that each cargo is covered by at most one ship and each ship is assigned to at most one route.

The LP relaxation of the set packing formulation will always provide the same or better lower bound compared to the LP relaxation of the three-index formulation.

3. Solution Method

Solving the set packing model (14)–(17) by listing all possible ship routes in P^s for all $s \in S$ and passing this problem to an IP solver seems impossible given the many possible routes for each ship and the many choices of sailing speeds for a single route. As a consequence, the model is solved using a heuristic branch-and-price algorithm (see e.g. Barnhart et al. 1998). The branch-and-price algorithm solves the linear programming (LP) relaxation of the problem (denoted as LP-SP in the following text), where (17) is replaced by

$$y_p^s \geq 0 \qquad \forall p \in P^s, s \in S.$$

This problem can be solved by column generation, where only a subset of the ship routes are firstly considered in the model and the rest of the routes that have the potential to improve the objective function are gradually generated and added to the model. Let $\bar{P}^s$ be the restricted set of routes for ship s, which can be initialized by single-cargo routes. The restricted LP with the restricted sets $\bar{P}^s$ is called *the master problem*. Solving the master problem gives dual variables π_i and λ^s corresponding to constraints (15) and (16), respectively. Using these dual variables, we can calculate the reduced cost of a path p for ship s as

$\hat{c}_p^s = c_p^s - \sum_{i \in N_0} a_{ip}\pi_i - \lambda^s$. If $\hat{c}_p^s \geq 0$ for all $s \in S$ and all $p \in P^s$ (that is, considering all feasible routes), then the current LP solution to the restricted problem is optimal for the full LP model. If there exist an $s \in S$ and a $p \in P^s$ such that $\hat{c}_p^s < 0$ then the corresponding variable has a chance of producing improved LP solution. It should be added to $\bar{P}^s$ and the LP should be resolved to obtain new dual variables. Finding the $s \in S$ and $p \in P^s$ that results in the lowest $\hat{c}_p^s$ is done by solving a *pricing problem.* In our case the pricing problem is an elementary shortest path problem with time windows and variable speeds. We solve a problem for each ship $s \in S$, aiming at finding the cheapest path from $o(s)$ to $d(s)$ satisfying the time windows. The information given by the dual variables is encoded in the arc costs of the shortest path problem. More precisely, in the shortest path problem related to ship s, the cost $\bar{c}_{ij}^{v_1 v_2}$ of an arc (i, j) with a traverse speed v_1 on the laden part of the journey, i.e. the sailing distance between cargo i's load and discharge ports, and speed v_2 on the ballast part of the travel (from node i to node j) is set to

$$\bar{c}_{ij}^{v_1 v_2} = \begin{cases} c_{is}^{v_1} g_i + e_s^{v_2} \cdot d_{ij} - \pi_i & \text{if } i \in N_0 \\ e_s^{v_2} \cdot d_{ij} - \lambda^s & \text{if } i = o(s) \end{cases}$$

By using these arc costs, it is easy to show that the cost of a path will equal the reduced cost of the corresponding variable. The resource constrained shortest path problem is usually solved by using labeling algorithms (Irnich and Desaulniers 2005). However, in this paper, we use a simple heuristic to ensure that the computational time of the pricing problem is short even for large instances. The heuristic is described in Section 3.2. Using a heuristic for solving the pricing problem implies that we do not know if the algorithm has solved LP-SP to optimality. Even if the heuristic cannot find a path with negative cost, we cannot rule out the existence of a path that could improve the current solution to LP-SP. However, the algorithm proceeds as if we had solved the LP to optimality. If the LP relaxation only takes integer values, the algorithm stops and outputs the solution. If some variables take fractional values, the algorithm will branch in order to find an integer solution. We choose to branch on the arcs used in the subproblem, i.e., whether the ships should use an arc or not. From an implementation point of view, it is handled in the way described by Desrochers et al. (1992).

The algorithm uses strong branching in order to decide which arc to branch on. A number, β, of branching candidates are evaluated by enforcing the branch and computing the resultant improvement in the lower bounds (Δ_1 and Δ_2) in the two child nodes. Following Linderoth and Savelsbergh (1999), the algorithm chooses the branch that maximizes

$$\alpha \min\{\Delta_1, \Delta_2\} + (1 - \alpha) \max\{\Delta_1, \Delta_2\}$$

where $0 \leq \alpha \leq 1$ is a parameter.

If the pricing problem were solved to optimality and the branch-and-bound algorithm were allowed to run until all relevant nodes in the search tree had been explored, it would either return the optimal solution or state that no feasible solution exists. Since the implemented algorithm only solves the pricing problem using a heuristic, we cannot guarantee that the optimal solution is found, but the solution quality is in general good as will be seen in Section 4.2.

After finishing the branch-and-bound search, the algorithm uses a generic IP solver to solve (14)–(17) using the generated columns. This can give improved solutions even when the branch-and-bound search has run to the end. This is because the heuristic branch-and-price algorithm can fathom a node erroneously due to the missing of an important column. If this column is generated at a later stage then solving the model with these additional generated columns may yield an improved solution.

3.1. Preprocessing

The purpose of the preprocessing is to reduce the solution space by identifying incompatibilities between the cargoes. Two cargoes are incompatible if it is infeasible to serve them on the same route without violating their time windows, i.e., $a_i + T_{ij}^{\min} > b_j$ or $a_j + T_{ji}^{\min} > b_i$, where $T_{ij}^{\min}$ denotes the minimum time required to serve cargo i and to travel from cargo i to cargo j. The first inequality states that cargo j can not be reached before the end of its time window even if a ship starts serving cargo i at the beginning of its time window, uses the minimum time to service cargo i and sail to cargo j. The second inequality describes the opposite. Any route that contains two incompatible cargoes is infeasible.

Let N_i^{INF}denote the set of cargoes that are incompatible with cargo i and $|N_i^{\text{INF}}|$ the number of cargoes incompatible with i. Figure 1 shows $\left|N_i^{\text{INF}}\right|$ for each cargo in an instance with 160 cargoes in a 90-day planning horizon. The cargoes (x-axis) are sorted by their $|N_i^{\text{INF}}|$ in a non-decreasing order. As can be seen from the figure, for most of the cargoes, the number of incompatible cargoes is larger than 60, and there are 11 cargoes, with more than 100 incompatible cargoes. These cargoes are characterized by a long distance between the load and discharge port and long service time, which makes it difficult for other cargoes to fit in the same route.

The result of the preprocessing is used in solving the pricing problem. The pricing heuristic will never try to put two incompatible cargoes into the same route. This helps to reduce the solution space and shorten the computational time of solving the pricing problem. The tests show that, for an instance with 100 cargoes and 20 ships within a 60-day planning horizon, the proposed preprocessing can reduce the computational time by more than 80% (from 46 seconds to 9 seconds).

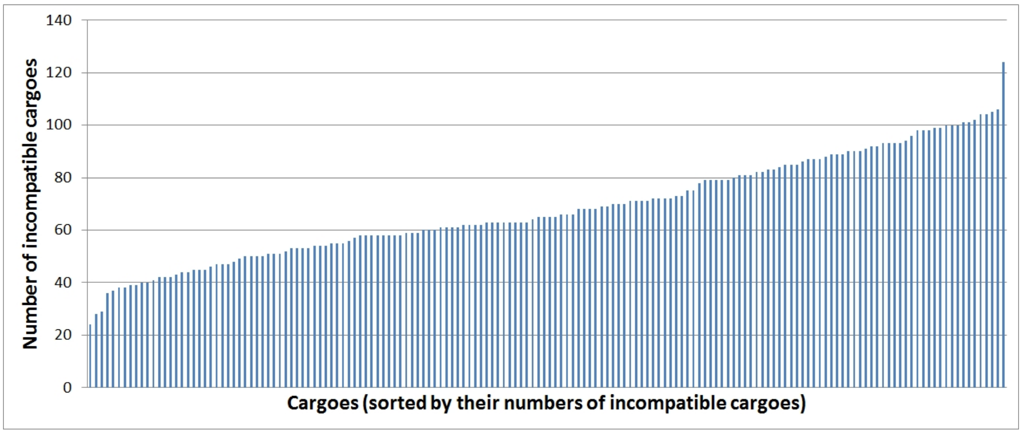

Figure 1: The number of incompatible cargoes for each cargo.

3.2. Heuristic algorithm for solving the pricing problem

Our pricing heuristic method is described in Algorithm 1. It initially makes a random permutation of the ships, which determines the order of processing.

For each ship s, the algorithm tries to find a list of routes, R, with negative reduced cost by means of an insertion heuristic. The insertion heuristic first initializes the route r with a seed cargo i (every cargo is treated as a seed node). Let the set of nodes N' denote the remaining candidate cargoes to be inserted, $N' = N_0 \backslash \{\{i\} \cup N_i^{\text{INF}}\}$. At each iteration, the algorithm attempts to insert every cargo $j \in N^{'}$ into every possible position on route r at every possible speed on the sailing legs from the precedent cargo to j, between the load and discharge ports of j, and from j to the successor cargo. The best cargo j^* with the minimun insertion cost is selected for insertion and N' is updated by removing cargo j^* and its incompatible cargoes. Whenever a node is inserted, the reduced cost of the resultant route r is updated. If it is negative, then the route is added to R. Node insertion and updating of N' and R are repeated until there are no nodes left in N'. The same insertion heuristic is repeated for every feasible seed cargo. If the resultant route list R for ship s is not empty, the pricing heuristic stops, filters R by removing the dominated routes in a heuristic way. Otherwise, the algorithm proceeds with the next ship, and repeats the same procedure until it finds a ship, for which the route list R is not empty, or until all ships have been tried.

Algorithm 1 Heuristic algorithm for the pricing problem.

```
S: the set of ships sorted in a random order;
R: the list of routes to be added into the master problem;
N_0: the set of all the cargoes
N_i^{INF}: the set of cargoes that are incompatible with cargo i;
R ← ∅
for (ship s ∈ S) do
    for (seed cargo i ∈ N_0) do
        Create an empty route r for ship s
        N' ← N_0;
        if (i can be inserted to r) then
            Insert i into route r;
            N' ← N' \ {{i} ∪ N_i^{INF}};
            if (r has negative reduced cost) then
                Update R;
            while N' is not empty do
                j* ← Find the best node in N' that can be inserted into r
                if j* is empty then
                    N' ← ∅
                else
                    Insert j into route r;
                    N' ← N' \ {{j*} ∪ N_{j*}^{INF}}
                    if (r has negative reduced cost) then
                        Update R;
    if R is not empty then
        Return R;
Return R;
```

4. Computational experiments

Our algorithm is implemented in C++ and executed on a computer with a 2.90GHz Intel Core i7-3520M processor that has two cores (and doubles the amount of virtual cores through the *hyper threading* feature). The computer has 8 GB of RAM. The model (1)–(13) is solved by CPLEX 12.5 on the same computer. CPLEX takes advantage of all available cores while our heuristic only uses one core. The data used for testing the algorithm is described in Section 4.1 and the computational results are presented in Section 4.2. The parameters α and β in strong branching were set to $\frac{3}{4}$ and 15, respectively, the same values as in Dabia et al. (2013) and Ropke (2013).

4.1. Data

The test instances are generated from real-life data provided by the tanker shipping operator. The distributions of the generated cargoes in the test instances approximate the observed distributions in the real-life dataset in terms of traveling distance, geographical location of the load and discharge ports, load and discharge times, and time windows. In practice, a preferred service (laden) speed, at which the cargo should be shipped, and the corresponding reward are fixed after a business negotiation between the cargo owner and the shipping company. In this work, in order to investigate the effect of variable speeds, we allow the cargoes to be served at different speeds and the corresponding rewards to be dependent on the selected sailing speeds. Let $v_i^{\text{pref}} \in V$ denote the preferred service speed of cargo i. The reward of cargo i at speed $v \in V$, r_i^v, is determined by the fuel consumption over the total sailing distance of the cargo and the deviation from the cargo's preferred speed, i.e. $|v - v_i^{\text{pref}}|$, multiplied by a random element for variation. The sailing speed is discretized in the real-life data and the set of speeds is $V = \{10, 10.5, ..., 16\}$ knots (1 knot equals 1.852 km per hour). As the fleet of ships is heterogeneous, the feasible sailing speeds and the corresponding fuel consumptions vary between ships. For example, a heavy and slow ships can sail between 12 and 15.5 knots, with a fuel consumption at 14 knots of 39.5 and 41.5 tons per day for ballast and laden sailing respectively, while a lighter ship consumes 29.5 and 34 tons per day at the same speed.

Thus, the test data are generated by approximating the long-term statistics of the cargoes and using the same practical sailing speed options and fuel consumption functions as in the real-life data. The dataset consists of twenty instances involving up to 250 cargoes and 32 ships in a 135-day planning horizon. The number of cargoes, the number of ships and the length of the planning horizon of each instance are given in Table 1.

Table 1: The ten instances generated from real-life data.

	Number of cargoes	Planning horizon (days)	Number of ships
c40_d30_s20	40	30	20
c40_d30_s32	40	30	32
c70_d45_s20	70	45	20
c70_d45_s32	70	45	32
c100_d60_s20	100	60	20
c100_d60_s32	100	60	32
c130_d75_s20	130	75	20
c130_d75_s32	130	75	32
c160_d90_s20	160	90	20
c160_d90_s32	160	90	32

4.2. Results

We test our algorithm on the generated instances, and compare the heuristic solutions to the optimal solutions. We also investigate the effects of allowing variable speeds and analyze how the fuel price affects the solutions.

The performance of the proposed heuristic method

The problem of simultaneously optimizing routing and sailing speed is solved by the proposed heuristic approach and by solving model (1)–(13) using CPLEX 12.5. A comparison between the two solution approaches is given in Table 2. Columns Z^H and Z^* are the profits obtained by the heuristic method and CPLEX respectively. The computational time in seconds and the number of served cargoes are given in columns T and n_s. The last column shows the optimality gap in percentage, which is calculated as $\frac{Z^H - Z^*}{Z^*} \cdot 100$. The instances of which CPLEX runs out of memory are indicated by "–" in the table. The limit on the computational time for the heursitc is set to 3600 seconds and for CPLEX 24 hours.

As can be seen from the table, CPLEX managed to find the optimal solutions for all the instances with up to 160 cargoes. Larger instances can be solved directly by CPLEX for our problem than for the vehicle routing problems. This is because, as mentioned earlier, our problem is a full-shipload shipping problem and the service time of a cargo, including the loading time, transportation time from pickup to delivery and unloading time, is relatively long compared to the pickup time window. Therefore the solution space is not as large as for a vehicle routing problem of the same size.

Our heuristic is able to find optimal solutions to three small instances and near-optimal solutions to the remaining instances. The optimality gap is consistently below 4.08% for the instances that CPLEX succeeded in finding the optimal solutions. In terms of computational time, the heuristic approach outperforms the exact method significantly. The average running time of the heuristic is less than 5% of that of CPLEX, and the heuristic managed to find solutions to all the instances. The numbers of served cargoes given by the two solution approaches are very close. In both heuristic and exact solutions, the ship's waiting time is approximately 15% of the entire planning horizon on average.

Table 2: The results obtained by the heuristic method and CPLEX for the ten test instances.

	Heuristic solution			Exact solution			
	Z^H	T (s)	n_s	Z^*	T (s)	n_s	Gap (%)
c40_d30_s20	5,497,357	0.2	20	5,497,357	3.0	20	0.00
c40_d30_s32	7,455,752	0.5	31	7,455,752	7.9	31	0.00
c70_d45_s20	8,222,284	2.5	33	8,222,892	52.8	33	-0.01
c70_d45_s32	11,735,410	3.3	49	11,735,410	117.6	49	0.00
c100_d60_s20	13,132,563	9.0	43	13,135,308	34.2	43	-0.02
c100_d60_s32	15,706,468	41.6	58	15,746,738	176.7	57	-0.26
c130_d75_s20	15,615,057	25.0	53	15,651,603	131.9	53	-0.23
c130_d75_s32	16,876,775	74.8	77	16,905,159	502.6	78	-0.17
c160_d90_s20	16,683,846	303.3	56	16,765,250	2,958.6	57	-0.49
c160_d90_s32	24,237,644	843.0	92	24,366,858	53,509.7	91	-0.53
Average	13,516,315	130.3	51.2	13,548,232	5,747.5	51.2	-0.17

In Table 3, we show detailed statistics for the heuristic. The first column shows the instance name. The second column shows the total time spent by the heuristic. The third column shows if the branch and bound search finished (X) or if it was stopped prematurely (-). The fourth column shows the number of branch-and-bound nodes explored. The fifth column shows the number of columns generated. And the last column shows the time spent in the pricing heuristic. It is obvious that most of the time is spent on the pricing problem, which means that only a small fraction of time is spent on solving LP and the necessary bookkeeping within the algorithm. It is also interesting to note that only a small number of branch-and-bound nodes have been searched.

The effect of speed optimization

In order to evaluate the effects of speed optimization, we compared the solutions to the multiple-speed case with those to the single-speed case. In the single-speed case, the laden sailing speed of a ship is equal to the preferred service speed of the cargo, and the ballast sailing speed is set to a constant speed, which is the smaller value of 12 knots (a realistic choice according to the company's statistic) and the ship's

Table 3: Detailed results for the heuristic

Instance	T_{tot} (s)	Fin.	#BB Nodes	#Columns	$T_{pricing}$ (s)
c40_d30_w1_s20	0.2	X	0	235	0.2
c40_d30_w1_s32	0.5	X	0	339	0.4
c70_d45_w1_s20	2.3	X	0	550	2.3
c70_d45_w1_s32	3.3	X	0	946	3.2
c100_d60_w1_s20	9.0	X	0	1112	8.9
c100_d60_w1_s32	41.6	X	2	1734	41.3
c130_d75_w1_s20	25.0	X	0	1810	24.9
c130_d75_w1_s32	74.8	X	0	2690	74.5
c160_d90_w1_s20	303.3	X	2	2185	302.8
c160_d90_w1_s32	843.0	X	2	3542	841.7

lowest feasible ballast speed, whereas in the multiple-speed case, both the laden and ballast speeds are variables. The profit, computational time and number of served cargoes for both cases are provided in Table 4.

For the multiple-speed case, column $\bar{v}_{\text{diff}}$ shows the average difference in knots between the actual service speed and the preferred speed over all the served cargoes. It is calculated as $\frac{\sum_{i \in N_S} |v_i - v_i^{\text{pref}}|}{n_s}$, where N_S is the set of served cargoes, n_s is the number of served cargoes, and v_i is the actual service speed of cargo i. The column Gap_1 presents the optimality gap of the single-speed case, which is calculated as $\frac{Z_S^H - Z_S^*}{Z_S^*} \cdot 100$. The column Gap_2 shows the gap between the heuristic solution to the multiple-speed case and that to the single-speed case, which is calculated as $\frac{Z_M^H - Z_S^H}{Z_S^H} \cdot 100$.

For the single-speed case, the heuristic finds the optimal solution for ten instances using on average 7% of the computational time used by CPLEX. By allowing speed variation, we gained a profit improvement ranging from 9.82% to 21.80% and an average improvement of 16%. We can also see that adding multiple speeds makes the problem significantly harder both for the MIP model and the heuristic branch-and-price method. Although the running time of the multi-speed heuristic approach increases compared to the single-speed heuristic method, it is still within an acceptable range, and shorter than that of CPLEX for the single-speed case for almost all the instances. In addition, the multiple-speed solutions yield on average 13% more served cargoes.

Table 4: A comparison between the multiple-speed case and the single-speed case.

	Heuristic solution (multiple-speed)				Heuristic solution (single-speed)			Exact solution (single-speed)				
	Z_M^H	T (s)	n_s	$\overline{v}_{\text{diff}}$	Z_S^H	T (s)	n_s	Z_S^*	T (s)	n_s	Gap$_1$	Gap$_2$
c40_d30_s20	5,497,357	0.2	20	1.35	4,636,034	0.03	18	4,636,034	2.4	18	0.00	18.58
c40_d30_s32	7,455,752	0.5	31	1.13	6,789,020	0.03	28	6,789,020	6.3	28	0.00	9.82
c70_d45_s20	8,222,284	2.5	33	2.00	6,750,725	0.05	26	6,750,725	22.1	26	0.00	21.80
c70_d45_s32	11,735,410	3.3	49	1.40	9,796,283	0.08	43	9,796,283	25.0	43	0.00	19.79
c100_d60_s20	13,132,563	9.0	43	1.30	11,438,435	0.09	40	11,438,435	31.7	40	0.00	14.81
c100_d60_s32	15,706,468	41.6	58	1.46	13,640,487	0.41	51	13,640,487	84.4	51	0.00	15.15
c130_d75_s20	15,615,057	25.0	53	1.35	13,554,505	0.19	49	13,554,505	167.9	49	0.00	15.20
c130_d75_s32	16,876,775	74.8	77	1.19	14,762,244	3.74	69	14,762,244	371.1	69	0.00	14.32
c160_d90_s20	16,683,846	303.3	56	1.46	14,139,457	0.44	51	14,139,457	372.4	51	0.00	18.00
c160_d90_s32	24,237,644	843.0	92	1.51	20,752,350	5.66	79	20,769,526	1,188.5	80	-0.08	16.79
Average	13,516,315	130.3	51.2	1.41	11,625,953	1.07	45.4	11,627,671	227.1	45.5	-0.01	16.43

Sensitivity to the fuel price

The price of fuel oil fluctuates in real life. Here we investigate how this affects the speed selection and profit by running the instance c130_d75_s32 with different fuel prices, ranging from 50% to 200% of the current price. The detailed results are illustrated in Table 5. The first column shows the price factor in percentage relative to the price used in the previous simulations. Column $n_s^>$ gives the number of cargoes served at a higher speed than preferred and column $n_s^<$ provides the number of cargoes served at a lower speed than preferred. The last column $\overline{v}$ is the average speed in knots for all served cargoes.

Figure 2 shows how the number of served cargoes, the number of cargoes served at a higher speed than preferred, and the number of cargoes served at a lower speed than preferred change as the fuel price increases from 50% to 200% of the current price. It can be seen that, as fuel price goes up, the number of served cargoes n_s generally drops because some cargoes are no longer profitable given the high fuel price. The tendency of serving the cargoes at lower speeds, as fuel price increases, is also clearly seen in the figure. The number of served cargoes is especially sensitive when the fuel price goes beyond the current price, whereas the number of cargoes served at a higher speed is very sensitive to fuel price throughout the whole range. The number of cargoes served at a lower speed keeps increasing until the fuel price reaches 160% of the current price and then drops. This is because in the beginning, it is preferred to serve the cargoes at lower speeds but when the fuel price reaches a high level, it is more desirable to ignore the unprofitable cargoes. The results in Table 5 show that when the price increases from 50% to 200% of the current price, the number of served cargoes drops by 72.6%, the percentage of cargoes served at a higher-than-preferred speed decreases from 86.90% to 8.70% and the percentage of cargoes served at a lower-than-preferred speed increases from 0 to 86.93%.

Figure 3 depicts the profit and the average speed as a function of fuel price. The average speed decreases approximately linearly as a function of the fuel price increases. The profit decreases very fast as the fuel price increases and the decrease rate gradually slows down. When the price increases from 50% to 200% of the current price, the profit is decreased by 96.8% and the average speed is decreased by 30.1%.

In summary, the fuel price has significant effects on the profit and the average sailing speed, which shows the importance and benefit of applying speed optimization in a market with changing fuel prices. However, in practice, if the fuel price increases significantly, the reward for serving each cargo is expected to increase as well, and thus, the increased cost will be shared with the cargo owners.

Table 5: The results of instance c130_d75_s32 with different fuel prices.

Price factor (%)	Z^H	n_s	$n_s^>$	$\frac{n_s^>}{n_s} \cdot 100$	$n_s^<$	$\frac{n_s^<}{n_s} \cdot 100$	$\bar{v}$ (knot)
50	34,375,102	84	73	86.90	0	0.00	15.23
60	30,418,937	84	66	78.57	0	0.00	14.79
70	26,691,163	83	63	75.90	1	1.20	14.47
80	23,193,509	81	50	61.73	5	6.17	14.04
90	19,976,088	80	49	61.25	7	8.75	13.93
100	16,876,775	77	45	58.44	12	15.58	13.66
110	14,119,083	68	33	48.53	19	27.94	13.39
120	11,887,260	63	25	39.68	22	34.92	13.12
130	9,901,069	57	22	38.60	22	38.60	12.85
140	8,059,775	54	19	35.19	25	46.30	12.42
150	6,339,031	55	19	34.55	27	49.09	12.31
160	4,823,284	47	12	25.53	30	63.83	11.93
170	3,594,923	41	10	24.39	27	65.85	11.50
180	2,591,039	33	7	21.21	24	72.73	11.29
190	1,785,860	30	4	13.33	23	76.67	11.07
200	1,090,434	23	2	8.70	20	86.96	10.64

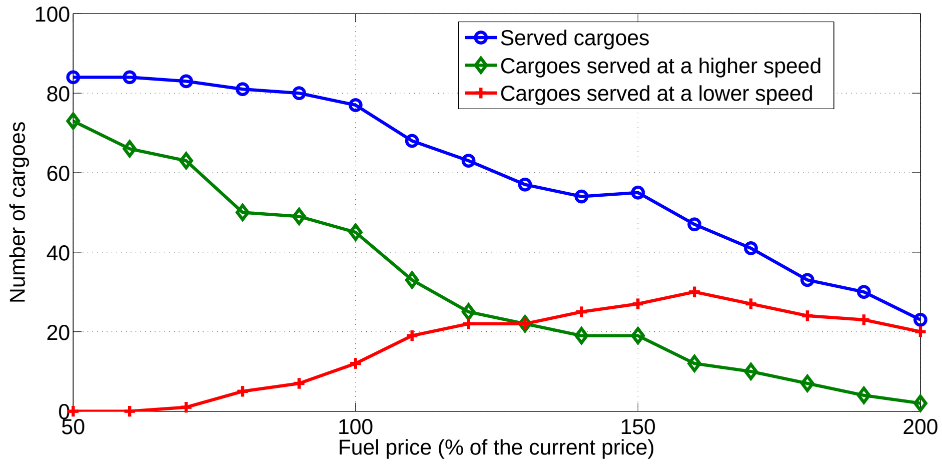

Figure 2: The number of served cargoes, the number of cargoes served at a higher-than-preferred speed and the number of cargoes served at a lower-than-preferred speed for different fuel prices.

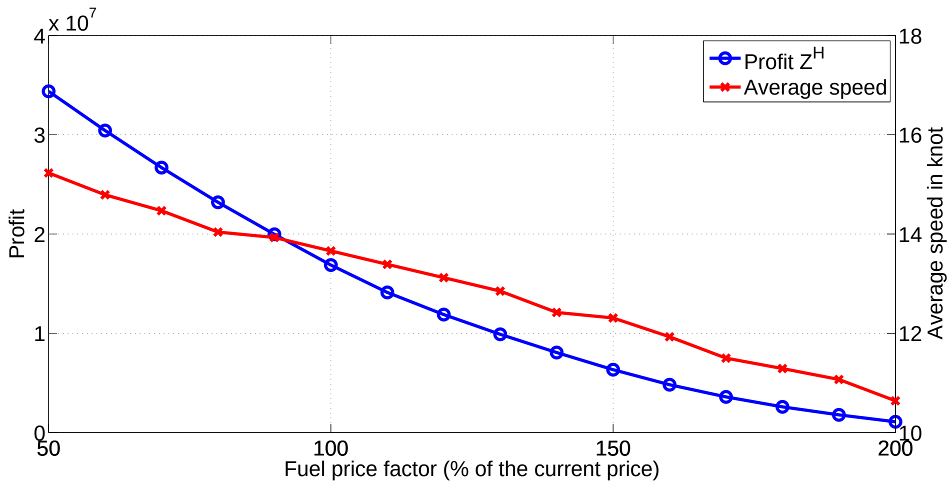

Figure 3: The profit and average speed in knots for different fuel prices.

5. Conclusion

In this paper, we have considered the simultaneous optimization of routing and sailing speed in the context of full-shipload tramp shipping encountered by a large Danish product tanker shipping operator. A number of cargoes with specified pick-up time windows can be transported from load to discharge ports using a fleet of heterogeneous ships of different speed ranges and fuel consumption. The ships can sail at certain discretized speeds. For each ship, the fuel consumption depends on the shipload and sailing speed, and is obtained from real-life data. This practical consideration is different from previous studies, where fuel consumption is load independent. The objective of the problem is to find the optimal route for each ship and the optimal sailing speed for each leg on the route to maximize the total profit of transportation.

We have presented a three-index formulation and a set packing formulation for this problem, and proposed a B&P algorithm with heuristic column generation and efficient data preprocessing. The algorithm is implemented and tested on instances generated by approximating long-term statistics and using the practical speed options and fuel consumption functions from real-life data. The computational results show that our heuristic algorithm is able to find optimal solutions to the small instances and near-optimal solutions to the large instances with optimality gaps consistently below 4.08%. The running time of the heuristic method is significantly shorter than that of CPLEX (on average around 5% of the latter). We have compared the solutions with speed optimization to those to a single-speed case, in which the cargoes can only be served at individual fixed speeds, and found that by allowing speed variation, we can gain on average 16% more profit and serve on average 13% more cargoes. We have also tested the algorithm with different fuel prices, ranging from 50% to 200% of the current fuel price, and observed that the solution is sensitive to fuel price changes. A higher fuel price leads to lower profit, fewer served cargoes and slower average sailing speed.

Acknowledgments

This work was supported by the Danish Maritime Fund. This support is gratefully acknowledged. The authors also thank the anonymous referee for the valuable comments that contributed to improve the quality of the paper.

Bibliography

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

G. Brønmo, B. Nygreen, and J. Lysgaard. Column generation approaches to ship scheduling with flexible cargo sizes. *European Journal of Operational Research*, 200:139–150, 2010.

G. Brønmo, M. Christiansen, K. Fagerholt, and B. Nygreen. A multi-start local search heuristic for ship scheduling - a computational study. *Computers & Operations Research*, 34(3):900–917, MAR 2007a. ISSN 0305-0548. doi: {10.1016/j.cor.2005.05.017}.

G. Brønmo, M. Christiansen, and B. Nygreen. Ship routing and scheduling with flexible cargo sizes. *Journal of the Operational Research Society*, 58(9):1167–1177, SEP 2007b. ISSN 0160-5682. doi: {10.1057/palgrave.jors.2602263}.

Ø. Buhaug, J. J. Corbett, Ø. Endresen, V. Eyring, J. Faber, S. Hanayama, D. S. Lee, D. Lee, H. Lindstad, A. Z. Markowska, A. Mjelde, D. Nelissen, J. Nilsen, C. Pålsson, J. J. Winebrake, W-Q Wu, and K. Yoshida, April 2009. Second IMO GHG study, International Maritime Organization (IMO) London, UK.

M. Christiansen, K. Fagerholt, B. Nygreen, and D. Ronen. Maritime transportation. In Cynthia Barnhart and Gilbert Laporte, editors, *Transportation*, volume 14 of *Handbooks in operations research and management science*, chapter 4, pages 189–284. Elsevier, 2007.

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

J. J. Corbett, H. Wang, and J. J. Winebrake. The effectiveness and costs of speed reductions on emissions from international shipping. *Transportation Research Part D-Transportation and Environment*, 14(8):593–598, DEC 2009. ISSN 1361-9209. doi: {10.1016/j.trd.2009.08.005}.

S. Dabia, S. Ropke, T. van Woensel, and T. De Kok. Branch and cut and price for the time dependent vehicle routing problem with time windows. 2013. Forthcoming.

M. Desrochers, J. Desrosiers, and M. Solomon. A new optimization algorithm for the vehicle routing problem with time windows. 40(2):342–354, 1992.

B. Eksioglu, A. V. Vural, and A. Reisman. The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering*, 57(4):1472–1483, 2009.

K. Fagerholt, G. Laporte, and I. Norstad. Reducing fuel emissions by optimizing speed on shipping routes. *Journal of the Operational Research Society*, 61(3, SI):523–529, MAR 2010. ISSN 0160-5682. doi: {10.1057/jors.2009.77}.

R. A. Gatica and P. A. Miranda. Special Issue on Latin-American Research: A Time Based Discretization Approach for Ship Routing and Scheduling with Variable Speed. *Networks & Spatial Economics*, 11(3, SI):465–485, SEP 2011. ISSN 1566-113X. doi: {10.1007/s11067-010-9132-9}.

M. M. Golias, G. K. Saharidis, M. Boile, S. Theofanis, and M. G. Ierapetritou. The berth allocation problem: optimizing vessel arrival time. *Maritime Economics and Logistics*, 11:358–377, 2009.

A. Henrik, M. Christiansen, and K. Fagerholt. The maritime pickup and delivery problem with time windows and split loads. *Information Systems and Operational Research*, 49(2):79–91, 2011.

S. Irnich and G. Desaulniers. *Shortest path problems with resource constraints*. Springer US, 2005.

J. E. Korsvik and K. Fagerholt. A tabu search heuristic for ship routing and scheduling with flexible cargo quantities. *Journal of Heuristics*, 16(2):117–137, APR 2010. ISSN 1381-1231. doi: {10.1007/s10732-008-9092-0}.

J. E. Korsvik, K. Fagerholt, and G. Laporte. A tabu search heuristic for ship routing and scheduling. *Journal of the Operational Research Society*, 61(4):594–603, APR 2010. ISSN 0160-5682. doi: {10.1057/jors.2008.192}.

J. E. Korsvik, K. Fagerholt, and G. Laporte. A large neighbourhood search heuristic for ship routing and scheduling with split loads. *Computers & Operations Research*, 38(2):474–483, FEB 2011. ISSN 0305-0548. doi: {10.1016/j.cor.2010.07.005}.

G. Laporte, P. Toth, and D. Vigo. Vehicle routing: historical perspective and recent contributions. *EURO Journal on Transportation and Logistics*, 2(1-2):1–4, 2013.

D.-Y. Lin and H.-Y. Liu. Combined ship allocation, routing and freight assignment in tramp shipping. *Transportation Research Part E-Logistics and Transportation Review*, 47(4):414–431, JUL 2011. ISSN 1366-5545. doi: {10.1016/j.tre.2010.12.003}.

J.T. Linderoth and M.W.P. Savelsbergh. A computational study of search strategies for mixed integer programming. 11(2):173–187, 1999.

H. Lindstad, B. E. Asbjørnslett, and A. H. Strømman. Reductions in greenhouse gas emissions and cost by shipping at lower speeds. *Energy Policy*, 39(6):3456–3464, JUN 2011. ISSN 0301-4215. doi: {10.1016/j.enpol.2011.03.044}.

I. Norstad, K. Fagerholt, and G. Laporte. Tramp ship routing and scheduling with speed optimization. *Transportation Research Part C-Emerging Technologies*, 19(5, SI):853–865, AUG 2011. ISSN 0968-090X. doi: {10.1016/j.trc.2010.05.001}. 4th International Workshop on Freight Transportation and Logistics, Cesme Izmir, TURKEY, MAY 26-29, 2009.

T. E. Notteboom. The time factor in liner shipping services. *Maritime Economics and Logistics*, 8:19–39, 2006.

OCEANA. Shipping industry on benefits of slower steaming. http://oceana.org/sites/default/files/o/fileadmin/oceana/uploads/Climate_Change/Shipping_Industry_on_Benefits_of_Slower_Steaming_July_2008.pdf, July 2008. Accessed on June 29, 2013.

H. N. Psaraftis and Kontovas. Speed models for energy-efficient maritime transportation: A taxonomy and survey. *Transportation Research Part C: Emerging Technologies*, 26(0):331 – 351, 2013. ISSN 0968-090X. doi: http://dx.doi.org/10.1016/j.trc.2012.09.012. URL http://www.sciencedirect.com/science/article/pii/S0968090X12001246.

D. Ronen. The effect of oil price on the optimal speed of ships. *Journal of the Operational Research Society*, 33(11):1035–1040, 1982.

D. Ronen. The effect of oil price on containership speed and fleet size. *Journal of the Operational Research Society*, 62(1):211–216, JAN 2011. ISSN 0160-5682. doi: {10.1057/jors.2009.169}.

S Ropke. Strong branching in branch-and-price-and-cut algorithms for vehicle routing problems. Technical report, DTU Management Science, 2013. Working paper.

S. C. Ryder and D. Chappell. Optimal speed and ship size for the liner trades. *Maritime Policy and Management*, 7 (1):55–57, 1980.

M. Stålhane, H. Andersson, M. Christiansen, J.-F. Cordeau, and G. Desaulniers. A branch-price-and-cut method for a ship routing and scheduling problem with split loads. *Computers & Operations Research*, 39:3361–3375, 2012.

UNCTAD. *Review of Maritime Transport 2012*. United Nations, 2013.

S. Wang and Q. Meng. Sailing speed optimization for container ships in a liner shipping network. *Transportation Research Part E-Logistics and Transportation review*, 48(3):701–714, MAY 2012. ISSN 1366-5545. doi: {10.1016/j.tre.2011.12.003}.