

Ann Oper Res (2013) 207:67–81
DOI 10.1007/s10479-011-0888-1

Routing and scheduling in project shipping

Kjetil Fagerholt · Lars Magnus Hvattum ·
Trond A.V. Johnsen · Jarl Eirik Korsvik

Published online: 19 May 2011

© The Author(s) 2011. This article is published with open access at Springerlink.com

Abstract Tramp shipping companies are committed to transport a set of contracted cargoes and try to derive additional revenue from carrying optional spot cargoes. Here, we present a real life ship routing and scheduling problem for a shipping company operating in project shipping, a special segment of tramp shipping. This segment differs from more traditional tramp segments, as the cargoes are usually transported on a one-time basis. Because of the special nature of the cargoes, complicating requirements regarding stowage onboard the ships and cargo coupling must be considered while determining routes and schedules for the ships in the fleet. A mathematical model is presented and a tabu search heuristic is proposed to solve the problem. Computational results show that the tabu search heuristic provides optimal or near-optimal solutions in a reasonable amount of time, and that it can give significant improvements to manual planning for the shipping company.

Keywords Maritime transportation · Tabu search · Stowage · Cargo coupling

K. Fagerholt (✉) · L.M. Hvattum

Department of Industrial Economics and Technology, Norwegian University of Science and Technology, Alfred Getz veg 3, 7491 Trondheim, Norway

e-mail: kjetil.fagerholt@iot.ntnu.no

L.M. Hvattum

e-mail: lars.m.hvattum@iot.ntnu.no

T.A.V. Johnsen · J.E. Korsvik

Norwegian Marine Technology Research Institute (MARINTEK), POB 4125 Valentinlyst, 7450 Trondheim, Norway

T.A.V. Johnsen

e-mail: trond.johnsen@marintek.sintef.no

J.E. Korsvik

Department of Marine Technology, Norwegian University of Science and Technology, 7491 Trondheim, Norway

e-mail: jarl.korsvik@marintek.sintef.no

1 Introduction

Maritime transportation is the major artery of international trade with an estimated share between 65% and 85% (UNCTAD 2007). Ship routing and scheduling constitutes one of the most important functions in many shipping companies, and proper routing and scheduling is therefore crucial, as a modest improvement in fleet utilization can result in large profit improvements. Decision support tools based on operations research are becoming more common within the shipping industry, but shipping companies are a heterogeneous group, and the set of restrictions that apply to the operations differ between each company. Hence, models and solution methods used to provide decision support require special adaptations to accommodate different segments of the shipping industry.

We study a special version of tramp shipping. A shipping company operating in the tramp market usually has a set of mandatory contract cargoes it is committed to carry, and will try to increase its revenue by transporting optional spot cargoes. The mandatory contract cargoes come from long-term agreements between the shipping company and the cargo owners. Each cargo in the given planning period must be picked up at its port of loading, transported and then delivered to its corresponding discharging port. There are time windows given, within which the loading of the cargoes must start, and there may also exist time windows for discharging. The shipping company controls a heterogeneous fleet of ships to transport the cargoes. There are compatibility constraints between ships and cargoes, since for instance some ships cannot enter all loading and discharging ports due to draft limitations.

The planners of a tramp shipping company solve, on a daily basis, a ship routing and scheduling problem which is similar to the *multi-vehicle pickup and delivery problem with time windows* described by Desrosiers et al. (1995). Generally, the ship capacities and the cargo quantities are such that the ships can carry multiple cargoes simultaneously. The challenge for the tramp shipping company is to select spot cargoes and construct routes and schedules in order to maximize profit. Here, the profit is defined as the income of all transported cargoes minus the variable sailing costs, which mainly consist of fuel and port costs.

In this paper, we describe and solve an important real life ship routing and scheduling problem that arises in a special segment of tramp shipping denoted as *project shipping*. This segment is also known as heavy duty and heavy lifting, and differs from more traditional tramp segments, such as wet and dry bulk shipping, as the cargoes are often unique and usually transported on a one-time basis. Such cargoes can for instance be parts of a process facility, reactors or yachts. In this segment there are requirements related to *cargo coupling* and *stowage* onboard the ships in addition to the above tramp shipping problem aspects. The stowage requirements arise because of the special characteristics of the cargoes. A project shipping company may need an engineering unit to calculate the feasibility of transporting their cargoes with respect to shape, stability, weight or the ships' heavy lifting capacities. Basically, a cargo can be placed onboard a given ship in two ways: on the deck or below deck in the ship's cargo hold. The engineering unit also calculates how much deck area or volume below deck a cargo requires. Together with the routing and scheduling decisions, the planners must therefore be able to evaluate the feasibility of transporting the cargoes which is again affected by whether the cargo is placed on or below deck. This separates project shipping from traditional tramp shipping, where the stowage problem is usually handled as an operational task and not as a part of the routing and scheduling. Although cargo placement

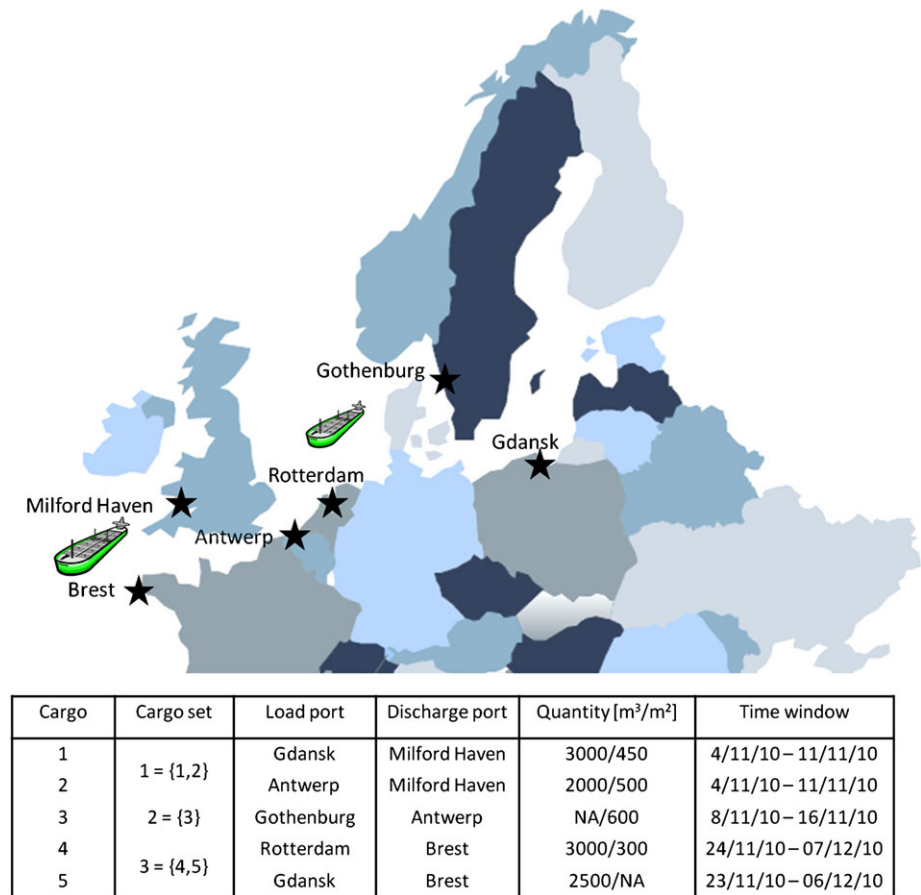


Fig. 1 Illustration of a small example problem

decisions are taken into account when planning, the engineering unit may still be required to make additional calculations so as to make sure that ship stability is maintained and that items put on deck do not block items put in the cargo hold.

The special nature of the cargoes also leads to some other requirements for the routing and scheduling decisions. For instance, the shipping company can receive a booking tender which consists of several cargoes from different origins to be delivered at one process facility together. This gives *coupling constraints* specifying that some of the cargoes cannot be evaluated individually, but rather must be evaluated as sets. It is also common to impose a requirement that all cargoes in a coupled set are serviced by the same ship if serviced at all.

Figure 1 shows a small example problem from Northern Europe with five cargoes and two ships with initial positions at sea. Each cargo requires a given volume below deck (given in cubic meters) or area on the deck (given in square meters). For some cargoes, the decision regarding whether to stow on or below deck is fixed (cargoes 3 and 5 in the example in Fig. 1). In the example we see that cargoes 1 and 2 belong to the same set. If cargo 1

is transported, then cargo 2 must also be transported and vice versa. The same applies to cargoes 4 and 5.

In addition to the above mentioned requirements, some cargoes may have *single service customer requirements*. This means that some cargoes must be transported directly from its loading port to its corresponding discharging port with no other cargoes onboard the ship at the same time, even though the cargo does not require the full capacity of the ship.

Most research studies on ship routing and scheduling problems consider exact methods. A common algorithm is to generate a set of feasible single ship schedules and select an optimal combination by solving a set partitioning problem, see for instance (Kim and Lee 1997; Bausch et al. 1998; Fagerholt 2001 and Brønmo et al. 2006). For larger problem instances, the solution approach based on generation of feasible single ship schedules may require too much computational time to make it useful in a practical planning situation. There have also been some contributions on local search-based heuristics for ship routing and scheduling problems. Brønmo et al. (2007) present a multi-start local search heuristic which applies to tramp ship routing and scheduling problems, while Korsvik et al. (2010) develop a tabu search heuristic for a similar problem. However, none of the above contributions deal with the special characteristics and requirements that arise in project shipping.

Several types of maritime stowage problems have been described in the literature, and their characteristics depend on the type of ship and cargo. Steenken et al. (2004) and Stahlbock and Voss (2008) give surveys on stowage of container ships. A stowage problem arising in bulk shipping is treated by Hvattum et al. (2009). RoRo ships carry vehicles and rolling equipment, and Øvstebø et al. (2011a) study the problem of loading and unloading such cargo on a fixed route. The stowage problems described in these contributions are all different from the one arising in project shipping. Combinations of stowage and routing have also been studied before. Fagerholt and Christiansen (2000a, 2000b) consider the integrated problem of routing and stowage in dry bulk shipping, while Øvstebø et al. (2011b) study the integrated problem for RoRo ships. The literature for combining routing and packing in other transport modes is growing quickly, but a recent survey is provided by Iori and Martello (2010).

The main contribution of this paper is the development of an efficient tabu search algorithm for ship routing and scheduling problems in project shipping. Section 2 gives a mathematical problem description. The proposed solution method is described in Sect. 3. A computational study, including tests on real data provided by a project shipping company, is presented in Sect. 4, while we draw some concluding remarks in Sect. 5.

2 Model formulation

We now present a model that also gives a detailed and precise problem definition. Section 2.1 introduces the notation that is used in the model, while Sects. 2.2–2.7 present and explain the objective function and the various constraints.

2.1 Notation

Sets and indices

V	Ships in the fleet indexed by v
N_v	Nodes compatible with ship v , indexed by i, j and k
N_P	Loading nodes, $N_P = \{1, 2, \dots, n\}$, where n is the number of cargoes available
N_D	Discharging nodes, $N_D = \{n + 1, n + 2, \dots, 2n\}$, where $n + i$ is the corresponding discharging node for loading node i
N_{Pv}	Loading nodes compatible with ship v , $N_{Pv} \subseteq N_P$
N_{Dv}	Discharging nodes compatible with ship v , $N_{Dv} \subseteq N_D$
N_C	Loading nodes for contracted cargoes, $N_C \subseteq N_P$
N_O	Loading nodes for optional cargoes, $N_O \subseteq N_P$
N_S	Loading nodes for the cargoes with single service requirement, $N_S \subseteq N_P$
K	Coupled optional cargo sets
N_{Km}	Loading nodes in coupled optional cargo set $m \in K$
A_v	Arcs (i, j) compatible with ship v

Parameters

R_i	Revenue for transporting cargo i (loading at node $i \in N_P$, transporting and discharging it at node $n + i$)
C_{ijv}	Cost of sailing from node i to j with ship v , including port and loading cost at node i
Q_i^A	Square meter of deck area required on the ship for cargo i if it is stowed on deck
Q_i^V	Volume required on the ship for cargo i if it is stowed below deck
Q_v^{ACAP}	Deck area capacity of ship v
Q_v^{VCAP}	Volume capacity of ship v
T_{ijv}	Sailing time between nodes i and j with ship v , including service time at node i
E_{iv}	Earliest start of service at node i for ship v
L_{iv}	Latest start of service at node i for ship v

Variables

x_{ijv}	Binary flow variable equal to 1 if ship v sails directly between nodes i and j , and 0 otherwise
t_{iv}	Start time of service at node i for ship v
q_m	Binary coupling variable equal to 1 if set N_{Km} is transported, and 0 otherwise
z_i	Binary stowage variable equal to 1 if cargo i is stowed on deck, and 0 if it is stowed below deck
l_{iv}^A	Area of load on deck (in square meters) of ship v when leaving node i
l_{iv}^V	Volume of load (below deck) on ship v when leaving node i

In addition to the above notation, we also define o_v and d_v , where o_v represents the initial open position for ship v , while d_v represent an artificial end node. The distances and costs to d_v from all other nodes that are connected to d_v in the network are set to zero.

2.2 Objective function

The objective function is defined as follows:

$$\max \sum_{v \in V} \sum_{i \in N_O \cap N_{P_v}} \sum_{j \in N_v} R_i x_{ijv} - \sum_{v \in V} \sum_{(i,j) \in A_v} C_{ijv} x_{ijv} \quad (1)$$

Objective function (1) maximizes the profit by taking the difference between the revenue from the optional spot cargoes and the variable sailing and port costs through the planning period. Since the contracted cargoes must be transported the revenue from these can be considered fixed. Therefore, we do not include the revenue from these in the objective function.

2.3 Ship routing constraints

The routing of the ships is restricted by constraints (2)–(6):

$$\sum_{v \in V} \sum_{j \in N_v | (i,j) \in A_v} x_{ijv} = 1 \quad i \in N_C \quad (2)$$

$$\sum_{v \in V} \sum_{j \in N_v | (i,j) \in A_v} x_{ijv} \leq 1 \quad i \in N_O \quad (3)$$

$$\sum_{j \in N_v} x_{o_v j v} = 1 \quad v \in V \quad (4)$$

$$\sum_{j \in N_v | (i,j) \in A_v} x_{ijv} - \sum_{j \in N_v | (i,j) \in A_v} x_{jiv} = 0 \quad v \in V, i \in N_v \setminus \{o_v, d_v\} \quad (5)$$

$$\sum_{i \in N_{D_v} \cup \{o_v\}} x_{id_v} = 1 \quad v \in V \quad (6)$$

$$\sum_{j \in N_v} x_{ijv} - \sum_{j \in N_v} x_{j,n+i,v} = 0 \quad v \in V, i \in N_{P_v} \quad (7)$$

Constraints (2) ensure that all contracted cargoes for the given planning period are transported, while constraints (3) state that the optional spot cargoes can be serviced at most once. Constraints (4)–(6) are the flow conservation constraints for each ship. Constraints (7) ensure that the same ship visits both loading node and the corresponding discharging node.

2.4 Ship load constraints

The load onboard the ships is restricted by constraints (8)–(16):

$$l_{o_v}^A = l_{o_v}^V = 0 \quad v \in V \quad (8)$$

$$l_{iv}^A + Q_j^A z_j \leq l_{jv}^A + (Q_v^{ACAP} + Q_j^A)(1 - x_{ijv}) \quad v \in V, j \in N_{P_v}, (i, j) \in A_v \quad (9)$$

$$l_{iv}^V + Q_j^V z_j \leq l_{jv}^V + (Q_v^{VCAP} + Q_j^V)(1 - x_{ijv}) \quad v \in V, j \in N_{P_v}, (i, j) \in A_v \quad (10)$$

$$l_{iv}^A - Q_j^A z_j \leq l_{j+n,v}^A + Q_v^{ACAP}(1 - x_{i,j+n,v}) \quad v \in V, j \in N_{P_v}, (i, j) \in A_v \quad (11)$$

$$l_{iv}^V - Q_j^V z_j \leq l_{j+n,v}^V + Q_v^{VCAP}(1 - x_{i,j+n,v}) \quad v \in V, j \in N_{P_v}, (i, j) \in A_v \quad (12)$$

$$\sum_{j \in N_v} Q_i^A x_{ijv} - \max\{Q_v^{ACAP}, Q_i^A\}(1 - z_i) \leq I_{iv}^A \leq \sum_{j \in N_v} Q_v^{ACAP} x_{ijv} \quad v \in V, i \in N_{Pv} \tag{13}$$

$$0 \leq I_{n+i,v}^A \leq \sum_{j \in N_v} (Q_v^{ACAP} - Q_i^A)x_{n+i,jv} \quad v \in V, i \in N_{Pv} \tag{14}$$

$$\sum_{j \in N_v} Q_i^V x_{ijv} - \max\{Q_v^{VCAP}, Q_i^V\}z_i \leq I_{iv}^V \leq \sum_{j \in N_v} Q_v^{VCAP} x_{ijv} \quad v \in V, i \in N_{Pv} \tag{15}$$

$$0 \leq I_{n+i,v}^V \leq \sum_{j \in N_v} (Q_v^{VCAP} - Q_i^V)x_{n+i,jv} \quad v \in V, i \in N_{Pv} \tag{16}$$

In constraints (8), we assume for simplicity that the ships have no load onboard when the ships are available in the planning period. Constraints (9) and (10) calculate the deck area load and volume load below deck onboard the ships in the loading nodes, respectively, while constraints (11) and (12) do the same in the discharging nodes. Some cargoes can only be stowed on deck or below deck, which can easily be ensured by fixing the z_i variables to either 1 or 0. Capacity constraints (both deck area and volume under deck) are given by (13)–(16) for the loading and discharging nodes, respectively. It is also common to have weight capacity constraints in ship routing and scheduling problems. However, in this project shipping routing and scheduling problem this is not a practical restriction and it is therefore not included in the formulation.

2.5 Time constraints

The scheduling of the ships is restricted by constraints (17)–(20):

$$t_{iv} + T_{ijv} \leq t_{jv} + (t_{iv} + T_{ijv})(1 - x_{ijv}) \quad v \in V, (i, j) \in A_v, i \neq o_v \tag{17}$$

$$t_{o_v,v} + T_{o_vjv} \leq t_{jv} + (t_{o_v,v} + T_{ijv})(1 - x_{ijv}) \quad v \in V, (i, j) \in A_v, i = o_v \tag{18}$$

$$E_{iv} \leq t_{iv} \leq L_{iv} \quad v \in V, i \in N_v \tag{19}$$

$$t_{iv} + \sum_{j \in N_v} T_{ijv}x_{ijv} - t_{n+i,v} \leq 0 \quad v \in V, i \in N_{Pv} \tag{20}$$

The linking of the routes and schedules is given by (17)–(18). Because of the inequality ships are allowed to wait at the nodes before start of service. Time window constraints are given by constraints (19), while constraints (20) force loading node i to be visited before the corresponding discharging node $n + i$.

2.6 Project shipping constraints

The requirements that are special for project shipping (in addition to the stowage constraints in Sect. 2.4) are given by constraints (21)–(25):

$$\sum_{v \in V} \sum_{i \in N_{Km}} \sum_{j \in N_v} x_{ijv} = |N_{Km}|q_m \quad m \in K \tag{21}$$

$$x_{i,n+i,v} = \sum_{j \in N_v} x_{ijv} \quad v \in V, i \in N_S \tag{22}$$

$$\sum_{j \in N_v} x_{ijv} = \sum_{j \in N_v} x_{k_jv} \quad v \in V, N_K \in K, i, k \in N_K \quad (23)$$

$$l_{iv}^A \leq z_i Q_i^A \quad v \in V, i \in N_S \quad (24)$$

$$l_{iv}^V \leq (1 - z_i) Q_i^V \quad v \in V, i \in N_S \quad (25)$$

The coupling of cargoes is given by constraints (21). The number of nodes visited within the set must either be equal to the cardinality of the set if a node in the set is serviced, or zero if the coupled set is not serviced at all. These constraints are only necessary for optional cargoes, since constraints (2) make sure that all contracted cargoes are transported.

The single service requirements are given by constraints (22)–(25). Constraints (22) ensure that if a ship services a cargo in the single service set, it must sail directly to that cargo's discharging node. Note that this can implicitly also be taken into account by only defining the x_{ijv} variables on the arcs going directly from i to $n + i$ if $i \in N_S$. Constraints (23) ensure that all cargoes within a coupled cargo set should be serviced by the same ship if serviced at all. Constraints (22) and (23) are not sufficient to ensure single service requirement. We must also add load constraints (24) and (25), which guarantee that the load onboard the ship after loading a single service cargo cannot be larger than the load of the cargo itself for deck area and volume under deck, respectively. This will ensure that the ship has no other cargoes onboard while transporting a cargo with a single service customer requirement, which is not guaranteed by constraints (22) and (23) only.

2.7 Variable domains

The variable domains are given by constraints (26)–(28):

$$x_{ijv} \in \{0, 1\} \quad v \in V, (i, j) \in A_v \quad (26)$$

$$q_m \in \{0, 1\} \quad m \in K \quad (27)$$

$$z_i \in \{0, 1\} \quad i \in N_P \quad (28)$$

3 Tabu search algorithm

We propose a computationally efficient tabu search algorithm for solving the routing and scheduling problem in project shipping. It uses the same structure as the tabu search algorithm of Korsvik et al. (2010), but incorporates several additional features specific to our problem. Starting from an initial solution s_0 , the algorithm moves at iteration t from the current solution s_t to the best solution \bar{s} in a neighbourhood $N(s_t)$ of s_t . An important feature of the approach is the possibility of considering infeasible solutions during the search. This relaxation mechanism facilitates the exploration of the solution space and is particularly useful for tightly constrained instances, which is the case for many ship routing and scheduling problems. To avoid cycling, solutions possessing some attributes of recently visited solutions are declared tabu for a number of iterations, unless they constitute a new incumbent. A continuous diversification mechanism reduces the likelihood of becoming trapped in a local optimum, and periodic reoptimization of the current or the best known solution is performed. The main steps of the tabu search are repeated for a preset number η of iterations. Similar tabu search algorithms are proposed by Cordeau et al. (2001) and Cordeau and Laporte (2003) for the vehicle routing problem and the dial a ride problem respectively. In the following subsections we describe the main elements of our tabu search heuristic in detail.

3.1 Relaxation mechanism

Let S denote the solutions satisfying constraints (1)–(12), (17)–(18), (20) and (26)–(28). However, we allow that the solutions violate constraints (13)–(16), (19) and (21)–(25), which represents capacity constraints, time windows constraints, coupling constraints and single service requirements, respectively. For a solution $s \in S$ let $c(s)$ denote the total sailing cost for the fleet of ships. If several of the available cargoes are rejected in solution s , we introduce a rejection cost for these cargoes. We set the rejection cost of an optional spot cargo i equal to its lost revenue R_i . For a mandatory contract cargo, we impose a large rejection cost to ensure that the cargo will be served. Whenever the shipping company can subcontract a cargo, i.e. it can hire a spot ship to transport the cargo, the rejection cost can be set to the cost of hiring the spot ship. The total cost for rejected cargoes in solution s is given by $c'(s)$. Let $u(s)$ denote the total violation of capacity constraints. The violations of capacity constraints are calculated on a route basis with respect to the ships capacities (for details see Sect. 3.8). The total violation of time window constraints is denoted by $w(s)$ and is equal to $\sum_{i=1}^{2n} (t_i - L_i)^+$, where $x^+ = \max\{0, x\}$ and n is the number of cargoes (we sum up to $2n$ since each cargo consists of both a loading and a discharge node). It should be observed that service at a node cannot start before the beginning of the time window. To handle the coupling constraints we assign a penalty φ_{mv} for transporting a given cargo set m with ship v . During the search the penalties φ_{mv} are adjusted so that cargoes from m are serviced by the same ship most of the time. For details on how the penalties are adjusted, see Sect. 3.9.

Potential moves are evaluated using the cost function to be minimized:

$$f(s) = c(s) + c'(s) + \alpha u(s) + \beta w(s) + \sum_{v \in V} \sum_{m \in K} \sum_{i \in N_{K_m}} \sum_{j \in N_v} \varphi_{mv} x_{ijv} \quad (29)$$

where the parameters α , β and φ_{mv} are positive and are dynamically adjusted during the search.

3.2 Neighbourhood structure

With each solution $s \in S$ is associated an attribute set $A(s) = \{(i, v) : i = 1, \dots, n; v = 1, \dots, m\}$ where (i, v) means that cargo i is assigned to ship v . The rejected cargoes in solution s are assigned to a dummy ship (or a spot ship). The neighbourhood $N(s)$ of a solution consists of all solutions that can be obtained by removing an attribute (i, v) from $A(s)$ and replacing it with another attribute (i, v') , where $v' \neq v$. Removing an attribute (i, v) consists of removing the loading and discharge nodes of cargo i from the route of ship v . This route is then reconnected by linking the predecessor and successor of the deleted nodes. Inserting nodes i and $(n + i)$ in the route of ship v' is performed in order to minimize the total increase in cost function $f(s)$ by using simple insertions (i.e., the ordering of the nodes already in the route of ship v' remains unchanged). When cargo i is removed from the route of ship v , its reinsertion in that route is forbidden for the next θ iterations by assigning a tabu status to the attribute (i, v) . Through an aspiration criterion, the tabu status of an attribute can be revoked if the search reaches a solution of smaller cost than the best known solution having the given attribute.

3.3 Diversification strategy

To diversify the search, any solution $\bar{s} \in N(s)$ such that $f(\bar{s}) \geq f(s)$ is penalized by a factor proportional to the frequency of addition of its distinguishing attribute and a scaling factor.

We let ρ_{iv} be the number of times attribute (i, v) has been added to the solution during the search, and t represents the current iteration number. If (i, v) is the attribute that must be added to the current solution s to obtain the new non-improving solution \bar{s} , the penalty

$$p(\bar{s}) = \lambda(c(\bar{s}) + c'(\bar{s}))\sqrt{nm}\rho_{iv}/t \quad (30)$$

is added to $f(\bar{s})$ when evaluating the cost of \bar{s} . The scaling factor $(c(\bar{s}) + c'(\bar{s}))\sqrt{nm}$ introduces a correction that adjusts the penalty with respect to total solution cost and instance size as measured by number of possible attributes. Finally, the parameter λ is used to control the intensity of the diversification. These penalties have the effect of driving the search process towards less explored regions of the solution space whenever a local optimum is reached.

3.4 Initial solution

To construct an initial solution s_0 we first sort the list of unassigned cargoes in increasing order of the start of the time window for the loading node. Each cargo is then sequentially assigned to the ship minimizing the value of the cost function, and the associated nodes i and $(n + i)$ are inserted in the best position in the partially constructed route.

3.5 Tabu search iterations

The tabu search algorithm starts from the initial solution s_0 and selects, at iteration t , the best non-tabu solution $\bar{s} \in N(s_t)$ with respect to the objective function $f(\bar{s}) + p(\bar{s})$. Initially, $\alpha = \beta = 1$ while all φ_{mv} are set equal to zero. After each iteration, the values of the parameters α and β are modified. If the current solution is feasible with respect to capacity constraints, the value of α is divided by $1 + \delta$, where $\delta \geq 0$, to make it less costly to visit solutions violating constraints of that type. If the current solution is infeasible regarding capacity constraints α is multiplied by $1.5 + \delta$, and it becomes more expensive to violate constraints of that type. The same rule is also applied to the parameter β associated with violation of time window constraints. The parameters φ_{mv} are changed during the search in order to force the coupled cargoes to be transported with the same ship (see description in Sect. 3.9). Insertion and deletion costs are not fully recomputed at every iteration. Since each exchange involves only two routes, insertion and deletion costs for all remaining routes are still valid after the exchange has been performed. The algorithm is repeated for η iterations. The best feasible solution s^* encountered during the search is identified.

3.6 Periodic route reoptimization

Every κ iterations, intra-route exchanges are performed. The loading and discharging nodes of all cargoes are sequentially removed from their current route and are then reinserted in the route of the same ship in order to minimize the value of $f(s)$. To provide some form of periodic intensification, intra-route exchanges are also performed whenever a new best solution is identified during the search.

3.7 Neighbourhood reduction

To reduce the size of the neighbourhood considered at every iteration, the following rule is used when evaluating the impact of inserting the two nodes i and $(n + i)$ of cargo i into the route of ship v . First insert the loading node i of the cargo in the best position in the route.

Then, fixing the loading node in its best position, all subsequent insertions of the discharge node ($n + i$) are evaluated. This rule has a significant effect on the computational time as it reduces the time for evaluating the insertion of cargo i in the route of ship v from $O(n^2)$ to $O(n)$. The best insertion of a given loading node, the one that minimize the value of $f(s)$, is dependent on the values of α and β because these parameters are modified after each iteration of the search. As a result, the best insertion position of the node can change from one iteration to the next even if the route is unchanged. In order to reduce computation time we do not recompute the insertion costs in unchanged routes, even if there may conceivably exist better insertions given the new values of α and β . A complete recomputing of the insertion costs is only performed every ω iterations.

3.8 Stowage subproblem

When considering a fixed route, that is where all the routing variables x_{ijv} , all the scheduling variables t_{iv} and all the cargo coupling variables q_m are fixed, the stowage problem that remains is to decide for each node visited whether the cargo is to be placed above or below deck. The objective is to minimize the total violation of the capacity constraints, with separate weights for area violations, volume violations and single service violations. This problem where only the z_i variables are free (together with the auxiliary l_{iv}^A and l_{iv}^V variables) is still NP-hard, as can be shown through a reduction from the *partition problem* (PP), which is known to be NP-complete (Gary and Johnson 1979): In PP we are given a set $B = \{b_1, \dots, b_n\}$ of n numbers, and must determine whether there exists a partition of B into two subsets B_1 and B_2 such that $\sum_{j \in B_1} b_j = \sum_{j \in B_2} b_j$. From an instance of PP we can generate an instance of the stowage problem where a single ship visits n loading nodes in succession (and thereafter the corresponding n discharging nodes), where $Q_v^{ACAP} = Q_v^{VCAP} = \sum_{j \in B} \frac{b_j}{2}$ and $Q_j^A = Q_j^V = b_j$. Clearly, a feasible solution to the PP exists only if a stowage solution exists with no violation of either constraints governing the area above deck or constraints governing the volume below deck. Hence, the stowage problem is NP-hard, even though it may be less complex than other stowage problems occurring in maritime transportation, such as in Avriel et al. (1998), Delgado et al. (2009) and Hvattum et al. (2009).

The following algorithm is used to find stowage decisions that minimize the total violation of capacity constraints: A fixed route of a ship (or route segment starting and ending with the ship being empty) is considered node by node, and for each node we create labels with three values, namely the current load above deck, the current load below deck and the sum of capacity violations incurred so far. For loading nodes the labels are created by extending each label from the previous node by either loading above or below deck, and for discharging nodes the labels are created by adjusting the current load based on the corresponding loading node label. For an exact algorithm we may only apply dominance rules for nodes where the ship is empty, but for a heuristic algorithm we also consider dominance for non-empty ships: if a label has higher or equal values for all the current loads and the sum of violations so far, it is removed and not considered again. When reaching the final node of the route (or segment), the label with the lowest sum of capacity violations is used as a starting point and the stowage decisions are found by backtracking through the labels.

3.9 Coupling constraint

The term $\sum_{v \in V} \sum_{m \in K} \sum_{i \in N_{K_m}} \sum_{j \in N_v} \varphi_{mv} x_{ijv}$ in the move evaluation function (29) is used to force the coupled cargoes to be transported with the same ship. In the initial solution there

Table 1 Description of 14 test instances of varying size

No.	Ships	Cargoes	Cargo sets
1	3	6	4
2	2	8	5
3	3	8	5
4	3	10	4
5	3	10	6
6	4	10	6
7	3	12	8
8	4	12	8
9	4	14	8
10	5	18	11
11	5	20	13
12	6	30	19
13	6	40	26
14	8	63	38

might be violation of coupling constraints. For each of the violated coupling constraints we choose the ship v containing most cargoes from the given coupled set and assign $\varphi_{mv} = 0$ for this ship. For all other ships a high penalty is assigned for transporting cargoes belonging to the coupled set. After some iterations in the search the cargoes will then be moved to ship v . When a coupling constraint is satisfied we gradually decrease the penalty for transporting the cargoes with a different ship. After a few iterations it may then become profitable to move cargoes from the coupled set to a different ship. When a cargo is moved to a new ship v' the penalty $\varphi_{mv'}$ is set equal to zero and high penalties are assigned for the other ships in order to gradually gather all the cargoes from m on ship v' .

4 Computational study

In this section we describe a computational study on the project shipping routing and scheduling problem, showing that the tabu search heuristic is able to quickly find high quality solutions on relevant problem instances based on data from a real world project shipping company.

A set of 14 test instances used in the study is described in Table 1. The instances have a varying number of ships, from two to eight, and a varying number of cargoes, from six to 63. Since some of the cargoes are coupled, we also report the number of cargo sets, where each cargo set contains cargoes that are coupled. Some of the cargo sets contain only a single cargo.

To enable an assessment of how well the tabu search performs, the mathematical model in Sect. 2 has been implemented in Mosel, and the 14 instances have been solved using Xpress 20 using a two hour time limit. The tests have been run on a computer with an Intel Core2 Duo 2.66 GHz CPU. Table 2 reports the initial LP-relaxation, the best bound at the end of the search, the best solution found, the gap between the best bound and the best solution, and the times in seconds to reach a feasible solution, the best solution and to prove optimality. It may be noted that the LP-relaxation is very weak and the model can be solved to optimality for small instances. There are some additional instances where Xpress finds good solutions, but for larger instances the exact solver does not even find any feasible solutions.

Table 2 Results of running Xpress using a time limit of two hours

No.	LP-relaxation	Best bound	Best solution	Gap	Seconds to feasible	Seconds to best	Seconds to optimality
1	5,256,638	4,732,756	4,732,756	0.0%	0.1	0.7	7.1
2	7,614,915	5,914,563	5,914,563	0.0%	3.1	170.3	5758.5
3	7,618,581	7,190,892	7,190,892	0.0%	1.3	98.0	513.7
4	4,066,972	3,133,510	3,133,510	0.0%	1.9	2.2	129.1
5	4,974,285	3,826,624	3,826,624	0.0%	3.6	25.8	127.9
6	4,998,759	4,177,192	4,177,192	0.0%	0.3	26.1	630.1
7	10,817,582	10,128,483	8,682,041	16.7%	60.9	5131.8	NA
8	10,945,603	10,258,363	9,959,807	3.0%	572.2	680.1	NA
9	9,912,482	9,438,833	9,047,644	4.3%	4.8	6364.9	NA
10	12,392,195	11,808,827	NA	NA	NA	NA	NA
11	15,118,321	14,475,315	11,178,887	29.5%	1732.8	7107.0	NA
12	21,104,308	20,448,860	NA	NA	NA	NA	NA
13	26,751,969	26,287,692	NA	NA	NA	NA	NA
14	42,395,432	41,764,364	NA	NA	NA	NA	NA

Table 3 Results of running the tabu search ten times, using either 1000 or 10,000 iterations

No.	1000 iterations			10,000 iterations			TS best/ XPRESS
	Best solution	Average solution	Average time (s)	Best solution	Average solution	Average time in seconds	
1	4,732,756	4,732,756	7.0	4,732,756	4,732,756	69.1	1.0000
2	5,914,563	5,914,563	9.5	5,914,563	5,914,563	94.0	1.0000
3	7,190,892	7,190,892	9.2	7,190,892	7,190,892	93.6	1.0000
4	3,120,913	3,120,913	11.1	3,120,913	3,120,913	112.0	0.9960
5	3,826,624	3,826,624	11.0	3,826,624	3,826,624	108.0	1.0000
6	4,177,192	4,177,192	10.7	4,177,192	4,177,192	103.3	1.0000
7	8,683,460	8,683,460	14.4	8,683,460	8,683,460	139.0	1.0002
8	9,960,280	9,960,280	15.2	9,960,280	9,960,280	145.9	1.0000
9	9,048,296	9,048,296	17.3	9,048,296	9,048,296	162.8	1.0001
10	8,741,703	8,688,618	18.6	8,741,703	8,741,703	185.0	NA
11	11,467,091	11,393,601	22.2	11,467,091	11,467,091	220.8	1.0258
12	16,961,882	16,664,222	27.5	16,972,808	16,951,413	272.0	NA
13	21,871,551	21,428,294	71.3	21,871,551	21,851,947	697.4	NA
14	36,867,958	36,633,223	151.4	36,964,752	36,934,720	1596.9	NA

Finally, results using the tabu search are reported in Table 3. The tabu search has been run for 1000 and 10,000 iterations, with ten different random seeds for each instance. Table 3 shows both the best solution and the average solution found over the ten different random seeds, as well as the average time in seconds taken. These tests were executed on a computer with an Intel Core2 Duo 2.20 GHz CPU.

Comparing Xpress and the tabu search we see that Xpress proves the optimal solution on six instances and that the tabu search is able to find the same solution on five of these instances. For instance number four the tabu search finds a solution that is only slightly worse. The tabu search is very robust, and finds the same solution starting from all ten random seeds after less than 1000 iterations. For four additional instances Xpress is able to find a feasible solution, without proving optimality, but for all of these four instances the tabu search on average produces better solutions after just 1000 iterations. For the remaining four instances, Xpress does not find feasible solutions, and the gaps between the best upper bounds by Xpress and the solutions found by the tabu search are quite large (between 13 and 35%). However, the initial gaps produced from Xpress are quite large on all the instances, hence it is possible that the final gaps between the tabu search solutions and the upper bounds of Xpress are mostly due to having poor upper bounds.

It should be noted that using 10,000 iterations give not much compared to using only 1000 iterations. Therefore, it seems that 1000 iterations are sufficient for any practical application. This also gives a running time less than 3 minutes for solving the biggest instances.

5 Conclusion

In this paper we have introduced and solved a real routing and scheduling problem that arises in the project shipping segment, a special segment in tramp shipping. This segment differs from more traditional tramp segments, as the cargoes are usually transported on a one-time basis. The characteristics of the cargoes require stowage planning to be integrated with the scheduling and routing decisions. We presented a mathematical model and suggested a tabu search heuristic to solve the problem. A computational study was performed on 14 test instances based on data from a shipping company. The results show that the proposed tabu search heuristic provides optimal or near-optimal solutions to the smallest instances which Xpress can solve. For medium-sized instances Xpress can find feasible solutions but cannot prove optimality, and for these instances the tabu search heuristic finds better solutions. For the largest instances Xpress cannot even find feasible solutions. However, comparisons with solutions produced manually by planners in the shipping company indicate that the solutions have high quality and give significant improvements. Another good characteristic with the tabu search is that it seems to be very robust and produce solutions of very similar quality even with different random seeds. It also seems that for any practical application, 1000 iterations (giving a running time of less than 3 minutes even for the largest instance) are sufficient. A version of the tabu search heuristic described in this paper has been integrated as a solver in a prototype decision support system used by several shipping companies (Fagerholt 2004).

Acknowledgements This research was carried out with financial support from the DESIMAL project supported by the Research Council of Norway. We are also grateful to the shipping company that provided real data for the test cases used in the computational study. Thanks are due to two referees for their valuable comments.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

- Avriel, M., Penn, M., Shpirer, N., & Witteboon, S. (1998). Stowage planning for container ships to reduce the number of shifts. *Annals of Operation Research*, 76, 55–71.

- Bausch, D. O., Brown, G. G., & Ronen, D. (1998). Scheduling short-term marine transport of bulk products. *Maritime Policy and Management*, 25, 335–348.
- Brønmo, G., Christiansen, M., Fagerholt, K., & Nygreen, B. (2007). A multi-start local search heuristic for ship scheduling—A computational study. *Computers & Operations Research*, 34, 900–917.
- Brønmo, G., Christiansen, M., Fagerholt, K., & Nygreen, B. (2006). Ship routing and scheduling with flexible cargo sizes. *Journal of the Operational Research Society*, 58, 1167–1177.
- Cordeau, J.-F., & Laporte, G. (2003). A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research. Part B: Methodological*, 37, 579–594.
- Cordeau, J.-F., Laporte, G., & Mercier, A. (2001). A unified tabu search heuristic for vehicle routing problems with time windows. *The Journal of the Operational Research Society*, 52, 928–936.
- Delgado, R., Jensen, M., & Schulte, C. (2009). Generating optimal stowage plans for container vessel bays. In *The proceedings of the 15th international conference on principles and practice of constraint programming (CP-09)*, LNCS Series, vol. 5732 (pp. 6–20). Berlin: Springer.
- Desrosiers, J., Dumas, Y., Solomon, M. M., & Soumis, F. (1995). Time constrained routing and scheduling. In M. O. Ball, T. L. Magnanti, C. L. Monma, & G. L. Nemhauser (Eds.), *Handbooks in operations research and management science: network routing* (pp. 35–139). Amsterdam: North-Holland.
- Fagerholt, K. (2001). Ship scheduling with soft time windows—an optimization based approach. *European Journal of Operational Research*, 131, 559–571.
- Fagerholt, K. (2004). A computer-based decision support system for vessel fleet scheduling—experience and future research. *Decision Support Systems*, 37, 35–47.
- Fagerholt, K., & Christiansen, M. (2000a). A combined ship scheduling and allocation problem. *The Journal of the Operational Research Society*, 51, 834–842.
- Fagerholt, K., & Christiansen, M. (2000b). A travelling salesman problem with allocation, time window and precedence constraints—an application to ship scheduling. *International Transactions in Operations Research*, 7, 231–244.
- Gary, M. R., & Johnsen, D. S. (1979). *Computers and intractability: a guide to the theory of NP-completeness*. San Francisco: Freeman.
- Hvattum, L. M., Fagerholt, K., & Armentano, V. A. (2009). Tank allocation problems in maritime bulk shipping. *Computers & Operations Research*, 36, 3051–3060.
- Iori, M., & Martello, S. (2010). Routing problems with loading constraints. *TOP*, 18, 4–27.
- Kim, S.-H., & Lee, K.-K. (1997). An optimisation-based decision support system for ship scheduling. *Computers & Industrial Engineering*, 33, 689–692.
- Korsvik, J. E., Fagerholt, K., & Laporte, G. (2010). A tabu search heuristic for ship routing and scheduling. *The Journal of the Operational Research Society*, 61, 594–603.
- Stahlbock, R., & Voss, S. (2008). Operations research at container terminals: a literature update. *OR-Spektrum*, 30, 1–52.
- Steenken, D., Voss, S., & Stahlbock, R. (2004). Container terminal operation and operations research—a classification and literature review. *OR-Spektrum*, 26, 3–49.
- UNCTAD (2007). Review of maritime transport, 2007. United Nations, New York and Geneva.
- Øvstebø, B. O., Hvattum, L. M., & Fagerholt, K. (2011a). Optimization of stowage planning for RoRo ships. *Computers & Operations Research*, 38, 1425–1434.
- Øvstebø, B. O., Hvattum, L. M., & Fagerholt, K. (2011b). Routing and scheduling of RoRo ships with stowage constraints. Forthcoming in *Transportation Research Part C*.