A shortest path heuristic for evaluating the quality of stowage plans in Roll-on Roll-off liner shipping

Jone R. Hansen[1(✉)], Kjetil Fagerholt[1,2], and Magnus Stålhane[1]

[1] Department of Industrial Economics and Technology Management, Norwegian University of Science and Technology, Norway: jone.hansen@ntnu.no

[2] SINTEF Ocean, Trondheim, Norway

Abstract. Roll-on Roll-off shipping companies transport rolling cargo, such as cars, trucks and large construction machines. When sailing, this type of cargo must be attached to the deck using chains, to prevent damaging the cargo. For each voyage including multiple port calls where the cargo is loaded/unloaded, an important decision is to decide where to place each vehicle (or unit), such that the time used on shifting is minimized. Shifting means temporarily moving some vehicles to make an entry/exit route for the vehicles that are to be loaded/unloaded at a given port. As the vehicles are securely fastened to the deck, shifting is a time-consuming procedure. We present the stowage plan evaluation problem which is to determine the optimal vehicles to shift at each port call, such that the time spent on shifting is minimized. Given a set of alternative stowage plans for a voyage, the results from the stowage plan evaluation problems are used to determine the best among these stowage plans. We present a shortest path based heuristic for solving the problem. Computational results show that the solution method is a powerful tool for comparing stowage plans, due to its fast computing times and high success rate, i.e. its ability to determine the better of two stowage plans.

Keywords: Maritime transportation, Stowage, Roll-on Roll-off

1 Introduction

Major improvements to the efficiency of maritime transportation have been made during the last decades due to operations research. However, compared to other segments in maritime transportation, the Roll-on Roll-off (RoRo) shipping segment has received little attention. RoRo vessels transport vehicles and other types of rolling material between different regions of the world according to predefined plans. Lower freight rates provide a challenging reality for the RoRo shipping companies, due to a surplus of tonnage in the world's deep sea fleet. We seek to improve the profitability of the RoRo segment, by introducing a method for evaluating different stowage plans. Better stowage plans may reduce the time used to load and unload vehicles, and hence, the time spent in port.

The problem addressed in this paper is the stowage plan evaluation problem (SPEP). In RoRo transportation, a feasible stowage plan is not as good as any other feasible stowage plan. A good stowage plan enables seamless loading and unloading of the carried rolling units (hereinafter referred to as vehicles) by using the least amount of time on moving vehicles unnecessarily, which is known as shifting. Shifting means temporarily moving some vehicles to make an entry/exit route for the vehicles that are to be loaded/unloaded at the given port. Given a stowage plan for a voyage, the objective of the SPEP is to minimize the time used on shifting at each port, by identifying the optimal vehicles to shift. The time use for shifting a vehicle is treated as a cost, such that the quality of a stowage plan is determined by its shifting cost, relatively to other stowage plans carrying the same vehicles.

Figure 1 shows the placement of vehicles on a deck, during the deep sea leg between Asia and Europe for a given voyage. Both stowage plans look structured, but there is a major difference in the shifting cost. At the first port call in Europe, all vehicles marked in blue are to be unloaded. In the upper stowage plan in Figure 1, all blue vehicles are placed in the bow of the ship. When arriving at the first port in Europe, two green and several yellow vehicles must be shifted in order to unload all the blue ones. The second stowage plan has all blue vehicles placed close to the ramp, and no shifts are required when unloading the vehicles.

Stowage planning has been widely studied in the context of container shipping, see for example [1, 5, 6]. Minimizing the number of shifts is an important objective, both in container shipping and RoRo transportation. In container shipping, the containers are stacked on top of one another. When dispatching a given container, containers stacked on top of it must be removed, i.e. shifted. Here, which containers to shift at each port call are implicitly given by the stowage plan. In RoRo transportation, which vehicles to shift to enable load-

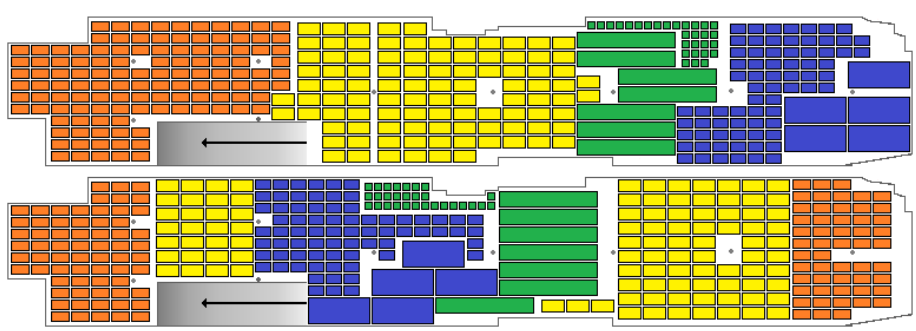


Fig. 1: Two different stowage plans for a given deck and cargo list. The exit ramp is marked with an arrow. Each colored square represents a vehicle and all vehicles with the same color are unloaded at the same port. Unloading sequence: Blue, green, yellow, orange. Thus, the blue vehicles are unloaded at the first unloading port and the orange ones at the last port on the voyage.

ing/unloading of the desired vehicles at each port call, must be determined. The problem of deciding which vehicles to shift at each port (SPEP) is, to the authors' knowledge, new to the OR literature. There are, however, publications dealing with the operational problem of creating stowage plans. Øvstebø *et al.* [4] consider the RoRo ship stowage problem (RSSP). For a ship set to sail on a given voyage, the problem is to decide which cargoes to carry, how many vehicles to carry from each cargo, and how to stow the vehicles carried during the voyage. They formulate the problem as a mixed integer programming (MIP) model and present a specially designed heuristic method to solve the problem. For modeling purposes, they divide each deck into several logical lanes, into which the vehicles are lined. The vehicles enter the each deck at stern and are unloaded according to the last in-first out (LIFO) principle. We argue that basing the shifting cost calculations on a LIFO principle is too crude. In practice, vehicles enter the decks using ramps placed somewhere on the deck and take the least inconvenient path from its placement to the ramp when unloaded.

Recently, Hansen *et al.* [3] presented a mathematical model for the two-dimensional RSSP, a simplification of the RSSP which arises if only one deck is considered. They argue that dividing the decks into lanes, as done in [4], may be too restricting, especially when the cargoes stowed are heterogeneous, i.e. they have different sizes and shapes. The authors propose different objective functions to influence the placement of the vehicles. While promising placement strategies are provided, the shifting cost of the stowage plans is not calculated. Hence, based on the reviewed literature, we identify the need for a method for evaluating and comparing different stowage plans carrying the same cargoes along a voyage.

The purpose of this paper is to present the stowage plan evaluation problem in RoRo transportation. We present a mathematical formulation of the problem and propose a shortest path based heuristic for solving the problem. Even though a solution to this problem may indicate which vehicles to shift at each port in a practical case, this is not the purpose of this contribution. The problem of deciding which vehicles to shift at each port is easily solved by the port workers, who drive the vehicles on and off the ship. However, the problem is crucial when deciding upon a stowage plan when planning a voyage. The SPEP is then solved for each proposed stowage plan, where the shifting cost is used to determine the best one. The stowage plans could both be provided by planners or a stowage plan generator/heuristic.

The remainder of the paper is structured as follows: We present the problem in Section 2. Section 3 presents a mathematical formulation of the problem. Then, in Section 4, we explain our solution approach. Computational results are presented in Section 5 and concluding remarks are given in Section 6.

2 Problem description

The SPEP considers a RoRo ship carrying a given set of cargoes along a voyage with a predefined set of loading and unloading ports to visit. We take as input a given stowage plan for the voyage, which states the number of cargoes, where to load and unload the cargo, the number of vehicles in each cargo, dimensions of the vehicles, and where each vehicle is placed on the ship. Given this stowage plan, the SPEP is to determine which vehicles to shift at each port call, to enable all vehicles that are to be loaded/unloaded to reach their destination on the deck (if loading) or to exit the ship (if unloading). Thus, the problem is to determine an entry and exit route for each vehicle placed on the deck. An entry route for a given vehicle is defined as the path from the entry/exit ramp to the location it is to be placed on the deck, and vice versa for an exit route. Each possible route is associated with a shifting cost, which depends on both the number of vehicles placed along the route and the vehicle's size. This cost is not necessarily a real cost, but it reflects the cost of time used to move the vehicle. The cost of shifting a vehicle varies, as larger vehicles usually require more effort to move. Objects, such as pillars and ramps, and weight restrictions on the deck limit the possible paths a vehicle can use when entering or exiting the deck. The shifting cost of a stowage plan is given by the cost of shifting all blocking vehicles at each port call, and the objective is to minimize this shifting cost.

Figure 2 presents an example of how the shifting cost is evaluated for different vehicles in different settings, and two loading ports are considered. For the first loading port, the entry route for one vehicle from cargo 2 is shown. The vehicle passes through two vehicles from cargo 1, but the vehicles are not shifted. As these vehicles are loaded in the same port, there always exists a loading sequence where the loaded vehicles do not block each other. In this example, this can be achieved by loading cargo 2 first, and then cargo 1. Thus, the shifting cost should only be accounted for if the blocking vehicle is loaded at a previous port call, or unloaded at a later port call. Note that as each deck is empty at the first loading

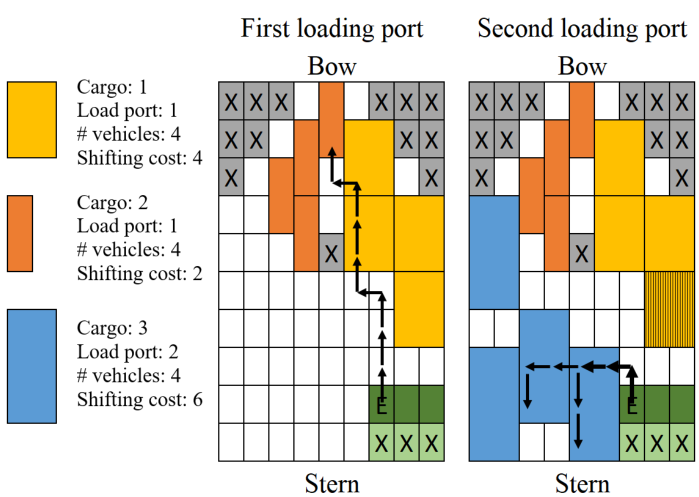


Fig. 2: Possible loading routes for some vehicles at different ports. Entry point is marked with an E. The X's indicate unusable space. Vehicles marked with vertical lines must be shifted.

port, the shifting cost for the first port will always be zero. This also applies to the last port call, as all remaining vehicles are unloaded at this port.

Another important aspect is that each shift is only accounted for once at each port. At the second loading port in Figure 2, two vehicles from cargo 3 use partly the same route, indicated by the arrows. Both vehicles' entry route requires that a vehicle from cargo 1 is shifted, as it is loaded at a previous port. As the vehicle already is shifted to make way for one of the vehicles from cargo 3, e.g. the one furthest to the left, there is no reason to place it back on the deck before the other vehicle from cargo 3 has been placed on the deck. Thus, the shifting cost from the two routes add up to 4, and not 8, which is the sum of the shifting cost for each of the two vehicles.

So to summarize, the SPEP deals with evaluating given stowage plans by determining which vehicles to shift at each port call to minimize the shifting cost of the plan. We consider the stowage plan for a single deck, as we assume that there exists an open path from the entry point of the ship to each deck. Then, when solving for several decks, the total shifting cost is then given by the sum of shifting costs for each deck.

3 Mathematical formulation

Building on the previous work in [3], the SPEP is based on a grid representation of the given deck, where $\mathcal{I}$ is the set of rows and $\mathcal{J}$ the set of columns over a deck, indexed by i and j, respectively. A square (i, j) represents a physical area on the deck where the vehicles may be placed, where square (1,1) is defined as the square located at the stern of the ship's port side. We define the parameter E_{ij} to be 1 if square (i, j) is the entry/exit point on the deck. Let $\mathcal{C}$, indexed by c, be the set of cargoes carried along the voyage. Each cargo c consists of N_c identical vehicles (or units). The set of vehicles $\mathcal{V}_c$ includes all vehicles v from cargo c. The length and width of each vehicle in cargo c, in squares, are given by S_c^L and S_c^W, respectively. Let $\mathcal{P}$ be the set of ports visited along the voyage, indexed by p. Each cargo $c \in \mathcal{C}$ is to be loaded at port P_c^L and unloaded at port P_c^U. We assume a given stowage plan for the voyage is provided, and define parameter P_{ijcvp} to be 1 if vehicle v from cargo c occupies square (i, j) when departing from port p. Further, let P_{ijcvp}^C be 1 if the lower left corner of vehicle v from cargo c is placed in square (i, j) when departing port p. The feasible stowage plan is now given by the parameters P_{ijcvp} and P_{ijcvp}^C.

To represent the SPEP, we present a fixed-charge multicommodity network flow formulation of the problem and define the following additional notation. Let $\mathcal{N}$ be the set of all squares (i, j) and $\mathcal{N}_c \subseteq \mathcal{N} \backslash \mathcal{U}_c$ be the set of squares (i, j) a vehicle from cargo c may use on its entry/exit route. The set $\mathcal{U}_c$ includes all squares that cannot be used by vehicles from cargo c, which typically are squares where pillars and other blocking objects are placed and squares where the weight restrictions are violated. Each cargo c has a graph $\mathcal{G}_c = (\mathcal{N}_c, \mathcal{A}_c)$ associated with it. The set of squares $\mathcal{N}_c$ and the set of arcs $\mathcal{A}_c \subset \mathcal{N}_c \times \mathcal{N}_c$ define the feasible movements for the vehicles in cargo c. Further, the set $\mathcal{C}_p$ includes all cargoes placed on the deck at port p. This set can further be divided into two disjoint sets: $\mathcal{C}_p^R$ includes all cargoes placed on the deck at port p, given that the port is either the loading port or the unloading port of cargo c, i.e. $p = P_c^L$ or $p = P_c^U$. Thus, this set includes all cargoes that are to be routed on or off the ship in the given port p. Next, the set $\mathcal{C}_p^N$ includes all cargoes placed on the deck at port p, given that the port is neither the loading port nor the unloading port of cargo c, i.e. $p \neq P_c^L$ and $p \neq P_c^U$. If any of these cargoes is shifted at port p, a shifting cost C_c^S is imposed. The shifting cost is based on the vehicle's area, i.e. $C_c^M = S_c^L S_c^W$, as it is usually more time-consuming to move larger vehicles than smaller ones. It is assumed that a shifted vehicle is moved off the deck during the port call and returned to the same location after the loading/unloading.

Let $\mathcal{B}_{ij}$ be the set of all neighboring squares to square (i, j), i.e. $\mathcal{B}_{ij} = \{(i+1, j), (i-1, j), (i, j+1), (i, j-1)\}$. A vehicle is allowed to move one square forward, backward, left or right from each square (i, j). By allowing sideways movement, some of the proposed entry and exit routes may be infeasible in practice, as most vehicles have a given turning radius and are unable to move sideways. However, it should be emphasized that the stowage plan evaluation is mainly conducted to compare alternative stowage plans and not to actually determine the optimal loading and unloading routes for each vehicle. Hence, it can be argued that this modeling choice is reasonable and sufficient.

We associate with each square $(i, j) \in \mathcal{N}$, cargo c, and port p, an integer number D_{ijcp} representing its supply/demand. If $D_{ijcp} > 0$, square (i, j) is a supply square for cargo c at port p; if $D_{ijcp} < 0$ square (i, j) is a demand square for cargo c at port p with a demand of $-D_{ijcp}$; and if $D_{ijcp} = 0$, square (i, j) is a transshipment square for cargo c at port p. If we are to load four vehicles from cargo 3 at port 4 and the entry point is square (1,1), then we have a supply of four, represented by $D_{1123} = 4$. Further, we have a demand of -1 at the squares where the vehicles are to be stowed, given by the parameter P_{ijcvp}^C for each of the vehicles, i.e. $D_{ijcp} = \sum_{v \in \mathcal{V}_c} -P_{ijcvp}^C$. Similarly, for a cargo's unloading port, the exit square gets a demand of $-N_c$, and $D_{ijcp} = 1$ for all squares where the lower left corner of a vehicle from cargo c is stowed. We define the parameter A_{ijcdvp} to be 1 if a vehicle from cargo c, placed in square (i, j), uses a square where vehicle v from cargo d is placed at port p. If a 2×2 sized vehicle from cargo c temporarily uses square (1,1) for its entry route, this vehicle also uses the squares (1,2), (2,1) and (2,2). If a vehicle v from cargo d is placed in any of these four squares, then $A_{11cdvp} = 1$, which imply that the vehicle v from cargo d must be shifted, if a vehicle from cargo c uses square (1,1).

The arc flow variable f_{ijklcp} represents the flow sent from square (i, j) to a neighboring square (k, l) of cargo c at port p. Finally, binary variables y_{cvp} take value 1 if vehicle v from cargo c is shifted at port p.

The model is solved for every port except the first and the last port along the voyage, since no vehicles are shifted in these ports. Let $\mathcal{P}^S = \mathcal{P} \backslash \{\text{first port, last port}\}$ be the set of ports where shifting may occur. The resulting shifting cost for a voyage is the sum of the shifting costs for each port. We can now formulate the SPEP problem for each $p \in \mathcal{P}^S$ as the following mathematical program:

$$\text{SPEP}(p \in \mathcal{P}^S)\text{: min } z_p = \sum_{c \in \mathcal{C}_p^N} \sum_{v \in \mathcal{V}_c} C_c^M y_{cvp} \tag{1}$$

subject to

$$\sum_{(k,l) \in \mathcal{B}_{ij}} f_{ijklcp} - \sum_{(k,l) \in \mathcal{B}_{ij}} f_{klijcp} = D_{ijcp} \quad c \in \mathcal{C}_p^R, (i,j) \in \mathcal{N}_c \tag{2}$$

$$A_{klcdvp} \sum_{(i,j) \in \mathcal{B}_{kl}} f_{ijklcp} \leq M_c y_{dvp} \quad c \in \mathcal{C}_p^R, (k,l) \in \mathcal{N}_c, d \in \mathcal{C}_p^N, v \in \mathcal{V}_d \tag{3}$$

$$f_{ijklcp} \geq 0 \quad c \in \mathcal{C}_p^R, ((i,j),(k,l)) \in \mathcal{A}_c \tag{4}$$

$$y_{cvp} \in \{0,1\} \quad c \in \mathcal{C}_p^N, v \in \mathcal{V}_c \tag{5}$$

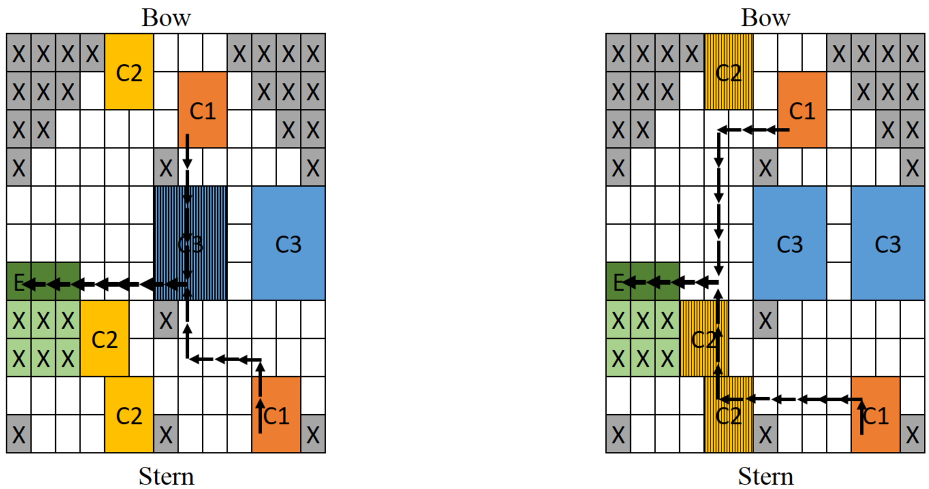
The objective function (1) is to minimize the cost of shifting vehicles at port p. The flow balance constraints (2) state that the outflow minus inflow must equal the supply/demand of the square i for each cargo c. Constraints (3) are the capacity constraints. If a given vehicle v from cargo d blocks the flow of vehicles from cargo c into square (k,l) at port p, given by A_{klcdvp}, then the blocking vehicle v from cargo d must be shifted to enable flow into the square, i.e. the shifting variable $y_{dvp} = 1$. An upper bound on M_c is given by N_c, i.e. the number of vehicles in cargo c. Constraints (4) and (5) define each variable's domain.

Figure 3a illustrates an optimal solution to the SPEP, obtained by solving model (1)-(5), for an example instance. Here, exit routes for the two vehicles of cargo 1 are to be decided, for a given unloading port. Vehicles from cargoes 2 and 3 must be shifted if routes passing through them are used, as they are to be unloaded at a later port call. The cheapest way to unload the two vehicles from cargo 1 is to shift the marked vehicle from cargo 3. This vehicle has a shifting cost of 9, i.e. the vehicle's area, resulting in a shifting cost of 9 for this port.

4 Shortest path solution method

When determining a stowage plan by using either an exact or a heuristic solution method, the SPEP becomes an important sub-problem that may have to be solved a large number of times to evaluate and compare the shifting costs of alternative stowage plans. Therefore, it can be important to solve the SPEP very quickly. The SPEP model, defined in Section 3, consists of a large number of the continuous flow variables, but relatively few binary variables (one for each vehicle that may be shifted). Due to this, small instances are easily solved by a commercial solver. However, for the large grid resolutions required for solving realistically sized problems, even solving the LP-relaxation of the SPEP model can be too time-consuming for practical use. Thus, we propose a heuristic method based on solving shortest path problems, which can be solved efficiently.

The SPEP deals with deciding which vehicles to shift, considering all cargoes that are to be loaded or unloaded at a given port. Instead of considering all cargoes that are to be loaded/unloaded at a port simultaneously, we consider one vehicle at the time. Thus, for a given vehicle in both its loading and unloading



(a) Optimal solution from the SPEP model

(b) Feasible routes, proposed by the heuristic in Section 4

Fig. 3: Exit routes for the two vehicles in cargo 1 (C1) are to be decided, for a given unloading port. Vehicles marked with vertical lines are shifted.

port, the problem is to decide which vehicles to shift, such that this vehicle may be loaded/unloaded. The objective is to minimize the shifting cost. This problem is equivalent to finding the cheapest entry/exit path, where the cost of moving from one square to another is dependent on whether we need to shift any vehicles due to the move, i.e. a shortest path problem (SPP).

When using the shortest path approach to solve the example instance in Figure 3b, we evaluate the exit routes for each vehicle individually. This results in a total shifting cost of 12, i.e. three shifted vehicles with an area of four squares. The solution to the left has shifting cost of 9, as explained in the previous section. It can be shown that the shortest path approach gives an upper bound on the shifting cost for a given port.

The remainder of this section is structured as follows. Section 4.1 describes the procedure of creating the graphs on which the SPPs are solved. Then, in Section 4.2, the procedure of solving a shortest path problem on the graphs are discussed and we present some additional strategies for improving the upper bound on the shifting cost.

4.1 Creating the graphs

For this SPP heuristic, we are not concerned with finding the shortest path in distance, but rather the cheapest path, i.e. the entry/exit route that gives the lowest contribution to the shifting cost. The cheapest paths can be found by solving a one-to-one shortest path problem for each vehicle's loading and unloading port. It is not possible to solve a single one-to-many SPP for each port, where the cheapest path for each entering or exiting vehicle is calculated, as edge costs are dependent on the size of the vehicle traversing the edge. We explain this using a 2×2 sized vehicle and a 4×4 sized vehicle, both initially placed with their lower left corner in square $(1,3)$, as illustrated in Figure 4. Then, we evaluate the edge cost associated with moving one square to the left for both vehicles. For the smallest vehicle, the edge cost is zero, as moving from square $(1,3)$ to $(1,2)$ does not impose any shifts. However, the largest vehicle uses a square where another vehicle is placed. The edge cost equals the cost of shifting the blocking vehicle, which in this case is six, i.e. the vehicle's area.

Since the edge costs depend on the size of the cargo, a graph $\mathcal{G}_{cp} = (\mathcal{N}_c, \mathcal{A}_c)$ must be created for every cargo, for their loading and unloading port $\mathcal{P}_c^{LU} = \{P_c^L, P_c^U\}$. We do not need to create a graph for each vehicle, as all vehicles within a cargo have the same dimensions. Let $\mathcal{N}_c$ be the set of squares (i,j) a vehicle from cargo c may use on its entry/exit route. The set of edges $\mathcal{A}_c$ defines the feasible movements for the vehicles, i.e. maximum four directed outgoing edges per square. We add the edge $((i,j),(k,l))$ to the set of edges $\mathcal{A}_c$, for all $(i,j) \in \mathcal{N}_c$ and $(k,l) \in \mathcal{B}_{ij}$.

Each time an edge is added to the set $\mathcal{A}_c$, the corresponding edge cost C^E_{ijklcp} is calculated. For this edge $((i,j),(k,l))$, let $\mathcal{V}^I$ be the set of vehicles that may be shifted and occupy any of the squares $(\bar{i},\bar{j})$, $\bar{i} = i..i + S^L_c - 1, \bar{j} = j..j + S^W_c - 1$. Similarly, let $\mathcal{V}^N$ be the set of vehicles that may be shifted and occupies any of the squares $(\bar{k},\bar{l})$, $\bar{k} = k..k+S^L_c-1, \bar{l} = l..l+S^W_c-1$. Then, all vehicles that must be shifted due to the move from (i,j) to (k,l) are given by the set $\mathcal{V}^S = \mathcal{V}^N \backslash \mathcal{V}^I$, i.e. all vehicles that occupy some of the squares in the new position which was not occupying any of the squares in the initial position. The edge cost C^E_{ijklcp} equals the cost of shifting all vehicles in the set $\mathcal{V}^S$.

Evaluating the move shown in Figure 4 for the large 4×4 vehicle, the cost of using edge $((1,3),(1,2))$ is six. The use of edge $((1,3),(1,2))$ implies that the lower left corner of the vehicle is moved from square (1,3) to (1,2), as shown in Figure 4. Initially, no vehicles that may be shifted are placed in the squares occupied by the vehicle, and the set $\mathcal{V}^I$ is therefore empty. After the move to square (1,2), square (4,2) is used by the moving vehicle. The blue 3×2 vehicle is placed in this square and must be shifted due to the move. Thus, the blue vehicle is added to the set $\mathcal{V}^N$. The set of vehicles to shift due to the move from

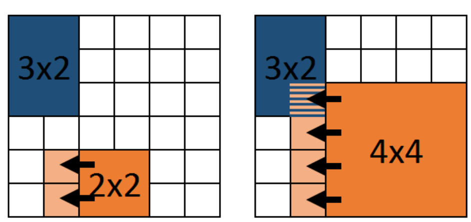


Fig. 4: The cost of moving a vehicle from a square to a neighbor square depends with the size of the moving vehicle.

square (1,3) to (1,2) are now given by the vehicles in the set $\mathcal{V}^S = \mathcal{V}^N \setminus \mathcal{V}^I$, which includes the blue vehicle with a shifting cost of six, giving a edge cost of six. Continuing the example, we now state that a new move from square (1,2) to square (1,1) is free, as the cost of shifting the blue vehicle is already accounted for by the move from square (1,3) to (1,2). For this new move, the blue vehicle initially occupies one of the squares used by the 4×4 vehicle and is added to the set $\mathcal{V}^I$. The blue vehicle is also added to the set $\mathcal{V}^N$, as it uses one or more of the same squares as the 4×4 vehicle after the move to square (1,1). As both sets include the blocking vehicle, the set $\mathcal{V}^S = \mathcal{V}^N \setminus \mathcal{V}^I$ is empty. Thus, no shifting cost is imposed due to this move. One final note regarding this procedure is that vehicles that are to be loaded or unloaded in the same port are never added to the sets $\mathcal{V}^I$ and $\mathcal{V}^N$. As mentioned in Section 2, there always exists a loading/unloading sequence, where none of the loading/unloading vehicles block the other loading/unloading vehicles.

4.2 Shifting cost calculations

Given the graphs created in the previous section, we can now calculate an upper bound on the shifting cost. For each vehicle at its corresponding loading and unloading port, a one-to-one SPP can be solved on the graph created for the cargo the vehicle belongs to. For a given vehicle at its loading port, the loading cost is given by the value of the vehicle's corresponding target node from the SPP solution, where the entry square is the start node. For the vehicle's unloading port, the unloading cost is given by the vehicle's target node, which is the exit square. The sum of the loading and unloading cost for all vehicles in all cargoes gives an upper bound on the shifting cost of the stowage plan. However, this upper bound can be poor, as several of the shifted vehicles may be accounted for more than once. This section presents three strategies which improve both the computational time and the solution quality of the shifting cost evaluation. The *Dijkstra-NoDec* procedure given in [2] is used for solving the SPPs.

4.2.1 Reducing the number of SPPs

In order to speed up the solution procedure, we propose to solve a one-to-many SPP for each cargo's loading and unloading port, instead of many one-to-one SPPs for each vehicle's loading and unloading port. This is possible, as all vehicles in a cargo have the same dimensions, and thus, the same edge costs. When solving for a loading port, the start node is set to entry square on the deck. The target nodes are the nodes where the lower left corner of the vehicles in the cargo are to be placed, provided by the stowage plan. The shortest path for each vehicle in the cargo is now given by the shortest path between the start node and the vehicles target node. When solving for an unloading port, the exact same procedure is used. We use the fact that the shortest path from the exit to a vehicle's location is always the same as the shortest path from the vehicle's location to the exit. So, for an unloading port, we also seek the shortest path from the exit square to each vehicle's location, which can be solved as a one-to-many SPP.

Even though each SPP takes more time to solve, as we now solve one-to-many SPPs, we have reduced the number of SPPs needed to be solved from two times the number of vehicles to two times the number of cargoes, which improves the overall computational time.

4.2.2 Route backtracking

We know that the vehicle's target node gives the cost of loading/unloading a vehicle. However, two or more vehicles may use a loading or unloading route which shifts the same vehicles. If the loading cost of both vehicles A and B are six, this could for example imply that a given vehicle C, with a shifting cost of six, is shifted for both vehicles A and B. We use the fact that each vehicle is only shifted once at each port and backtrack the loading/unloading routes for all vehicles loaded/unloaded in at a port, recording the unique shifted vehicles at each port. A better bound is then given by the sum of the cost of shifting each of the recorded vehicles. In the example above, the bound now becomes six, and not 12, where the latter is the sum of vehicle A and B's loading cost.

4.2.3 Dynamically updating the edge costs

At each port, there are usually several cargoes that are to be loaded or unloaded, and we solve an SPP for each of these, one at the time. By continuously sharing information about which vehicles that are shifted, we can improve the upper bound of the shifting cost. As an example, assume two cargoes are to be loaded at a port. An SPP is solved for the first cargo, and three vehicles must be shifted. When solving the SPP for the second cargo, we can now set the edge costs where these three vehicles are placed to 0, as they must be shifted regardless of the solution for the second cargo. Now, the SPP solution for the second cargo may utilize this and choose alternate routes for the vehicles. The routes may be more expensive when considering only this cargo, but as these vehicles are shifted anyway, the routes contribute less to the overall shifting cost. When imposing this strategy, the order in we solve the SPPs becomes important. Based on preliminary testing, the cargoes should be ordered based on the area of the vehicles they consist of. Consider a large truck driving from A to B. If we shift the vehicles blocking the truck's route, both the truck and a small car can use this route. This does not apply the other way around. Thus, evaluating the cargoes with the largest vehicles first is both logical and computational promising.

Dynamically updating the edge costs can also be used to improve the shifting cost estimation within each cargo. We know that as soon as a vehicle has found its shortest path to the entry square, all other vehicles in the cargo could also use this path as part of their route, with no additional cost, as the vehicles have the same size. Thus, each time the shortest path between the start node and a target node has been found during the one-to-many SPP procedure, we force this path to be used by the vehicle placed in the corresponding target node. Then, an edge is added between this target node and the start node, with an edge cost of zero. This means that the vehicles placed at the remaining target nodes could either find the shortest path to the entry square or the shortest path to one of the fixed vehicle's target nodes. This strategy is especially valuable when several vehicles within the same cargo are placed together. Once the shortest path for one of these vehicles to the exit is found, e.g. vehicle A, the other vehicles may move to vehicle A's location, and the shortest path is found. Note that if a vehicle is not placed nearby vehicle A, it can still use the path from vehicle A to the exit as a partial path of its route at no additional cost.

5 Computational study

The mathematical model presented in Section 3 is implemented in Mosel and solved using Xpress-IVE 1.28.12. The SPP heuristic described in Section 5 is coded in Java. All computational experiments have been run on a PC with Intel Core i7-6500U processor and 16 GB of RAM running Windows 10.

5.1 Test instances

Each instance represents a feasible stowage plan for a given deck, cargo list, and grid resolution, where the placement of each vehicle is given. The deck information gives the layout of the deck, location of entry/exit square, and gives the unusable spaces. Four decks are used in this computational study, where decks A and B are fictional, and decks C and D are real deck layouts. Four different grid resolutions are used, i.e. 50×19, 100×38, 150×56, and 200×75. The ratio between the number of rows and columns is based on the length and width of a car equivalent unit (CEU), which has an approximate length-width ratio of 4:1.5. Each stowage plan has a space utilization factor (SUF), i.e. the total area of all vehicles divided by the deck's area. The SUFs used are 0.8 and >0.9. When creating an instance, the vehicles are either randomly placed on the deck (R) or placed in a logical manner (L) according to objective function (3) in [3]. Finally, the cargo list contains information about the number of cargoes, loading and unloading port of each cargo, and each vehicle's size and weight. Two cargo lists are tested for each possible combinations of the mentioned aspects.

We identify a test instance by its name Rows-Columns-Deck-SUF-Placing-Cargo List. 150-56-D->0.9-R-2 means an instance with a grid resolution of 150×56 on the real deck D with cargo list number 2, where more than 90% of the deck's area is occupied by vehicles and the vehicles are randomly placed.

For the test instances, we have set the upper limit on computing time to 7,200 seconds for both the solution methods. As a SPEP problem is solved for each port except the first and last port, the computing time is distributed evenly among these ports. Thus, for an instance with six ports, the maximum computing time for each of the ports 2-5 are 1,800 seconds.

5.2 Comparison of SPEP model and SPP heuristic

To compare the performance of SPEP and the SPP heuristic, we have tested both methods on 64 instances. Table 1 presents a summary of the results, where the average results from cargo lists 1 and 2 for each instance are presented. As the SPEP model is solved for every port for each instance, a positive gap may be reported even though the solution time is less than the maximum computation time. The percentage of the gap closed is calculated as $(z_u - z_b)/z_b$ where z_u is the best solution and z_b is the best bound.

In Table 1, we have reported the SPP results from the SPP heuristic using all improvement strategies, denoted SPP_A. To evaluate the effect of the improvement strategies discussed in Section 4.2, we have tested the base heuristic (SPP_B), i.e. without any of the strategies, where the average gap and solution time are 488% and 0.55 seconds, respectively. These results are significantly worse than SPP_A's results and are therefore not reported in the table. For the instances solved to optimality by Xpress, SPP_A found solutions with an average gap of 8.5%. 37 of the 64 instances were solved to optimality by Xpress, using the SPEP model. The SPP heuristic found a solution proved optimal by the SPEP model in two of the 37 instances.

Even for the smallest grid resolution, the SPEP model's computing times are too long for practical use. The SPP heuristic performs significantly better, reducing the average solution times to less than 0.1 seconds, which is acceptable in its intended context, i.e. as a sub-problem when determining a stowage plan. For both methods, grid resolution has the highest impact on solution time.

Table 1: Average gaps and solution times per group of instances. Each group consists of two instances with different cargo lists.

Group of instances	Vehicles logically placed				Vehicles randomly placed			
	SPEP		SPP_A		SPEP		SPP_A	
	Avg. gap(%)	Avg. CPU (s)	Avg. gap(%)	Avg. CPU (s)	Avg. gap(%)	Avg. CPU (s)	Avg. gap(%)	Avg. CPU (s)
50-19-A->90	0.0	12.72	16.4	0.01	0.0	229.82	12.4	0.01
50-19-A-80	0.0	6.22	2.3	0.01	0.0	22.66	5.6	0.01
50-19-B->90	0.0	20.67	9.2	0.01	0.0	408.17	4.6	0.01
50-19-B-80	0.0	46.40	7.7	0.01	0.0	365.21	6.5	0.01
100-38-C->90	3.2	1050.93	6.6	0.03	0.0	397.58	7.7	0.03
100-38-C-80	0.0	299.22	8.1	0.03	0.0	391.90	8.1	0.04
100-38-D->90	8.5	840.57	13.0	0.04	23.6	3077.87	30.1	0.05
100-38-D-80	0.0	341.50	8.2	0.03	13.2	1763.38	17.8	0.05
150-56-C->90	0.0	116.74	8.8	0.08	0.0	877.34	8.5	0.09
150-56-C-80	0.0	95.81	1.8	0.07	4.5	1350.81	9.5	0.10
150-56-D->90	7.4	1224.54	28.3	0.08	75.2	4315.77	97.8	0.12
150-56-D-80	16.0	933.72	29.9	0.09	57.4	4240.85	79.6	0.12
200-75-C->90	15.8	4819.01	25.6	0.18	29.3	3457.04	39.9	0.16
200-75-C-80	6.8	3072.33	6.9	0.16	34.1	3451.87	38.8	0.17
200-75-D->90	58.3	5117.53	54.7	0.20	82.4	4629.67	75.3	0.23
200-75-D-80	22.7	2317.14	29.1	0.22	99.7	4882.36	85.9	0.22
Avg.	8.7	1269.69	16.0	0.08	26.2	2116.39	33.0	0.09

The columns to the left in Table 1 present the results for the instances where the vehicles are placed logically. Considering the SPEP model's results, we see that both the average solution time and the gap are lower when the vehicles are logically placed, than for the randomly generated stowage plans. For the SPP heuristic, we see that the solution times are approximately the same for both placement methods, while the gap is higher when vehicles are randomly placed.

5.3 Comparing stowage plans

We stress that the intended use of the SPP heuristic is to compare stowage plans, not determine the actual shifting cost of a stowage plan. Even though the gaps reported in the previous section was relatively high, this is acceptable as long as they are consistently low/high for all the stowage plans. Thus, the important aspect is that the solutions from the SPP heuristic follow the same trend as the SPEP model's solutions, with respect to the shifting cost evaluated. The quality of the SPP heuristic should be, and is here, determined by its ability to decide which is the better stowage plan, given a set of stowage plans.

To test the quality of the SPP heuristic, the following decision problem is used: Given two stowage plans A and B, which is better? The two stowage plans are evaluated by both the SPEP model and the SPP heuristic, where the better stowage plan is the one with the lowest shifting cost. The SPP heuristic succeeds if it reports the same best stowage plan as the SPEP model. Ten groups of instances are used to test the SPP heuristic's quality. All instances within each group have the same grid resolution, deck, SUF, and cargo list, but the vehicles' placement differ. Each group consists of 50 randomly generated stowage plans. The decision problem is asked for every unique pair of stowage plans out of the 50 plans, resulting in $C(50, 2) = 1225$ decision problems. The success rate equals the number of successful evaluations divided by the number of combinations.

Table 2: SPP heuristic's success rate (SR) per group of instances.

	SPP_B	SPP_A		
Group of instances	SR (%), $t = 0.0$	SR (%), $t = 0.0$	SR (%), $t = 0.025$	SR (%), $t = 0.05$
50-19-A-80-R-1	62.0	88.9	94.4	97.3
50-19-A-80-R-2	64.5	87.4	92.3	95.7
50-19-A->90-R-1	70.7	91.1	95.7	98.4
50-19-A->90-R-2	78.6	93.8	96.9	98.8
100-38-C-80-R-1	71.5	91.1	96.5	99.2
100-38-C-80-R-2	67.1	86.4	93.7	97.0
100-38-C->90-R-1	53.7	88.3	94.1	97.2
100-38-C->90-R-2	64.8	90.4	95.8	99.0
150-56-C-80-R-1	68.1	91.7	95.0	97.9
150-56-C->90-R-1	64.3	86.5	92.9	96.3
Avg.	66.5	89.6	94.7	97.7

Table 2 presents a summary of the test results. Without any improvement strategies, i.e. SPP_B, we see that the average success rate is 66.5%, which is not much better than random guessing (50%). However, with the improvement strategies (SPP_A), the average success rate improves to 89.6%. The two rightmost columns in Table 2, show the success rate when both answers to the decision problem are accepted if relative difference in shifting cost below a certain tolerance value t. The relative difference in shifting cost between stowage plans A and B is given by $|z(A) - z(B)|/\min\{z(A), z(B)\}$. Based on the results, SPP_A will successfully identify the better stowage plan 97.7% of the times on average, given that the two stowage plans have a relative difference in shifting cost greater than 5%. This is certainly an acceptable result for the method's intended use, i.e. as a subroutine in a solution method for generating stowage plans.

6 Concluding remarks

This paper has introduced the stowage plan evaluation problem, which is solved to compare different stowage solutions for a voyage in Roll-on Roll-off liner shipping. We have presented a mathematical formulation describing the problem. To efficiently solve the problem we proposed a shortest path based heuristic.

Our computational tests indicate that the shortest path based heuristic is a promising method for comparing different stowage plans. The problem of determining the better of two stowage plans was on average successfully solved 9 out of 10 times by the heuristic. In the case where stowage plans with less than 5% difference in shifting cost were considered equally good, the average success rate increased to 98%. In addition to a high average success rate, short computation times was reported for the proposed solution method. These promising results enable the solution method to be used as a subroutine in a stowage plan generator, which is an interesting venue for future research.

References

1. Ambrosino, D., Paolucci, M., Sciomachen, A.: Experimental evaluation of mixed integer programming models for the multi-port master bay plan problem. Flexible Services and Manufacturing Journal 27(2-3), 263–284 (2015)
2. Chen, M., Chowdhury, R.A., Ramachandran, V., Roche, D.L., Tong, L.: Priority queues and Dijkstra's algorithm. Computer Science Department, University of Texas at Austin (2007)
3. Hansen, J.R., Hukkelberg, I., Fagerholt, K., Stålhane, M., Rakke, J.G.: 2d-packing with an application to stowage in roll-on roll-off liner shipping. Lecture Notes in Computer Science 9855, 35–49 (2016)
4. Øvstebø, B.O., Hvattum, L.M., Fagerholt, K.: Optimization of stowage plans for roro ships. Computers & Operations Research 38(10), 1425–1434 (2011)
5. Pacino, D., Delgado, A., Jensen, R.M., Bebbington, T.: Fast generation of near-optimal plans for eco-efficient stowage of large container vessels. Lecture Notes in Computer Science 6971, 286–301 (2011)
6. Parreño, F., Pacino, D., Alvarez-Valdes, R.: A grasp algorithm for the container stowage slot planning problem. Transp. Res. Part E 94, 141–157 (2016)