Computers & Operations Research 133 (2021) 105345

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

A revised formulation, library and heuristic for a chemical tanker scheduling problem

Anurag Ladage [a,b,c,*], Davaatseren Baatar [b], Mohan Krishnamoorthy [d,e], Ashutosh Mahajan [c]

[a] IITB-Monash Research Academy, IIT Bombay, Powai, Mumbai 400076, India
[b] Department of Mathematics, Monash University, Melbourne, Victoria 3800, Australia
[c] Department of Industrial Engineering and Operations Research, IIT Bombay, Powai, Mumbai 400076, India
[d] School of Information Technology and Electrical Engineering, The University of Queensland, QLD 4072, Australia
[e] Department of Mechanical and Aerospace Engineering, Monash University, Clayton, VIC 3800, Australia

ARTICLE INFO

Keyword:
Integer programming
Maritime transportation
Ship routing
Tramp shipping
Scheduling with time-windows
Tank allocations

ABSTRACT

We study *a chemical tanker scheduling problem* in which a multi-compartment chemical tanker picks up chemicals (within specific time windows) and delivers them to their destinations. Additional safety and ship balancing requirements related to the compartment storage need to be met. We refer to this problem as the *single ship pick-up and delivery problem with pick-up time windows, tank allocations and changeovers* (s-PDP-TWTAC). We propose a formulation for the s-PDP-TWTAC that is significantly smaller in size when compared to the existing approaches. Even though our formulation requires significantly lesser computational effort than others, it is still challenging to solve for large realistic networks. We propose a linear programming based neighbourhood search heuristic to find good solutions to these problems. We also describe an instance generator for creating random but realistic instances, and a library of instances that have been created for testing and benchmarking.

1. Introduction

We consider a scheduling problem that the *tramp shipping industry* faces. Tramp ships are analogous to a cab service (Christiansen et al., 2004) as opposed to liner ships that can be viewed as a bus service. Tramp ships capable of transporting multiple non-mixable chemicals together are known as chemical tankers. Given a list of ports and potential cargoes, the problem is one of finding a schedule of a chemical tanker. A schedule consists of a subset of ports to visit, the planned arrival times at each port, cargoes to be serviced by the ship at each of these ports, and the cargo weights assigned to each compartment of the ship. According to Brooks and Faust (2018), the maritime industry accounts for about 80 % to 90 % of the total world merchandise trade by volume and 60 % of the total trade by value. Chemicals, including petroleum products, constitute an essential class of merchandise. A United Nations (2019) report states that the maritime industry transports nearly 200 million tons of chemicals annually.

Our goal is to develop a mathematical model capable of generating a schedule based on important operational and tactical constraints for a single chemical tanker. Empirical computational results described later show that even the improved model is intractable for medium-sized benchmark instances. Consequently, a systematic study of the formulation and solution techniques is required. Additionally, motivated by the unavailability of freely accessible data, we build an instance generator, and a library of instances, that may help in developing better models and solution techniques.

The problem definition is adopted from Neo et al. (2006). At any given time, the chemical tanker operator may have a list of unassigned cargoes (chemicals). The unassigned cargoes have their respective origin and destination ports. Additionally, these cargoes need to be picked up within a specific time window (if not, the operator could risk losing customers). The chemical tanker moves from one port to another, transporting multiple chemicals simultaneously. While transporting these several chemicals concurrently, the operator has to adhere to the numerous restrictions related to the storage of chemicals in the compartments. The chemicals need to be well-distributed in the chemical tanker's compartments to avoid imbalance during sailing. Finding profitable paths that satisfy the tactical and operational restrictions are essential for chemical tanker operators. We refer to this problem as the single ship pickup and delivery problem with pickup time windows, tank allocations and changeovers (s-PDP-TWTAC).

A chemical tanker is generally much smaller (Wang et al., 2018) than

* Corresponding author at: IITB-Monash Research Academy, IIT Bombay, Powai, Mumbai 400076, India.
E-mail address: anurag.ladage@monash.edu (A. Ladage).

https://doi.org/10.1016/j.cor.2021.105345
Received 3 April 2020; Received in revised form 16 March 2021; Accepted 22 April 2021
Available online 7 May 2021
0305-0548/

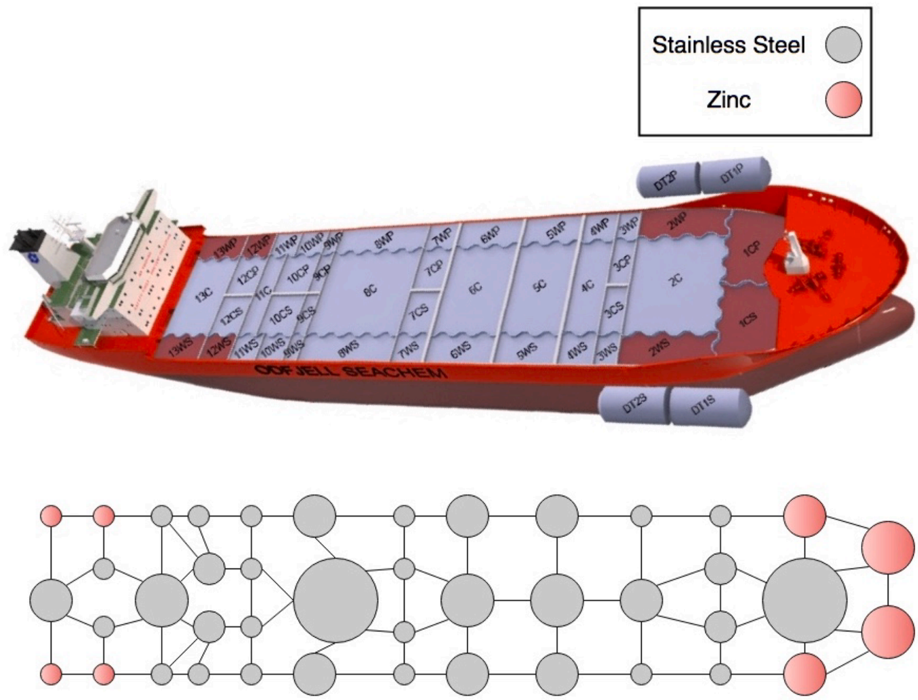

Fig. 1. Bow Cecil – chemical tanker – Odfjell Ship Management.

crude oil-carrying chemical tankers. Nonetheless, a chemical tanker is much more sophisticated and has a higher number of compartments. The number of compartments can typically vary from 10 to 55. Fig. 1 depicts the Bow Cecil chemical tanker used by the Odfjell shipping company[1]. It also shows a planar network representation of the ship's compartment structure.

A higher number of compartments allows for more flexible cargo-compartment allocation plans. This flexibility, in turn, increases the complexity of the problem. The compartment walls of the chemical tanker are made of stainless steel, zinc or epoxy. The compartment material might restrict the possible cargoes that can be stored in the compartment. Additionally, safety regulations may restrict the type of chemicals that may be stored in adjacent compartments. Furthermore, unbalanced distribution of the cargo weight in various compartments may lead to instability of the tanker. All of these additional restrictions in conjunction with the chemical tanker scheduling and time window requirements makes our proposed model quite complex.

This paper is structured as follows. Existing literature related to this problem is reviewed in Section 2. Section 3 describes the problem and modelling assumptions. Subsequently, we present our compact and revised formulation for the proposed single ship pick-up and delivery problem with pick-up time windows, tank allocations, and changeovers (s-PDP-TWTAC). Section 5 explains a linear relaxation-based heuristic for the s-PDP-TWTAC. Then, we describe an instance generator and a benchmark dataset for the s-PDP-TWTAC. We conclude by summarising the insights gained in this work and identifying some directions for future research.

2. Literature survey

Research in maritime transportation can be broadly classified into the liner shipping segment and the tramp shipping segment. One of the first maritime transportation review papers was published by Ronen (1983), Christiansen et al. (2004), Christiansen et al. (2004), Kjeldsen (2011), Kjeldsen (2011), Christiansen et al. (2013) and Brooks and Faust (2018) present excellent summaries of application of mathematical programming in both, liner as well as tramp shipping. Christiansen et al. (2007) specifically focuses on the evolution of scheduling problems in tramp shipping. Further, Christiansen et al. (2013) and Vilhelmsen et al. (2015) extensively review the research carried out in the field of tramp shipping since 1983. Together they review around 50 research papers, most of which have been published in the last decade. Our problem deals with the scheduling of chemical tankers, which is a special type of tramp ship. The literature primarily deals with two different variations of the problem of transporting bulk cargoes between their origins and destinations, respectively. Depending on the application, problems are either modelled as an extension of the travelling salesman problem with pickups and deliveries (TSPPD) or the pick-up delivery problem with time-windows (PDPTW). Our formulation is an extension of the PDPTW.

Extensions of the TSPPD optimise the intra-port movement of the chemical tankers and have time windows related to the loading/unloading of cargoes. TSPPD based formulations assume that all the berths (analogous to ports) have to be visited. Additionally, all cargoes are either loaded or unloaded at their origins and destinations, respectively. As output, the problem generates a sequence of berths to visit, and the arrival times at each berth. Unlike the second variation, the TSPPD based extensions also generate the sequence in which every cargo is either picked up or discharged. This makes the TSPPD based extensions different in comparison to the PDPTW based extensions.

A recent study extending the TSPPD based formulations has been carried out by Elgesem et al. (2018). They consider stochastic waiting times at different terminals within the port. Additionally, they perform experiments to empirically prove that including uncertainty into their models directly affect the optimal route of the chemical tankers. However, unlike our problem, Elgesem et al. (2018) do not generate a cargo-compartment allocation plan as part of their problem output.

Another study related to TSPPD based formulations is presented by Wang et al. (2018). There are some similarities between the problems presented in this paper, and the one presented by Wang et al. (2018). Similar to our problem, Wang et al. (2018) incorporate draft requirements and compartment-related decisions into their problem

[1] https://www.odfjell.com

definition. The compartment-related decisions include compartment capacities, ship stability criteria, cargo-cargo and cargo-compartment incompatibility norms. However, our problem allows more feasible cargo-compartment allocations by permitting cago swaps between the compartments. Further, because their formulation is based on TSPPD, some higher level decisions like generating an optimal set of ports and cargoes are fixed.

The second set of variations of the chemical tanker scheduling problems are an extension of the PDPTW (Sun et al., 2018) problem. These set of problems are defined over a set of ports (inter port), as opposed to a set of berths/terminals (intra port). In addition to generating a sequence of ports to visit, these set of problems also decide the subset of ports to visit, and the subset of cargoes to pick-up. Fagerholt and Christiansen (2000) propose a problem in dry bulk shipping by modifying the PDPTW formulation. They generate a feasible cargo-compartment allocation based on the capacities of the compartments. Yet, they do not consider tanker stability conditions and cargo-cargo incompatibilities.

Jetlund and Karimi (2004), and more recently by Lin and Liu (2011), Cóccola et al. (2015) and Cóccola and Méndez (2015) modify the PDPTW to propose a multi-ship pick-up delivery problem with pickup time windows. Their formulations generate feasible schedules for a fleet of heterogeneous chemical tankers, which takes into account all the decisions shown in the first column of Table 1. Hennig et al. (2015) and Homsi et al. (2020) solve the multi-ship pick-up delivery problem with pickup time windows with an added complexity of splitting cargoes between chemical tankers. Furthermore, Homsi et al. (2020) also includes time-windows related to drop-offs. However, the second variations of problems discussed until now do not consider the compartment-related decisions (mentioned earlier).

Neo et al. (2006) and Santos et al. (2020) have worked on PDPTW based formulations that incorporate the compartment-related decisions in their problem definition. The problem discussed by Santos et al. (2020) is one of transporting fertilisers (chemicals) from their origins to their destinations, respectively. Altough they include the compartment capacity into their problem, they do not consider the ship stability conditions, the incompatibility norms, the cargo distribution into multiple compartments, and the cargo swapping activity. Moreover, Santos et al. (2020) fix the cargoes that have to be picked up and they include soft time-windows as opposed to hard time-windows, considered by us. However, unlike us, they solve a multi-ship problem with split loads.

The problem discussed in this paper was introduced by Neo et al. (2006). As per our knowledge, it is the only other paper that combines higher level decisions with all the aforementioned compartment-related decisions. Higher level decisions include generation of the route of the ship, determining the set of unassigned cargoes to service, and minimising the time required to complete the voyage of the chemical tanker. Neo et al. (2006) establish the necessity to solve the combined problem by discussing a real-world case study. However, they do not validate their model using an extensive empirical study. Extending their work forward, we present a revised MILP formulation in Section 4.1. We model the changeover (cargo swaps) activity differently, which reduces the problem size and tightens the linear relaxation of the problem. Additionally, we model the changeover activity as a temporal activity in our model. We also generalise the definition of the pick-up time windows. We present a detailed discussion of the differences between the original formulation presented by Neo et al. (2006), and our revised formulation in Section 4.2.

Individually, the problem of generating a cargo-compartment plan based on the compartment-related decisions is complex. Hvattum et al. (2009) and Vilhelmsen et al. (2016) provide some insight into the problem complexity, and refer to the problem of generating cargo-compartment allocations as the *Tank Allocation Problem* (TAP). They introduce multiple variants of the TAP, and prove that the problem is NP-complete. Table 1 highlights the contributions in the literature as well as differentiates our problem from the rest. Having identified the gap in the literature, we present a brief description of our problem.

3. Problem description

The single ship pick-up and delivery problem with pick-up time windows, tank allocations and changeovers (s-PDP-TWTAC) models the scheduling of a chemical tanker on a network of ports. We consider a chemical tanker with a list of onboard chemicals (cargoes), which need to be transported to their destinations, respectively. At the same time, unassigned cargoes can be picked up by the chemical tanker. Our goal is to generate a schedule for the chemical tanker, which includes cargoes to be picked up, the ports to be visited, the sequence in which these ports are visited, the arrival times at each port, and a feasible cargo-compartment allocation. Our objective tries to maximise the difference

Table 1
Problem characteristics tackled by researchers working on problems similar to the s-PDP-TWTAC

Decisions optimised by researchers	Researchers tackling problems similar to the s-PDP-TWTAC					
	Jetlund and Karimi (2004), Lin and Liu (2011), Cóccola et al. (2015), Cóccola and Méndez (2015)	Hennig et al. (2015), Homsi et al. (2020)	Hvattum et al. (2009), Vilhelmsen et al. (2016)	Wang et al. (2018)	Fagerholt and Christiansen (2000)	Neo et al. (2006), series s-PDP-TWTAC (Our work)
Single ship			✓	✓	✓	✓
Multiple ships	✓	✓				
Port set	✓	✓			✓	✓
Port sequence	✓	✓		✓	✓	✓
Cargo pick-ups and drop-offs	✓	✓			✓	✓
Cargo pick-up and drop off sequence	✓	✓		✓	✓	✓
Pick-up time window	✓	✓		✓	✓	✓
Drop off time window		✓			✓	
Compartment capacity			✓	✓	✓	✓
Ship stability			✓	✓		✓
Ship's draft				✓		✓
Cargo split between ships		✓				
Cargo-cargo compatibility			✓	✓		✓
Cargo-compartment compatibility			✓	✓		✓
Cargo swapping						✓

between the revenue and four different costs. These costs include the time charter cost, the fuel cost incurred while travelling between ports, the fixed cost associated to a port call, and the cargo swapping cost incurred for every intra-compartment cargo swap.

To make the problem tractable, the intra port activities have been simplified. During each port visit a constant administrative time incorporates activities such as waiting time for berth allocation, repairs, re-fueling, security clearances, immigration procedures, and delays related to custom inspections. We assume half the administrative time is spent on security checks, following which the cargo-compartment assignments are decided. We term this point as the cargo-compartment assignment point. At this point, we decide the pick-up cargoes and generate a cargo-compartment plan. The cargo-compartment assignment point presented should lie within the pick-up time window of each of the cargoes. The pick-up time windows are specified in units of days (fractional days are allowed). Any cargo that is picked up has to be delivered within the time horizon. All the temporal inter-port and intra-port activities are performed sequentially, one after the other.

Every chemical tanker has a maximum draft limit, which limits its cargo carrying capacity. Moreover, a chemical tanker has multiple compartments or cargo holds. Each compartment can store at most one cargo, but a cargo can be distributed into multiple compartments. A loaded cargo can be moved to a different compartment at an additional cost and time. This movement provides more flexibility in picking up new cargoes. The cargo-compartment allocation plan must also take into account the ship balancing requirements, and the compartment capacities. As shown in Fig. 2, the cargoes have to be distributed within permissible limits of trim and heel.

We also consider the cargo-cargo compatibility criteria, which restricts the storage of certain chemicals in neighbouring compartments. Fig. 3 represents cargo-cargo compatibility for a set of cargoes through an illustrative graph. An edge in the graph means that the two cargoes can be stored in neighbouring compartments. Similarly, the cargo-compartment compatibility criteria restricts the cargo storage to a subset of chemical tanker compartments. The cargo-compartment compatibility can be represented by a bipartite graph, as shown in Fig. 4.

We believe that the compartment-related decisions are essential while delivering chemicals using chemical tankers. A chemical tanker can have different compartment structures, which dictate the compartment-related decisions. If the compartment-related decisions are ignored, one cannot state with certainty that the schedule will be feasible for a given chemical tanker.

We make the following assumptions in our model. These assumptions have been borrowed from Jetlund and Karimi (2004), Neo et al. (2006), Cóccola et al. (2015) and Cóccola and Méndez (2015). They are listed below.

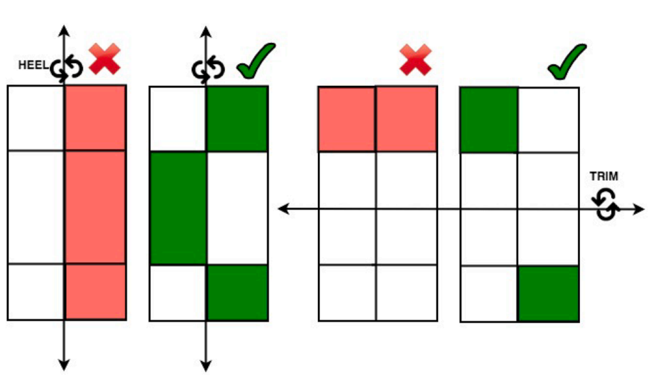

Fig. 2. Ship balancing requirements: This figure illustrates the trim and heel movements of the ship, along with the possible cargo arrangments affecting them.

- We make a simplifying assumption to fix the maximum number of port calls (sailing legs). However, even in the industry, the scheduler is required to generate a schedule for a fixed number of port calls. As such, this is a reasonable assumption.
- The chemical tanker may or may not pick-up all the unassigned cargoes at the port it visits.
- Cargoes cannot be delivered partially.
- The time for loading/unloading cargoes varies only with the total weight of the cargo.
- All port arrival and departure administrative activities are assumed to take 0.25 days.
- Four primary time-consuming activities, namely, traveling between ports, cargo loading, cargo unloading and cargo swapping are considered in our model. No two of these activities can be performed simultaneously.
- Each compartment can carry only one cargo at any given time. The cargo can be split into multiple compartments of the chemical tanker.
- Changeovers (rearranging) of loaded cargoes within the compartments of the ship are allowed. A fixed penalty cost (changeover cost) and changeover time is incurred every time an existing cargo is replaced by a different cargo within a compartment. The changeover cost and time are also incurred when an empty compartment is filled with a new cargo. We assume that cargoes can be offloaded from the chemical tankers during re-assignment of these cargoes, and then loaded again.
- Loaded cargoes can be re-assigned/swapped to compartments only at ports.
- Due to safety factors and storage norms, cargoes can only be placed in specific compartments (cargo-compartment compatibility constraints).
- Safety norms also impose certain restrictions on the placement of cargoes in neighbouring compartments, which we model as cargo-cargo compatibility constraints.
- The average speed (nm/hour) of the chemical tanker is assumed constant.
- Fuel consumption is assumed to vary linearly with the distance travelled independently of the load on the ship.
- A port can only be visited once in the planning period.

Further, with the help of an small example, we explain the problem and the importance of considering the compartment-related decisions. Fig. 5 helps us illustrate our example. At the start of the planning horizon, suppose that the chemical tanker is at Port 0 (Shanghai) and has a list of 7 ports that may be visited. We consider three different chemical tankers as shown in the bottom right corner of Fig. 5. Cargo *C5* is on board the chemical tankers at time zero. Additionally, cargoes, *C1* to *C4*, are the potential (unassigned) cargoes that are available at the ports. Attributes related to these cargoes, such as origin–destination ports, total volume, and pick-up time windows are displayed in Fig. 5. Fig. 5 also depicts the simplified port activities that have been considered in our problem.

Fig. 5 shows a sample schedule for chemical tanker 1. The cargo *C1* is picked up by the chemical tanker. At Shanghai, cargo *C5* is stored in compartments 2 and 4. At Port 1 (Hong Kong), the chemical tanker picks up cargo *C1*, which is stored in the third compartment. Finally, the voyage ends at Singapore where it delivers both the cargoes. However, due to the compartment structure and related constraints, the same schedule might become infeasible for chemical tanker 2 and 3.

Consider the second chemical tanker, which has two compartments. Each compartment has a storage capacity of 750 tonnes. If we ignore the chemical tanker stability criteria, the entire cargo might be assigned to either compartment 2 or compartment 3. This would jeopardize the safety of the chemical tanker. Consequently, the cargo *C5* is equally distributed in both the compartments of the chemical tanker 2 as shown in Fig. 5. Further on, if the cargo swapping is not allowed, the chemical

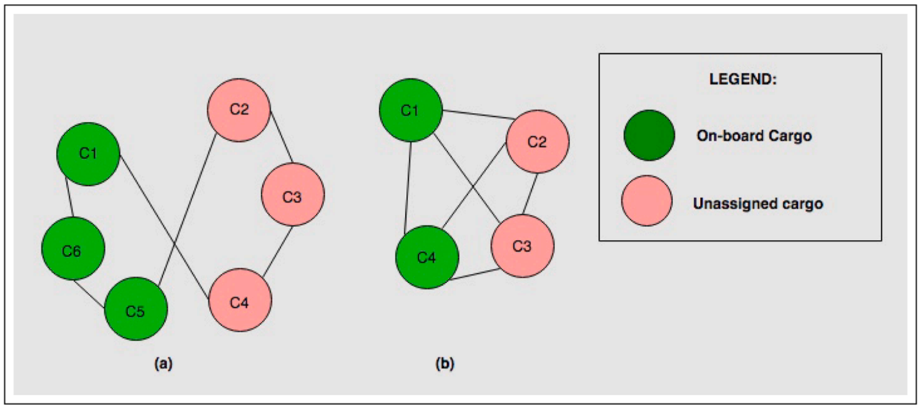

Fig. 3. Cargo-cargo compatibility graph: Instance (a) shows partial compatibility of cargoes while instance (b) shows complete compatibility of cargoes with each other.

tanker would reach port Hong Kong completely full. As a result, the cargo *C1* cannot be picked up, which makes the previous schedule infeasible for the second chemical tanker.

Let us now consider the structure of the third chemical tanker, in which the central compartment is coated with Epoxy. Observe that if both cargo *C1* and *C5* are incompatible with epoxy coated compartments, then the schedule generated for chemical tanker 1 becomes infeasible for chemical tanker 3. Finally, let us assume that cargo *C1* and *C5* are incompatible with each other. Meaning, both these cargoes cannot be stored in adjacent compartments. Then, the schedule presented in Fig. 5 becomes infeasible for chemical tanker 1. It is easy to observe that neglecting any of the above decisions might generate infeasible schedules for a chemical tanker.

We summarise our problem as follows. Given a set of ports and a set of cargoes, we try to identify the optimal schedule of the chemical tanker. An optimal schedule is one that would transport the most profitable cargoes while adhering to the various problem constraints. Our objective maximises the revenue earned by transporting unassigned (potential) cargoes and minimises the port cost, fuel cost, time chartering cost, and the changeover (cargo swapping) cost. The entire set of feasible cargoes need not be delivered. However, all the cargoes loaded on the ship are required to be delivered before the end of the planning horizon.

The primary decisions that affect the complexity of our problem are the finding of the set of ports to visit, the determining of the sequence in which these ports should be visited, the identification of the set of cargoes to transport, the assigning of the cargoes to compartments and the swapping of cargoes between compartments. The proposed problem is reducible to a *Hamiltonian path problem* by fixing all decisions except the routing of the ship. Thus, the problem is NP-hard. Our model can also be seen as a variation of *the Pickup and Delivery Problem with Time Windows (PDPTW)*, (Jetlund and Karimi, 2004), which itself is an

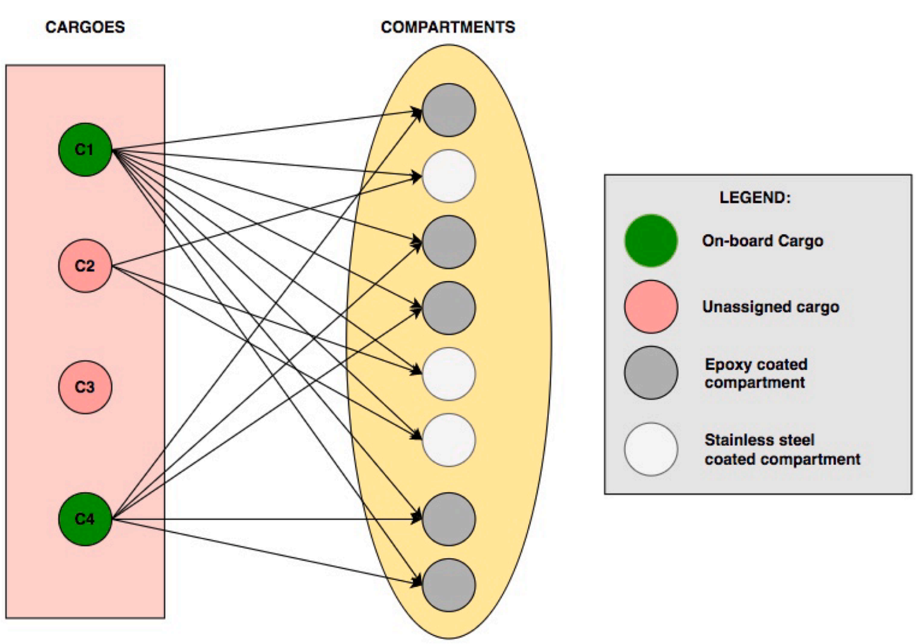

Fig. 4. Cargo-compartment compatibility graph: Direct connections between cargoes and compartment show compatibility while no connection reports incompatibility.

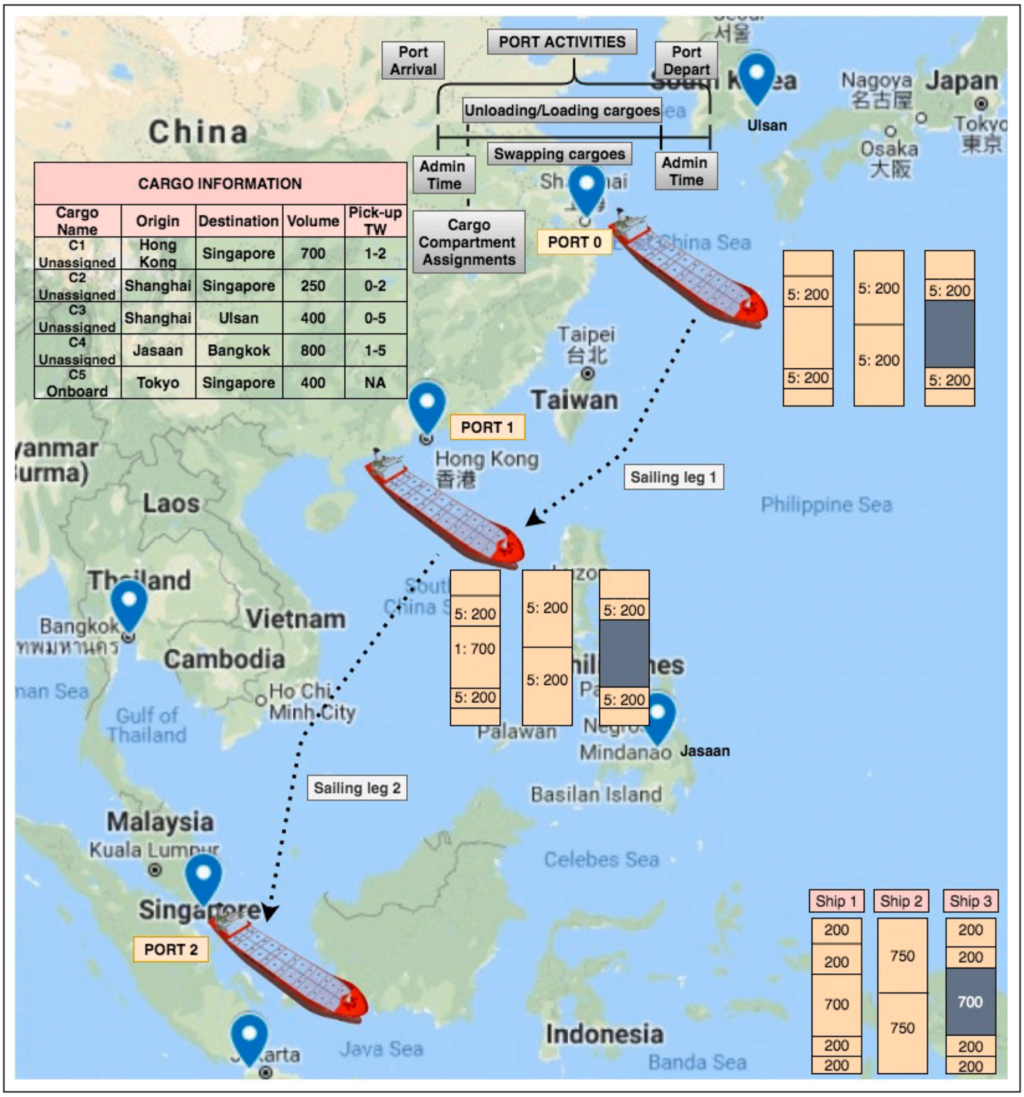

Fig. 5. Toy size illustration of chemical tanker scheduling activities.

extension of the vehicle routing problem. The tramp scheduling problem without compartment-related decisions is structurally very similar to the PDPTW, as defined by Sun et al. (2018). However, unlike the PDPTW, the s-PDP-TWTAC does not require the vehicle to return to its starting location, and all cargoes need not be served.

At every port, if the route of the chemical tanker and the cargoes to be transported are fixed, we are left with the decision of allocating cargoes to compartments. This sub-problem of cargo-compartment allocations is an extension of the generalised segregation storage problem (GSSP), which is also NP-complete (Barbucha, 2004). In the following section, we mathematically describe the various parameters and decision variables used in our formulation. We also present the mixed integer linear programming formulation used to model the problem explained in this section.

4. An MILP formulation

This section begins with a brief description of the revised (REV) formulation for the *the single ship pick-up and delivery problem with pick-up time windows, tank allocations and changeovers (s-PDP-TWTAC)*. Following this, we present a MILP for the s-PDP-TWTAC. Finally, we conclude this section by stating some key improvements proposed in our MILP model.

4.1. The revised formulation

The s-PDP-TWTAC revised (REV) formulation is described as follows. Let K be the set of indices of the sailing legs, and N^P be the set of feasible ports. The set of cargoes (N^G) divided into on-board cargoes (N^O) and unassigned cargoes (N^U). N^O are the cargoes that are on-board the chemical tanker at the beginning of the planning horizon. Unassigned cargoes (N^U) are the cargoes that can be potentially picked up to maximise the profit. The set N^H is the set of chemical tanker compartments.

Any cargo loaded on the ship has to be delivered. An unassigned cargo $j \in N^U$ can only be picked up within a specified time-window [T_j^E, T_j^L]. A cargo $j \in N^G$ is defined by characteristics like revenue obtained (R_j), origin (P_j^L), destination (P_j^D), weight (W_j) and density (ρ_j). A compartment $h \in N^H$ is defined by compartment volume (V_h), and lateral (κ_h) and longitudinal (ι_h) distance from the centre of the ship.

The set N_h^B defines the structure of the chemical tanker by listing the bordering compartments for every $h \in N^H$. The sets N_j^I and N_h^X helps us define the cargo-cargo incompatibility and the cargo-compartment incompatibility, respectively. The set N_j^I lists all the cargoes that cannot be stored beside the cargo $j \in N^G$. The set N_h^X includes cargoes that cannot be stored in compartment $h \in N^H$.

Given the starting port (P^I) and the set N^P, our model tries to find an optimal by maximising the difference in the revenue (R_j) and the four different costs; namely, the port cost (C_p^P), the fuel cost ($C_{pp'}^F$), the time chartered cost (C^T) and the changeover cost (C^S). The route of the ship is defined using decision variables like port arrival time (t_k) and routing variable ($z_{kpp'}$). The routing variable equals one if and only if the chemical tanker travels between ports $p, p' \in N^P$ at the end of sailing leg $k \in K$. If the number of profitable port calls are less than the maximum number of sailing legs (—K—), then the chemical tanker is forced to enter a dummy port. Once the chemical tanker enters the dummy port, it stays there till the end of its voyage. The sequence in which the cargoes are serviced are modelled using variables l_{kj} and u_{kj}. The variables l_{kj} and u_{kj} record the sailing leg $k \in K$ at the end of which a cargo is picked up and dropped off, respectively.

The decision variable c_{kjh} equals 1 if cargo $j \in N^G$ is stored in compartment $h \in N^H$ at the end of sailing leg $k \in K$. Moreover, if a cargo $j \in N^G$ is stored in compartment $h \in N^H$ then the variable w_{kjh} gives the cargo weight stored in the compartment. Further, we keep track of the total changeovers by defining variables b_{kjh} and r_{kjh}. We formally define all the sets, decision variables and parameters in Section 4.

Sets:

$$
\begin{aligned}
K &= \text{Set of indices of sailing legs}, \{0, \ldots, |\mathrm{K}|\},\\
N^P &= \text{Set of ports},\\
N^G &= \text{Set of all cargoes/goods. Includes cargo 0, a dummy cargo for modelling},\\
N^O &= \text{Set of cargoes already on} - \text{board the chemical tanker at time zero}, N^O \subset N^G,\\
N^U &= \text{Set of potential cargoes that can be picked up}, N^U \subset N^G,\\
N_j^I &= \text{Set of cargoes incompatible with cargo} j \in N^G, N_j^I \subset N^G,\\
N^H &= \text{Set of compartments (cargo holds)in the ship},\\
N_h^B &= \text{Set of neighbouring/bordering compartments for compartment} h \in N^H, N_h^B \subset N^H,\\
N_h^X &= \text{Set of cargoes that cannot be stored in compartment} h \in N^H, N_h^X \subset N^G.
\end{aligned}
$$

Indices:

$$
\begin{aligned}
p, p', p'\prime &= \text{Index for port},\\
k, k' &= \text{Index for sailing leg (Index 0 indicates that the chemical tanker is at its starting port)},\\
j &= \text{Index for cargo}(j = 0 \text{signifies dummy cargo with no weight and no incompatibilities}),\\
h, h' &= \text{Index for compartment (cargo hold)}.
\end{aligned}
$$

Revised decision variables:

$$
\begin{aligned}
t_k &= \text{Port arrival time of the chemical tanker at the destination of leg k} \in K(\text{Continuous}),\\
z_{kpp'} &= \text{1 if chemical tanker at the end of leg k} \in \mathrm{K}\backslash\{0\} \text{departed from port p} \in N^P \text{ and}\\
&\quad \text{arrived at p}' \in N^P(\text{Binary}),\\
l_{kj} &= \text{1 if the chemical tanker at the end of leg k} \in K \text{loads cargo j} \in N^U(\text{Binary}),\\
u_{kj} &= \text{1 if the chemical tanker at the end of leg k} \in K \text{unloads cargo j} \in N^U(\text{Binary}),\\
c_{kjh} &= \text{1 if the chemical tanker at the end of leg k} \in K \text{carries cargo j} \in N^G \textit{in} \text{ compartment}\\
&\quad \text{h} \in N^H(\text{Binary}),\\
w_{kjh} &= \text{Weight of cargo j} \in N^G \text{ assigned to compartment h} \in N^H \text{ of chemical tanker at end}\\
&\text{of leg k} \in K(\text{Continuous}),\\
b_{kjh} &= \text{1ifthechemicaltankerattheendoflegk} \in \text{Kreplacesanycargoj} \in N^G(\text{otherthanitself})\\
&\quad \text{with cargo j} \in N^G\backslash\{0\} \text{ in compartment h} \in N^H(\text{Binary}),\\
r_{kjh} &= \text{1 if chemical tanker at end of leg k} \in \mathrm{K}\backslash\{0\} \text{ removes cargo j} \in N^G\backslash\{0\} \text{ in compartment}\\
&\quad \text{h} \in N^H(\text{Binary}).
\end{aligned}
$$

Parameters:

$P^I =$ Starting port of the ship, $P^I \in N^P$,
$P_j^L =$ Loading port for cargo $j \in N^U, P_j^L \in N^P$,
$P_j^D =$ Discharge port for cargo $j \in N^G, P_j^D \in N^P$,
$|N^P| =$ Dummy Port, $|N^P| \in N^P$,
$R_j =$ Revenue that can be obtained if cargo $j \in N^G$ is transported by the chemical tanker,
$C_{pp'}^F =$ Cost for travelling between ports $p \in N^P and p' \in N^P$,
$C_p^P =$ Port cost incurred on visiting port $p \in N^P$,
$C^T =$ Cost of time charter of the chemical tanker per day,
$C^S =$ Cost per changeover/swap including cleaning, labour, etc. related to swapping cargoes within compartments. Also represents the cost incurred if a cargo $j \in N^G \backslash \{0\}$ is filled in an empty compartment,
$W_j =$ Weight of the cargo $j \in N^G$,
$\rho_j =$ Density of the cargo $j \in N^G$,
$V_h =$ Volume of compartment $h \in N^H$,
$T_j^E =$ Earliest pick-up time for cargo $j \in N^U$,
$T_j^L =$ Latest pick-up time for cargo $j \in N^U$,
$T_j^P =$ Time required to pick-up cargo $j \in N^U$,
$T_j^D =$ Time required to discharge cargo $j \in N^G$,
$T_{pp'}^T =$ Travel time between port $p \in N^P$ and $p' \in N^P$,
$T^S =$ Time per changeover/swap, which includes the time taken to clean the compartment and swap the cargoes within compartments. Also represents the cleaning time elapsed if a cargo $j \in N^G \backslash \{0\}$ is filled in an empty compartment,
$T_1^A =$ Waiting time for berth allocation, security clearances, immigration procedure,
$T_2^A =$ Time delays incurred due to repairs, bunkering, and customs inspections,
$T^A =$ Total administrative time, $T^A = T_1^A + T_2^A$,
$\kappa_h =$ Lateral distance from compartment $h \in N^H$ to the centre of the chemical tanker,
$\iota_h =$ Longitudinal distance from compartment $h \in N^H$ to the centre of the chemical tanker,
$\alpha =$ Maximum absolute permissible trim causing moment of the chemical tanker,
$\beta =$ Maximum absolute permissible heel causing moment of the chemical tanker,
$DC =$ Draft constant. The total allowable draft(in tonnes)for the chemical tanker (tonnes),
$M =$ A suitably large number for modelling binary decisions.

The objective function of our formulation is as follows:

$$\textbf{Maximise} \quad \sum_{j \in N^U}\left(R_j W_j \sum_{k \in K \backslash \{|K|\}} l_{kj}\right) - \sum_{p \in N^P}\sum_{p' \in N^P}\left(C_{pp'}^F \sum_{k \in K \backslash \{0\}} z_{kpp'}\right) - \left(C^T\left(t_{|K|} + T^A\left(1 - \sum_{p \in N^P} z_{|K|p|N^P|}\right) + \sum_{p \in N^P}\sum_{j \in N^O}\left(T_j^D z_{|K|pP_j^D}\right) + \sum_{j \in N^U}\left(T_j^D u_{|K|j}\right)\right)\right)$$
$$- \sum_{p' \in N^P}\left(C_{p'}^P \sum_{k \in K \backslash \{0\}}\sum_{p \in N^P} z_{kpp'}\right) - \left(C^S \sum_{k \in K}\sum_{j \in N^G \backslash \{0\}}\sum_{h \in N^H} b_{kjh}\right) + \sum_{j \in N^O} R_j W_j - C_{P^I}^P \tag{1}$$

The first term calculates the total revenues generated by picking up a subset of unassigned cargoes. The second term calculates the fuel cost, which is a function of the route of the ship. The next term including C^T, calculates the total cost of chartering the chemical tanker. C^T is affected by all the temporal actions that the chemical tanker performs. Thus, total C^T is calculated by combining the arrival time at the last port ($t_{|K|}$), with the temporal port activities performed at the last port (T^A and the total unloading time of all the cargoes discharged at the last port)). The total time spent at the port is zero if the last port visited is a dummy port. The succeeding term, calculates the total port cost C^P, which is incurred for every port visited by the chemical tanker. Moreover, the changeover cost is calculated by summing up the total number of changeovers (b_{kjh}).

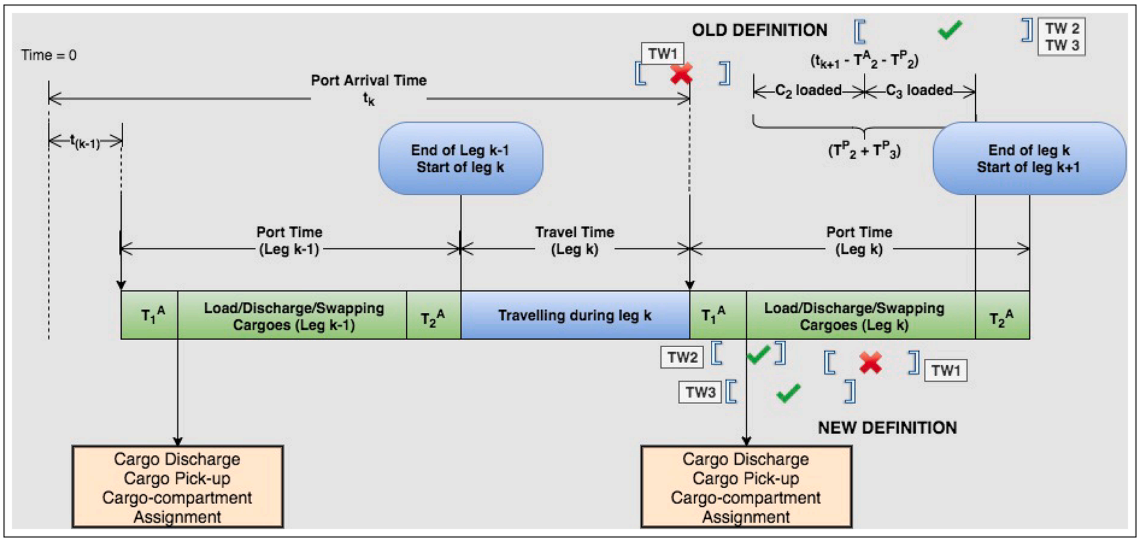

Fig. 6. The figure represents all the chemical tanker activities and the new definition of pick-up time windows.

Finally, the last two terms of the Eq. 1 calculate the revenue obtained from the onboard cargoes, and port cost (C^P) related to visiting the immediate destination. As a result, the objective function (1) tries to increase the revenue earned by servicing the cargoes. Simultaneously, the objective function tries to reduce the travel cost, time chartered cost, port cost and changeover cost.

$$\sum_{p\in N^P} z_{kpp'} = \sum_{p''\in N^P} z_{(k+1)p'p''} \forall k \in K \backslash \left\{0, |K|\right\}, p' \in N^P, \tag{2}$$

$$\sum_{k\in K\backslash\{0\}} \sum_{p'\in N^P} z_{kpp'} \leqslant 1 \forall p \in N^P \backslash \left\{|N^P|\right\}, \tag{3}$$

$$\sum_{k\in K\backslash\{0\}} \sum_{p\in N^P} z_{kpp'} \leqslant 1 \forall p' \in N^P \backslash \left\{|N^P|\right\}, \tag{4}$$

$$\sum_{k\in K\backslash\{0\}} \sum_{p\in N^P} z_{kp,P^D_j} = 1 \forall j \in N^O \backslash \left\{P^D_j = P^I\right\}, \tag{5}$$

Constraints (2) to (5) define the path of the ship. Constraint (2) ensures that the ship must leave every port it visits, except the last one. However, if the chemical tanker enters the dummy port $|N^P|$ it has to stay there for rest of the voyage. We enforce this during pre-processing by fixing all the routing variables ($z_{k|N^P|p'}, p' \in P\backslash\{|N^P|\}$) to zero. These routing variables correspond to all the arcs originating from dummy port to all other ports. Constraints (3) and (4) together enforce the assumption that a chemical tanker can visit any port at most once. Constraint (5) imposes the condition that discharge ports of each on-board cargo must be visited. Next we formulate constraints related to the pick-up and delivery of cargoes.

$$l_{(k-1)j} \leqslant \sum_{p\in N^P} z_{kP^L_j p} \forall k \in K \backslash \left\{0\right\}, j \in N^U, \tag{6}$$

$$u_{kj} \leqslant \sum_{p\in N^P} z_{kpP^D_j} \forall k \in K \backslash \left\{0\right\}, j \in N^U, \tag{7}$$

$$l_{kj} \leqslant \sum_{k'\in K\backslash\{k'\leqslant k+1\}} u_{k'j} \forall k \in K \backslash \left\{|K|\right\}, j \in N^U, \tag{8}$$

$$\sum_{k\in K} u_{kj} = \sum_{k\in K} l_{kj} \forall j \in N^U, \tag{9}$$

$$u_{kj} \geqslant -1 + \sum_{k'\in K} l_{k'j} + \sum_{p\in N^P} z_{kpP^D_j} \forall k \in K \backslash \left\{0\right\}, j \in N^U, \tag{10}$$

Constraints (6) and (7) ensure that the unassigned cargoes are picked up and dropped off at their corresponding loading and unloading ports. Constraint (8) states that an unassigned cargo can be dropped off only after pick-up. Constraints (9) and (10) ensure that an unassigned cargo can be discharged at the end of leg k if and only if it was picked up and its discharge point is visited at the end of leg k. Next, we model the constraints that deal with the temporal activities of the ship.

$$t_k \geqslant \left(T^E_j - T^A_1\right) l_{kj} \qquad \forall k \in K \backslash \left\{|K|\right\}, j \in N^U, \tag{11}$$

$$t_k \leqslant \left(T^L_j - T^A_1\right) l_{kj} + M\left(1 - l_{kj}\right) \qquad \forall k \in K \backslash \left\{|K|\right\}, j \in N^U, \tag{12}$$

$$\begin{aligned} t_{(k+1)} \geqslant{} & t_k + T^A\left(1 - \sum_{p\in N^P} z_{(k+1)|N^P|p}\right) + \sum_{j\in N^O}\left(T^D_j \sum_{p\in N^P} z_{(k+1)P^D_j p}\right) + \sum_{j\in N^U}\left(T^P_j l_{kj}\right) \\ & + \sum_{j\in N^U}\left(T^D_j u_{kj}\right) + T^S \sum_{j\in N^G\backslash\{0\}} \sum_{h\in N^H} b_{kjh} + \sum_{p\in N^P}\sum_{p'\in N^P}\left(T^T_{pp'} z_{(k+1)pp'}\right) \quad \forall k \in K \backslash \left\{|K|\right\}, \end{aligned} \tag{13}$$

Constraint (11) enforces the condition that if a cargo is picked up then the cargo-assignment time ($t_k + T^A_1$) should be greater than the earliest pick-up time (T^E_j). Similarly, Constraint (12) states that if a cargo is picked up then the cargo-assignment time ($t_k + T^A_1$) should be less than the latest pick-up time (T^L_j). Constraint (13) makes sure that the port arrival time (t_{k+1}) during the sailing leg ($k+1$) is greater than the addition of the port arrival time (t_k) during leg k, the administrative time (T^A) during leg k (if the present port is not the dummy port), all the loading and unloading times for the cargoes picked up and dropped off during leg k, the cargo swapping time (T^S) elapsed during leg k, and the travel time during leg (k + 1). The constraints for allocating cargoes to compartments are as follows:

$$\sum_{h\in N^H} c_{0jh} \geqslant 1 \forall j \in N^O \backslash \left\{ P_j^D = P^I \right\}, \tag{14}$$

$$\sum_{h\in N^H} c_{0jh} \geqslant l_{0j} \forall j \in N^U, \tag{15}$$

$$\sum_{h\in N^H} c_{kjh} \geqslant \sum_{h\in N^H} \frac{c_{(k-1)jh}}{|N^H|} - \sum_{p\in N^P} z_{kpP_j^D} \forall k \in K \backslash \left\{0\right\}, j \in N^O \backslash \left\{ P_j^D = P^I \right\}, \tag{16}$$

$$\sum_{h\in N^H} c_{kjh} \geqslant \sum_{h\in N^H} \frac{c_{(k-1)jh}}{|N^H|} + l_{kj} - u_{kj} \forall k \in K \backslash \left\{0, |K|\right\}, j \in N^U, \tag{17}$$

$$\sum_{h\in N^H} c_{kjh} \leqslant |N^H| \sum_{k'\in K\backslash\{k'>k\}} l_{k'j} \forall k \in K, j \in N^U, \tag{18}$$

$$\sum_{k'\in K\backslash\{k'<k\}} \sum_{h\in N^H} c_{k'jh} \leqslant |K||N^H| \left(1 - \sum_{p\in N^P} z_{kpP_j^D}\right) \forall k \in K \backslash \left\{0\right\}, j \in N^G \backslash \left\{0\right\}, \tag{19}$$

Constraints (14) and (15) enforce the cargo allocations during leg 0. Constraint (14) makes sure that all the on-board cargoes are assigned to at least one compartment unless they are delivered at the immediate destination. In the same way, Constraint (15) makes sure that if an unassigned cargo is picked up during leg 0 then it is assigned to at least one compartment. The Constraints (16) and (17) together maintain the continuity of unassigned and on-board cargoes respectively. Constraints (16) and (17) are trivially satisfied if the cargoes are dropped off during the present leg. However, if the on-board cargoes remains loaded on the ship during the present leg, then the Constraint (16) makes sure that the cargo is assigned to at least one compartment. Similarly, Constraint (17) makes sure that the unassigned cargoes are assigned to at least one compartment till they are on the ship. Finally, Constraints (18) and (19) enforce the fact that cargoes cannot be assigned to compartments before they are picked up and after they are dropped off. Some additional cargo assignment constraints are as follows:

$$\sum_{j\in N^G} c_{kjh} = 1 \forall k \in K, h \in N^H, \tag{20}$$

$$c_{(k-1)jh} + b_{kjh} = c_{kjh} + r_{kjh} \forall k \in K \backslash \{0\}, j \in N^G \backslash \{0\}, h \in N^H, \tag{21}$$

$$b_{0jh} = c_{0jh} j \in N^G \backslash \{0\}, \forall h \in N^H, \tag{22}$$

$$c_{kjh} + \sum_{j'\in N_j^I} c_{kj'h'} \leqslant 1 \qquad \forall k \in K \backslash \left\{|K|\right\}, j \in N^G \backslash \left\{0\right\}, h \in N^H, h' \in N_h^B, \tag{23}$$

Constraint (20) makes sure that either the compartment is empty or it has exactly one cargo in it. Constraint (21) help us keep track of changeovers (cargo swapping) within every compartment during consecutive legs. Recall that if a compartment is empty we assume it has cargo 0. Constraint (21) and a negative objective function co-efficient makes sure that the variable b_{kjh} equals 1 if a compartment is filled with different cargoes in succeeding sailing legs, or an empty compartment is filled with new cargo. Moreover, the variable r_{kjh} ensures that the variable b_{kjh} takes on value 0, when $c_{(k-1)jh}$ equals 1 and c_{kjh} equals 0. Constraint (22) is a special case of Constraint (21). It tracks the number of changeovers at the end of sailing leg 0, which is equal to the number of filled compartments of the chemical tanker. Constraint (23) imposes the cargo-cargo compatibility criteria. The following constraints implement the cargo weight per compartment related restrictions.

$$w_{kjh} \leqslant V_h \rho_j c_{kjh} \forall k \in K \backslash \{|K|\}, j \in N^G \backslash \{0\}, h \in N^H, \tag{24}$$

$$\sum_{h\in N^H} w_{kjh} = W_j \sum_{k'\in K\backslash\{k'>k\}} \left(l_{k'j} - u_{k'j}\right) \forall k \in K \backslash \left\{|K|\right\}, j \in N^U, \tag{25}$$

$$\sum_{h\in N^H} w_{kjh} = W_j \left(1 - \sum_{k'\in K\backslash\{0,k'>k+1\}} \sum_{p\in N^P} z_{k'P_j^D p}\right) \forall k \in K \backslash \left\{|K|\right\}, j \in N^O, \tag{26}$$

$$\sum_{h\in N^H} \sum_{j\in N^G\backslash\{0\}} w_{kjh} \leqslant DC \forall k \in K, \tag{27}$$

$$-\alpha \leqslant \sum_{h\in N^H} \sum_{j\in N^G\backslash\{0\}} w_{kjh} \iota_h \leqslant \alpha \forall k \in K, \tag{28}$$

$$-\beta \leqslant \sum_{h\in N^H} \sum_{j\in N^G\backslash\{0\}} w_{kjh} \kappa_h \leqslant \beta \forall k \in K, . \tag{29}$$

Constraint (24) ensures that the weight of the cargo assigned to the compartment can be at most equal to the maximum capacity of the compartment. Constraint (25) makes sure that the total weight of the unassigned cargo distributed in various compartments is equal the total weight of that cargo between pick-up and delivery. Constraint (26) forces the same condition on the on-board cargoes. Constraint (27) makes sure that the total weight allocated to the chemical tanker is less than the draft constant. Constraint (28) and (29) are ensure that the maximum allowable trim and heel moments are not exceeded.

4.2. Key differences between s-PDP-TWTAC formulations: existing vs. revised

We now describe the key advantages of our model over other existing ones. First, several decision variables defined by Jetlund and Karimi, 2004 and Neo et al., 2006 have been eliminated/fixed. Second, we capture the changeover (cargo swapping) activities in a new way, which reduces the complexity of the problem. Third, we propose a different approximation of the pick-up time windows for better modelling. Thus, our model is more realistic and at the same time, more tractable than the earlier ones. Next, we describe the improvements implemented by us.

4.2.1. Eliminating/fixing of decision variables from the existing model

A MILP solver performs advanced pre-processing automatically. However, they only look at the mathematical formulation, and have no knowledge about the application and the model. As a result, the solver sometimes can not do model level or application specific reformulations. We also see a substantial improvement in the running time with the proposed reformulations which shows that the solvers are unable to discover and deploy the proposed reformulation techniques. These modifications, even-though elementary, can be overlooked by the reader. Please refer to Section 4 for all the definitions. We eliminate the following decisions from the model.

1. The port 0 (immediate destination) of the ship is given. Therefore, during leg 1, we eliminate all arcs not originating from port 0.
- $z_{1pp'} = 0 \ \forall p \in P \backslash \{P^I\}, p' \in P,$

2. If immediate destination of the ship is equal to the loading port of certain cargoes, then the cargo can only be picked up at the end of leg 0. Consequently, the cargo pick-up variable for these cargoes is eliminated for legs greater than 0.
- $l_{kj} = 0 \quad \forall k \in K \backslash \{0\}, j \in N^U \backslash \{j \big| P^I \neq P_j^L\},$

3. The cargo-compartment incompatibility states that incompatible cargoes cannot be stored within certain compartments. This restric-

tion can easily be enforced by eliminating following variables from the model.

- $c_{kjh}, w_{kjh}, b_{kjh}, r_{kjh} = 0 \ \forall k \in K, j \in N_h^X, h \in N^H$,

4. At the end of the planning horizon all the cargoes need to be delivered. Thus, the following decisions can be fixed to zero.
- $l_{|K|j}, u_{|K|j} = 0 \quad \forall j \in N^U$,
- $c_{|K|jh}, w_{|K|jh}, b_{|K|jh}, r_{|K|jh} = 0 \quad \forall j \in N^G, h \in N^H$,

5. Eliminate all cargo-related decision variables if immediate destination of the ship is equal to the discharge port of these cargoes.
- $l_{kj} = 0 \quad \forall k \in K, j \in N^U \backslash \{j \middle| P^I \neq P_j^D\}$,
- $u_{kj} = 0 \quad \forall k \in K \backslash \{0\}, j \in N^U \backslash \{j \middle| P^I \neq P_j^D\}$,
- $c_{kjh}, w_{kjh}, b_{kjh}, r_{kjh} = 0 \quad \forall k \in K, j \in N^G \backslash \{0, j \middle| P^I \neq P_j^D\}, h \in N^H$,

4.2.2. Remodelling of the changeover decision variables

The formulation presented by Neo et al. (2006) captures the changeover activity using a four indexed decision variable. They define a changeover variable $m_{kjj'h} = 1$ if at the end of leg $k \in K$ cargo $j \in N^G$ is replaced with cargo $j' \in N^G$ in compartment $h \in N^H$.

In contrast, we model the changeover activity using three indexed variables b_{kjh}. The two indexed variables are an extension of the on/off variable idea that is presented in Schwindt et al., 2015. As a result, the changeover activity can be captured by a significantly reduced number of variables. Moreover, empirical tests (Section 7) indicate that our formulation yields tighter linear relaxations than the existing formulation presented by Neo et al. (2006).

4.2.3. Generalising the definition of the pick-up time windows

The pick-up time windows, as defined by Jetlund and Karimi (2004), Neo et al. (2006) and Cóccola et al. (2015), had some practical limitations. Their definition stated that if a cargo is being picked up, its latest pick-up time should be greater than the port arrival time plus half of the administrative time. Additionally, the definition also stated that the earliest pick-up time should be less than the port departure time minus half of the administrative time and loading time of that cargo.

Jetlund and Karimi (2004), Neo et al. (2006) and Cóccola et al. (2015) present the following constraints for the pick-up time windows:

$$t_{k+1} \geqslant \left(T_j^E + T_2^A + T_j^P\right) l_{kj} + \sum_{p \in P} \sum_{p' \in P} T_{pp'}^T z_{kpp'} \quad \forall k \in K \backslash \{|K|\}, j \in N^U,$$

$$t_k \leqslant \left(T_j^L - T_1^A\right) l_{kj} + M\left(1 - l_{kj}\right) \quad \forall k \in K \backslash \{|K|\}, j \in N^U.$$

We elaborate the need for our approximation with a small example and Fig. 6. Assume that we have three cargoes. Let cargo 2 and cargo 3 have the same pick-up time-windows. The loading times for cargo 2 and cargo 3 are T_2^P and T_3^P. If the existing definition of time windows is considered, then both cargo 2 and cargo 3 can be picked up. Further, the assumption that the cargoes are loaded consecutively would result in either one of the cargoes extending outside the pick-up time window. Such a situation might frequently occur in practical instances. As a result, we re-define the pick-up time-windows. According to the revised definition, cargo can be picked up only if the cargo-assignment point (Fig. 5) lies within the pick-up time window. Fig. 6 shows a comparison of both the definitions. This approximation captures more generalised real-world instances.

Even-though the model presented in Section 4 is cleaner and smaller than the existing models, it is still difficult to solve even for medium-sized test instances. In order to find good feasible solutions faster, we propose a heuristic in the following section.

5. A neighbourhood search based heuristic

We propose a preliminary construction heuristic to find a good feasible solution. Our heuristic first solves a linear programming (LP) relaxation of the MILP. Solving an LP is usually much faster than MILP. If the LP relaxation is infeasible, MILP is also infeasible. Otherwise, we fix a large number of variables and solve a much smaller MILP. The heuristic presented in this section is a modification of the *Relaxation Enforced Neighbourhood Search* (RENS) heuristic introduced by Berthold (2007), and the *Relax and Fix* (RaF) heuristic implemented by Rodrigues et al. (2016) and Santos et al. (2020).

According to Santos et al. (2020), the RaF heuristic is effective on problems that can be divided into n sets of integer variables. For example, Rodrigues et al. (2016) define the sets of variables (to relax) based on time intervals. On the other hand, Santos et al. (2020) divide the variables based on heirarchy of decisions like routing variables, cargo assignment variables and so on. The n sets of variables are disjoint sets. In every iteration, the RaF heuristic solves the MIP formulation by relaxing a set of integer variables. Solutions generated in the previous iterations are provided as initial solution to the solver. According to Santos et al. (2020), some constraints are also relaxed for every iteration to reduce the problem complexity. However, as constraints are relaxed a heuristic is required to repair any infeasible solution that is generated on solving the sub-problems.

There are some similarities between the RaF heuristic implementations and our heuristic because all of them solve a relaxation to provide insight for the overall problem. Additionally, similar to Santos et al. (2020), we divide the decision variables into different sets based on a heirarchy of decisions. Unlike Rodrigues et al. (2016) and Santos et al. (2020) we solve the linear programming (LP) relaxation, which relaxes all of the integer variables and includes all problem constraints. Additionally, instead of relaxing, we fix a subset of integer variables to reduce the complexity of the MIP formulation.

Berthold (2007) also implement a heuristic which uses the LP relaxation to reduce the problem complexity of the MIP formulation. Based on the LP solution, the MIP is solved over this restricted feasible region to generate a local optimum. Consequently, we introduce an adaption of the RENS heuristic, which solves the revised formulation over a restricted feasible region. The feasible region is restricted by updating the bounds of a subset of integer variables based on the LP solution.

Our bound update rule is derived from the structural analysis of the problem. Specifically, the update rule eliminates ports (except the dummy port ($|N^P|$)) that are not visited by the chemical tanker in the LP optimal solution. Let $(\bar{l}, \bar{u}, \bar{z}, \bar{c}, \bar{w}, \bar{b}, \bar{r})$ be the LP relaxation of the revised formulation presented in Section 4. Let, $U_{kpp'} = 1$ be the upper bound on $z_{kpp'}$. Mathematically, the upper bounds are updated as follows:

Bound update Rule:

$$\textit{If} \sum_{k' \in K \backslash \{0\}} \sum_{p' \in N^P \backslash \{|N^P|\}} \left(\bar{z}_{k'pp'} + \bar{z}_{k'p'p}\right) = 0,$$

$$\textit{Then} \sum_{k' \in K \backslash \{0\}} \sum_{p' \in N^P \backslash \{|N^P|\}} \left(U_{k'pp'} + U_{k'p'p}\right) = 0 \quad \forall p \in N^P \backslash \left\{\left|N^P\right|\right\}.$$

Constraint (5) ensures that any feasible solution of the LP relaxation will always contain discharge ports of the onboard cargoes. As a result, the heuristic will terminate with at least one feasible solution if the optimal solution of the LP relaxation is found.

Multiple reasons make this heuristic a viable option for our problem. The primary reason being that the formulation we propose in this paper has a tighter linear relaxation than the previous formulation presented in the literature. Since our LP relaxation is closer to the convex hull of MILP feasible region, its neighbourhood should provide a reasonable starting solution.

A unique feature of our heuristic is the bound update rule that is based on the problem structure. As part of the structural analysis, we tried to fix various groups of decision variables. Once certain groups of variables were fixed, we analysed their effect on parameters like the total solution time, the number of nodes explored, and the initial relative gap. We carried out certain experiments that fixed the ports to visit (not the order in which these ports should be visited), or the entire ship route was fixed, or the cargoes to be served were fixed. Restricting other decision did not lead to a substantially simplified MILP. Out of the three decision sets, fixing either the cargoes or the ship's route made the ship's moment extremely restrictive in the temporal plane. Additionally, we observed that a significant number of route defining constraints (Constraints (2)–(7)) were active in the optimal basis of the linear relaxation in all of the benchmark instances.

Furthermore, we observed that restricting the feasible set of ports (not the sequence in which these ports should be visited) significantly reduced the MILP termination time. Additionally, letting MILP decide the sequence of ports increased the feasible region of the problem substantially when compared to fixing the exact route of the ship. The flowchart for the heuristic is presented in Fig. 7.

We postpone our discussion on the performance of this heuristic until Section 7. We next describe our instance generator, which was used to generate the instances used for testing.

6. Instance generator

Benchmark data sets in maritime transportation research are scarce. Only a few researchers have presented reusable benchmark datasets. Brouer et al. (2011) present a benchmark dataset for liner shipping network design models. Their dataset is composed of data from the liner company, Maersk Line. Similarly, Papageorgiou et al. (2014) and Hemmati et al. (2014) present an extensive list of real-world benchmark data for maritime inventory routing problems and tramp scheduling problems. Hemmati et al. (2014) develop their data to represent various shipping segments based on factors like the deep sea or short sea and full-load or mixed-load problems. However, certain limitations restrict the use of their data to our problem. For example, they do not provide data related to the operational facets of our problem, such as the volume of compartments, compartment materials, compartment dimensions, cargo-cargo, and cargo-compartment compatibility. In order to overcome this limitation, we introduce an instance generator that is based on real-world data and parameters. The instance generator code and instances are publicly available online[2].

Our instance generator is built in the R programming language. It has three main components, the core data folder, the instance generation engine, and the input parameter file. The core data contains static data used by the instance generator to create the final problem-specific instances. The instance generator engine is the actual code responsible for producing problem instances by processing the core data based on the specifications from the user. Finally, the input parameter file allows the user to select different parameter settings for the instances being produced. Fig. 8 outlines the structure of our instance generator.

6.1. Instance format

A single instance generated by the instance generator consists of four files; namely, the ship data file, the onboard cargo data file, the unassigned cargo data file and the problem data file. The ship data file consists of all the ship-related data like ship number, ship name, ship structure, port cost, time-chartered cost, and so on. The onboard cargo data file and the unassigned cargo data file consists of cargo data like cargo number, cargo weight, origin, destination, cargo-cargo compatibility restrictions, etc. Finally, the problem data file consists of miscellaneous problem-related data like the total number of ports, port names, port distances and administrative time. Fig. 9 presents a complete list of data included in a single instance.

[2] https://ladageanurag.shinyapps.io/s-PDP-TWTAC/

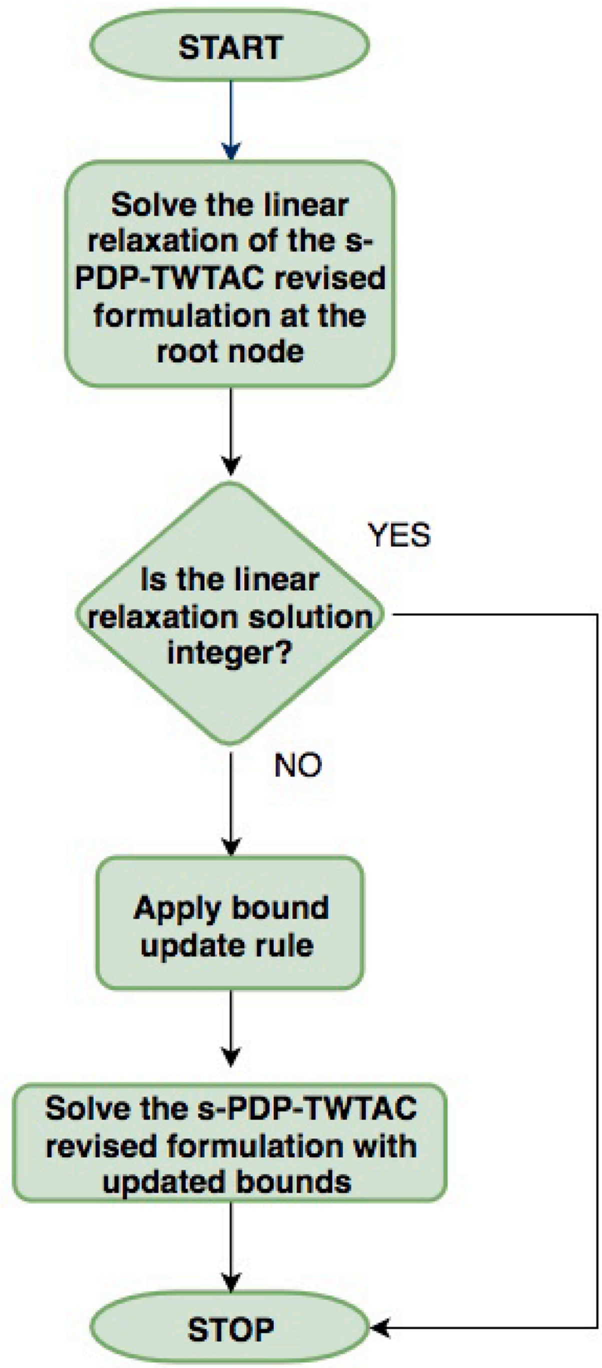

Fig. 7. Flowchart for the neighbourhood search based heuristic.

6.2. Core data for generator

The generator relies on the core data to create instances. The core data includes the data collected by us and can be enhanced by the user. The core data consists of 38 structurally different ships and four networks of ports. The designs of the chemical tankers are largely based on Odfjell's chemical tanker fleet[3]. The number of compartments on the chemical tanker ranges from 16 to 52. The compartment walls are made of stainless steel, zinc or epoxy. Network data consists of nautical

[3] https://www.odfjell.com/tankers/our-fleet/

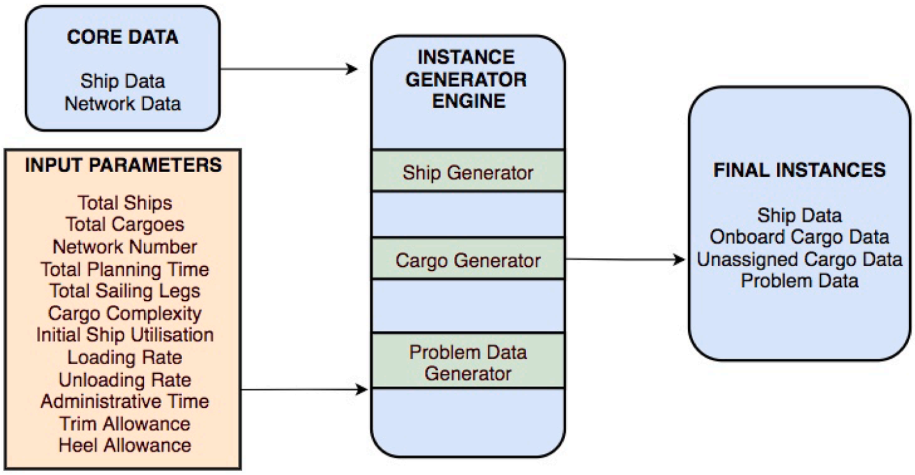

Fig. 8. Principle components of the instance generator.

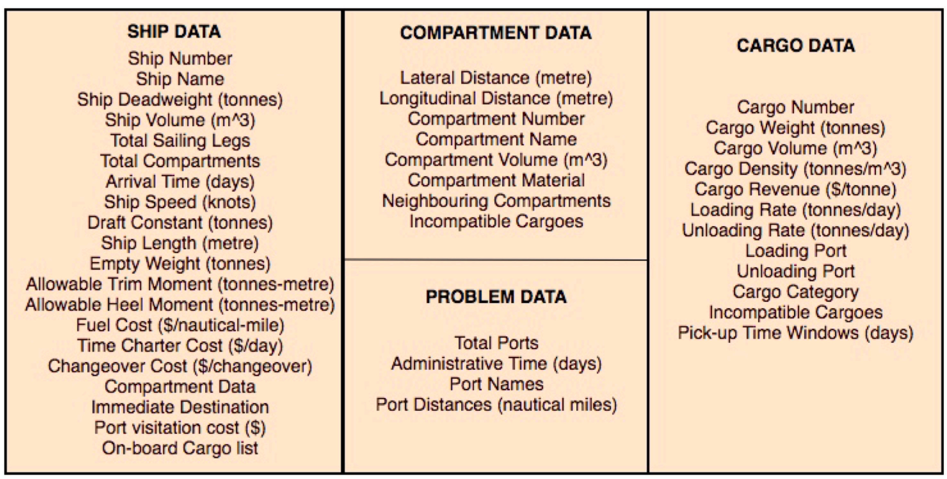

Fig. 9. Complete list of data generated for a single instance.

distances between ports[4]. Network 1 is borrowed from Jetlund and Karimi, 2004. Network 2 consists of the 98 busiest ports of 2015 as specified by American Association of Port Authorities[5]. Network 3 consists of the top 47 busy ports in the Asia Region, as published by the International Association of Ports and Harbour[6]. Finally, Network 4 consists of the busiest ports in the year 2016 for the NAFTA region[7]. The NAFTA region consists of ports in the USA, Mexico and Canada. A user can add additional chemical tankers and network-related data to this core data. The instance generator engine reads this core data and generates instances.

6.3. Input parameters file

Our model depends on many parameters. To keep our instance generator flexible, we have provided multiple value levels for each parameter. The list of all input parameters that can be specified by the user are as follows:

- total_ships: This parameter allows the user to specify the total number of instances that need to be generated. If a single value (n) is provided, then the generator engine randomly selects *n* different ship data files to create *n* single ship instances. If a list is supplied, then only those ships are used to create instances. When a list is provided, the number of single ship instances generated is equivalent to the length of the list. By default, a maximum of 38 (the default number of different ships included in the core data) instances can be generated if all other input parameters are fixed.
- total_cargoes: This specifies the total numbers of cargoes (both on-board and unassigned) that need to be generated. The number of cargoes can be from one to infinity. Practically, values between 50 and 150 might be interesting, depending on market conditions.
- network_number: Enables the user to select any single network, for instance generation. The network number can be between 1 and 4. The four networks have 36, 98, 46 and 48 ports respectively.
- total_planning_time: The total short-term problem planning horizon. Specifying this parameter ensures that the pick-up time windows of all the unassigned cargoes start before the parameter value. The length of the pick-up time windows varies randomly from 3 days to 7 days. The maximum and minimum time-windows lengths are determined from the literature. We use the value 30 for our tests.
- cargo_complexity: This parameter can take values, 1, 5 or 10. Each value varies the percentage of cargoes that belongs to each category. Setting this parameter value to 1 makes sure that 20% of the cargoes belong to category 1, 30% of the cargoes belong to category 2, 20%

[4] http://ports.com/sea-route/
[5] http://aapa.files.cms-plus.com/Statistics/WORLD%20PORT%20RANKINGS%202015.xlsx
[6] http://www.iaphworldports.org/iaph/wp-content/uploads/WorldPortTraffic-Data_for_IAPH_using_LL_data_2017_Final.pdf
[7] http://aapa.files.cms-plus.com/Statistics/NAFTA%20REGION%20CONTAINER%20TRAFFIC%20PORT%20RANKING%202016_T3.pdf

of the cargoes belong to category 3, while the rest of the cargoes belong to category 4. Similarly, setting this parameter to 5 segregates the cargoes into four categories by {40%,30%,20%,10%} percentage split. Furthermore, setting this parameter to value 10 yields a cargo split that follows {70%,10%,10%,10%} percentage split. All four cargo categories are described in Section 6.4.

- totallegs: This number specifies the total number of legs per ship. If total_ships is a list, then the totallegs parameter also needs to be a list of the same size separated by spaces. If a single number is specified, then all the ships with as many numbers of legs are generated. MILP solution time grows exponentially as the number of sailing legs increase. The suggested range of values is from 7 to 15.
- ship_util_level: This parameter specifies the maximum allowable chemical tanker utilisation at the beginning of the time horizon. For example, setting the value to 0.5 assumes that a maximum of half of the chemical tanker can be filled up with onboard cargoes in the generated data. The minimum and maximum values for this parameter are 0 and 1.
- loading_rate: Specifies the loading rate for all the cargoes. The default tested value is 4800 tonnes/day, which is borrowed from Jetlund and Karimi, 2004.
- unloading_rate: Specifies the unloading rate for all the cargoes. Default tested value is 4800 tonnes/day, which is borrowed from Jetlund and Karimi, 2004.
- administrative_time: The default tested value is 0.25 days as stated by Jetlund and Karimi, 2004.
- Alpha and beta: Absolute maximum allowable trim and heel moments in tonnes-metre. The tested value for both parameters in our experiments is 1 tonnes-metre.

6.4. *Instance generator engine*

The instance generator engine takes multiple input parameters that are provided by the user through a text file. A detailed description of all the input parameters is provided in Section 6.3. A high-level pseudo-code of the algorithm, which is used to generate the problem instances is presented using Algorithm 1. A single problem instance comprises of a chemical tanker data file, an on-board cargoes data file, an unassigned cargoes data file, and the problem data file. The chemical tanker data file and both the cargo data files store chemical tanker and cargo-related information, respectively. The problem data file stores port-related information that includes the list of ports in the network and distances (nautical miles) between them. The problem data file also includes the administrative time constant.

The *readInputFile()* processes the inputs that are provided by the user. Subsequently, the *generateShips()* and *generateCargoes()* functions generate the interim chemical tanker data file and the cargo data file. The *generateCargoes()* sub-routine is capable of generating many cargoes infinitely; it can generate as many as four different categories of cargoes. The first category can be stored in any compartment and has no conflict with any other cargo category. Cargoes in category two have conflicts with cargoes of category three. Further, the cargoes in category three have conflicts with cargo categories two and four. Additionally, the cargoes in category three cannot be stored in epoxy-coated compartments. Finally, the cargoes in category four also have conflicts with cargo category three, and can only be stored in compartments that are made of stainless steel.

The cargo data file that is generated by *generateCargoes()* acts as an input to the *modifyToOnboardCargoes()* and *modifyToUnassignedCargoes()* functions. Both these functions generate the final instance files for on-board cargoes and unassigned cargoes. The final list of onboard cargoes has to be generated such that there is at least one cargo-compartment allocation by weight, which respects the chemical tanker stability requirements and the compartment capacities. For this purpose, the sub-routine *solveWeightAssignmentsLP()* solves a linear program (Eqs. (30)–(33)) below. This linear program tries to maximise the weight in each compartment (w_h) while satisfying the compartment capacity constraint (31) and chemical tanker stability constraints (32, 33). Parameter ρ_{min} equals the minimum density amongst all the cargo generated for that instance. Rest of the parameters used in the linear program are already defined in Section 4.

$$\text{Maximise}: \sum_{h\in N^H} w_h \tag{30}$$

$$\text{Subjectto}: 0 \leqslant \quad w_h \quad \leqslant V_h\rho_{min} \qquad \forall h \in N^H, \tag{31}$$

$$-\alpha \leqslant \sum_{h\in N^H} w_h \iota_h \leqslant \alpha, \tag{32}$$

$$-\beta \leqslant \sum_{h\in N^H} w_h \kappa_h \leqslant \beta. \tag{33}$$

Subsequently, cargo numbers are assigned to weights using the *assignCargoNumbersToWeights()* sub-routine. The *assignCargoNumbersToWeights()* sub-routine takes into consideration all the compatibility constraints to give a list of on-board cargoes with at least one feasible cargo-compartment assignment allocation.

Algorithm 1. Instance Generator Engine

```
procedure GENERATE_INSTANCES(CORE_DATA_DIRECTORY, OUTPUT_DIRECTORY)
    readInputFile()
    generateShips()
    generateCargoes() → Returns cargo_data_file
    modifyToOnboardCargoes(cargo_data_file)
      |- solveWeightAssignmentsLP()
      |- assignCargoNumbersToWeights()
    modifyToUnassignedCargoes(cargo_data_file)
    modifyShipData()
    generateProblemData()
end procedure
```

The generator then modifies the chemical tanker data file to include the list of on-board cargoes, immediate destination and port costs through the *modifyShipData()* routine. The *modifyShipData()* routine completes the chemical tanker data instance file. Finally, it generates the problem instance file using the *generateProblemData()* function. In the next section, we discuss computational experiments on instances obtained from the generator.

7. Computational study

The computational study is divided into two main parts. First, we discuss the effects of improvements in the model formulation. Second, we present a secondary study related to the performance of the proposed heuristic.

We generated 200 test instances for our experiments in the following way. A default seed value of 10 and *total_ships* input parameter value of 38 were provided to our generator to obtain 1,672 (44 different instance sets and 38 chemical tankers) random instances. To keep the number of test instances reasonable, we selected a subset of 13 chemical tankers (Table A.2), with the most diverse characteristics. We narrowed down our test set by randomly selecting 200 test instances in such a way that there is at least one instance from each of the 44 instance sets, and at least one for each of the 13 ships. The instances are named *INST_SET_SHIP*. *SET* denotes the instance set number for a given instance. *SHIP* denotes the chemical tanker number that is used in that particular instance. For example, instance *INST_1_1* would belong to the instance set 1, and model ship 1 (BOW MEKKA) operations. Table A.1 lists the different input parameter values used to generate 44 instance sets. Tables A.1 and A.2 also tabulate some solution-related statistics, which will be discussed in Section 7.2.

Instances with the same instance set number have identical input parameters (mentioned in Section 6.3). However, every instance within the same instance set has different chemical tanker characteristics. Additionally, even though the input parameter value *total_cargoes* is same for an instance set it only defines the cardinality of the cargo set. Individual cargoes differ in terms of cargo characteristics like total weight, density, origin, destination and pick-up time-windows.

The generator was run using R (version 3.6.1) and RStudio (version 1.1.383). All subsequent tests are carried out using the Cplex 12.7.1 MILP solver. Each instance was solved using 4 cores of the Xeon-E5-2667-v3 3.20 GHz CPU and 8 GB RAM. We used C++11 standard libraries and Cplex Concert Technology libraries to construct all the formulations.

We have uploaded along with the instance generator all the 200 test instances[8]. Logs and solution files for the Cplex run are available online. The Cplex run includes solving the REV formulation presented in Section 4 using the Cplex solver for a CPU time limit of 86,400 s, and the default MIP gap tolerance of 0.01 %. First two sets of experiments described in Sections 7.1 and 7.2 have the changeover time (T^S) set to zero. The changeover cost (C^S) is also set to 0 during leg 0. These settings enable us to compare our formulations with the existing ones. We also discuss the effect of non-zero changeover time (T^S) in Section 7.3.

For the Cplex run, the total number of variables varies between 23,115 and 4,97,749, while the total number of constraints vary between 24,440 and 5,23,392. It can be further observed that Cplex terminated with an optimal solution for 115 test instances with an average CPU time of 18,186 s, and a feasible solution for another 85 test instances. Amongst the 85 instances, Cplex found a Cplex gap (%) of less than 10 % for 3 instances, between 10 % and 50 % for 24 instances, and greater than 50 % for 58 instances. Additionally, our experiments showed that the Cplex run with pre-processing went out of memory for only 2/200 test instances. However, the Cplex runs without pre-processing went out of memory for a total of 77/200 instances. Results related to the performance of the heuristic will be discussed in Section 7.2. The next section presents our primary empirical study, which compares two different formulations of the s-PDP-TWTAC.

7.1. Effects of improvements in the model formulation

We now compare our revised (REV) formulation and an existing (OG) formulation of the s-PDP-TWTAC. In order to make a fair comparison, we make use of the new approximation of time windows in both the formulations. We refer to the formulation presented by Neo et al., 2006, which is altered with our definition of pick-up time windows as the original formulation (OG). Further, we refer to the formulation presented in Section 4 as the revised formulation (REV). As the solution times are large we limit our comparative study to 30 instances selected from the above set of 200 instances.

The OG and the REV formulations are compared on problem size, and their linear relaxations. The LP relaxation provides an upper bound to the optimal value of the model. Lower the upper bound, tighter is the relaxation and closer to the integer feasible points. Fig. 10 reports the percentage reduction from OG to REV in the number of variables and constraints. It was observed that the problem size decreased drastically for all the instances when the REV formulation is used. 21 out of 30 instances show a reduction of at least 90 % in the total number of variables, and other instances show a reduction by at least 76 %. Further, the total constraints decrease by 15 % to 23 %. The LP relaxation of both the formulations could be solved within the time limit for 22 out of 30 instances. Fig. 11 compares the LP relaxation value of OG and REV formulations on these 22 instances. It shows that both the solution time and the upper bound decreases for our revised (REV) formulation. In the remaining 8 instances, the LP relaxation of the OG formulation ran out of memory. In contrast, solver managed to solve the REV formulation without any memory issues.

Now we compare performance parameters such as the total solution time and the relative gap at time limit for the two MILP formulations (Table 2). The second column (*Cplex Status*) reports *OOM* if the MILP solver ran out of memory. The column that is labelled, *Best Objective*, lists the best lower bound obtained at termination. The total solution time at termination for all the 30 instances is reported in the succeeding column (*Total CPU Time*). Additionally, the *Relative Gap* column presents the relative gap (%) between the upper and lower bound on the optimal value reported by Cplex at termination.

Out of the 30 instances, 18 instances ran out of memory (8 GB) without discovering any feasible integer solution when solved using the OG formulation. In contrast, the REV formulation stays within memory limits and finds at least one feasible integer solution for all 30 instances. Additionally, REV formulation finds the optimal solution in 15 instances (within the time limit), while the OG formulation terminates with optimality in only six instances. We observe that REV formulation is much faster (up to 15 times) and memory efficient as compared to the OG formulation. We now describe the performance of the proposed heuristic next.

7.2. Revised formulation results: Cplex vs. Heuristic

All the 200 test instances were used to perform this empirical computational study. The primary goal of this experiment is to present the results of solving the s-PDP-TWTAC revised formulation with the proposed heuristic. The heuristic is run for a CPU time of 86,400 s. The heuristic also terminates if the Cplex reported relative gap (%) is less than 0.01 %. We also make some preliminary comparisons between the Cplex run and the heuristic run. Further, we discuss the sensitivity of some of the performance parameters with respect to the input parameters used to generate the test instances. The performance parameters include the Gap (%) and the total CPU time (sec) of both the Cplex run and the heuristic run. The comparison is aimed at understanding why

[8] https://ladageanurag.shinyapps.io/s-PDP-TWTAC/

the MILP solver takes a long time. A solver might be slow because it is not able to find a good solution early on. It might also be slow because it is unable to prove that the solution is optimal. Even though our heuristic finds better quality solutions faster for several instances, the Cplex run can solve the problem exactly for some others. Further, the Cplex run also generates an upper bound for the overall problem that our heuristic does not.

We report some of the solution statistics in Tables A.1 and A.2. As explained earlier, the first four columns of Tables A.1 and A.2 give the instance set characteristics and the chemical tanker characteristics, respectively. In Table A.1, Column *Instance per set* gives the number of test instances (out of 200) that belong to each instance set. Similarly, in Table A.2, Column *instance per ship* gives the number of test instances (out of 200) for every chemical tanker. In both tables, Columns *Avg. variables* to *Avg. Heur CPU time* tabulate the corresponding solution statistics. Table A.1 presents average solution statistics for every instance set. Likewise, Table A.2 report solution statistics averaged for each of the chemical tankers.

We record four performance parameters, the Heur Gap (%), the Cplex Gap (%), the Cplex CPU time (sec) and the Heur CPU time (sec). The *Heur Gap (%)* presents the percentage difference between the upper bound obtained during the Cplex run, and the lower bound obtained from the heuristic run. Moreover, the *Cplex Gap (%)* is the percentage difference between the Cplex upper and lower bounds. Both the gaps (%) are with respect to the absolute values of the upper bounds generated by Cplex. Lower the gap better is the performance of the run. The *Cplex Gap (%)* and the *Heur Gap (%)* for a given instance are calculated as follows:

$$\mathrm{CplexGap}\left(\%\right) = \frac{\mathrm{Cplexupperbound} - \mathrm{Cplexbestobjective}}{\mathrm{Abs(Cplexupperbound)}} \times 100$$

$$\mathrm{HeurGap}\left(\%\right) = \frac{\mathrm{Cplexupperbound} - \mathrm{Heuristicbestobjective}}{\mathrm{Abs(Cplexupperbound)}} \times 100$$

Subsequently, the average performance measures reported in Tables A.1 and A.2 are calculated as follows.

$$\mathrm{Avg.\,Cplex\,Gap}(\%) = \frac{\mathrm{Total\,Cplex\,Gap}(\%)\,\mathrm{for}\,n\,\mathrm{instances}}{\mathrm{n}}$$

$$\mathrm{Avg.\,Heur\,Gap}(\%) = \frac{\mathrm{Total\,Heur\,Gap}(\%)\,\mathrm{for}\,n\,\mathrm{instances}}{\mathrm{n}}$$

$$\mathrm{Avg.\,Cplex\,CPU\,time(sec)} = \frac{\mathrm{Total\,Cplex\,run\,CPU\,time(sec)\,for}\,n\,\mathrm{instances}}{\mathrm{n}}$$

$$\mathrm{Avg.\,Heur\,CPU\,time(sec)} = \frac{\mathrm{Total\,heuristic\,run\,CPU\,time(sec)\,for}\,n\,\mathrm{instances}}{\mathrm{n}}$$

Table A.1 reports *n* in Column *Instances per set*, while Table A.2 reports it in Column *Instances per ship*. The heuristic run terminates with either no integer solution (Phase I OOM), local optimal solution (Local Optimal) or a feasible integer heuristic solution (Phase II OOM or time limit). The local optimal solution is the best possible solution that can be generated by the heuristic without hitting the time limit or the memory limit. It may not be the optimal solution of the REV formulation.

Out of the 200 instances, our heuristic terminated due to local optimality in 157 instances, due to feasibility and time limit (Phase II time limit or OOM) in 33 instances, and due to Phase I OOM issue in 10 instances. A run terminates with *Phase I OOM* if memory is exhausted while solving the linear relaxation of the problem. On the other hand, the Phase II time limit or OOM termination occurs if the Phase II MILP does not terminate within the time limit or the solver runs out of memory, respectively.

Fig. 12 presents the Gap (%) and CPU time (sec) of both the runs. The horizontal axis plots instances in increasing order of Cplex Gap (%). The Gap (%) of both the runs are plotted in Chart 1, while the second chart plots the CPU time (sec) for both runs. We classify the 200 test instances into two sets. Set I (118 instances) includes all the instances with Cplex Gap (%) less than 1 %, while Set II (82 instances) includes all the

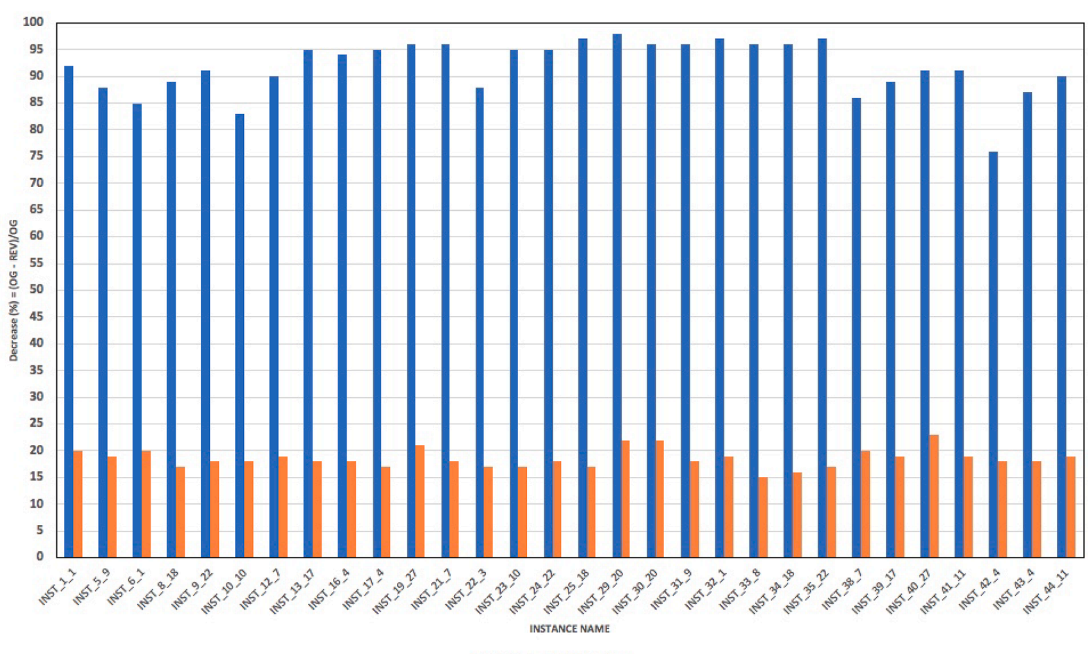

Fig. 10. Percentage reduction in the number of variables and constraints - OG formulation vs. REV formulation.

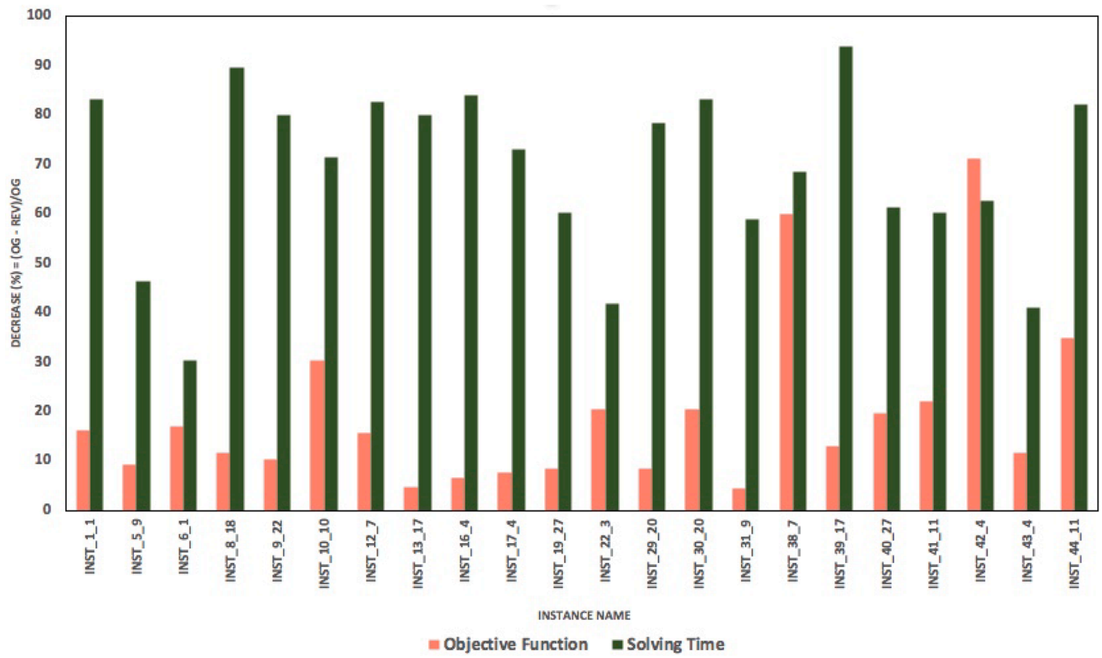

Fig. 11. LP Relaxation - Objective value and solution time comparison - OG formulation vs. REV formulation.

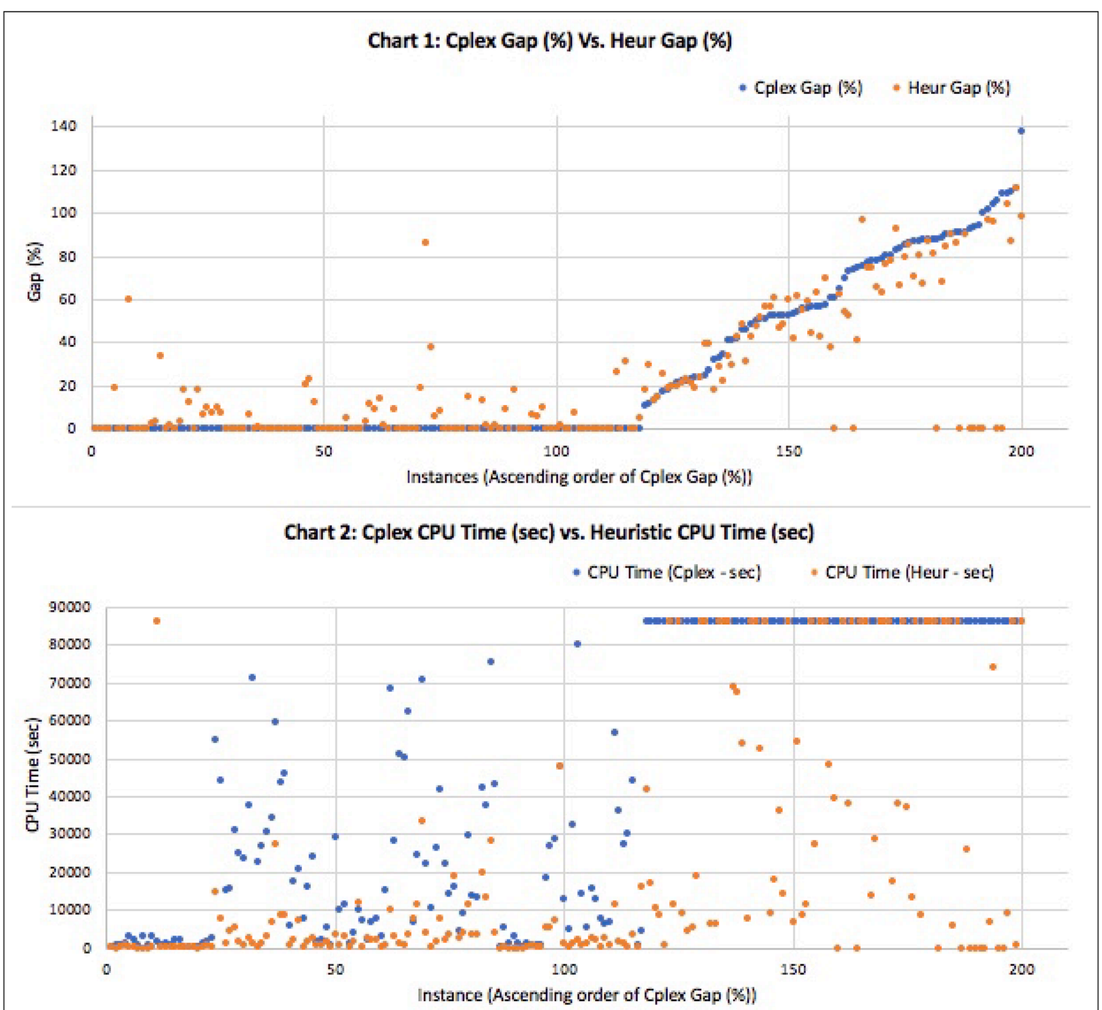

Fig. 12. Comparison of the Gap (%) and the CPU time (sec) of the Cplex and heuristic runs as the problem difficulty (Cplex Gap (%)) increases.

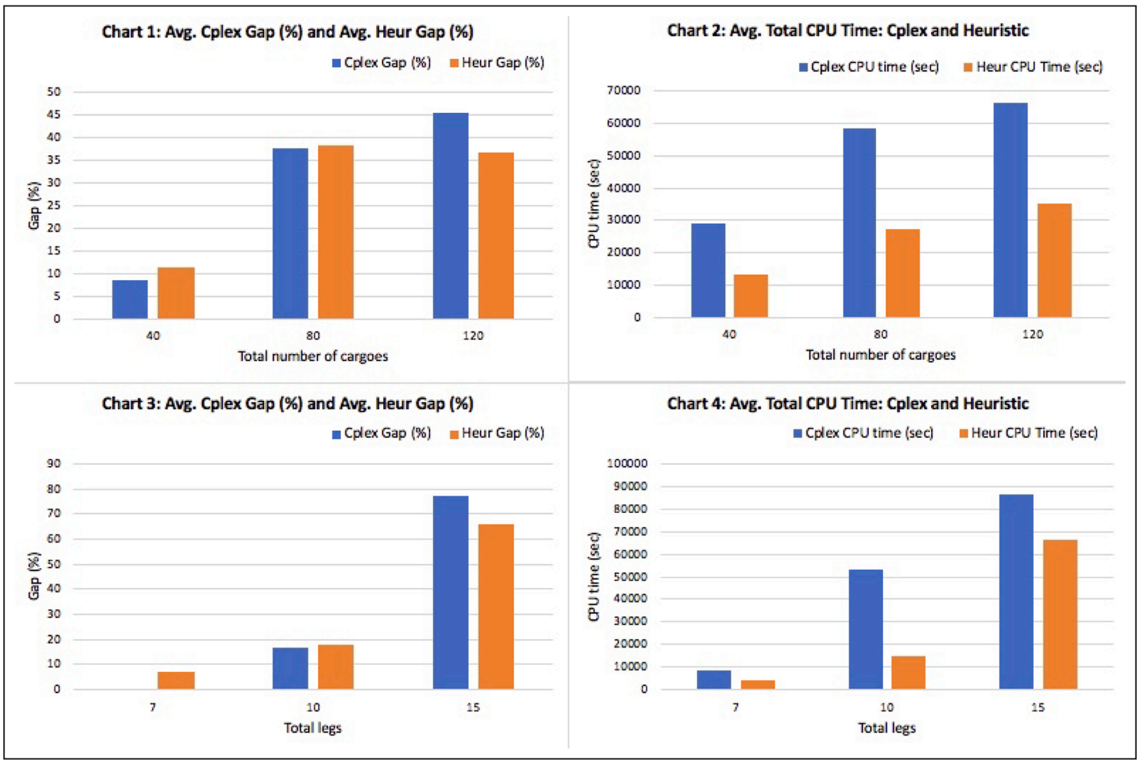

Fig. 13. The effect of the total number of cargoes and the maximum number of sailing legs on the average performance parameters.

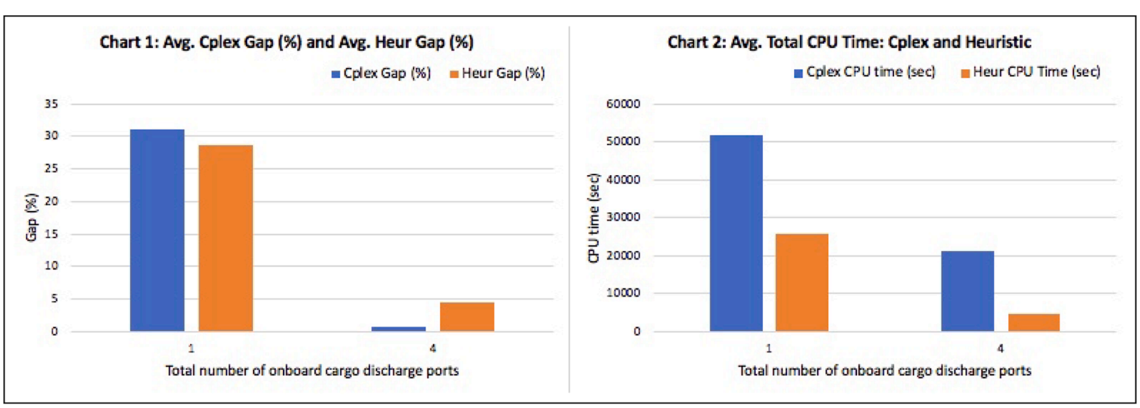

Fig. 14. The effect of the total number of onboard discharge ports on the average performance parameters.

Table 2
Comparison of problem status, best integer solution, total solution time and relative gap between the original and the revised formulation.

Instances	Cplex Status		Best Objective		Total CPU Time		Relative Gap	
	OG	REV	OG	REV	OG (sec)	REV (sec)	OG (%)	REV (%)
INST_1_1	Optimal	Optimal	1904910	1904910	5651	412	0.01	0.01
INST_5_9	Optimal	Optimal	995262	995262	72102	31006	0.01	0.01
INST_6_1	Feasible	Optimal	1537990	1537990	86400	25274	54.563	0.01
INST_8_18	OOM	Optimal	No Sol	2269880	718	20858	Inf	0.01
INST_9_22	OOM	Feasible	No Sol	3239300	594	86400	Inf	53.261
INST_10_10	Feasible	Feasible	407682	372226	86400	86400	607.687	689.32
INST_12_7	OOM	Feasible	No Sol	1172500	779	86400	Inf	104.12
INST_13_17	OOM	Optimal	No Sol	2801950	1054	2288	Inf	0.01
INST_16_4	OOM	Optimal	No Sol	1512150	1538	15294	Inf	0.01
INST_17_4	OOM	Feasible	No Sol	1829870	834	86400	Inf	109.954
INST_19_27	Feasible	Feasible	3821410	4684360	86400	86400	82.4	37.607
INST_21_7	OOM	Feasible	No Sol	1531150	14	86400	Inf	314.955
INST_22_3	OOM	Feasible	No Sol	−281065	70	86400	Inf	1220.24
INST_23_10	OOM	Feasible	No Sol	2951890	30	86400	Inf	132.28
INST_24_22	OOM	Feasible	No Sol	793339	30	86400	Inf	695.712
INST_25_18	OOM	Optimal	No Sol	3769140	37	70888	Inf	0.01
INST_29_20	OOM	Optimal	No Sol	3798930	270	75667	Inf	0.01
INST_30_20	OOM	Optimal	No Sol	2219180	110	43172	Inf	0.01
INST_31_9	OOM	Feasible	No Sol	5863370	747	86400	Inf	15.953
INST_32_1	OOM	Feasible	No Sol	2450400	33	86400	Inf	109.332
INST_33_8	OOM	Feasible	No Sol	991297	33	86400	Inf	771.241
INST_34_18	OOM	Feasible	No Sol	581309	1	86400	Inf	1035.6
INST_35_22	OOM	Feasible	No Sol	−585672	34	86400	Inf	1740.83
INST_38_7	Optimal	Optimal	−1167910	−1167910	6477	5420	0.01	0.01
INST_39_17	Optimal	Optimal	1218010	1218010	16594	1014	0.01	0.01
INST_40_27	Optimal	Optimal	623656	623656	1640	750	0.01	0.01
INST_41_11	Feasible	Feasible	25338.7	200491	86400	86400	3081.64	0.05
INST_42_4	Feasible	Optimal	−811647	−781745	86400	32743	51.717	0.01
INST_43_4	Optimal	Optimal	2836360	2836360	76294	15602	0.01	0.01
INST_44_11	Feasible	Optimal	470923	505503	86400	36092	199.387	0.01

instances with Cplex Gap (%) greater than 1%.

The heuristic terminated with a solution equivalent to the Cplex optimal solution for 63 instances, which are a subset of Set I. For these 63 instances, the total solution time reduced by 71.62 %. Within Set I, the Cplex run terminated with a lower Gap (%) when compared to the heuristic run for 54 instances. For the instances in Set I, the total solution time of the heuristic run increased for 6 instances with an average of 786.67 %, while it decreased for 112 instances with an average of 79.87 %, when compared to the Cplex run.

Within Set II, the heuristic could not find a solution for 10 instances. For rest of the 72 instances (within Set II), compared to the Cplex run, the heuristic run terminated with a better, same and worst lower bound for 49, 3 and 20 instances, respectively. For Set II, an average Cplex gap of 59.95 % was observed. In comparison, the heuristic run resulted in an average gap (%) of 55.74 %. Moreover, the CPU time (sec) for the heuristic run decreased by 41.75 % when compared to the Cplex run for the instances in Set II. In summary, Fig. 12 shows that for instances in Set I, Cplex finds better quality solutions than the heuristic. However, as the Cplex Gap (%) increases the heuristic consistently finds better quality solutions compared to Cplex. Moreover, for majority of the test instances, the heuristic run terminates faster than the Cplex run.

Further, we discuss the sensitivity of the performance parameters, namely; the Gap (%) and the total CPU time of both the Cplex run and the heuristic run with respect to the different input parameters. The Gap (%) of both the runs is calculated with respect to the upper bound reported by Cplex. The primary input parameters considered for this study are the total number of cargoes, the total number of ports, the maximum number of sailing legs, and the number of discharge ports of the onboard cargoes. Moreover, some secondary input parameters like the total number of compartments, the ship speed, the draft constant, the fuel cost, the time charter cost, and the average compartment volume were also considered during this study.

Anaysis using multiple linear regression was performed to explore the effects of the input parameters on the performance parameters. Some of the primary input parameters significantly affect the performance parameters. Figs. 13 and 14 help us illustrate this claim. However, the performance parameters seem to be insensitive to the secondary input parameters. Figs. 13 and 14 classify the test instances into different categories based on the input parameters. The vertical axis in these figures presents the average of the performance parameters. For example, the first chart in Fig. 13 differentiates the test instances based on the total number of cargoes on the horizontal axis. Similarly, the vertical axis presents the average Gap (%).

In Fig. 13, Chart 3 and 4 show that both the average solution quality and the average total CPU time worsen with the increase in the maximum number of sailing legs. Similarly, both the Cplex performance parameters deteriorate with the increase in the maximum number of legs. Additionally, even though the average total CPU time for the

Table 3
Effect of changeover time on the revised formulation (30 instances).

	Cplex status count		Avg. CPU	Avg. Gap	Obj change
	Optimal	Feasible	time (sec)	(%)	Avg. (%)
REV ($\hat{T}S$ = 0)	15	15	55749	234.35	NA
REV ($\hat{T}S$ = 1 h)	13	17	66130	1073.07	31.9

heuristic run worsens with the total number of cargoes, the solution quality does not.

Charts 1 and 2 in Fig. 14 illustrate that the increase in the total number of onboard cargo discharge ports significantly improves the Gap (%) and the total CPU time related to both the runs. This effect is correct because the total number of onboard cargo discharge ports reduces the flexibility of the route of the chemical tanker. As per the problem definition, all onboard cargoes must be delivered. Consequently, their corresponding discharge ports have to be visited. Thus, a higher number of different discharge ports of onboard cargoes reduces the number of new ports on the route of the ship. This reduces the feasible region of the problem. Additionally, our preliminary sensitivity analysis shows that the performance parameters were not affected by the number of ports or the number of compartments.

7.3. Effect of changeover time (T^S) on the REV formulation

We perform this study on the same 30 instances shortlisted in Section 7.1 (Table 2). We solve the instances with $T^S = 0$ and $T^S = 1$ h. Additionally, for $T^S = 1$ runs we include the changeover cost at the end of leg 0. Both runs were restricted to 86400 s of CPU time per instance. Summary of these results is presented in Table 3. The performance measures listed in Table 3 are calculated as follows:

$$\text{Avg. CPU time} = \frac{\text{Total CPU time for n instances}}{\text{n}}$$

$$\text{Avg. Gap } (\%) = \frac{\text{Total Cplex Gap } (\%) \text{ for n instances}}{\text{n}}$$

$$\text{Obj change } (\%) = \frac{ABS(OBJ_{T^S=0} - OBJ_{T^S=1})}{ABS(OBJ_{T^S=0})} \times 100$$

$$\text{Avg. Obj change } (\%) = \frac{\text{Total Obj change } (\%) \text{forninstances}}{\text{n}}$$

For $T^S = 0$ runs, Cplex terminates with an optimal solution for 15 instances and a feasible solution for the remaining 15 instances. On the other hand, $T^S = 1$ runs terminated with an optimal and feasible solution for 13 and 17 instances, respectively. We observe that the average CPU time (sec) for the 30 instances increases by 18.6 % for $T^S = 1$ run. The overall Avg. Gap (%) also increases for $T^S = 1$ runs. The Avg. Obj change (%) is around 31.9 %, with a high standard deviation of 63 %. For 21/30 instances, the Obj change (%) is less than 5 %. In 7 instances, the Obj change (%) was more than 50 %. Thus, there seems to be a moderate effect of including T^S on both the solution time and the optimal value.

8. Conclusions and future work

We present a revised (MILP) formulation that incorporates compartment-related decisions. Compartment-related decisions include compartment capacities, ship stability criteria, cargo-compartment and cargo-cargo incompatibility norms. These decisions directly impact the higher level scheduling decisions like determining the route of the chemical tanker and cargo pickups. Our revised formulation substantially reduces the number of decision variables and constraints. The revised formulation performs better than the existing formulation that is available in the literature in terms of memory requirements, solution quality and Cplex solution time. We hope our instance generator and the library of instances will be used by other researchers to improve the models and solution techniques.

The proposed heuristic shows that the existing heuristics in general-purpose MILP solvers are insufficient to tackle the s-PDP-TWTAC problem. The proposed LP-based neighborhood search heuristic gives better results than a commercial MILP solver but still may be slow for practical purposes. Using an LP-based neighborhood search heuristic, we restrict the feasible region using some knowledge of the continuous variables. Additionally, the REV formulation gives a tighter LP relaxation than the OG formulation. This made the LP relaxation a good candidate to define a restricted neighborhood. However, Phase I (solving the LP relaxation) also impedes the heuristic's overall performance. We hope to improve this heuristic in the future.

It can be concluded from Fig. 12 that the heuristic consistently finds better quality lower bounds in comparison to the Cplex run as the problem complexity (Cplex gap (%)) increases. Additionally, we conclude that the Cplex and heuristic performances deteriorate with the increase in the number of sailing legs and the total number of cargoes. The Cplex and heuristic performance also improves with the number of different discharge ports for onboard cargoes. Preliminary study showed that all other inputs to the instance generator did not affect the gaps and total solution time (both runs) significantly. However, additional study using a significantly higher number of test instances is required to discover better inputs to performance relationships.

We have made some simplifying assumptions to make the problem tractable. The most important of these simplifications are the in-port activities of the chemical tanker. For example, a port has multiple berth or terminals. Thus, once the chemical tanker enters a port, it is required to visit different berths in order to load/unload the cargoes or to refuel. Incorporating these into the problem definition along with partial cargo pickups, and multiple discharge ports for a single cargo are some other interesting practical considerations, which would increase the robustness of the problem.

Section 7.3 shows that the inclusion of changeover time affects the average solution time and the objective value moderately. In practice, the changeover time (T^S) may vary based on various factors like the number of cleaning cycles and the type of cargo stored in the compartment. Additional study is required to determine its effect on the problem.

Another extension of our work would be to scale the present model to optimise an entire fleet of heterogeneous chemical tankers. Having multiple ships increases the complexity significantly. We are also trying to gain a better insight into other input factors that could affect the complexity of the problem. Subsequently, we hope to introduce further instances for the multi-ship version of the s-PDP-TWTAC.

CRediT authorship contribution statement

Anurag Ladage: Conceptualization, Data Curation, Formal Analysis, Investigation, Methodology, Visualization, Writing - original draft, Writing - review & editing. **Davaatseren Baatar:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Supervision, Validation, Writing - review & editing. **Mohan Krishnamoorthy:** Conceptualization, Funding acquisition, Methodology, Project administration, Supervision, Writing - review & editing. **Ashutosh Mahajan:** Conceptualization, Funding acquisition, Methodology, Project administration, Supervision, Writing - review & editing.

Acknowledgments

We thank the editor and the reviewers of this journal for their insightful and detailed suggestions to improve this paper. Their suggestion led to improvements in the formulation, computational results and presentation of the paper.

Appendix A

Table A.1
Instance set data combined with the average (per instance set) statistics related to Cplex and heuristic runs of the revised formulation.

Instance set	Total cargoes	Network number	Total legs	Onboard cargoes discharge port	Instances per set	Avg. variables	Avg. constraints	Avg. Cplex gap (%)	Avg. heur gap (%)	Avg. Cplex CPU time (sec)	Avg. heur CPU time (sec)
1	40	1	7	1	6	43408	57191	0.01	3.25	1448.83	366.83
2	40	2	7	1	6	94383	60193	0.01	10.16	1681.83	14530.67
3	40	3	7	1	8	45221	55980	0.01	7.92	970.25	127.88
4	40	4	7	1	4	46720	51995	0.01	7.66	2600	4358.5
5	40	1	10	1	5	49667	59382	0.01	8.47	32214.4	6831.6
6	40	2	10	1	6	136299	83130	3.71	3.67	44494.67	1866
7	40	3	10	1	6	60921	63053	0.01	1.42	40243.83	9357
8	40	4	10	1	6	71844	83683	0.01	0.01	15380.5	2397.83
9	40	1	15	1	5	100603	132176	24.8	20.79	86400	70942.6
10	40	2	15	1	4	194300	94044	101.22	96.37	86400	22569.25
11	40	3	15	1	3	114784	135634	45.25	30.72	86400	86400
12	40	4	15	1	2	114085	132965	46.19	40.92	86400	77056.5
13	80	1	7	1	4	83335	118948	0.01	13.99	2498	738.75
14	80	2	7	1	4	136394	124260	0.01	0.01	13012.75	1903
15	80	3	7	1	5	87365	120124	0.01	1.04	6140.8	3676.4
16	80	4	7	1	3	72798	94775	0.01	8.04	8695.67	1076
17	80	1	10	1	6	98349	134064	17.5	20.21	73734.83	12582.5
18	80	2	10	1	6	187005	159942	35.43	37.19	80524.83	14011.67
19	80	3	10	1	3	141346	194443	43.48	45.35	86400	48502.33
20	80	4	10	1	4	112200	146745	26.04	31.32	71406.75	5128.75
21	80	1	15	1	4	165962	238134	83.68	81.54	86400	86400
22	80	2	15	1	6	264903	200870	100.18	90	86400	37930
23	80	3	15	1	5	196374	264101	68.95	61.75	86400	76789
24	80	4	15	1	1	233813	316200	87.43	86.67	86400	86400
25	120	1	7	1	3	150964	223035	0.01	0.01	34131	17464.33
26	120	2	7	1	4	162734	169457	0.01	35.7	25290.25	3549.5
27	120	3	7	1	4	101858	137909	0.01	3.53	14367.5	6825.5
28	120	4	7	1	6	147804	211763	0.01	2.57	24387.83	9310.83
29	120	1	10	1	3	171276	290309	27.11	28.46	82822.33	67004.67
30	120	2	10	1	6	257975	280855	61.66	53.16	79195.33	20739
31	120	3	10	1	6	148034	207094	27.54	31.04	86400	34733
32	120	4	10	1	3	192374	284153	52.74	53.07	86400	36389.67
33	120	1	15	1	6	238604	341598	82.46	66.87	86400	73790.67
34	120	2	15	1	4	374820	400572	90.65	82.87	86400	86400
35	120	3	15	1	5	275870	395457	91.68	60.31	86400	86400
36	120	4	15	1	5	244908	327946	82.79	78.08	86400	53841.33
37	40	1	7	4	1	53353	83854	0.01	0.01	503	68
38	40	2	7	4	5	92015	58793	0.01	5.9	2208.4	127.8
39	40	3	7	4	2	56571	63176	0.01	0.01	1000	574.5
40	40	4	7	4	2	54644	65740	0.01	3.45	659.5	172.5
41	40	1	10	4	5	61922	82451	0.02	4.36	41816.4	21566.6
42	40	2	10	4	5	132411	78300	0.01	1.99	28926	1101
43	40	3	10	4	7	66584	79313	3.2	3.37	20168.57	2177.43
44	40	4	10	4	6	68680	85378	0.01	9.69	32624.83	2976.33

Table A.2
Ship data combined with average (per ship) statistics related to Cplex and heuristic runs of the revised formulation.

Ship numbers	Ship name	Draft constant (tonnes)	Total compartments	Ship speed (knots)	Instances per ship	Avg. variables	Avg. constraints	Avg. Cplex gap (%)	Avg. heur gap (%)	Avg. Cplex CPU time (sec)	Avg. Heur CPU time (sec)
1	BOW MEKKA	11176	52	14.3	16	152081	209057	20.78	19.18	41222.12	25090.25
3	BOW HECTOR	8154	16	14.2	15	84040	58615	17.43	26.97	41849	9133.67
4	BOW SANTOS	5027	22	14.1	18	105427	88765	29.21	34.03	49803.89	21144.22
7	BOW FAGUS	11176	52	14.3	18	164351	215119	22.03	21.48	46607.94	27238.88
8	BOW ATLANTIC	4541	24	13.6	16	107265	114122	34.74	37.67	52575.31	18861.38
9	BOW KISO	8793	16	13.8	16	79702	64199	21.15	23.54	39526.56	10960.44
10	BOW FORTUNE	10915	47	14.3	17	149452	180304	22.09	16.16	42259.41	16540.25
11	BOW SAGA	10262	40	14.6	17	128695	172658	22.02	22.13	46645.71	22842.81
17	BOW ARCHITECT	7691	28	14.5	13	123080	123345	29.04	30.09	47945.46	27024.69
18	BOW CEDAR	11176	52	14.3	15	208423	241611	31.08	16.82	50513	28112.42
20	BOW LIND	13416	29	13.7	11	111102	154050	24.6	20.99	50484.27	34533.27
22	BOW FIRDA	10915	47	14.3	19	190641	239079	43.95	27.94	59761.16	37577.53
27	BOW HERON	9450	31	14.5	9	98211	101828	13.05	15.52	26950.22	10811.44

Appendix B. Supplementary data

Supplementary data associated with this article can be found, in the online version, athttps://doi.org/10.1016/j.cor.2021.105345.

References

Barbucha, D., 2004. Three approximation algorithms for solving the generalized segregated storage problem. Eur. J. Oper. Res. 156, 54–72.

Berthold, T. (2007). Rens-relaxation enforced neighborhood search.

Brooks, M.R., Faust, P., 2018. 50 Years of Review of Maritime Transport, 1968–2018: Reflecting on the Past. Exploring the Future, Technical Report, UNCTAD.

Brouer, B.D., Pisinger, D., Spoorendonk, S., 2011. Liner Shipping Cargo Allocation with Repositioning of Empty Containers. INFOR 49, 109–124. https://doi.org/10.3138/infor.49.2.109. URL:http://www.utpjournals.press/doi/abs/10.3138/infor.49.2.109.

Christiansen, M., Fagerholt, K., Ronen, D., 2004. Ship Routing and Scheduling: Status and Perspectives. Transp. Sci. 38, 1–18. https://doi.org/10.1287/trsc.1030.0036. URL:http://pubsonline.informs.org/doi/abs/10.1287/trsc.1030.0036.

Christiansen, M., Fagerholt, K., Nygreen, B., & Ronen, D. (2007). Chapter 4 Maritime Transportation. In C.B. a. G. Laporte (Ed.), Handbooks in Operations Research and Management Science (pp. 189–284). Elsevier volume 14 of Transportation. URL: http://www.sciencedirect.com/science/article/pii/S0927050706140049.

Christiansen, M., Fagerholt, K., Nygreen, B., Ronen, D., 2013. Ship routing and scheduling in the new millennium. Eur. J. Oper. Res. 228, 467–483. https://doi.org/10.1016/j.ejor.2012.12.002. URL:http://www.sciencedirect.com/science/article/pii/S0377221712009125.

Cóccola, M.E., Méndez, C.A., 2015. An iterative milp-based approach for the maritime logistics and transportation of multi-parcel chemical tankers. Comput. Ind. Eng. 89, 88–107.

Cóccola, M.E., Dondo, R., Méndez, C.A., 2015. A MILP-based column generation strategy for managing large-scale maritime distribution problems. Comput. Chem. Eng. 72, 350–362. https://doi.org/10.1016/j.compchemeng.2014.04.008. URL:http://linkinghub.elsevier.com/retrieve/pii/S0098135414001252.

Elgesem, A.S., Skogen, E.S., Wang, X., Fagerholt, K., 2018. A traveling salesman problem with pickups and deliveries and stochastic travel times: An application from chemical shipping. Eur. J. Oper. Res. 269, 844–859.

Fagerholt, K., Christiansen, M., 2000. A combined ship scheduling and allocation problem. J. Oper. Res. Soc. 834–842.

Hemmati, A., Hvattum, L.M., Fagerholt, K., Norstad, I., 2014. Benchmark suite for industrial and tramp ship routing and scheduling problems. INFOR 52, 28–38.

Hennig, F., Nygreen, B., Furman, K., Song, J., 2015. Alternative approaches to the crude oil tanker routing and scheduling problem with split pickup and split delivery. Eur. J. Oper. Res. 243, 41–51. https://doi.org/10.1016/j.ejor.2014.11.023. URL:http://linkinghub.elsevier.com/retrieve/pii/S0377221714009461.

Homsi, G., Martinelli, R., Vidal, T., Fagerholt, K., 2020. Industrial and tramp ship routing problems: Closing the gap for real-scale instances. Eur. J. Oper. Res. 283, 972–990.

Hvattum, L.M., Fagerholt, K., Armentano, V.A., 2009. Tank allocation problems in maritime bulk shipping. Comput. Oper. Res. 36, 3051–3060. https://doi.org/10.1016/j.cor.2009.02.002. URL:http://linkinghub.elsevier.com/retrieve/pii/S0305054809000422.

Jetlund, A.S., Karimi, I., 2004. Improving the logistics of multi-compartment chemical tankers. Comput. Chem. Eng. 28, 1267–1283. https://doi.org/10.1016/j.compchemeng.2003.08.009. URL:http://linkinghub.elsevier.com/retrieve/pii/S0098135403002102.

Kjeldsen, K.H., 2011. Classification of ship routing and scheduling problems in liner shipping. INFOR 49, 139–152. URL:http://www.utpjournals.press/doi/abs/10.3138/infor.49.2.139.

Lin, D.-Y., Liu, H.-Y., 2011. Combined ship allocation, routing and freight assignment in tramp shipping. Transp. Res. E 47, 414–431.

Neo, K.-H., Oh, H.-C., Karimi, I., 2006. Routing and cargo allocation planning of a parcel tanker. Comput. Aid. Chem. Eng. 21, 1985–1990.

Papageorgiou, D.J., Nemhauser, G.L., Sokol, J., Cheon, M.-S., Keha, A.B., 2014. Mirplib–a library of maritime inventory routing problem instances: Survey, core model, and benchmark results. Eur. J. Oper. Res. 235, 350–366.

Rodrigues, V.P., Morabito, R., Yamashita, D., Da Silva, B.J., Ribas, P.C., 2016. Ship routing with pickup and delivery for a maritime oil transportation system: Mip model and heuristics. Systems 4, 31.

Ronen, D., 1983. Cargo ships routing and scheduling: Survey of models and problems. Eur. J. Oper. Res. 12, 119–126.

dos Santos, P.T.G., Kretschmann, E., Borenstein, D., & Guedes, P.C. (2020). Cargo routing and scheduling problem in deep-sea transportation: Case study from a fertilizer company. Computers & Operations Research, (p. 104934).

Schwindt, C., Zimmermann, J., et al., 2015. Handbook on project management and scheduling, vol. 1. Springer International Publishing, Cham.

Sun, P., Veelenturf, L.P., Hewitt, M., Van Woensel, T., 2018. The time-dependent pickup and delivery problem with time windows. Transp. Res. B 116, 1–24.

United Nations, 2019. Review of maritime transport 2019. United Nations, UNCTAD 2019. URL:https://unctad.org/en/PublicationsLibrary/rmt2019_en.pdf.

Vilhelmsen, C., Larsen, J., Lusby, R.M., 2015. Tramp Ship Routing and Scheduling-Models. Methods and Opportunities, Technical Report DTU Management Engineering.

Vilhelmsen, C., Larsen, J., Lusby, R., 2016. A heuristic and hybrid method for the tank allocation problem in maritime bulk shipping. 4OR 14, 417–444.

Wang, X., Arnesen, M.J., Fagerholt, K., Gjestvang, M., Thun, K., 2018. A two-phase heuristic for an in-port ship routing problem with tank allocation. Comput. Oper. Res. 91, 37–47.