Optimal dual cycling operations in roll-on roll-off terminals

Jia, Beizhen; Tierney, Kevin; Reinhardt, Line Blander; Pahl, Julia

Published in:
Transportation Research Part E: Logistics and Transportation Review

DOI:
10.1016/j.tre.2022.102646

Publication date:
2022

Document Version
Early version, also known as pre-print

Citation for published version (APA):
Jia, B., Tierney, K., Reinhardt, L. B., & Pahl, J. (2022). Optimal dual cycling operations in roll-on roll-off terminals. *Transportation Research Part E: Logistics and Transportation Review*, *159*, Article 102646. https://doi.org/10.1016/j.tre.2022.102646

Optimal Dual Cycling Operations in Roll-on Roll-off Terminals

Beizhen Jia[a], Kevin Tierney[b], Line Blander Reinhardt[c], Julia Pahl[d]

[a] *Aalborg University, 2450 Copenhagen SV, Denmark*
[b] *Decision and Operation Technologies Group, Bielefeld University, 33615 Bielefeld, Germany*
[c] *Roskilde Universitet, 4000 Roskilde, Denmark*
[d] *University of Southern Denmark, 5230 Odense, Denmark*

Abstract

Roll-on roll-off (RoRo) shipping plays an important role in freight transport on the European continent, and is faced with the challenge of reducing its CO_2 emissions while increasing its efficiency. Dual cycling, in which loading and discharging processes are carried out simultaneously, achieves this goal by reducing the turnaround time of vessels in port and thus the CO_2 output of handling equipment in port and fuel consumption through slow steaming at sea. Optimizing the dual cycling operations on RoRo vessels has not yet been investigated in the literature. This paper presents the novel RoRo dual cycling problem (RRDCP), and formulates it using integer programming (IP) with the objective to minimize the total makespan of discharging and loading operations. We further prove that the RRDCP is NP-complete by a reduction from a general machine scheduling problem, and introduce a novel heuristic to solve the problem called a generalized random key algorithm (GRKA). We evaluate the IP model and GRKA approach on both generated and industrial instances, showing that the GRKA heuristic finds optimal or near-optimal solutions to real-world problems in just seconds. We provide managerial insights on industrial instances, which indicate that our approach leads to a reduction in fuel consumption and CO_2 emissions of up to 25% for RoRo operations.

Keywords: RoRo shipping, dual cycling, scheduling complexity

1. Introduction

Short sea *roll-on roll-off* (RoRo) shipping is a major transportation mode in Europe, especially in countries with long shorelines or many islands. In particular, in nations such as Italy, Denmark, Norway, Sweden, and Greece, the share of national seaborne transport is relatively high, ranging from 14% to 26% in 2019 (EUROSTAT, 2020; UNCTAD, 2019). RoRo shipping is commonly operated in liner shipping mode, i.e., according to fixed schedules where vessels transport a wide range of "rolling" general cargo, such as cars, trucks, heavy rolling machinery, trailers with or without an engine, etc. (see also Hansen et al., 2019). The ports in the European Union handling most RoRo cargo in 2018 include Immingham, UK, (30%), Genova, Italy, (19%), and London, UK, (15%).

RoRo shipping can substitute emission-intensive transport modes such as trucks (Chandra et al., 2016), making RoRo shipping a potential path to increased sustainability in the EU. An early study by Hjelle (2010) shows that transport by vessels may not be more emission efficient than trucks in places where the road capacity is abundant, the vessels are not filled to capacity and when distances by sea and by road are similar, but also show that the combination of vessel and truck often is the most viable in cases where distances can be shortened by sailing. Ng (2009) independently comes to similar conclusions when investigating the competitiveness of short sea shipping (SSS). More recent work by Christodoulou et al. (2019) show that the RoRo transport can be integrated with rail as well, thus ensuring sustainable intermodal transport chains.

Reduced turnaround times of vessels are both critical for vessel operators and for ports in terms of cost effectiveness (Pahl and Cordova, 2020; Si et al., 2018). This is achieved not only through streamlined administrative services, but also by effective planning for time-efficient discharging and loading operations (Nguyen and Kim, 2010). This translates into port efficiency (Strandenes, 2004; Pahl and Cordova, 2020), which permits vessels to sail slower during their sea voyages. Since the fuel consumption of vessels is roughly cubic in the vessel speed (Psaraftis and Kontovas, 2013), slower speeds can lead to significantly increased fuel efficiency and, thus, reduced CO_2-emissions. Moreover, more efficient terminals have increased availability and better utilization of yard space that allows for higher throughput rates and better overall competitiveness (UNCTAD, 2019).

This work focuses on unloading and loading RoRo ships in shuttle services in which the vessel is completely turned over at each port call, in contrast to, e.g., Puisa (2021). Loading and discharging RoRo cargo ships carrying trailers is often performed by tugs that are trailer heads operated by terminal staff. The number of tugs deployed for terminal operations and the sequence of discharging/loading of decks and trailers have a significant impact on the total amount of time a ship stays in the terminal, i.e., the turnaround time. It is currently common in the industry for loading operations to commence only after the vessel or a deck is fully discharged. In this paper, we model and solve the *RoRo dual cycling problem* (RRDCP). The RRDCP is concerned with discharging and loading trailers in a minimal amount of time from a RoRo vessel subject to precedence constraints on the discharging and loading order. To the best of our knowledge, this is the first time that a model has been formulated for dual cycling in RoRo shipping. To this end, our paper presents the following novel components.

- We provide an integer programming model of the RRDCP.

- We reduce a general machine scheduling problem to the RRDCP.
- We introduce a new metaheuristic paradigm that generalizes biased random key genetic algorithms (Gonçalves and Resende, 2011) and solve the RRDCP to near optimality in just a few seconds.

Our models and experimental analysis are based on real-world data from a European RoRo shipping company. Based on this data, we perform a managerial assessment of dual cycling using our technique and show that applying dual cycling operations on RoRo vessels can lead to a fuel consumption reduction.

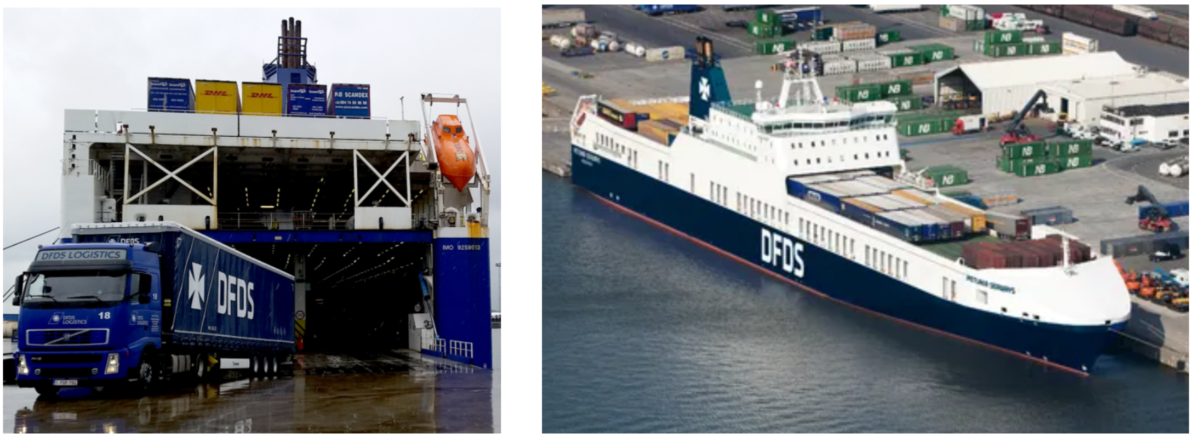
The rest of the paper is organized as follows. We provide a brief problem description in Section 2, followed by a review of related literature in Section 3. The RRDCP and IP model is defined and formulated in Section 4, followed by a discussion on its complexity from a scheduling point of view in Section 5. In Section 6 a random key heuristic is introduced and the results are presented in Section 7. We conclude the paper with a discussion and outlook in Section 8.

2. Problem description

RoRo shipping can encompass a wide variety of rolling cargo. In this work, we focus on unaccompanied trailers that are loaded and discharged using tugs as seen in Figure 1. A RoRo vessel is made up of multiple decks, with each deck having different height and weight restrictions (see also Fischer et al., 2016). Decks are divided into lanes that are marked with horizontal lines parallel to the length of the vessel. The main deck has one or more external ramps connected to the shore that serve as the bridge on to and off the vessel. Most RoRo vessels only have one external ramp connected to the main deck as this requires less port infrastructure for servicing vessels. Cargo needs to pass over the external ramp connected with the main deck and is transported to other decks through internal ramps. We note, however, that other configurations do exist (multiple ramps, etc.) and providing decision support in these cases may require algorithmic adjustments to the approaches we describe in this work.

In this paper, we focus on RoRo vessels with one external ramp that is as wide as the vessel, and provides multi-lane access from the shore, as shown in Figure 1. Once internal ramps between decks are free of cargo, the loading and discharging of the connected decks can take place simultaneously and independently. The waiting time of tugs at internal ramps is negligible as traversing the ramp costs very little time. Hence, it is sufficient to analyze each deck individually for planning loading and discharging operations. We further assume that stability while loading is ensured by the anti-heeling systems of the ship, as the precedence constraints prevalent in industry prevent a single side of the ship from being completely loaded before the other side. Moreover, the ship operator will not load the same side of multiple decks at the same time. We emphasize that the current planning processes of the industrial partner of this work do not consider stability as a concern during loading and discharging of their vessels, thus stability constraints are not included in this problem.

For arriving vessels, a loading plan is developed at the terminal. The loading plan indicates the positions on board the vessels that are to be filled with specific trailers and ensures the vessels' stability while underway. Loading and discharging operations are subject to precedence rules,



(a) A RoRo vessel is loaded from the rear with a trailer by a tug.

(b) Aerial view of the upper deck of a RoRo vessel.

Figure 1: RoRo vessels from the company DFDS.

which are based on safety requirements and the layout of a specific deck. For example, a trailer cannot be discharged before the trailer in front of it and the trailer in front of it to the starboard side are both discharged. However, when loading, trailer positions belonging to the lane farthest starboard will have the trailer in front of it and the one in front of it to the port side as their precedence constraints.

Figure 2 shows an upper deck (the deck above the main deck) of an example vessel. Tugs pulling trailers enter through the main deck from the quay through the rear of the vessel. The deck shown is loaded through an internal ramp; tugs drive on to the deck and begin loading the trailer slots starting at the rear of the vessel. In this example there is also a ramp continuing up to an upper deck, so this deck would be filled before loading the deck shown. Note that some vessels allow for ramps to different decks directly from the quay and the models in this work all support this. The rectangles represent positions that can be loaded with trailers. For instance, trailer position A can only be loaded if positions B and C have both been loaded. When discharging, position E can only be discharged when positions D and F no longer have trailers in them. In this example, a dual cycle could be performed as follows. A tug delivers a trailer to be loaded to position A, drives to position E and unloads the trailer there. Of course, this assumes the precedence constraints for both positions A and E are satisfied. Moreover, trailer positions cannot be loaded if the trailer already in that position has not yet been discharged.

The traditional way of handling unaccompanied cargo in short sea RoRo shipping is to deploy a number of *tugmasters* driving tugs, which are highly maneuverable trucks, to discharge each single trailer from the vessel to the yard and to load export trailers from the yard to the vessel once the overall discharging operation is finished. As a result, tugmasters travel empty 50% of the time which is highly inefficient in terms of utilization time and energy consumption. Figure 3 shows data from a case study with our industrial partner that quantifies the activity time for RoRo cargo operations at the terminal. The figure presents the discharge and loading times for 11 voyages collected from a leading RoRo shipping company. The average total operational time indicates

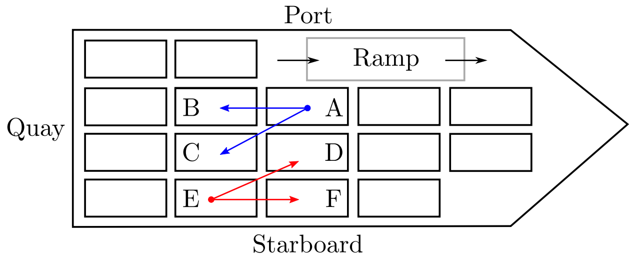


Figure 2: Simplified deck illustration showing precedence constraints for discharging (red) and loading (blue).

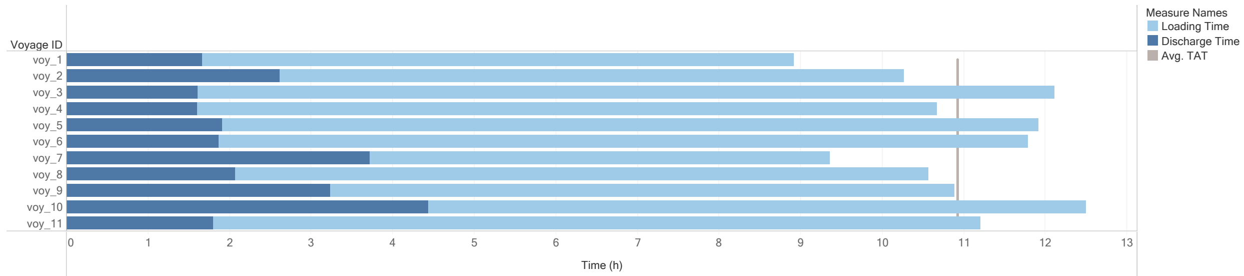


Figure 3: Turnaround time (TAT) at the terminal.

that almost 11 hours are spent on loading and discharging operations, where approximately 2 hours are used for discharging and 9 hours for loading operations. Note that the loading time is determined by several components, including the working efficiency, the imbalance in number of trailers to be discharged and loaded, whether cargo arrives at the terminal on time, and the break times required by the labor union.

Reducing the number of empty tug trips has a direct impact on reducing vessel turnaround times. This can be achieved by dual cycling. In dual cycling, after loading a trailer, tugs drive to a trailer that should be discharged and remove it from the vessel. Figure 4 illustrates the difference between two cargo handling strategies, with Figures 4(a) and 4(b) displaying the options for trailer handling without and with dual cycling, respectively. Although the advantages of dual cycling are obvious, it is not possible to perform it with every discharge and load operation. RoRo vessels must first be emptied sufficiently to start dual cycling. That is, the precedence constraints must be satisfied. In most cases, this means dual cycling can begin once a trailer position is unconstrained by both the discharge and loading precedence constraints. The starting time of dual cycling thus heavily depends on the size and structure of the vessel decks where trailers are located.

A final assumption of the problem we consider is that the parameters are deterministic. We assume the planning of trailer locations on the ship and quay has been completed before operations start, and that there are no online adjustments necessary. In particular, the loading and unloading times are clearly stochastic parameters. However, trying to forecast these values is very difficult and there is little we can do to handle this uncertainty in planning. Since the solution procedures we provide are extremely fast, usually requiring only several seconds per deck, recourse actions can be performed in real time.

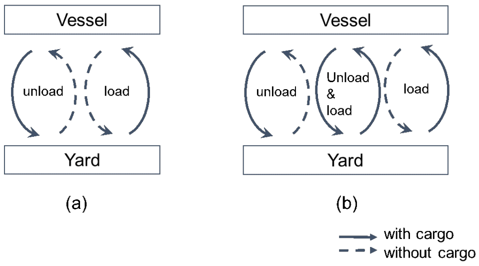


Figure 4: Comparison between a traditional cargo operation strategy (a) and a dual cycling strategy (b).

3. Literature review

Within seaborne transport, the area of liner shipping has been subject to a significant amount of research. In regard to discharging and loading vessels, the literature has mainly focused on minimizing container handling times, driving distances of port material handling equipment, as well as reducing empty moves of various kinds of container terminal equipment (Nguyen and Kim, 2010). Recently, interest in RoRo operations has grown, but nonetheless, only a limited number of areas within the RoRo field has been examined. For instance, the area of RoRo stowage is investigated regarding both optimizing stability, energy efficiency, and safety (Jia et al., 2020; Jia and Fagerholt, 2021), as well as packing and shifting costs (Hansen et al., 2016). Moreover, Jia et al. (2019) analyze and estimate the discharge time of cargo units on a RoRo vessel based on different loading positions. Besides, there is research dealing with routing and scheduling of RoRo vessels (Øvstebø et al., 2011) as well as analysis of effects of sulphur emission limits on RoRo operations provided by Zis and Psaraftis (2017).

Dual cycling has primarily been developed and applied in the area of container handling. For instance, the quay crane double cycling problem (QCDCP) at container terminals has been investigated previously by many researchers who typically concentrate on the optimization of a single quay crane. In this regard, the first academic paper addressing QCDCP is provided by Goodchild and Daganzo (2006). The problem solved concerns the optimal load ordering of container stacks assuming non-preemptive stack operations. In their study, a simple scenario is investigated and formulated as a two-machine flow shop scheduling problem. They argue that their problem can be solved by Johnson's rule (Johnson, 1954). Although Goodchild and Daganzo (2006) consider both loading and discharging operation as unit time operations, they do not consider precedence relations in their problem, but only take into account one crane operating on a single row of a vessel.

The model from Goodchild and Daganzo (2006) has been extended in a number of ways. Song and Kwak (2009) find a general rule for the optimal starting point of dual cycling for a QCDCP for a single crane to avoid delays to the start of dual cycling due to containers blocking each other. Zhang and Kim (2009) formulate a MIP model and develop a local search-based heuristic for the general QCDCP also considering hatch covers. Ku and Arthanari (2016) review the weaknesses of a selection of existing models for the multi-QCDCP and call attention to real-world requirements that need to be included as constraints in models to enhance their applicability. In Zheng et al. (2020), the problem considered by Goodchild and Daganzo (2006) is transformed to a problem with stack-wise precedence constraints. The problem with stack-wise precedence can be converted into m precedence chains, where m is the number of stacks and each container has at most one predecessor and at most one successor. Zheng et al. (2020) develop a polynomial time algorithm for this single crane case that shares similarities with a simple lane case of the problem considered here with only a single tug operating. With the same concept for the purpose of improving loading and unloading efficiency, double cycling in RoRo shipping requires researchers' attention. One could relate the problem to a multi-QCDCP with empty crane moves and more complex precedence constraints not restricting operations to one bay or stack. Thus the two problems are very different in essence and, to the best of our knowledge, the complexity of the RoRo dual cycling problem is unknown.

In addition to the problem-specific focus of dual cycling research for quay cranes, its relation to machine scheduling has been investigated. Goodchild and Daganzo (2006) point out that the dual cycling problem is a type of machine scheduling problem; see, e.g., Prot and Bellenguez-Morineau (2018) for an overview. This provides us with inspiration for reducing machine scheduling to the RRDCP which we discuss further in Section 5 on the complexity of RRDCP.

Finally, we note that there has been work on the trajectories for loading RoRo ships, as well as on stowage. Man et al. (2020) investigates methods for calculating the best and safest trajectories for loading large (out of gauge) cargo on RoRo ships without collision. In the same realm, Oucheikh et al. (2021) proposes a deep reinforcement learning approach where artificial intelligence agents represent tug masters. The agents learn to navigate the environment of loading and unloading processes in a 3D-learning framework aiming at avoiding collisions with static as well as dynamic obstacles. In terms of stowage, Chen et al. (2021) determine a loading sequence for cars, and Bayliss et al. (2021) solves a stochastic vehicle loading problem for ferries.

4. RRDCP Statement and Mathematical Formulation

We move from our general description of the short-sea RoRo problem in Section 2 to describe the RRDCP in more detail and formulate it mathematically. We first discuss assumptions necessary for modeling the RRDCP in terms of the spatial layout of the vessel and the precedence constraints, followed by an IP model.

4.1. RRDCP Assumptions

A deck is divided into slots that fit standardized trailers in terms of their size. We assume the trailers are homogeneous in size, but note that in reality the lengths and heights may vary. Irregular-sized cargo can be treated as taking multiple slots and easily incorporated by altering the precedence matrix. In addition, we consider only unaccompanied trailers since self-driving units do not require extra labor.

The handling of unaccompanied trailers is performed by a tug driver with a tug. The related tug operations can be divided into five types of actions. A tug driver at one stage can either:

1. drive from the quay to the vessel with a trailer (loading),

2. drive from the quay to the vessel without a trailer,

3. drive from the vessel to the quay with a trailer (discharging),
4. drive from the vessel to the quay without a trailer, or
5. stay idle on quay.

Each action requires a single unit of time. The reason for this is that it greatly simplifies the checking of precedence constraints and the overall modeling of the problem, as we do not need to consider time as a continuous property. Furthermore, some trailers will take more or less time to load or discharge than others, and it is difficult to predict since the time varies from trailer to trailer depending on the driver, traffic on board and in the terminal, weather conditions, trailer location on board, etc. We abstract from various factors in this article, but discuss the applicability of the model further in Section 8.

We assume that tugs do not interfere with each other while pulling trailers to or from the vessel, i.e., adjacent trailers may be loaded or removed within a single time unit. All tugs start on the quay and must drive on to the vessel. In reality, tugs are assigned to specific decks, and once a deck is emptied, the tug can be assigned to another deck. However, since we model decks independent from each other, we assume a fixed number of tugs over the planning horizon of a deck. Note that tugs are not allowed to idle on the vessel and the number of tugs that can service the vessel is limited and fixed throughout the overall discharging/loading operation. We further note that there is a maximum number of tugs allowed on a deck at any given time to prevent collisions and the safety of the vessel. We abstract from pauses for drivers as union regulations are different from port to port.

The objective of the RRDCP is to minimize the total operational time of loading and discharging a vessel, i.e., makespan minimization subject to precedence rules. The precedence rules are described with a precedence matrix $\mathcal{P}$ for discharging and loading of the form $(i, i') \in \mathcal{P}$ and $(j, j') \in \mathcal{P}$, respectively. That is, trailer i must be discharged before i' can be discharged, and trailer j must be loaded before j' can be loaded. Furthermore, a trailer cannot be loaded into its designated position on the vessel until the trailer in that slot is discharged. However, the discharge and subsequent loading of a trailer in the same designated position may take place in the same time step. Note that this is possible through the use of two tugs; one to first discharge a trailer and one to deliver a new trailer immediately following the discharge.

4.2. Mathematical Formulation

Sets and Parameters

$\mathcal{S}$	Set of trailer positions to be discharged from vessel to quay, indexed by i
$\mathcal{Q}$	Set of trailer positions to be loaded from quay to vessel, indexed by j
$\mathcal{T}$	Set of time steps, $\mathcal{T} = \{1, \ldots, \lvert\mathcal{S}\rvert + \lvert\mathcal{Q}\rvert\}$, indexed by t
$\mathcal{P}$	Set of trailer handling precedence pairs, as described above
k	Total number of tugs

Variables

x_{it}	Equals one if trailer (position) $i \in \mathcal{S}$ is discharged in time step $t \in \mathcal{T}$ and zero otherwise
y_{jt}	Equals one if trailer (position) $j \in \mathcal{Q}$ is loaded in time step $t \in \mathcal{T}$ and zero otherwise
wsq_t, wqs_t	Number of tugs traveling from/to vessel to/from quay without a trailer in time step $t \in \mathcal{T}$, respectively
wqq_t	Number of tugs idling on the quay in time step $t \in \mathcal{T}$
u	Makespan

Objective and Constraints

$$\min z = u \tag{1}$$

s.t.

$$u \geq ty_{jt} \qquad \forall j \in \mathcal{Q}, t \in \mathcal{T} \tag{2}$$

$$u \geq tx_{it} \qquad \forall i \in \mathcal{S}, t \in \mathcal{T} \tag{3}$$

$$u \geq (|\mathcal{Q}| + |\mathcal{S}|)/k \tag{4}$$

$$u \leq 2(|\mathcal{Q}| + |\mathcal{S}|) \tag{5}$$

$$\sum_{t\in\mathcal{T}} x_{it} = 1 \qquad \forall i \in \mathcal{S} \tag{6}$$

$$\sum_{t\in\mathcal{T}} y_{jt} = 1 \qquad \forall j \in \mathcal{Q} \tag{7}$$

$$\sum_{t\in\mathcal{T}} tx_{it} \leq \sum_{t\in\mathcal{T}} (t-1)x_{i't} \qquad \forall (i,i') \in \mathcal{P} \tag{8}$$

$$\sum_{t\in\mathcal{T}} ty_{jt} \leq \sum_{t\in\mathcal{T}} (t-1)y_{j't} \qquad \forall (j,j') \in \mathcal{P} \tag{9}$$

$$\sum_{t\in\mathcal{T}} tx_{it} \leq \sum_{t\in\mathcal{T}} ty_{it} \qquad \forall i \in \mathcal{S} \tag{10}$$

$$\sum_{i\in\mathcal{S}} x_{it} + wsq_t = \sum_{j\in\mathcal{Q}} y_{jt-1} + wqs_{t-1} \qquad \forall t = 2, \ldots, |\mathcal{T}| \tag{11}$$

$$\sum_{i\in\mathcal{S}} x_{it} + wsq_t + \sum_{j\in\mathcal{Q}} y_{jt} + wqs_t + wqq_t = k \qquad \forall t = 1, \ldots, |\mathcal{T}| \tag{12}$$

$$\sum_{i\in\mathcal{S}} x_{it} \leq k/2 \qquad \forall t = 1, \ldots, |\mathcal{T}| \tag{13}$$

$$\sum_{j\in\mathcal{Q}} y_{jt} \leq k/2 \qquad \forall t = 1, \ldots, |\mathcal{T}| \tag{14}$$

$$\sum_{j\in\mathcal{Q}} y_{j1} + wqs_1 + wqq_1 = k \tag{15}$$

$$x_{i1} = 0 \qquad \forall i \in \mathcal{S} \tag{16}$$

$$wsq_1 = 0 \tag{17}$$

$$x_{it}, y_{jt} \in \{0, 1\} \tag{18}$$

$$wsq_t, wqs_t, wqq_t, u \in \mathbb{Z}_{\geq 0} \tag{19}$$

The objective (1) is to minimize the total operational time of the loading and discharging of a vessel, i.e., the makespan (u). It is defined as the latest time in which either a loading or discharging operation occurs, as shown in constraints (2) and (3). Note that tx_{it} and ty_{jt} equal the time when the trailer i or j is discharged or loaded, respectively, and zero otherwise. Furthermore, constraint (4) provides a lower bound on the makespan if all tugs (k) were able to perform dual cycling on all tasks (quay positions $\mathcal{Q}$ and ship positions $\mathcal{S}$). We impose an upper bound on the makespan through constraint (5), which is the time used for single cycling with one tug. Constraints (6) and (7) make sure that each trailer position is discharged and loaded only once, respectively. Constraints (8) to (10) enforce the precedence restrictions between loading-loading, loading-discharging, and discharging-discharging operations, respectively. Specifically, for any discharging precedence pair $(i, i') \in \mathcal{P}$, constraints (8) ensure that the time step in which i is discharged must be strictly before the time step in which i' is discharged. The same logic of constraining the loading sequence is applied in constraints (9). Constraints (10) make sure that a trailer position cannot be loaded with cargo before it is emptied from discharging. However, this can happen within the same time step given the consideration that the tug performing the discharging task is available onboard and ready to unload cargo by the end of last time step. Alternative formulations of the precedence constraints have been considered, but we note that this one has the best performance. Constraints (11) and (12) link tug movement with the loading and discharging operations. Constraints (11) state that either with cargo or empty, the total number of tugs traveling from ship to quay at this time step (t) should equal to the total number of tugs traveling from quay to ship at last time step ($t-1$). Constraints (12) state that the total number of tugs performing loading and discharging tasks, traveling between ship and quay empty, and waiting at quay should equal to the number of tugs deployed (k). The maximum number of tugs on board at each time step, usually set as half of the total number of working tugs, is enforced by constraints (13) and (14). We set the number of tugs to $k/2$ as dividing the tugs equally to enter and exist the ship ensures there are no wasted time steps at the end of the plan. Finally, we assign starting values to the variables in the first time step in constraints (15), (16) and (17). Note that y_{j1} being zero is implied if no loading is possible for dual cycling from the beginning.

Discharging and loading feasibility. We now identify the conditions for the precedence graph under which all positions on the vessel can be discharged and loaded. First, $\mathcal{P}$ must be directed acylic. Second, we need to ensure that no positions on the vessel are "forgotten" and could end up blocking a portion of the vessel once they are loaded.

For discharging, let $\mathcal{S}' = \{i' \mid \nexists (i, i') \in \mathcal{P}, i', i \in \mathcal{S}\}$ be the set of positions that will be discharged first, i.e., no positions require the positions in $\mathcal{S}'$ to be discharged before they can be discharged. The vessel can be fully discharged without blockage if all positions $\mathcal{S} \setminus \mathcal{S}'$ are *reachable* from the positions $\mathcal{S}'$, i.e., there is a path in $\mathcal{P}$ connecting positions that will be discharged first to those that will be discharged last. An equivalent argument can be made for loading, with $\mathcal{Q}' = \{j \mid \nexists (j, j') \in \mathcal{P}, j', j \in \mathcal{Q}\}$ and all positions $\mathcal{Q} \setminus \mathcal{Q}'$ must be reachable from $\mathcal{Q}'$ in $\mathcal{P}$.

5. Complexity of the RRDCP

As mentioned previously, to the best of our knowledge there is no study of the complexity of the RRDCP. In the following, we show that the decision version of the RRDCP is NP-complete by reduction from a general scheduling problem generally denoted by $P|prec, p_j = 1|C_{max}$ (Lawler et al., 1993). To do this, we first formally define the RRDCP as a scheduling problem. We are given the following sets and parameters:

(a) A set $\mathcal{J}$ of loading and discharging tasks that must be completed $\mathcal{J} = \{j_1, \ldots, j_h, j_{h+1}, \ldots j_n\}$, where h is the number of loading tasks and $n-h$ is the number of discharging tasks with the additional sets $\mathcal{J}_l = \{j_1, \ldots, j_h\}$ and $\mathcal{J}_d = \{j_{h+1}, \ldots, j_n\}$,

(b) a partial order $\prec$ on $\mathcal{J}$,

(c) a unit time duration for each loading task and discharge task (including the transportation of the cargo from port to vessel and from vessel to port, respectively), denoted by $W(j_i) = 1$,

(d) a number of tugs that each can complete at most one task j_i at a time t.

(e) a change cost $C_{lm} \in \{0, 1\}$ in which a tug performing a loading task j_l followed by a loading task $j_m, l, m \leq h$ must use one unit time $C_{lm} = 1$ to drive empty from the vessel to the port between the two tasks. Furthermore, a tug performing a discharge task j_p followed by a discharge task j_q must also use one unit time $C_{pq} = 1$ to drive empty from the port to the vessel between the two tasks. This is also called a sequence dependent setup time.

Given these sets and parameters, we can now describe the RRDCP as follows:
(D1): *General dual-cycling decision problem.* Given (a),(b),(c),(d), (e), and a completion time t does a total function $f(j) \longrightarrow \{0, ...t-1\}, \forall j \in \mathcal{J}$, exist such that:

(i) if $j \prec j'$, then $f(j) + W(j) \leq f(j')$ (precedence constraints are satisfied),

(ii) $\forall j \in J, f(j) + W(j) \leq t$ (all tasks are completed before the time t),

(iii) for $0 \leq i < t$, there are at most k elements in $\mathcal{J}$ for which it holds that $f(j) \leq i < f(j) + W(j)$ (ensure at most k tugs are used), and

(iv) if $f(j) + W(j) = f(j')$ then ($j \in \mathcal{J}_l$ and $j' \in \mathcal{J}_d$) or ($j \in \mathcal{J}_d$ and $j' \in \mathcal{J}_l$).

Lenstra and Rinnooy Kan (1978) determine the complexity of different scheduling problems with precedence constraints, although none of problems consider sequence dependent setup times included in (D1). Minimizing the makespan is denoted as C_{max} by Lenstra and Rinnooy Kan (1978), where in this case C_{max} is greater than the completion time of any job j. Thus $C_{max} \leq f(j) + W(j)$ for all $j \in \mathcal{J}$.

The aim of the dual cycling problem is to find the minimum completion time of the last tug. To show that the dual-cycling problem is NP-complete, we consider the scheduling problem. This is done here by considering the scheduling problem defined by Ullman (1975) and later denoted $P|prec, p_j = 1|C_{max}$ by Lawler et al. (1993). To prove that (D1) is NP complete, we make a reduction to the decision scheduling problem which in Ullman (1975) is defined as follows:

We are given the following sets and parameters:

(a) a set $\mathcal{J} = \{j_1, \ldots, j_n\}$ of jobs.

(b) a partial order $\prec$ on $\mathcal{J}$.

(c) a weight function $W(j_i) = 1$.

(d) a number of processors k is represented by the number of tugs used.

(S1): *The scheduling decision problem.* Given (a),(b),(c) and (d), does a total function $f(j) \longrightarrow \{0, ...t-1\}, \forall j \in J$ exist?

Lawler et al. (1993) prove that (S1) is NP-complete by reduction from the clique problem.

The reduction from (S1) to (D1) is straightforward. We simply set $C_{lm} = 0$ for all l and m thus making (e) from (D1) redundant. It is also straight forward to see that a solution to this problem would also present a solution to (S1) and that the translation between the problems can be done in linear time. Furthermore, (D1) is clearly in NP, as a solution to D1 can be trivially checked for correctness in polynomial time. Thus we have shown that (D1) is NP-complete when the number of tugs are provided as input. However, if the number of tugs to use is considered as a decision variable, the problem is still open as noted by Prot and Bellenguez-Morineau (2018). As with all NP-complete problems, special cases of the problem can be polynomial time solvable and as mentioned by Ullman (1975), polynomial time algorithms might exist for specific numbers of tugs. Such polynomial time algorithms have been proven to exist for problems with two machines (Ullman, 1975). Moreover, special graph structures may also lead to polynomial time algorithms (Hu, 1961).

In conclusion, an m parallel machine scheduling problem for unit time jobs with arbitrary precedence constraints is NP-complete. Therefore, the RRDCP is NP-complete. However, for cases satisfying conditions of a bounded path and limited degree in the precedence graph, polynomial time algorithms exist (Aho and Mäkinen, 2006). In RRDCP problems the degree is often limited to between two and eight. However, bounding the path length can be more challenging.

6. A Random-Key Heuristic for the RRDCP

Our heuristic approach to solving the RRDCP is based on the fact that constructing a feasible solution is possible in linear time. Indeed, the RRDCP is naturally oriented towards construction methods, as it is not clear how to define a neighborhood for local search approaches that would not break a sequence of load and discharge movements. Our heuristic is a generalization of the biased random-key genetic algorithm (BRKGA) metaheuristic (see Gonçalves and Resende, 2011), in which we replace the genetic algorithm with a different metaheuristic method. Replacing the GA with other continuous search paradigms allows us to achieve better performance and leaves flexibility to the user as to how to solve a problem. The RRDCP is well-suited to random-key approaches since it is essentially an ordering problem, and random-key approaches are highly effective at learning orders of operations (Gonçalves et al., 2011; Soares and Carvalho, 2020). We describe a heuristic approach, which we call a generalized random-key algorithm (GRKA), for solving the RRDCP. After describing the GRKA framework, we discuss the RRDCP-specific *decoder* required by the GRKA, which is a parameterized, greedy construction heuristic. Finally, we present some complexity results regarding the decoder indicating that not all RRDCP problems are actually hard.

6.1. GRKA

The GRKA consists of two components, as shown in Figure 5: an unconstrained, continuous optimizer (despite the name, unconstrained, continuous optimizers support lower and upper bounds on the variable domains) and a decoder. The unconstrained optimizer must be able to solve a continuous optimization problem in which the objective function is given by a black box (i.e., there is no derivative) and the input is constrained to within the hypercube $[0, 1]^n$. Numerous algorithms exist for this task, such as particle swarm optimization (PSO) (Kennedy and Eberhart,

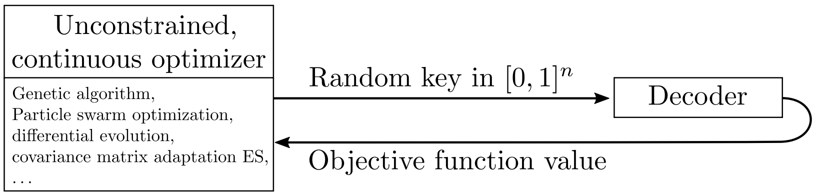


Figure 5: Overview of the GRKA Method.

1995), differential evolution (DE) (Price, 2013), covariance matrix adaptation evolutionary strategies (CMA-ES) (Hansen, 2006), or the GA used in the BRKGA (Gonçalves and Resende, 2011).

The central component of the GRKA is the *random key*, which is a vector of values in the range [0, 1]. The random key connects the problem-independent optimizer to the problem-specific *decoder*, in that the optimizer queries the value of the objective function from the decoder for a given random key. That is, the random key is directly modified by the continuous optimizer (CMA-ES, PSO, etc.). The decoder is a parameterized construction heuristic, and the only set of parameters it accepts is the random key. The decoder must use the information given in the random key to create a solution to the problem. For example, in the traveling salesperson problem, each node could be assigned to an entry in the random key and the decoder would build a tour by sorting the nodes by the value of their random key entry and adding them to the tour in that order. The optimizer can thus influence the tour's construction by raising or lowering the random key entry of each node.

A key advantage of random-key heuristics over other types of metaheuristics is that they make it possible to solve discrete, constrained optimization problems with continuous optimizers in an unconstrained setting. A further advantage is that the problem modeler does not need to think about metaheuristic details and can instead focus on specifics of the problem at hand. A disadvantage is that delta evaluation (i.e., incremental evaluation) of the objective function is not possible, meaning fewer solutions can be explored than in local search techniques.

6.2. RRDCP Decoder

We design an RRDCP decoder that accepts a random key with a length equal to the number of trailers to discharge and load. The decoder works by iteratively checking whether there are any trailers that can be discharged or loaded without violating the precedence constraints, and loading/discharging as many as possible according to the position of the tugs. The decoder prefers trailers with a lower random key entry if there are more trailers to load/discharge than tugs available. The available tugs are divided into two groups and it is assumed that these two groups switch positions on the vessel in each time step, i.e., one group drives on the vessel and the other one drives off. To simplify the algorithm, we assume the total number of tugs k is even, however the algorithm works for any number of tugs.

Algorithm 1 accepts the random key, R and the parameters S, Q, P, k as defined in the mathematical model. First, the random key is adjusted to the range $[0, 1)$ so that it can be used for sorting in the following step. Next, all trailers to be loaded and discharged are inserted into the priority queues D^S and D^Q, respectively, with the priority of each entry equal to the number of descendants in the precedence graph P plus the value of the random key. Note that ties in the random key are unlikely due to the precision of the floating point numbers, but should they occur, they can be broken arbitrarily.

Algorithm 1 Greedy RRDCP Decoder.

```
1: function RRDCP-DECODER(R, S, Q, P, k)
2:     Remap R to the range [0, 1)
3:     D^S ← S, sorted by |{(i, i′) ∈ P | i ∈ S}| + R(i) (ascending)
4:     D^Q ← Q, sorted by |{(j′, j) ∈ P | j ∈ Q}| + R(i) (ascending)
5:     t ← 1                                                          ▷ Time counter
6:     while |D^S| > 0 or |D^Q| > 0 do
7:         U ← Pop the top min{|D^S|, k/2} dischargeable trailers from D^S   ▷ Unload step
8:         Update D^Q and D^S for all u ∈ U
9:         L ← Pop the top loadable min{|D^Q|, k/2} trailers from D^Q        ▷ Load step
10:        Update D^Q for all l ∈ L
11:        t ← t + 1
12:    return t
```

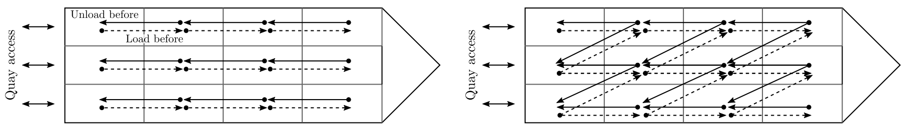
The main loop of the construction approach begins on line 6, which iterates until all trailers are loaded or discharged. On the following line, we pop the top $k/2$ *dischargeable* trailers, or however many are left, from the priority queue and insert them into U for processing. A trailer can be discharged if all trailers preceding it in P are already discharged (i.e., its priority value in the priority queue is less than one). With the trailers in U discharged, we can now update the priority queues, thus decrementing the priority value of any trailer connected to a trailer in U in the precedence graph.

Having determined which trailers will be discharged, we now check whether we can perform any loading operations on line 9. As with discharging, we only load trailers that are ready to be loaded according to the precedence graph. The construction procedure continues until every trailer is loaded and discharged, incrementing the time at the end of each iteration, and returning the time t as the objective function value.

6.3. Special Cases and Lower Bounds

In this section, we investigate a few special cases of the RRDCP in which a simple, polynomial time algorithm can solve the RRDCP optimally, as well as lower bounds on the number of time steps. While the cases we show on their own are not necessarily realistic, many real problems could contain these cases as subproblems once some trailers are loaded and discharged. We focus in particular on two cases of the RRDCP in which we assume the vessel has m *lanes*, each with a capacity of n trailers with $n \geq m$, and that there are an even number $k \leq m$ of tugs available. Assume the vessel is accessed at the beginning of each lane. With an even number of tugs, we always cycle half of the tugs on/off the vessel in each time step as in the decoder. This means the first step is always to drive $k/2$ tugs on to the vessel and let $k/2$ tugs idle at the quay. We ignore this step in our calculations of the number of steps below.

Figure 6 shows the two cases we examine in this subsection. In the *simple lanes* setting, as shown in Figure 6a, the precedence graph only contains arcs within a single lane between adjacent trailers. The precedence constraints in simple lanes just ensure that lanes are discharged in the order of the trailers in the lane, and loaded in reverse order and each lane's precedence graph is disjoint from other lanes. Figure 6b shows the *adjacent lane* setting, in which to discharge a trailer, both the trailer preceding it in the lane and the preceding trailer in the lane immediately to the left must be removed first. This precedence constraint structure is rather realistic, however we note real instances have varying lane lengths and obstacles, which we do not consider here.



(a) Simple lanes. (b) Adjacent lane precedence.

Figure 6: Simplified, but still realistic RRDCPs.

We first introduce two propositions related to when dual cycling can first begin, and how many loading steps still must be performed after dual cycling ends.

Proposition 1. *There are at least $n-1$ time steps of discharging before dual cycling begins.*

Proposition 2. *There are at least $n-1$ loading after the last dual cycle occurs.*

Both propositions follow from the lane length n: dual cycling can only start once an entire lane is empty, and the last lane is loaded when there is nothing left to discharge. Note that in both cases we have $n-1$ and not n since dual cycling begins/ends with the last slot in a lane.

6.3.1. Simple Lanes

We now prove a lower bound on the number of time steps necessary for the simple lanes case, and show that for even k where $m \mod (k/2) = 0$, the lower bound represents the number of steps in the optimal solution. Let T_{mnk} be the minimum number of time steps required to fully discharge and load a vessel in the simple lanes case with k tugs.

Theorem 1. *For simple lanes with even k, $n \geq m$, $T_{nmk} \geq n + \left\lceil \frac{nm}{k/2} \right\rceil - 1$*

Proof. Proof From Proposition 1 there are at least $n-1$ time steps before any dual cycling can occur. Clearly it is not possible to perform better than discharging the remaining $mn-n-1$ slots during dual cycling and thus while loading. There are nm slots remaining to be loaded. Since we can load at most $k/2$ trailers at each time step, it will take at least $\frac{nm}{k/2}$ time steps until all trailers are loaded. It is not possible to perform a fractional time step, therefore we must round this value up and get $\lceil \frac{nm}{k/2} \rceil$. Combining the first $n-1$ discharges with the loading while discharging the remaining, we get $T_{nmk} \geq n + \left\lceil \frac{nm}{k/2} \right\rceil - 1$. □ □

Theorem 2. *Let k be even and let $m \mod (k/2) = 0$ then for simple lanes, $T_{nmk} = n + \frac{nm}{k/2} - 1$*

Proof. Proof Since $k < m$ it is clearly possible to discharge the first $k/2$ lanes in parallel. This takes $n-1$ steps (Proposition 1), dual cycling then begins by discharging the next $k/2$ lanes while loading the first $k/2$. This pattern repeats until there are no more lanes to discharge. The final $k/2$ lanes can be loaded in parallel. Since we load $k/2$ trailers in every time step starting from time step $n-1$ until all trailers are loaded, we require $\frac{nm}{k/2}$ time steps until they are all loaded. Thus, $T_{nmk} = n + \frac{nm}{k/2} - 1$. □ □

We now have an algorithm that provides an optimal number of time steps when m is divisible by $k/2$. When m is not divisible by $k/2$, we note that the optimal solution is generally only a few time steps away from the bound shown above, but we do not prove these case

6.3.2. Adjacent Lanes

In the adjacent lane case, we must consider the discharge and load order imposed by neighboring lanes. The trailers are discharged and subsequently loaded in a pattern that can be best described as a triangle. We note that our bound is not tight; there are many cases where an extra time step or two are necessary because not all $k/2$ tugs can be used in every time step. We leave this as an open problem, but note that we do not believe finding the optimal solution is computationally difficult on these problems.

Theorem 3. *For adjacent lanes with even k, $n \geq m$, $T_{nmk} \geq \frac{m(2n-m+1)}{k} + \left\lceil \frac{nm}{k/2} \right\rceil - 1$.*

Proof. Proof The lower bound on the number of moves in the adjacent lane case has two terms representing discharging and loading. As in our previous proofs, we need to know how many discharges are performed until we begin double cycling. To discharge the last trailer in the first lane, we must discharge both the "triangle" of trailers m rows above it, and the $n-m$ by m rectangle of trailers. Mathematically, there are $m(n-m)$ trailers in the top rectangle, and $\frac{m(m+1)}{2}$ trailers extending up from the last slot in the first lane. We divide this numbers of trailers through by our available tugs and get $\frac{m(n-m)}{k/2} + \frac{\frac{m(m+1)}{2}}{k/2}$. In the same time step as the last discharge of the first lane we can begin dual cycling. There are nm trailers to load, meaning we need at least $\lceil \frac{nm}{k/2} \rceil$ time steps. Putting all these terms together, and subtracting one since they overlap by one time step for dual cycling, we get the following, which we can simplify into the following inequality: □

$$T_{nmk} \geq \frac{m(n-m)}{k/2} + \frac{\frac{m(m+1)}{2}}{k/2} + \left\lceil \frac{nm}{k/2} \right\rceil - 1 = \frac{m(2n-m+1)}{k} + \left\lceil \frac{nm}{k/2} \right\rceil - 1$$

□

7. Computational Results

In this section, we report the computational results of the GRKA compared with standard solver solutions for the RRDCP. We first describe the test instances and their precedence rules, followed by computational results on various combinations of deck layouts and trailer precedence. We also address the magnitude of benefits from conducting dual cycling operations in real life using empirical data at the end of the section. The mathematical model is coded in the Julia Language with JuMP (Dunning et al., 2017), solved in Gurobi 9.0 (Gurobi Optimization, Inc., 2020) with default settings. The GRKA is coded in Python 3. All tests are run on machines with dual 16 core Intel Xeon E5-2670 processors at 2.6 GHz with 64 GB of memory in total.

7.1. Test Instances

Vessels can have vastly different configurations for the stowage of trailers, thus we design a benchmark of 40 instances to consider varying sizes and properties of the vessels. The benchmark of instances, the data generator, and our heuristic solution procedure will be made available on github upon publication of this work. The 40 test instances are generated with a lane structure through a combination of ten different deck layouts and four sets of precedence rules. Deck layouts are randomly generated of various sizes from 67 trailers to 113 trailers. These are based on two classes of deck layouts, denoted by M and L, where M represents a deck layouts with a maximum capacity of 79 trailers in 13 rows in the longest lane, and a deck width of 7 lanes at the widest row. The class L represents deck layouts with a maximum capacity of 113 trailers in 15 rows in the longest lane and 9 lanes on the widest row. We assume the ramp for accessing the trailers is at the back of the vessel, i.e., the stern, although we note that on our industrial instances access from the midship and the bow is also realistic. We categorize each instance as $\{M, L\}$-B-$\{N, S, O\}$, where B is the number of blocked slots, e.g., due to support pillars, and the final parameter $\{N, S, O\}$ describes whether there are no blocked slots (N), blocked slots in the shape of a square (S), or a blocked slot structure difficult to describe (O).

We provide four sets of general rules set up for testing regarding the structure of the precedence constraints based on a general understanding of the discharging and loading operations. Table 1 provides the rules. Note that the rules do not apply to the first row for loading and the last row for discharging since nothing blocks them from being loaded or discharged, respectively. Moreover, regarding rules mp, ms and mps, for trailers located at the most port or starboard side, the blocking slots are the front and the front to a feasible side, to ensure a safety space of two slots in front of the trailer being loaded or discharged. An example of the precedence relations based on rule ms is illustrated in Figure 7. As stated above for rule ms, additional clearance is needed from the starboard side of trailers. Therefore, for trailers loaded at slots 5-7, the preceding slots are (9,10), (10,11) and (11,12), respectively. As for the trailer on the most starboard side at slot 8, the blocking slots are 12 and 11.

In addition to our generated instances, we also collect instances based on a short sea RoRo vessel operating from Vlaardingen, Netherlands to Immingham, United Kingdom. The vessel contains four decks (from bottom to top): the lower hold (lh), main deck (md), upper deck (ud), and weather deck (wd), each with different capacities, layouts and ramp positions. Each industrial

Table 1: Precedence of Slot (r,c) in Various Precedence Rules.

Rule	discharging	Loading	Precedence.
m	(r+1,c)	(r-1,c)	Front slot.
mp	(r+1,c),(r+1,c-1) or (r+1,c+1)	(r-1,c),(r-1,c+1) or (r-1)(c-1)	Front and its immediate port slots.
ms	(r+1,c),(r+1,c+1) or (r+1,c-1)	(r-1,c),(r-1,c-1) or (r-1,c+1)	Front and its immediate starboard slots.
mps	(r+1,c),(r+1,c-1) and/or (r+1,c+1)	(r-1,c),(r-1,c+1) and/or (r-1,c-1)	Front, its immediate port and starboard slots.

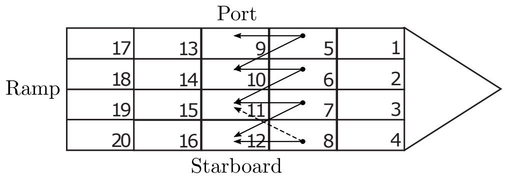


Figure 7: Example of discharging with rule ms.

instance is denoted by *deck name-capacity-ramp location*. Precedence constraints are developed based on a combination of rules illustrated in Table 1, with additional inputs from the stowage planner at the company. We assume full loading and discharge for both generated and industrial instances. All instances are solved with four tugs.

7.2. Computational Experiments

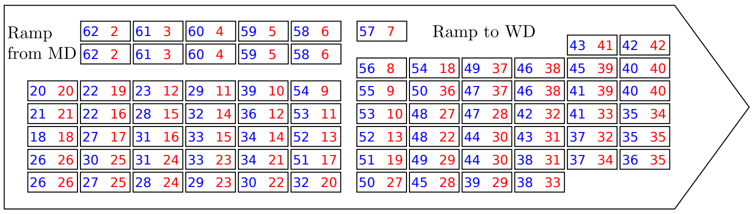
We now present the computational results on our test instances. Table 2 shows the minimum solution value of Gurobi and GRKA with several different options for the continuous optimizer: BRKGA (Gonçalves and Resende, 2011), PSO (Kennedy and Eberhart, 1995), DE (Price, 2013), random construction, and CMA-ES (Hansen, 2006). We chose these continuous optimizers for the GRKA as they represent a wide range of effective strategies for continuous black-box optimization. We allow Gurobi to have 48 wall hours of solving time with up to 8 threads, while all runs of GRKA are performed single threaded with up to 60 seconds of CPU time, although it never needs more than 10 seconds. We run GRKA five times for each instance and continuous optimizer combination and report the best value. We note that the performance of the GRKA is relatively stable over the five runs. However, given the low CPU time of the approach, it is clearly feasible to run multiple copies of GRKA in parallel and take the best value.

We first note that CMA-ES matches or exceeds the best solution found by Gurobi on all instances except one in only a few seconds of CPU time, compared to the hours of time needed by Gurobi. Furthermore, Gurobi finds no solution on six instances of the dataset, further emphasizing the need for a heuristic.

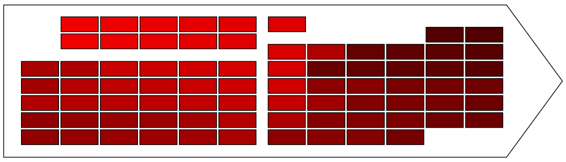
Table 2: Minimum value found over five runs of each metaheuristic implemented in GRKA and a single execution of the Gurobi solver. Note that Gurobi runs that did not find the optimal solution are marked with a star *.

Instance	Ruleset	BRKGA	PSO	DE	Random	CMA-ES	Gurobi	CMA-ES gap to Gurobi
M-0-N	m	**52**	53	**52**	53	**52**	**52**	0.000
	mp	68	68	68	68	68	68*	0.000
	mps	80	80	80	80	80	80	0.000
	ms	68	68	68	68	68	68	0.000
M-9-S	m	47	47	47	47	**46**	**46**	0.000
	mp	60	60	60	60	60	60	0.000
	mps	72	72	72	72	72	72*	0.000
	ms	62	62	62	62	62	62*	0.000
M-11-O	m	**45**	46	46	46	**45**	**45**	0.000
	mp	58	58	58	58	58	58*	0.000
	mps	68	68	68	68	68	68	0.000
	ms	59	59	59	59	59	59*	0.000
M-19-O	m	**44**	45	45	45	**44**	**44***	0.000
	mp	58	58	58	58	58	58	0.000
	mps	68	68	68	68	68	68	0.000
	ms	61	61	61	61	61	61*	0.000
L-0-N	m	72	**71**	73	73	**71**	72*	-0.014
	mp	**96**	**96**	97	97	**96**	-	-
	mps	114	114	114	114	114	-	-
	ms	**96**	**96**	97	97	**96**	-	-
L-20-S	m	**60**	61	61	61	**60**	**60***	0.000
	mp	**76**	**76**	**76**	77	**76**	**76***	0.000
	mps	94	94	94	94	94	94*	0.000
	ms	82	82	82	82	82	82*	0.000
L-24-O	m	59	59	59	59	**57**	58*	-0.017
	mp	**72**	**72**	73	**72**	**72**	**72***	0.000
	mps	90	90	90	90	90	90*	0.000
	ms	79	79	79	79	79	79*	0.000
L-16-O	m	63	63	63	63	**62**	63*	-0.016
	mp	84	84	84	84	84	-	-
	mps	98	98	98	98	98	98*	0.000
	ms	84	84	84	84	84	-	-
L-18-O	m	61	61	62	61	**60**	**60***	0.000
	mp	75	75	75	75	**74**	83*	-0.108
	mps	93	93	93	93	93	93*	0.000
	ms	82	**81**	82	82	82	82*	0.000
L-20-O	m	**56**	**56**	**56**	57	57	**56***	0.018
	mp	**69**	**69**	**69**	**69**	**69**	70*	-0.014
	mps	84	84	84	84	84	-	-
	ms	**75**	**75**	**75**	**75**	**75**	76*	-0.013
lh-38-stern	Industry	38	38	38	38	38	38	0.000
md-67-stern		**58**	**58**	59	**58**	**58**	59*	-0.017
ud-77-midship		**62**	63	63	64	**62**	**62***	0.000
wd-80-bow		60	61	61	61	**59**	**59**	0.000

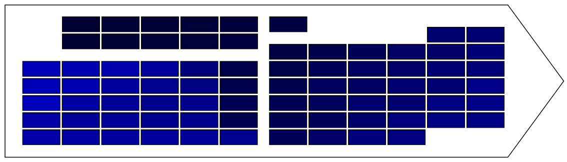
Using GRKA with the random setting, i.e., the decoder is just run with random input until the timeout is reached, results in solutions that are not far away from the best found by CMA-ES or Gurobi. This is likely due to the fact that at least some parts of the problem are easy to solve. In other words, once some decisions are made about the order some trailers are loaded/discharged,



(a) Loading sequence (blue) and discharging sequence (red).



(b) Discharging sequence shaded from the first discharge (light) to the last discharge (dark).



(c) Loading sequence shaded from the first loads (light) to the last load (dark).

Figure 8: An optimal solution to ud-77-midship requiring 62 steps.

the rest of the problem is likely solvable to optimality almost regardless of the random key values. Nonetheless, the advantage of using CMA-ES or one of the other optimizers is clear, as they are able to focus on key choice points and find correct decisions that lead to high quality solutions.

7.3. Solution Analysis

We analyze an optimal solution to the instance ud-77-midship (upper deck with 77 trailer slots and ramp located amidship) to gain further insight into the solution procedure. Figure 8 shows the time step in which each slot is loaded (blue) or discharged (red), along with a visual representation of the discharging and loading. Note that time step 1 is used to bring two of the four tugs on to the ship to begin discharging. Deck access is provided in the rear port section of the vessel, where a ramp extends up from the lower deck. Furthermore, a portion of the forward port deck is blocked by a ramp up to the weather deck.

The solution to this particular instance benefits heavily from dual cycling. With no dual cycling, 84 time steps would be needed to discharge and load the 42 trailer positions. The first dual cycle begins at time step 18 in the middle of the rear of the ship. Note that the dual cycle involves a single trailer position, that is, a tug discharges the trailer at time step 18 and another tug immediately follows that discharge by loading a trailer in the same spot. Notice that future dual cycles do not necessarily concern the same trailer slot, e.g., at time step 22 trailers are loaded towards the rear port of the vessel, and discharged from the middle of the ship.

7.4. Empirical Analysis

We further investigate the impact of dual cycling optimization by benchmarking the heuristic results of the four industrial instances against the current operational mode, i.e., single cycling, as shown in Table 3. Our goal is to quantify how much slower a RoRo vessel can sail due to the time savings of dual cycling at the quay while maintaining its original schedule. In this way, the same amount of cargo can be transported for less CO_2 output and costs. Our results show clearly that dual cycling creates significant savings in the number of tug moves. The larger the

Table 3: Empirical results and estimated time savings (estimated with 5 minutes per move).

Instance	Total Operational Time		Savings	
	Single Cycle	GRKA(CMA-ES)	Moves	Time*
lh-38-stern	45	38	7	35
md-67-stern	71	58	13	65
ud-77-midship	82	62	20	100
wd-80-bow	90	59	31	155

deck is, the higher the degree of dual cycling is possible. In the case of the studied vessel, dual cycling reduces the total tug moves by 71, which is equivalent to a time savings of 355 minutes when assuming 5 minutes per move. Note the time savings are distributed across the four tugs in operation, resulting in an average savings of more than 88 minutes, thus enabling an equally sized reduction in turnaround time.

A shorter turnaround time gives vessels more time and flexibility during their sea voyage, and thus can reduce fuel consumption and CO_2 emissions through slow steaming. Recall that the admiralty coefficient A is a constant for a given vessel that approximates the relationship of displacement ∇, vessel speed V and engine power P (Man Diesel & Turbo, 2011), $A = \frac{\nabla^{\frac{2}{3}} \times V^3}{P}$. The amount of fuel consumed can be estimated by the power used and vessel speed can be approximated by voyage distance and time at sea. Thus, fuel savings through slow steaming can be estimated for any given displacement, resulting in this case in a fuel and emission reduction of nearly 25%. Shorter turnaround times also lead to lower labor costs and potentially more vessels can be accommodated at existing berths. Furthermore, high tug utilization with less empty movements indicates less fuel consumption and emission from the tugs, shorter working hours for the tug drivers and less labor costs for the terminal operators.

8. Conclusion

In this paper, we address dual cycling for RoRo vessels. We presented a mathematical formulation of the RRDCP in the form of an integer programming problem, and showed that the decision version of the RRDCP is NP-complete. Since the integer program we developed requires significant time to solve, we introduce a novel heuristic based on generalizing the BRKGA as our solution approach. We showed that our approach can solve real instances to optimality in many cases, and near optimality in the rest, using only several seconds of CPU time. We further presented the benefit of implementing dual cycling with our algorithm in an industrial case. The empirical results show a significant saving in tug moves and turnaround time, allowing for an estimated reduction in fuel consumption and CO_2 emission by 25%. Given the speed of the heuristic, we expect that our approach can easily be integrated into companies' systems and terminal operations for decision support.

For future work, we intend to investigate how to increase the model realism with regards to the time discretization in our model. Assuming that time is discretized, with each tug movement taking one unit of time regardless of the cargo type and cargo location on board or in terminal, is a key limitation of this work. Future work will thus examine alternative time discretizations or removing the discretization entirely. Another aspect that will be considered is the robustness of the solution with regard to dynamic or realtime information.

Acknowledgements

The research was partially supported by European Interreg Baltic Sea Region Program (ECO-PRODIGI project) and the RoRoGreen Project funded by the Innovation Fund Denmark. We wish to thank our industrial partner DFDS for their openness and innovation and valuable support throughout the conduct of research. We thank the Paderborn Center for Parallel Computation (PC^2) for the use of the OCuLUS cluster.

References

Aho, I., Mäkinen, E., 2006. On a parallel machine scheduling problem with precedence constraints. Journal of Scheduling 9, 493–495. doi:10.1007/s10951-006-8499-4.

Bayliss, C., Currie, C.S., Bennell, J.A., Martinez-Sykora, A., 2021. Queue-constrained packing: A vehicle ferry case study. European Journal of Operational Research 289, 727–741. doi:https://doi.org/10.1016/j.ejor.2020.07.027.

Chandra, S., Christiansen, M., Fagerholt, K., 2016. Combined fleet deployment and inventory management in roll-on/roll-off shipping. Transportation Research Part E: Logistics and Transportation Review 92, 43–55. doi:10.1016/j.tre.2016.03.014.

Chen, X., Li, F., Jia, B., Wu, J., Gao, Z., Liu, R., 2021. Optimizing storage location assignment in an automotive ro-ro terminal. Transportation Research Part B: Methodological 143, 249–281. doi:https://doi.org/10.1016/j.trb.2020.10.009.

Christodoulou, A., Raza, Z., Woxenius, J., 2019. The integration of roro shipping in sustainable intermodal transport chains: The case of a north european roro service. Sustainability (Switzerland) 11. doi:10.3390/su11082422.

Dunning, I., Huchette, J., Lubin, M., 2017. JuMP: a modeling language for mathematical optimization. SIAM Review 59, 295–320. doi:10.1137/15M1020575.

EUROSTAT, 2020. Statistics Explained. URL: `https://ec.europa.eu/eurostat/statistics-explained/pdfscache/6652.`

Fischer, A., Nokhart, H., Olsen, H., Fagerholt, K., Rakke, J.G., Stålhane, M., 2016. Robust planning and disruption management in roll-on roll-off liner shipping. Transportation Research Part E: Logistics and Transportation Review 91, 51–67. doi:10.1016/j.tre.2016.03.013.

Gonçalves, J.F., Resende, M.G., 2011. Biased random-key genetic algorithms for combinatorial optimization. Journal of Heuristics 17, 487–525.

Gonçalves, J.F., Resende, M.G., Mendes, J.J., 2011. A biased random-key genetic algorithm with forward-backward improvement for the resource constrained project scheduling problem. Journal of Heuristics 17, 467–486.

Goodchild, A.V., Daganzo, C.F., 2006. Double-cycling strategies for container ships and their effect on ship loading and unloading operations. Transportation Science 40, 473–483. doi:10.1287/trsc.1060.0148.

Gurobi Optimization, Inc., 2020. Gurobi optimizer 9.0 reference manual. URL: `http://www.gurobi.com`.

Hansen, J.R., Fagerholt, K., Meisel, F., Rakke, J.G., 2019. Planning interrelated voyages with separation requirements in roll-on roll-off shipping. EURO Journal on Transportation and Logistics 8, 633–659. doi:10.1007/s13676-019-00140-x.

Hansen, J.R., Hukkelberg, I., Fagerholt, K., Stålhane, M., Rakke, J.G., 2016. 2D-packing with an application to stowage in roll-on roll-off liner shipping, in: International Conference on Computational Logistics, Springer. pp. 35–49.

Hansen, N., 2006. The CMA Evolution Strategy: A Comparing Review. Springer, Berlin, Heidelberg. pp. 75–102.

Hjelle, H., 2010. Short sea shipping's green label at risk. Transport Reviews 30, 617–640. doi:10.1080/01441640903289849.

Hu, T.C., 1961. Parallel sequencing and assembly line problems. Operations Research 9, 841–848. doi:10.1287/opre.9.6.841.

Jia, B., Fagerholt, K., 2021. Step-wise stowage planning of roll-on roll-off ships transporting dangerous goods. Maritime Transport Research 2, 100029. doi:10.1016/j.martra.2021.100029.

Jia, B., Fagerholt, K., Reinhardt, L.B., Rytter, N.G.M., 2020. Stowage planning with optimal ballast water, in: International Conference on Computational Logistics, Springer. pp. 84–100.

Jia, B., Rytter, N., Reinhardt, L., Haulot, G., Billesø, M., 2019. Estimating discharge time of cargo units – a case of ro-ro shipping, in: International Conference on Computational Logistics, Springer. pp. 122–135.

Johnson, S.M., 1954. Optimal two- and three-stage production schedules with setup times included. Naval Research Logistics Quarterly 1, 61–68.

Kennedy, J., Eberhart, R., 1995. Particle swarm optimization, in: Proceedings of ICNN'95-International Conference on Neural Networks, IEEE. pp. 1942–1948.

Ku, D., Arthanari, T.S., 2016. On double cycling for container port productivity improvement. Annals of Operations Research 243, 55–70. doi:10.1007/s10479-014-1645-z.

Lawler, E., Lenstra, J., Rinnooy Kan, A., Shmoys, D., 1993. Sequencing and scheduling: Algorithms and complexity. Handbooks in Operations Research and Management Science 4, 445–522. doi:10.1016/S0927-0507(05)80189-6.

Lenstra, J.K., Rinnooy Kan, A.H.G., 1978. Complexity of scheduling under precedence constraints. Operations Research 26, 22–35. doi:10.1287/opre.26.1.22.

Man, R., Bak, J., Son, C., Kim, S., Jung, Y., 2020. Automatic path generation for tractor-trailers according to the ramp slope of ro-ro ships. Journal of Computational Design and Engineering 8, 316–329. doi:10.1093/jcde/qwaa080.

Man Diesel & Turbo, 2011. Basic principles of ship propulsion.

Ng, A., 2009. Competitiveness of short sea shipping and the role of port: The case of north europe. Maritime Policy and Management 36, 337–352. doi:10.1080/03088830903056983.

Nguyen, V.D., Kim, K.H., 2010. Minimizing empty trips of yard trucks in container terminals by dual cycle operations. Industrial Engineering and Management Systems 9, 28–40.

Oucheikh, R., Löfström, T., Ahlberg, E., Carlsson, L., 2021. Rolling cargo management using a deep reinforcement learning approach. Logistics 5, 10. doi:10.3390/logistics5010010.

Øvstebø, B.O., Hvattum, L.M., Fagerholt, K., 2011. Routing and scheduling of roro ships with stowage constraints. Transportation Research Part C: Emerging Technologies 19, 1225–1242. doi:10.1016/j.trc.2011.02.001.

Pahl, J., Cordova, M., 2020. A secure bet in the maritime supply chain, in: Handbook of Research on the Applications of International Transportation and Logistics for World Trade. IGI Global, pp. 330–353. doi:10.4018/978-1-7998-1397-2.ch018.

Price, K.V., 2013. Differential evolution, in: Handbook of Optimization. Springer, pp. 187–214.

Prot, D., Bellenguez-Morineau, O., 2018. A survey on how the structure of precedence constraints may change the complexity class of scheduling problems. Journal of Scheduling 21, 3–16. doi:10.1007/s10951-017-0519-z.

Psaraftis, H.N., Kontovas, C.A., 2013. Speed models for energy-efficient maritime transportation: A taxonomy and survey. Transportation Research Part C: Emerging Technologies 26, 331–351.

Puisa, R., 2021. Optimal stowage on ro-ro decks for efficiency and safety. Journal of Marine Engineering & Technology 20, 17–33.

Si, S.L., You, X.Y., Liu, H.C., Zhang, P., 2018. DEMATEL technique: A systematic review of the state-of-the-art literature on methodologies and applications. Mathematical Problems in Engineering 2018, 1–33. doi:10.1155/2018/3696457.

Soares, L.C.R., Carvalho, M.A.M., 2020. Biased random-key genetic algorithm for scheduling identical parallel machines with tooling constraints. European Journal of Operational Research 285, 955–964.

Song, J.H., Kwak, K.S., 2009. Optimization of double cycling in container ports. Journal of Navigation and Port Research 33, 127–134. doi:10.5394/KINPR.2009.33.2.127.

Strandenes, S.P., 2004. Port pricing structures and ship efficiency. Review of Network Economics 3. doi:10.2202/1446-9022.1047.

Ullman, J.D., 1975. Np-complete scheduling problems. Journal of Computer and System Sciences 10, 384–393. doi:10.1016/S0022-0000(75)80008-0.

UNCTAD, 2019. Review of Maritime Transport. United Nations Conference on Trade and Development. United Nations Publications, 300 East 42nd Street, New York, New York 10017, United States of America. URL: https://unctad.org/en/pages/PublicationWebflyer.aspx?publicationid=2563.

Zhang, H., Kim, K.H., 2009. Maximizing the number of dual-cycle operations of quay cranes in container terminals. Computers & Industrial Engineering 56, 979–992. doi:10.1016/j.cie.2008.09.008.

Zheng, F., Pang, Y., Liu, M., Xu, Y., 2020. Dynamic programming algorithms for the general quay crane double-cycling problem with internal-reshuffles. Journal of Combinatorial Optimization 39, 708–724.

doi:10.1007/s10878-019-00508-9.
Zis, T., Psaraftis, H.N., 2017. The implications of the new sulphur limits on the european ro-ro sector. Transportation Research Part D: Transport and Environment 52, 185–201.