University of Groningen

A Decision-Support Framework for the Stowage of Maritime Containers in Inland Shipping

Fazi, Stefano

Published in:

Transportation Research Part E: Logistics and Transportation Review

DOI:

[10.1016/j.tre.2019.09.008](https://doi.org/10.1016/j.tre.2019.09.008)

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version

Publisher's PDF, also known as Version of record

Publication date:

2019

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Fazi, S. (2019). A Decision-Support Framework for the Stowage of Maritime Containers in Inland Shipping. *Transportation Research Part E: Logistics and Transportation Review*, 131, 1-23.

<https://doi.org/10.1016/j.tre.2019.09.008>

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

A decision-support framework for the stowage of maritime containers in inland shipping

Stefano Fazi

University of Groningen, Faculty of Economics and Business, Nettelbosje 2, 9747 AE Groningen, the Netherlands

ARTICLE INFO

Keywords:

Inland shipping
Stowage
Containers
Dry-port
Stability
Metaheuristic

ABSTRACT

The typical voyage of a barge transporting maritime containers inland, consists of visiting a set of terminals where loading/unloading operations occur. The required stowage plans must meet stability requirements during all transport phases and avoid costly re-handling operations. In this paper, we formalize this problem and consider a typical dry-port transport system. We propose a comprehensive mathematical model that seeks to maximize stowed containers. To solve it, we develop a hybrid metaheuristic approach, based on local search and an industrial solver. Numerical experiments, based on real-world data, provide insights into the performances of the proposed framework and current stowage practices.

1. Introduction

In several regions worldwide, inland shipping plays a leading role in the transport of maritime containers inland. Typically, the reason is the presence of navigable canals and good infrastructures, such as dry-ports, that enable a far reach into the hinterland and consequently an easier generation of economies of scale. In the Netherlands, authorities are promoting this modality with clear modal split goals. The expectation is that inland shipping will reach a volume of 8 million TEU (twenty-foot equivalent unit) per year in 2035, covering 45% of total container discharge. The response of terminal operators and shipping companies is to invest in new technologies to improve container handling and operational efficiency that are both largely affected by loading and unloading operations (Port of Rotterdam, 2015; Port of Rotterdam, 2018; Dkhil et al., 2018).

Operationally, transport by barge, see Fig. 1, comes with a set of problems requiring both expertise and support systems (Fazi et al., 2015). In this regard, the stowage of a set of containers on board is challenging. In general, stowage is defined as the decision as to the exact position of containers on a ship in order to guarantee stability during navigation and handling operations. Unlike maritime container ships, barges have a light structure and, typically, have limited or no ballast tanks to compensate for imbalances, making their stability particularly sensitive to stowage conditions (Li et al., 2017). Moreover, the inland shipping industry stresses capacity utilization over schedules, due to the competition with alternative modes of transport, such as truck or rail, and the resulting need to generate economies of scale and reduce costs (Zuidwijk and Veenstra, 2014). Hence, stowage plans need to be devised to maximize the number of transported TEU so as to avoid extra costly re-handling during loading and unloading operations, for example, when it comes to reaching containers stowed below others to be discharged at later ports (i.e., over-stowage) (Li et al., 2018).

In this paper, we will formalize the stowage problem for inland shipping and propose a decision-support framework. We will provide a description of a typical transport system consisting of a dry-port, serving as collection point for import and export containers, and several seaport terminals, where loading/unloading operations occur. Furthermore, we include accurate stability

E-mail address: s.fazi@rug.nl.

<https://doi.org/10.1016/j.tre.2019.09.008>

Received 10 February 2019; Received in revised form 9 September 2019; Accepted 14 September 2019

Available online 27 September 2019

1366-5545/ © 2019 Elsevier Ltd. All rights reserved.

Fig. 1. Barges sailing on the Rhine river. Container barges can typically carry between 20 and ~400 TEU, which can be accessed from the top.

guidelines. Next, we will develop two novel mathematical formulations on a container level in order to model relevant decisions. The goal of the first model is to maximize the number of containers on board of a barge, visiting a given sequence of terminals, while guaranteeing stability during navigation and avoiding over-stowage. The purpose of the second model is to formulate a feasible handling sequence of containers to ensure stability during loading/unloading. For solving the first model, we propose a tailored hybrid Variable Neighborhood Search metaheuristic, endowed with a labeling mechanism to ensure no over-stowage. The second model takes as its input the feasible stowage from the first model, and then a MILP solver is called on the fly to find a feasible loading/unloading sequence.

By means of numerical experiments we will validate our framework and test the performances of the proposed solution method against the mathematical model and current stowage practices. The results show the limitations of trial-and-error procedures, and the possibility to increase barge utilization. As support for this study, we propose the case of inland shipping and related regulations in the Netherlands, for which real-world data on containers and a Dutch barge is available.

This paper is structured as follows. In Section 2, we review the existing literature and clarify the contribution. In Section 3, the problem is described in detail. In Section 4, we develop the mathematical models. In Section 5, we describe the solution method. In Section 6, we show the experiments to analyze the performances of our method and provide practical insights on current stowage practices. Section 7 concludes the paper with a few observations.

2. Related literature

Our problem is a special case of the “container stowage problem” (CSP), which we define as the problem of assigning containers to vessel positions. In general, the assignment must fulfill stability requirements, while minimizing over-stowage and handling time at the terminal. Although in some papers CSP may refer to specific cases of the problem, in this paper we use it as the overarching term.

2.1. The container stowage problem for container ships

In real life, the CSP for maritime container ships is typically made up of two phases. In the first phase, a class based stowage plan is generated by the liner shipper, where container types are assigned to vessel positions for each leg of a route; the typology refers to weight and size. The goal of the liner shipper is to achieve stability, while maximizing vessel utilization and minimizing over-stowage. In the second phase, the terminal can decide on the match between container types and actual containers with the goal of minimizing the handling time to load and unload the containers; the decision is based on the position of the containers at the yard (Monaco et al., 2014). In some cases, the liner shipper completes the stowage plan without knowing the yard’s status.

In the literature, the CSP has been studied from very different angles and only few papers have analyzed the problem in the terms just described (Iris and Pacino, 2015; Monaco et al., 2014).

As far as the first phase is concerned, the common objective is to minimize the shifts on board. Avriel et al. (1998) propose a simplified problem (a ship with a single bay) and ignore stability requirements; a binary linear programming formulation and a heuristic procedure are developed as well. Dubrovsky et al. (2002) develop a genetic algorithm for the same problem. They also include stability constraints by penalizing excess weight on either side of the ship. In more recent studies, Ding and Chou (2015) and Roberti and Pacino (2018) develop heuristics and exact methods that outperform the one found in Avriel et al. (1998). In other papers addressing the liner shipper’s problem, the problem is further deconstructed into two phases: strategic and tactical. In the strategic phase (Master planning), containers are allocated to blocks with the aim of reducing occupied cargo spaces, hatch-lid movements, over-stowage, etc. In the tactical phase (Slot planning), containers are assigned to specific slots with the aim of minimizing re-handling and a mix of containers with different destinations. In Wilson and Roach (1999), a stack of containers must also be placed by non-decreasing weight from the bottom up. The first phase is solved via branch-and-bound, and the second via tabu search. Kang and Kim (2002) consider approximate linear formulas for stability constraints, and the number of shifts is found via approximation. The authors develop a greedy heuristic for the strategic problem and a tree search method for the tactical problem. In the same line of research, Pacino et al. (2011) tackle the strategic phase, here called master planning, which becomes input for the model presented in Delgado et al. (2012); both papers solve their respective problems with commercial solvers. In terms of stability, Pacino et al. (2011) constrain each block’s total weight as being within pre-calculated limits, whereas Delgado et al. (2012) limit the weight and height of each stack. Parreño et al. (2016) extend the problem definition in Delgado et al. (2012) by including IMO containers. The aim of the problem is to minimize overstows and maximize the number of stowed containers. The authors develop an Integer Programming

formulation and a Greedy Randomized Adaptive Search Procedure (GRASP) algorithm to obtain fast solutions. Finally, [Zhang and Lee \(2016\)](#) propose a multi-objective problem that maximizes ship stability and minimizes containers re-handles. A variant of the non-dominated sorting genetic algorithm combined with a local search component is proposed.

As far as the second phase is concerned, studies analyze the problem from the point of view of terminals and also tackle loading sequencing. [Ambrosino et al. \(2004\)](#) develop a mathematical model to decide the stowage plan of a container ship, including stability guidelines and physical constraints. The model, called the master bay planning problem (MBPP), considers a given sequence of docks where only unloading operations are performed, and the goal is to minimize the time for loading the containers on board. The model is solved by first grouping containers based on their destinations and reducing the solution state space, and, next, by solving the model with a commercial solver. The study is later extended with an improved heuristic approach in [Ambrosino et al. \(2006\)](#). [Sciomachen and Tanfani \(2007\)](#) deal with the MBPP, including the maximization of crane productivity. The authors relate the problem to the 3D bin packing problem (3D-BPP) and develop a heuristic inspired by an existing branch-and-bound methodology for the 3D-BPP. [Imai et al. \(2006\)](#) studied the relationship between the loading-related re-handle and the ship stability for mega container ships. Unloading-related re-handles was intentionally left out of the scope since it makes the problem substantially more difficult. They defined the problem as a multi-objective integer programming and proposed the weighting method to generate noninferior solutions. The objective function adds three elements for stability and one element as an approximation for the number of re-handles to retrieve the container from the yard. [Monaco et al. \(2014\)](#) treat the problem of how to load containers on a container ship, with the aim of reducing handling operations and the time required to access containers from the quay yard area and bring them on board. The stowage must respect basic stability guidelines and a given approximated allocation scheme. The authors propose a two-step heuristic algorithm to solve the problem. Finally, [Iris et al. \(2018\)](#) integrate the assignment and scheduling of transfer vehicles and container load sequencing for the allocation of containers to a ship's slots. The aim is to reduce the service time for the handling equipment and to meet loading deadlines. The model is solved heuristically and relaxed for benchmarking purposes.

2.2. The container stowage problem in inland shipping

Stowage plans in inland shipping involve different problems and objectives than maritime container ships. First of all, barge stability is very sensitive to container positioning on board and navigation rules are very strict. Moreover, barges usually do not have any ballast to stabilize the cargo, as is the case for maritime vessels, which use water to compensate for weight imbalance ([Pacino et al., 2012](#)). Third, barge operators need to avoid over-stowage, since the additional costs may severely put the economy of scale at risk ([Li et al., 2017](#)).

Besides those differences, stowage planning in inland shipping may be considered as an integrated Master and Slot planning problem for maritime ships; thereby more related to the first phase in CSP, where shipping liners define stowage plans to maximize utilization and reduce over-stowage while ensuring stability. With concern to the second phase, barges do not make an interesting case for terminal operators for the minimization of handling time compared to large vessels. The reason is twofold. First, the number of containers handled is low. Second, over-stowage in inland shipping is limited, and thus, reducing already to a minimum the handling effort.

Although barge transport is gaining a leading role in inland container transport in several regions worldwide, literature lacks studies specific to this ship category. In fact, available papers on inland shipping refer generically on small container ships with no hatch covers. Barges typically also have no hatch covers, but structurally they are flat-bottomed ship that are meant to sail just rivers and canals. In principle, the main difference may be related to the way stability measurements are computed. The papers reviewed in this section explicitly refer to inland shipping and related problems, but do not mention the term “barge”.

[Yaagoubi et al. \(2016\)](#) tried to combine the CSP with the Travelling Salesman Problem, but proposed a very simple version of the problem. Given a loading plan for the ship, the objective is to jointly minimize routing costs and handling costs. The formulation consists of a single ship with: one container stack, one type of container, and with each destination port requiring the delivery of only one container. [Li et al. \(2017\)](#) research the problem of container ship stowage on the full route of the Yangtze River focusing on inland container ships. The problem is deconstructed into two steps. In the first, the assignment of containers to bays aims at reducing the number of bays occupied and the longitudinal weight difference. In the second, the assignment of containers to slots aims at reducing over-stowage. A GRASP algorithm is developed.

More recently [Li et al. \(2018\)](#) formulated the CSP for inland shipping on the Yangtze River considering a multi-port stowage plan that aims to maximize capacity utilization by ensuring that the least number of stacks is used. The problem considers the weight of the containers to be uncertain due to logistical limitations at the Yangtze River terminals. The model ensures that no over-stowage occurs by allowing stacks to be composed only of containers of the same origin and destination. Stability constraints concern the maximum weight of each stack, and longitudinal (trim) and transversal (list) weight tolerance, as proposed in [Pacino et al. \(2011\)](#). A hybrid neighborhood search algorithm is developed and tested against exact methods.

2.3. Contribution

In this paper, our main aim is to formalize a typical stowage problem for inland shipping, and develop a comprehensive decision-support framework. The problem considers multi-port stowage planning in a dry-port setting, with deterministic containers' features and barges as ship category and takes the point of view of barge operators. Contrary to [Li et al. \(2018\)](#), we consider container weights to be known in advance, since accurate weighing technology has emerged and is already available in several western ports ([Port of Rotterdam, 2017](#)). Next, we also allow containers with different origins and destinations to be in the same stack. Furthermore, in [Li](#)

et al. (2018) the stability is imposed by limiting excess weight on one of the sides of the ship, whereas our approach consists of linking the stowage to the actual stability formulas, since there is relevant interplay. A similar attempt can be found in the PhD thesis of Delgado (2013); however, the stability formulas do not consider the center of gravity of each stowed containers, but rather consider it per location, namely for adjacent sets of stacks. Finally, we compute feasible sequences of loading and unloading operations to keep the ship stable. A similar attempt may be found in Kim et al. (2004) and Meisel and Wichmann (2010).

With regard to modeling, we propose a 5 index binary variable $\delta_{xyz,i}^c$ which indicates whether a container c has been placed into slot xyz in leg i . This modeling approach can also be found in Ambrosino et al. (2006). In terms of constraints and level of definition of stowage plans, Ambrosino et al. (2006) and (Parreño et al., 2016), for example, also consider actual positions of 20-foot and 40-foot containers.

3. Problem description

In this section, we give details of our problem. We will first outline the setting. Next, we will describe the decision making process. Finally, notions of ship stability are introduced.

3.1. Problem setting

We consider the transportation of maritime containers between a seaport and a dry-port by barge. A dry-port is an inland terminal directly connected to the seaport. Its strategic position in the hinterland makes it a key center for the collection of large volumes of containers. This favors the use of high-capacity modes of transport, since large volumes are consolidated and diverted to a central terminal and regular shuttle services are activated. We refer to Roso et al. (2009) for an excellent description, and to Crainic et al. (2015) and Fazi and Roodbergen (2018) for recent studies.

Barges can be described as ships with a flat-bottomed hull, without hatch covers, and built for sailing calm waters, such as rivers and canals. There exist several types of barges, such as crane, dry-bulk, and push barges. In this study, we will refer to self-propelled barges, equipped for container transport. A drawing of this type of barges is displayed in Fig. 2.

In the proposed setting, a barge follows a predefined rotation, that is, route, starting from the dry-port and consisting of a sequence of sea terminals where containers are loaded and unloaded, and finally ending at the dry-port. We will consider two classes of containers: import and export. Export containers, initially available at the dry-port, need to be moved to their terminals of destination at the seaport. Import containers need to be picked up at their terminals of origin at the seaport and then moved to the dry-port. See Fig. 3 for a simplified representation. Although exchange of containers between sea terminals of a seaport can occur, this is typically not a task for barges transporting containers from and to the hinterland. Hence, in this study we will not consider this possibility. The interested reader is referred to Schepler et al. (2017) and Lee et al. (2012) for studies related to the management of multi-terminal systems and inter-terminal transshipment of containers.

3.2. Transport planning

A typical inland shipping planning phase involves two main entities: barge planners and captains. Barge planners, following the guidelines of the shippers, develop a list of containers to be transported in a voyage by a specific barge, available at the dry-port. In this phase, planners will only take into account the maximum capacity of the ship and make sure that the round trip is composed of a limited number of stops. After receiving the list of containers, the captain of the barge starts the stowage-planning phase, namely he/she will decide for each transport leg the location of the containers on board. The goal of the captain is to maximize the number of stowed containers while avoiding over-stowage and guaranteeing stability during the voyage. If rejection of containers occurs during this phase, the barge planners are notified and plan those containers in a different manner.

Commonly, stowage plans are still generated manually by the captain of the barge. The captain tries several configurations for each transport leg by trial and error on stowage simulation software, if available, which accepts the stowage plan as soon as the tried configuration reaches an acceptable stability (Gumus et al., 2008). To produce a reasonable plan, the captain follows basic guidelines based on experience. For example, a common strategy is to dedicate certain areas of the ship for containers destined to the same terminal. If this number is large and the number of containers to be loaded back is small, it is likely that the barge will experience disequilibrium. Obviously, such rough procedures may lead to container rejections, especially when the captain has little experience with stowage plans, the captain has never sailed on a specific ship, the number of containers is large or close to maximum capacity, and the established voyage has several stops.

In this paper, we take the point of view of the captain. We consider an initial list of containers and a rotation as input. The output is a stowage plan for each transport leg.

Fig. 2. Drawing of a self-propelled barge.

Fig. 3. An example. A barge departs from the dry-port and visits a set of numbered sea terminals, where loading and unloading operations occur. Numbers in the barge's slots represent containers and their related sea terminal. In particular, underlined numbers refer to import containers, which are loaded at the sea terminal of origin. Non-underlined numbers refer to export containers.

3.3. Stowage plans

The position of a container on a ship is usually considered according to its location in terms of rows, bays, and tiers. In a barge, a bay, referred to as x , expresses the longitudinal position; the row, referred to as y , the latitudinal position; and the tier, referred to as z , represents the vertical position on the ship. We cite [Ambrosino et al. \(2004\)](#) as an excellent representation of this.

For convenience, and given the fact that containers are standardized units, it is typical to represent the deck of a ship as divided into rectangular slots. Some slots could be dedicated to either 20-foot/40-foot containers, or special ones, such as 45-foot, or containers with dangerous cargo, or reefer containers, for the proximity to electric plugs. On a slot, located in a defined x and y position, containers can be stacked and form tiers.

3.3.1. General stability considerations

Main stability guidelines are usually the same for every ship, although a few formulas change when considering either box-shaped or triangular-shaped vessels. We refer to [Barrass \(2000\)](#) and [Barrass and Derrett \(2011\)](#) for two excellent books on ship stability. In this section, we will discuss the main stability indicators that are computed in the proposed setting.

Among others, there are three main factors to be calculated to assess stability: the center of gravity, the angle of list, and the trim. The first two items concern the transverse (or latitudinal) stability, whereas the latter the longitudinal stability. Other forces, such as Shearing forces and Bending moments concern the structural robustness of the ship and are more relevant in case of rough weather and waves ([The Naval Arch, 2017](#)); therefore, they will not be considered in this setting.

Looking at a ship from the stern in a perfect upright position, by tracing an imaginary line up from the keel (K), perpendicular to the surface of the earth, we encounter the center of buoyancy (B), the center of gravity (G), and the metacenter (M). The center of gravity, G , is an imaginary point where the whole mass of the ship is concentrated for the gravitational force to act and pull the ship down. The center of buoyancy, B , is the point where the force of buoyancy pushes the ship upwards to keep it afloat. The metacenter, M , is a property of the ship and is an abstract reference point where the ship starts oscillating as soon as there is a displacement. Consequently, the distance between K and M , denoted as $\overline{KM}$, is equal to $\overline{GM} + \overline{KG}$.

The first requirement for a stable ship is that G should be low and below M . As soon as weight is placed on the ship, G and M tend to shift upwards and downwards, respectively. To prevent G rising too much, the heavier cargo should be placed at the bottom of the ship, to keep a low center of gravity: this is to prevent the ship from inclining too much to port or starboard due to a non-centered force, that is, wind, pilot-boats, containers etc. In general, an optimal G should be within a certain range, which differs per ship. On one hand, a small $\overline{GM}$ will make the ship dangerously unstable, in jargon “tender”, on the other hand, a large $\overline{GM}$ (i.e. $\overline{KG}$ small) will make the ship “stiff” with an uncomfortable jerky movement while sailing ([Barrass, 2000](#)), though, in essence, very stable. In the scientific literature on stowage planning, the approach is to maximize $\overline{GM}$, see [Imai et al. \(2006\)](#) and [Kang and Kim \(2002\)](#). In practice, the approach is twofold. Either the computed $\overline{GM}$ is within an established range, or $\overline{KG}$ is less than an established point $\overline{KG}_{max}$, which is a precautionary measurement that considers external factors, such as wind, and is typically a function of the draft

Fig. 4. A basic requirement for stability.

(Barrass, 2000; Papanikolaou, 2018). See Fig. 4. In order to compute $\overline{KG}$, we can use the following formula:

$$\overline{KG} = \frac{\text{Sum of the moments about K}}{\text{Sum of the weights}} = \frac{\overline{KG}_0 W_0 + \sum_{i \in C} W_i (kg_i)}{W_0 + W_c} \quad (1)$$

where $\overline{KG}_0$ is the $\overline{KG}$ of the empty ship. W_0 is the weight of the empty ship. kg_i is the height of the center of gravity of a placed container i of weight W_i , computed from the keel of the ship; if the container is placed, for example, in the second tier, kg_i includes the height of the container placed below. Finally, $W_c = \sum_{i \in C} W_i$ is the total weight of loaded containers, defined by set C . We notice that $\overline{KG}$ will tend to increase in those cases where heavier containers have larger kg values.

A second requirement for stability involves the angle of list, which is the angle to which the ship heels to either port or starboard. As soon as a weight is positioned off-center, the gravitational force $\vec{G}$ will shift towards the displacement and the ship will start inclining proportionally. To reach a stable floatation, the buoyancy force $\vec{B}$ will also shift towards the displacement until it is in line with $\vec{G}$; this stabilization will result in an extra inclination of the ship. See Fig. 5.

The inclination of the ship when G is below M is called “angle of list”, otherwise “angle of loll”. A low $\overline{KG}$ prevents the angle of loll, which is hazardous for stability, and also limits the shift of $\vec{G}$ due to off-centered forces. The angle of list is minimum when there is not too much excess load on one of the sides of the ship. The formula related to the angle of list, θ , is as follows:

$$\tan(\theta) = \frac{\sum_{i \in C} W_i (y_i - y_0)}{(W_0 + W_c) \overline{GM}} \quad (2)$$

where y_i is the y-coordinate of the placed i^{th} container, y_0 is the central coordinate. $\overline{GM}$ is the distance between G and M , where $\overline{GM} = \overline{KM} - \overline{KG}$. $\overline{KM}$ is a value that depends on the loaded weight and is computed during construction phase of the ship. Hence, it must be reported in the available documentation.

Finally, the trim concerns longitudinal stability and is the difference between the forward (bow side) and aft (stern side) drafts, where draft is the vertical distance between the keel and the waterline. When the ship is inclined towards the aft, it is said that it has a trim of x meters by aft, otherwise by forward. The formula related to the angle of trim, α , is as follows:

Fig. 5. The extra weight entails an imbalance. Its effect is increased as the ship seeks for stability when floating.

$$\tan(\alpha) = \frac{\sum_{i \in C} W_i (x_i - x_0)}{MCT/L} \quad (3)$$

where x_i is the x-coordinate of the i^{th} container, x_0 is the x-coordinate of the initial center of gravity and finally, MCT is the moment required to change the trim by one centimeter and L is the length of the ship between perpendiculars.

Some additional guidelines for stowage are the following points. (1) It is not possible to place a 20-foot container on top of a 40-foot one, since 40-foot containers have a weaker structure than 20-foot ones and may be damaged (Parreño et al., 2016). (2) In case of bridges, the weight of the barge should be large enough to guarantee the passage beneath. This obviously depends also on the height of the stacks, that is, layers of containers on the ship. (3) Open top containers cannot support other containers. (4) Reefer containers have specific locations on the ship, in proximity to electric plugs.

3.3.2. Stability and stowage related considerations for container barges

Container barges follow the reported generic stability guidelines. Among the main elements to assess a barge stability, KM and $KG_{\max}$ are values that cannot be easily computed. KM is a particular measurement that depends on the construction of the ship and the experimental calculations are typically reported in an available manual. With concern to $KG_{\max}$, the values are calculated with a complex formula that includes several components, such as lateral wind pressure and free fluids moments. For this research a manual of a real-world barge is available. Here, KM values depend on the loaded weight, whereas $KG_{\max}$ values depend on the number of tiers, the presence of high-cube containers, and the total loaded weight. For example, a barge loaded with 3 layers consisting of 40-foot standard containers has a higher $KG_{\max}$ than another with 3 layers including 40-foot high-cube containers with the same weight, since high-cube containers have a higher center of gravity.

With concern to trim, for barges, depending on the sailing direction on the river, it may be preferable for the trim to be slightly by forward or aft in order to guarantee smoother navigation and consequently less oil consumption; this information has been provided by experienced sailors.

Another important stability element for barges is to maintain stability during handling operations at the terminals (Maritime New Zealand, 2006). For example, the trim and/or the list may be negatively affected if too much cargo is discharged on one side only. Hence, it is also crucial to compute a sensible loading/unloading sequence, although this is typically not considered in practice.

4. Mathematical formulations

In this section, we present the two models characterizing the decision support framework. The first model computes the stowage and aims to maximize the number of TEU on board while ensuring stable navigation. The second model computes, given a feasible stowage plan, a feasible loading/unloading sequence to maintain the barge stable during handling operations.

4.1. A stowage model for stable navigation

The input of our problem is a list of candidate containers and a rotation generated by barge planners. The planners only make sure that the rotation satisfies the TEU capacity of the barge after loading/unloading operations at each terminal, and leave to the captain the task of finding a feasible stowage plan. Other relevant parameters are the structural specifications of the barge.

We consider a list of candidate containers C , composed of two subsets: $\mathcal{E}$ for export containers, and $\mathcal{I}$ for import. Export containers can be loaded on the barge at the dry port and discharged at their sea terminal of destination, whereas import containers can be loaded at their sea terminal of origin and discharged at the dry port. Each container also features a weight W_i with $i \in C$.

Let us also define a barge with a stowage space divided in slots, where X , Y and Z are the sets for the slots' coordinates, respectively for bays, rows, and tiers. With concern to bays, each slot may be occupied by two 20-foot containers or one 40-foot (or 45 foot in some cases), always placed with the short side parallel to the bow. The address number of a slot depends on the system adopted by the barge operator. We follow the system as in Ambrosino et al. (2004). Specifically, each 20-foot bay is numbered with an odd number, and two consecutive odd bays conventionally yield one even bay, for the stowage of 40-foot containers (or 45). Hence, we define bays from 1 to $|Y|$, excluding indexes multiple of 4. Henceforth, for conciseness, we will focus on 20-foot and 40-foot containers, whose subsets are indicated respectively by $\mathcal{T}$ and $\mathcal{F}$, and their corresponding bays by $\mathcal{X}^{\mathcal{T}}$ and $\mathcal{X}^{\mathcal{F}}$.

The barge transports containers through a predefined route, which starts from the dry-port and ends to it. In this regard, we define a set of nodes (i.e., docks) N indicating the dry-port and the sea terminals. The dry-port has index 0, whereas the other terminals are assigned an incremental index based on their appearance in the route. Finally, we define a parameter D_{ij} which equals 1, if container $i \in C$ is destined to (for $i \in \mathcal{E}$) or has to be picked up at (for $i \in \mathcal{I}$) dock j , with $j \in N \setminus \{0\}$.

The main goal of our problem is to stow as many candidate containers as possible. The main constraints are: to make sure the stowed containers are transported from origin to destination seamlessly; to avoid over-stowage when unloading containers; to guarantee stability during all transport phases; to construct a sensible stowage without floating containers; to avoid other unrealistic situations.

To model our decision, we define a main binary decision variable $\delta_{xyz,j}^i$, with x , y and z being the 3-dimensional coordinates, $i \in C$, $j \in N$. This variable takes value 1 if container i is stowed in slot xyz in the transportation leg successive to node j . Other variables concern the computation of $KG_{\max}$ and $\overline{KM}$ for each transport leg, as described in Section 3.3.2. $\overline{KM}$ depends on the loaded weight. $KG_{\max}$ depends on the loaded weight, number of tiers occupied, and presence or not of high-cube containers. Hence, u_r^j is a binary variable indicating a level $r \in R$ of weight in the leg after j , with R being the set of ranges of weight defined by a minimum

Table 1
Sets, parameters and variables.

Sets	
C	Set of containers
N	Set of docks, index 0 for the dry-port
$\mathcal{E}$	Set of export containers
$\mathcal{I}$	Set of import containers
$\mathcal{F}$	Set of 40-foot containers $\mathcal{F} \subset C$
$\mathcal{T}$	Set of 20-foot containers $\mathcal{T} \subset C$
X, Y, Z	Sets of the 3-D coordinates
$\mathcal{X}^{\mathcal{F}}$	Set of bays for 40-foot containers
$\mathcal{X}^{\mathcal{T}}$	Set of bays for 20-foot containers
R	Ranges of weights for KG_{max} computation
Parameters	
dX_x	Distance of bay x from the stern of the ship, $x \in X$
dY_y	Distance of row y from the port side of the ship, $y \in Y$
min_r, max_r	Respectively minimum weight and maximum weight for range r , $r \in R$
D_{ij}	1 if container i is destined to/originated at dock j , 0 otherwise
W_i	Weight of a container i
W_0	Weight of the empty barge
H_i	Height of a container i
H^f	Distance from the keel to the stowage floor of the ship
T_i	0 if container i is standard, 1 if high-cube
U_i	0 if container i is 20 TEU, 1 if 40 TEU
KG_{max}^{zcr}	KG_{max} value for a stowage with top layer $z \in Z$, presence of high-cube containers $c \in \{1, 2\}$, weight level $r \in R$
KM^k	KM value for weight level k
M	A large value
Variables	
$\delta_{xyz,j}^i$	Binary variable indicating whether container $i \in C$ is in slot xyz , $x \in X, y \in Y, z \in Z$, and in the leg after dock $j \in N$
$h_{i,j}$	Height of a stowed container from the keel, $i \in C, j \in N$
w_j	Cargo weight in the leg after $j \in N$
l_j^z	Binary variable indicating whether layer z is the top layer, for $z \in Z, j \in N$
s_j^1	Binary variable, equals 1 if only standard containers are in the stowage, for $j \in N$
s_j^2	Binary variable, equals 1 if there is at least a high-cube container, for $j \in N$
u_j^r	Binary variable indicating a level of weight for the stowage in leg j , for $r \in R$, for $j \in N$
kg_{max-j}	KG_{max} value in leg j , for $j \in N$
$kg_{i,j}$	KG value of container i in the leg after j , for $i \in C, j \in N$
km_j	KM value in the leg after j , for $j \in N$

(min_r) and maximum (max_r) value; l_j^z is a binary variable indicating whether layer $z \in Z$ is the top occupied layer in the leg after j ; s_j^2 is a binary variable indicating, if 1, that there is at least a high-cube container in the leg after j , whereas s_j^1 is 1 if there are only standard containers in the stowage.

A list of all sets, variables and parameters is reported in Table 1.

We now present the mathematical formulation. Henceforth, for conciseness, $\sum_{xyz}$ will stand for $\sum_{x \in X} \sum_{y \in Y} \sum_{z \in Z}$, and $\forall xyz$ will stand for $\forall x \in X, y \in Y, z \in Z$. This simplification will apply for all possible combinations of indexes. We start with the objective function:

$$\max \sum_{xyz} \sum_{i \in \mathcal{E}} \delta_{xyz,0}^i \left(1 + U_i \right) + \sum_{xyz} \sum_{i \in \mathcal{I}} \delta_{xyz,|N|}^i \left(1 + U_i \right) \quad (4)$$

The objective function (4) maximizes the number of TEU placed on board; hence, two 20-foot containers count as one 40-foot container. We notice that the summations are over the first and last leg of the tour, as they contain all stowed containers, export and import containers respectively.

The following first set of constraints concerns the right physical positioning of a container on board.

$$\text{s. t.} \quad \sum_{x \in \mathcal{X}^{\mathcal{T}}} \sum_{yz} \delta_{xyz,j}^i \leq 1 \quad \forall i \in \mathcal{T}, j \in N \quad (5)$$

$$\sum_{x \in \mathcal{X}^{\mathcal{F}}} \sum_{yz} \delta_{xyz,j}^i = 0 \quad \forall i \in \mathcal{T}, j \in N \quad (6)$$

$$\sum_{x \in \mathcal{X}^F} \sum_{yz} \delta_{xyz,j}^i \leq 1 \quad \forall i \in \mathcal{F}, j \in N \quad (7)$$

$$\sum_{x \in \mathcal{X}^T} \sum_{yz} \delta_{xyz,j}^i = 0 \quad \forall i \in \mathcal{F}, j \in N \quad (8)$$

$$\sum_{i \in \mathcal{F}} \delta_{xyz,j}^i + 0.5 \sum_{i \in \mathcal{T}} \delta_{(x+1)yz,j}^i + 0.5 \sum_{i \in \mathcal{T}} \delta_{(x-1)yz,j}^i \leq 1 \quad \forall x \in \mathcal{X}^F, yz, j \in N \quad (9)$$

$$\sum_{i \in \mathcal{T}} \delta_{xyz,j}^i \leq 1 \quad \forall x \in \mathcal{X}^T, yz, j \in N \quad (10)$$

$$0.5 \sum_{i \in \mathcal{T}} \delta_{x+1yz-1,j}^i + 0.5 \sum_{i \in \mathcal{T}} \delta_{x-1yz-1,j}^i + \sum_{i \in \mathcal{F}} \delta_{xyz-1,j}^i - \sum_{i \in \mathcal{F}} \delta_{xyz,j}^i \geq 0 \quad \forall x \in \mathcal{X}^F, y, z > 1, j \in N \quad (11)$$

$$\sum_{i \in \mathcal{T}} \delta_{xyz-1,j}^i - \sum_{i \in \mathcal{T}} \delta_{xyz,j}^i \geq 0 \quad \forall x \in \mathcal{X}^T, y, z > 1, j \in N \quad (12)$$

From constraints (5)–(8), 20-foot containers can be placed only once in 20-foot spots, and cannot be placed in 40-foot containers slot, and inversely for 40-foot containers. With (9) we impose that a slot can be occupied either by a 40-foot container or by at most two 20-foot containers; these constraints also impose that a 40-foot spot is occupied at most by a 40-foot container. Inequalities (10) make sure that a 20-foot spot is occupied by one 20-foot only. Constraints (11) and (12) impose respectively that a 40-foot is supported by one 40-foot or two 20-foot containers, and that a 20-foot is only supported by another 20-foot. Note that in (11), we write $x \in \mathcal{X}^F$; hence for those summations with $i \in \mathcal{T}$, we have $x + 1$ and $x - 1$, following the system used in Ambrosino et al. (2004).

The second set of constraints concerns the presence of export and import containers on board and the imposition of no over-stowage.

$$\sum_{xyz} \delta_{xyz,j}^i \leq 0 \quad \forall i \in \mathcal{E}, j \in N/\{0\}; D_{ij} = 1 \quad (13)$$

$$\sum_{xyz} \delta_{xyz,j-1}^i \leq 0 \quad \forall i \in \mathcal{I}, j \in N/\{0\}; D_{ij} = 1 \quad (14)$$

$$\delta_{xyz,j-1}^i = \delta_{xyz,j}^i \quad \forall xyz, i \in \mathcal{C}, j \in N/\{0\}; D_{ij} = 0 \quad (15)$$

$$\sum_{s \in \mathcal{C}} \sum_{h=z+1}^{|Z|} \delta_{xyh,j-1}^s \left(1 - D_{sj}\right) \leq 1 - \delta_{xyz,j-1}^i \quad \forall x \in \mathcal{X}, y, z < |Z|, i \in \mathcal{E} \\ j \in N: D_{ij} = 1 \quad (16)$$

$$\sum_{s \in \mathcal{F}} \sum_{h=z+1}^{|Z|} \delta_{x+1yh,j-1}^s \left(1 - D_{sj}\right) \leq 1 - \delta_{xyz,j-1}^i \quad \forall x \in \mathcal{X}^T, y, z < |Z|, i \in \mathcal{E} \cap \mathcal{T}, \\ j \in N: D_{ij} = 1 \quad (17)$$

$$\sum_{s \in \mathcal{F}} \sum_{h=z+1}^{|Z|} \delta_{x-1yh,j-1}^s \left(1 - D_{sj}\right) \leq 1 - \delta_{xyz,j-1}^i \quad \forall x \in \mathcal{X}^T, y, z < |Z|, i \in \mathcal{E} \cap \mathcal{T}, \\ j \in N: D_{ij} = 1 \quad (18)$$

Constraints (13) impose that an export container must not be on board in the leg after its discharge node. Analogously (14) impose that an import container must not be on board in the leg before its pickup node. Equalities (15) make sure that for two consecutive legs of the route, excluding either the ones with “0” or the “reference” node ($D_{ij} = 1$) as intermediate node j , the position of the container on board must be the same. From inequalities (16)–(18) over-stowage is avoided. Specifically, an export container destined to a dock j in tier $z < |Z|$ should not have any container s above, unless container s has to be discharged too at j . (16) are specific for 40-foot containers, whereas (17) and (18) are for 20-foot containers. As import containers are all discharged at the dry-port, they are not subject to over-stowage.

The next constraints, from (19)–(22), concern the computation of the height of a positioned container from the keel. (19) and (20) are related to the computation for 40-foot containers, whereas (21) and (22) for 20-foot containers.

$$h_{i,j} \geq H^f + H_i + \sum_{h \in Z: h < z} \sum_{w \in \mathcal{F}} \delta_{xyh,j}^w H_w + \sum_{h < z} \sum_{w \in \mathcal{T}} \delta_{x-1yh,j}^w H_w - \left(1 - \delta_{xyz,j}^i\right) M \\ \forall x \in \mathcal{X}^F, y, z, i \in \mathcal{F}, j \in N \quad (19)$$

$$h_{i,j} \leq H^f + H_i + \sum_{h \in Z: h < z} \sum_{w \in \mathcal{F}} \delta_{xyh,j}^w H_w + \sum_{h < z} \sum_{w \in \mathcal{T}} \delta_{x-1yh,j}^w H_w + \left(1 - \delta_{xyz,j}^i\right) M$$

$$\forall x \in \mathcal{X}^f, y, z, i \in \mathcal{F}, j \in N \quad (20)$$

$$h_{i,j} \geq H^f + H_i + \sum_{h \in Z: h < z} \sum_{w \in \mathcal{T}} \delta_{xyh,j}^w H_w - \left(1 - \delta_{xyz,j}^i\right) M$$

$$\forall x \in \mathcal{X}^f, y, z, i \in \mathcal{T}, j \in N \quad (21)$$

$$h_{i,j} \leq H^f + H_i + \sum_{h \in Z: h < z} \sum_{w \in \mathcal{T}} \delta_{xyh,j}^w H_w + \left(1 - \delta_{xyz,j}^i\right) M$$

$$\forall x \in \mathcal{X}^f, y, z, i \in \mathcal{T}, j \in N \quad (22)$$

Note that in (19) and (20) the height of a 40-foot container is computed by including the height of the containers below. If the container is supported by two 20-foot containers, we take arbitrarily the height of the container at the fore part of the slot.

The following inequalities concern the computation of necessary elements to evaluate stability.

$$w_j = \sum_{xyz} \sum_{i \in C} \delta_{xyz,j}^i W_i \quad \forall j \in N \quad (23)$$

$$kg_{i,j} = h_{i,j} - 0.5H_i \leq \sum_{xyz} \delta_{xyz,j}^i M \quad \forall i \in C, j \in N \quad (24)$$

$$\sum_{xy} \delta_{xy,z,j}^i \leq \sum_{h=z}^{|Z|} l_j^h \quad \forall i \in C, z > 1, j \in N \quad (25)$$

$$\sum_{z=2}^{|Z|} l_j^z = 1 \quad \forall j \in N \quad (26)$$

$$\sum_{xyz} T_i \delta_{xyz,j}^i \leq s_j^2 \quad \forall i \in C, j \in N \quad (27)$$

$$s_j^1 + s_j^2 = 1 \quad \forall j \in N \quad (28)$$

$$w_j \geq \min_r - (1 - u_j^r)M \quad \forall j \in N, r \in R \quad (29)$$

$$w_j \leq \max_r + (1 - u_j^r)M \quad \forall j \in N, r \in R \quad (30)$$

$$\sum_{r \in R} u_j^r = 1 \quad \forall j \in N \quad (31)$$

$$kg_{\max-j} \leq KG_{\max}^{zcr} + (1 - l_j^z)M + (1 - s_j^c)M + (1 - u_j^r)M$$

$$\forall z > 1, c \in \{1, 2\}, r \in R, j \in N \quad (32)$$

$$km_j \leq KM^r + (1 - u_j^r)M \quad \forall r \in R, j \in N \quad (33)$$

$$km_j \geq KM^r - (1 - u_j^r)M \quad \forall r \in R, j \in N \quad (34)$$

With (23) the loaded weight in the leg after terminal j is computed. (24) compute the KG for each container, assuming that the center of gravity of each container is always at the middle of its height. From constraints (25) and (26) we compute values l_j^z to keep track of the highest occupied tier in the leg after j . We note that $KG_{\max}$ values are usually not available for stowage plans consisting of only one tier. Moreover, the model is free to choose for l_j^z any tier larger than the highest occupied one; however, larger tiers entail smaller $KG_{\max}$ values. From constraints (27) and (28) we compute the variables indicating the presence of high-cube containers. From (29)–(31) we compute the corresponding range of weight for the loaded weight. Finally, with (32), we compute the quantity $kg_{\max-j}$ for each transport leg, and with (33) and (34) the KM values.

With the following constraints, we constrain KG values, angles of list and trim.

$$GM_j = km_j - \frac{\overline{KG}_0 W_0 + \sum_{xyz} \sum_{i \in C} W_i kg_{i,j}}{W_0 + w_j} \quad \forall j \in N \quad (35)$$

$$kg_{\max-j} \geq \frac{\overline{KG}_0 W_0 + \sum_{xyz} \sum_{i \in C} W_i kg_{i,j}}{W_0 + w_j} \quad \forall j \in N \quad (36)$$

$$\eta \geq \frac{\sum_{xyz} \sum_{i \in C} W_i \delta_{xyz,j}^i (dY_y - dY_0)}{(W_0 + w_j) GM_j} \quad \forall j \in N \quad (37)$$

$$\gamma \geq \frac{\sum_{xyz} \sum_{i \in C} W_i \delta_{xyz,j}^i (dX_x - dX_0)}{MCT/L} \quad \forall j \in N \quad (38)$$

With equalities (35) we compute GM . With inequalities (36) we constrain KG for each transport leg. Finally, with constraints (37) and (38) we compute respectively the tangent values of the angles of list ($\tan(\theta)$) and trim ($\tan(\alpha)$), which are limited by η and γ . We notice the non-linearity of the constraints, from (35)–(37), which can be linearized by using auxiliary real variables. We refer the interested reader to Appendix A to find the linearization technique for constraints (36). Finally, all variables are real, except for the binary variables summarized in Table 1.

Other constraints or features can be added to this formulation. For the sake of conciseness, we provide the following list:

- Minimum weight in a specific transport leg for bridges
- Maximum weight in a specific transport leg for structural limits of the ship or river depth
- A container can be supported only by a heavier one
- Specific constraints for special containers: reefer, open top, 45-foot

4.2. A model for a stable barge during handling

The second model concerns the computation of a feasible handling sequence that maintains the barge stable while processed by the quay-crane. For this model, variable $\delta_{xyz,j}^i$ is a constant input from the previous stowage model. In turn, other stowage-related variables must be defined in order to check intermediate stowage configurations during loading/unloading. In particular, assuming that at a terminal one container at a time is handled, the number of intermediate stowage plans equals the number of containers handled, say T , minus one, since the last handling operation will generate the final configuration. See Fig. 6 for a graphical example.

Hence, let us define a variable $\Delta_{xyz,k}^{i,t}$ which indicates for a terminal k , the position of container i during handling phase t , where $t \in \{0 \dots T\}$. For $t = 0$ and $t = T$ we have respectively $\delta_{xyz,k-1}^i = \Delta_{xyz,k}^{i,0}$ and $\delta_{xyz,k}^i = \Delta_{xyz,k}^{i,T} \quad \forall xyz, i \in C$. Also, we identify with $\bar{E}$ and $\bar{I}$ respectively the subsets of export and import containers handled at terminal k , and we define $C^k = \bar{E} \cup \bar{I}$. Finally, let p_i be a variable which compute the number of times a container i is in stowage during the handling phases $t \in \{1 \dots T-1\}$.

We now present the mathematical model for a specific terminal k .

$$\sum_{xyz} \sum_{t=1}^{T-1} \Delta_{xyz,k}^{i,t} = p_i \quad \forall i \in C^k \quad (39)$$

$$p_i = T - 1 \quad \forall i \in C^k / (\bar{E} \cup \bar{I}) \quad (40)$$

$$\Delta_{xyz,k}^{i,t} \geq \Delta_{xyz,k}^{i,t+1} \quad \forall i \in \bar{E}, t \in 0 \dots T-1, xyz \quad (41)$$

$$\Delta_{xyz,k}^{i,t} \leq \Delta_{xyz,k}^{i,t+1} \quad \forall i \in \bar{I}, t \in 0 \dots T-1, xyz \quad (42)$$

Fig. 6. Stowage phases during handling operations. In this example, 3 containers must be discharged and 2 loaded.

$$p_i + 1 \leq p_c + (1 - \Delta_{xyz,k}^{c,0})M + (1 - \Delta_{xyh,k}^{i,0})M \quad \forall i \in \bar{\mathcal{E}} \cap \mathcal{T}, c \in \bar{\mathcal{E}} \cap \mathcal{T}: c \neq i$$

$$x \in \mathcal{X}^T, y, z < |Z|, h > z \quad (43)$$

$$p_i + 1 \leq p_c + (1 - \Delta_{xyz,k}^{c,T})M + (1 - \Delta_{xyh,k}^{i,T})M \quad \forall i \in \bar{\mathcal{I}} \cap \mathcal{T}, c \in \bar{\mathcal{I}} \cap \mathcal{T}: c \neq i$$

$$x \in \mathcal{X}^T, y, z < |Z|, h > z \quad (44)$$

$$p_i + 1 \leq p_c + (1 - \Delta_{xyz,k}^{c,0})M + (1 - \Delta_{xyh,k}^{i,0} - \Delta_{x+1,yh,k}^{i,0} - \Delta_{x-1,yh,k}^{i,0})M \quad \forall i \in \bar{\mathcal{E}}, c \in \bar{\mathcal{E}} \cap \mathcal{F}: c \neq i, x \in \mathcal{X}^F$$

$$y, z < |Z|, h > z \quad (45)$$

$$p_i + 1 \leq p_c + (1 - \Delta_{xyz,k}^{c,T})M + (1 - \Delta_{xyh,k}^{i,T} - \Delta_{x+1,yh,k}^{i,T} - \Delta_{x-1,yh,k}^{i,T})M \quad \forall i \in \bar{\mathcal{I}}, c \in \bar{\mathcal{I}} \cap \mathcal{F}: c \neq i, x \in \mathcal{X}^F$$

$$y, z < |Z|, h > z \quad (46)$$

$$\sum_{i \in \mathcal{F}} \Delta_{xyz,k}^{i,t} + 0.5 \sum_{i \in \mathcal{T}} \Delta_{(x+1)yz,k}^{i,t} + 0.5 \sum_{i \in \mathcal{T}} \Delta_{(x-1)yz,k}^{i,t} \leq 1 \quad \forall t \in 1 \dots T-1, x \in \mathcal{X}^F, yz \quad (47)$$

$$p_i \neq p_c \quad \forall i \in \bar{\mathcal{E}}, c \in \bar{\mathcal{E}}: c \neq i \quad (48)$$

$$p_i \neq p_c \quad \forall i \in \bar{\mathcal{I}}, c \in \bar{\mathcal{I}}: c \neq i \quad (49)$$

$$p_i \neq T-1-p_c \quad \forall i \in \bar{\mathcal{E}}, c \in \bar{\mathcal{I}} \quad (50)$$

With constraints (39), variable p_i is computed for each container i present in the stowage plans of terminal k ; this subset is identified with C^k . Equalities (40) impose that containers that are not to be loaded/unloaded at terminal k must be in stowage during the whole handling period. Inequalities (41) and (42) make sure that: we do not place containers in different slots than the ones decided upfront; once a container is either loaded or unloaded it will respectively stay in or out for the remaining handling time. From constraints (43)–(46), we impose that if containers i and c are in the same stack and c is below i , then $p_c > p_i$. (43) and (44) are specific for 20-foot containers. With (47), we impose that a slot cannot be occupied at the same time by multiple containers. Finally, from constraints (48)–(50), we impose that containers are handled at different times.

In terms of stability calculations, KG , list and trim restrictions have to be considered. In this regard, constraints from (19)–(38) have to be replicated for $t \in \{1 \dots T-1\}$ and variable $\delta_{xyz,j}^i$ substituted with $\Delta_{xyz,j}^{i,t}$. Finally, the same linearization technique must be performed.

From the model, there are two ways to determine the loading/unloading sequence. The first is to compare variables $\Delta_{xyz,k}^{i,t}$ and $\Delta_{xyz,k}^{i,t+1}$ for each time $t \leq T-1$ and check which container i is added or removed at time $t+1$. The second alternative is to check for how many handling steps p_i container i is in stowage. The interested reader can find the full model in Appendix C.

5. Solution framework

To solve the stowage problem proposed, we will develop a hybrid framework consisting of a metaheuristic and an MILP solver. The metaheuristic is the main entity that searches for feasible stowage plans. The MILP solver is called upon at the end of the computation to compute a feasible handling sequence for the final stowage. The next subsections will then be dedicated to the description of the metaheuristic approach. First, we will describe the conditions necessary to construct feasible solutions, and then we will go on to describe the algorithms.

Henceforth, and in this section only, a solution is intended to be feasible as long as containers are positioned correctly, that is, without floating or (partially) unsupported containers, and if there is no over-stowage (Constraints (16)–(18)). Therefore, when constructing a “feasible” solution, we will not consider stability requirements. In turn, the metaheuristic will construct other feasible stowage plans and will stop as soon as all stability measurements are respected.

5.1. Solution properties

Let us consider a given route made of a certain number l of legs. An important property of this problem is that, in the first and last legs, the stowage consists of only export and only import containers (Constraints (13) and (14)), respectively. Moreover, due to the fact that over-stowage is not permitted, the position of a placed container is fixed during its journey (Constraints (15)). Hence, by constructing the stowage plans of the first and last legs, we can extrapolate the stowage plans for the intermediate legs.

In order to generate feasible stowage plans in the first and last leg - say, 1 and l - the following two necessary conditions must be met.

Condition 5.1. An export container i is correctly positioned in the stowage plan of leg 1 if: in leg l , all import containers in the same stack have a reference dock that equals or is successor in the route to the one of i ; in leg 1, container i supports containers with a reference dock that equals or is a predecessor in the route to the one of i ; and, in leg 1, container i is stowed above containers with a reference dock that equals or is a successor in the route to the one of i .

Condition 5.2. An import container i is correctly positioned in the stowage plan of leg l if: in leg 1, all export containers in the same stack have a reference dock that equals or is a predecessor in the route to the one of i ; in leg l , container i supports containers with a

reference dock that equals or is a successor in the route to the one of i ; and, in leg l , container i is stowed above containers with a reference dock that equals or is a predecessor in the route to the one of i .

5.2. A Variable Neighborhood Search (VNS) metaheuristic

Here we will describe the development of a Variable Neighborhood Search (VNS) metaheuristic. VNS is a popular stochastic algorithm that explores a set of predefined neighborhoods, chosen randomly or systematically, to find and escape local optima. For a description and general templates, we refer to [Talbi \(2009\)](#). The proposed algorithm chooses neighborhoods based on fixed probability and also contains operators that are activated after a set number of iterations. Furthermore, the algorithm is endowed with a mechanism based on the Boltzman distribution for solution acceptance. Finally, labels are used to impede the generation of infeasible solutions.

The method will start with the generation of a feasible solution and, by means of two operators, will investigate the solution state space. In terms of objective function, a cost is added for exceeding the stability measurements previously described. Thereby, the proposed algorithm will try to minimize the number of non-stowed containers and the excess for the stability requirements. Hence, the ideal optimum, or lower bound, of any instance would be zero.

We will first introduce the greedy algorithm for the initial solution and, after that, the metaheuristic. In the appendix, the interested reader can find the pseudo-codes for the algorithms and a flow chart.

5.2.1. A greedy algorithm to generate an initial solution

The generation of a feasible solution is inspired by a practical greedy algorithm. Captains typically try to group containers belonging to the same dock and stow them on a section of the deck. However, generally speaking, it is possible to stow a container in another section, as long as conditions 5.1 and 5.2 are respected. This strategy should prevent over-stowage, although it may be ineffective in terms of stability if the number of containers loaded and unloaded at a dock are very different.

The algorithm starts off by first focusing on the stowage of the export leg, namely the first leg of the route. We first sort export containers by the inverse order of related destination docks in the route. Next, for each dock, the related subset of containers is sorted by non-increasing weight. The aim of this sorting is threefold: first, to create stacks with containers mainly belonging to the same dock; second, to fill any incomplete stacks with containers that are going to be dropped earlier in the route in order to avoid over-stowage; and, third, to make sure that the weight of containers in the stacks decreases from the bottom. After this sorting, each container in the subset is allocated one-by-one to available slots (i.e., combinations of x and y coordinates) in the ship, if this satisfies the physical constraints.

Once the procedure has processed all export containers, each occupied slot will be tagged with an “export” label. This label refers to the dock farthest from the start of the route, the one where an export container in the stack has to be unloaded. Given the fact that slots can be occupied by 20-foot containers, a tag is also assigned to each half of the slot. This label is necessary so that, during the insertion of import containers, it will prevent an import being inserted - say, one to be loaded at dock j - in a slot where export containers will be unloaded at docks $i > j$.

Next, the greedy algorithm will focus on stowing import containers in the import leg, that is, in the last section of the route. This procedure is analogous to the previous one for export containers, except for sorting first containers by the order of related origin docks in the route. In addition, a container is stowed on a slot only if the “export” label of the stack is less or equal to the dock of the pointed import container. At the end of this iteration, each slot with a formed stack will receive an “import label”. This label refers to the dock nearest to the start of the route, where an import container in the stack has to be loaded.

Once export and import containers are allocated, the greedy algorithm creates the intermediate stowage plans and finally computes the stability features. If a stability constraint is not respected, an associated penalty will be assigned as a cost. This will allow the metaheuristic to also take account of infeasible solutions that do not respect stability. If all containers are allocated and stability is achieved, the procedure stops. Otherwise, the metaheuristic will be initiated.

5.2.2. Metaheuristic

The metaheuristic is endowed with four operators to modify the current solution and search the solution state space. These operators can be assigned to two classes. The first class, “Search” operators, includes two operators that modify the current stowage plan without changing the number of containers on board. The second class, “Insertion and Removal” operators, includes two operators that try to change the number of containers on board after specific conditions are met. Finally, the metaheuristic stops when the stowage meets the stability requirements.

5.2.3. Search operators

The first operator, O_1 , of this class swaps stacks. The operator first selects a non-empty stack randomly. Next, another stack, which can be empty, is randomly selected. After that, the two stacks are swapped. This change will affect all transport legs. Since the position of the containers does not change for the tier, the kg values will not change. On the other hand, trim and list may change. To be more specific, the trim will change if the new stack positions differ in terms of bays, whereas the list will change when the positions differ in terms of rows.

The second operator, O_2 , changes container positions by either swapping containers or moving a container into another stack. If the first container is an export, the operator will focus on the first leg of the route; if it is an import, then on the last. After a container i

is selected, a stack, different from the one where i was initially placed, is tried out. The operator will first check whether the container is eligible for insertion in that stack, taking into consideration the labels created to avoid over-stowage. In particular, if a container is an export, it can be moved to a stack only if the corresponding stack in the final leg has a label that is greater or equal to the index of the dock where the container has to be discharged; vice versa, for import containers. Next, the operator will evaluate whether the container can just be moved or swapped with another when the stack is full or when a tier is composed of a spare 20-foot container. Finally, the operator tries for a maximum number of times to find a feasible move. This operator is particularly useful either when containers are not inserted yet or when kg restrictions are not met. This may, however, also affect the trim and list, although these effects may be marginal compared to a “stack” exchange. Furthermore, it may also be used for shaking up the current solution structure, by accepting all changes proposed for a set number of iterations.

The two operators are selected according to a fixed probability distribution. Once selected, the operator will operate first on the stowage of the initial and final leg, and finally will update the stowage of the intermediate legs. Once a new solution is generated, it is accepted if better, or it is either discarded or accepted if worse, with the probability for this provided by the Boltzmann distribution (Kirkpatrick et al., 1983). Starting from an initial solution s with cost $C(s)$, a new feasible solution s' with cost $C(s') > C(s)$ is either rejected or accepted with a probability computed as $P(T) = \min\left\{e^{\frac{C(s) - C(s')}{T}}, 1\right\}$, where T is the parameter called “Temperature,” which is fixed throughout the search in a multi-start approach, with different values run in parallel.

5.2.4. Insertion and removal operators

When the greedy algorithm does not succeed in packing all containers in the initial solution, the metaheuristic activates the insertion operator for a given number of iterations. More specifically, the procedure shakes the solution with the “Search” operators and tries randomly to include the non-stowed containers. This operator is integrated within O_2 , which will also randomly choose containers left out. The aim of the insertion operator is to reach a “feasible” stowage plan with all containers on board.

Once the iterations for the insertion operator are completed or when the greedy algorithm has found a “feasible stowage” with all containers on board, the metaheuristic activates the removal operators. In this stage, the metaheuristic tries for a specific number of iterations to reach stability with the given set of containers. At the end of the iterations, if stability is not reached, the metaheuristic removes the heaviest container from the stowage in the transport leg most affected by stability requirements. After its removal, the metaheuristic again runs for a fixed number of iterations.

6. Numerical section

In this section, we will test the performance of the metaheuristic and CPLEX 12.6 on a set of instances. We will solve the first model with both solution methods, and compare the results in terms of solution quality and timing. In addition, the second model will be implemented to come up with the final solution of the metaheuristic to check whether a feasible handling sequence exists. Thus, the second model can be considered as an extra component of the decision support system that will discard the metaheuristic solution in case stability during handling cannot be attained.

The experimental framework is based on the case study of a Dutch inland terminal, active in the Brabant region, and acting as barge operator. Several voyages during the week to the Port of Rotterdam area are the norm, and more than 30 terminals are visited in total. For this experiment, we will focus on a specific ship, whose technical documentation is available. From this point on, we will refer to this ship as the “VM.” The technical features comprise ship sizes, TEU capacity, bare-boat weight W_0 , KG_{max} , and KM values. The VM can handle 108 TEU in total, to be stowed in 6 bays, 3 rows, and in 3 tiers at most. It can be classified as small-size barge (Pielage et al., 2007). The main features are summarized in Appendix B.

Our test consists of a set of 25 randomly generated instances, along with 5 instances drawn from a real-world data set. The random instances provide an indication as to the performance of the metaheuristic against CPLEX 12.6 and the behavior of different parameterizations. The real instances provide insights into current stowage practices and an indication of the savings possible by implementing such a methodology for the proposed stowage problem.

6.1. Metaheuristic parameterization, CPLEX and technical remarks

The greedy algorithm and the metaheuristic are coded in C++ and run on an Intel(R) Core(TM)2 DUO machine with 2.93 GHz and 8.00 GB RAM of memory. The metaheuristic has a few parameters that need to be set: number of iterations, acceptance criteria, and probability for operator selection.

For the class “Search” operators, we will test several fixed probabilities in order to choose one of the two operators. To be specific, the probability of choosing the first operator, swapping stacks, is in the range of 0, 10, 30, 50, 70 and 90. We will not test probability 100%, since the first operator is not able to change KG values. Finally, operator O_2 will try to find a feasible move for a maximum of 10 times.

As for the insertion operator, if not all containers are initially stowed, the algorithm runs for 1 million iterations, during which the insertion operator can be activated with a certain probability. Once these iterations are terminated or when the initial solution already contains all containers in stowage, the algorithm runs for sets of 1 million iterations. After each set, if a feasible stowage arrangement is not found, the removal operator removes a container. This number of iterations was deemed workable during a preliminary experimental phase.

The acceptance criteria are based on the Boltzmann distribution, and several values for T are tried in parallel. The preliminary

Table 2

Results on a set of randomly generated instances. The “GAP” column reports the gap between the metaheuristic solution and the best integer solution found by CPLEX. Optimal solutions are highlighted in boldface.

#	TEU	MH		CPLEX - 3600 s				CPLEX - 7200 s				GAP
		Solution	Time	Best integer	Best Node	gap	Time	Best integer	Best Node	gap	Time	
1	168	168	0.01	168	168	0.00	243.70	168	168	0.00	250.00	0.00
2	174	174	0.01	174	174	0.00	116.92	174	174	0.00	119.54	0.00
3	175	175	0.01	175	175	0.00	454.88	175	175	0.00	453.80	0.00
4	176	176	0.09	176	176	0.00	300.25	176	176	0.00	304.49	0.00
5	176	176	2.62	175	176	0.01	3093.30	175	176	0.01	3164.05	0.01
6	176	176	0.02	169	176	0.04	2404.83	175	176	0.01	7200.58	0.01
7	177	177	0.04	177	177	0.00	857.39	177	177	0.00	873.80	0.00
8	178	178	0.01	176	178	0.01	2016.41	178	178	0.00	6498.99	0.00
9	182	182	87.20	153	182	0.19	3410.18	180	182	0.01	4819.76	0.01
10	185	185	0.02	185	185	0.00	859.89	185	185	0.00	868.26	0.00
11	186	182	118.60	156	186	0.19	1088.22	179	186	0.04	6638.66	0.02
12	189	189	0.01	189	189	0.00	2158.71	189	189	0.00	2202.45	0.00
13	189	187	125.72	185	189	0.02	1414.01	185	189	0.02	1410.69	0.01
14	189	187	128.36	156	189	0.21	2372.71	182	189	0.04	7200.19	0.03
15	190	190	0.01	180	190	0.06	3600.29	180	190	0.06	3546.70	0.06
16	191	191	0.05	187	191	0.02	1110.12	187	191	0.02	1172.94	0.02
17	191	191	0.13	188	191	0.02	978.53	188	191	0.02	964.16	0.02
18	191	191	0.02	188	191	0.02	3379.40	188	191	0.02	3253.91	0.02
19	192	192	0.04	192	192	0.00	1365.45	192	192	0.00	1391.00	0.00
20	193	193	30.52	191	193	0.01	1490.36	191	193	0.01	1516.36	0.01
21	194	194	0.01	194	194	0.00	2202.61	194	194	0.00	2185.85	0.00
22	195	187	119.72	143	193	0.35	2027.95	187	193	0.03	7200.24	0.00
23	196	170	124.42	0	177	-	3600.21	147	177	0.20	4914.24	0.16
24	197	191	1118.70	193	197	0.02	2398.87	193	197	0.02	2408.61	-0.01
25	202	199	121.01	199	202	0.02	3600.19	199	202	0.02	3546.72	0.00

experimentation showed stable results with $T \in [0.1, 100]$.

Finally, the main model will be integrated in CPLEX using the Concert Technology in C++ . We will not include the second model in CPLEX in the experimentation, since this would only increase the size of the problem without producing any valuable information in terms of performance. Due to large memory requirements, for CPLEX, we used a high performance machine, with a 24 Intel Xeon 2.5 GHz core and 128 GB of internal memory. Overall, a number of ten parallel threads was found to be the best performing. We will test two different time limits: 1 h and 2 h.

6.2. Results on randomly generated instances

The instances have a certain variety in terms of: presence of 20-foot containers, number of stops, ratio of import/export containers, and ratio of regular/high-cube containers. Instances were generated according to the following distributions:

- Number of containers: *Unif* [95, 110]
- Route stops: *Unif* [2, 6]
- Weight (tons): *Unif* [10, 35]

Furthermore, each container has a 20% probability of being 20-foot and an equal probability of being either import or export. Finally, 40-foot containers have a 70% chance of being high-cube. These probabilities follow the trend of the available data set. Based on our instances, the number of variables $\delta_{vz,j}^i$ can be expected to be about 100,000.

Results for the 25 instances are shown in Table 2. For each instance, we indicate the total number of TEU. For the metaheuristic, labeled as “MH,” we report the best solution found among all parameterizations, along with related timing to reach the solution. For CPLEX, we show the results of the two experimentations with 1-h and 2-h time limits, and for each we report the best integer solution (lower bound for the maximization problem), the best node (upper bound), and the timing when the best integer solution is found during the set time frame. These values are automatically displayed by CPLEX. Finally, in the last column (GAP), we show the gap between the best solutions from CPLEX and the metaheuristic.

These results show very good performance for the proposed algorithm. Overall, the average gap between the CPLEX best solution and the metaheuristic one is 1.3%. The maximum gap is 16%, and, in one case, CPLEX finds a better solution with gap of -1%. For Instance 17, CPLEX cannot find a meaningful integer solution before the time limit of 1 h. The time performance is notable. In general, the metaheuristic is able to find its best feasible solution within seconds, whereas CPLEX requires several minutes and in most cases more than an hour. This is a crucial aspect to consider, given the limited time frame for a captain to make a stowage plan. For some instances, the metaheuristic takes several seconds before stopping the computation; this is because the algorithm needs to remove containers from the stowage to reach stability and the removal operator is activated only after a fixed number of iterations.

Table 3

Results of the metaheuristic for different parameterizations.

Instance	T = 0.1				T = 50				T = 100				SA			
	p = 0	p = 10	p = 50	p = 90	p = 0	p = 10	p = 50	p = 90	p = 0	p = 10	p = 50	p = 90	p = 0	p = 10	p = 50	p = 90
5	168	176	176	174	166	174	174	174	170	174	174	173	168	174	175	172
6	157	176	176	176	166	176	176	176	159	176	176	176	160	176	176	176
9	143	182	174	176	151	178	174	182	155	180	180	180	160	176	176	176
23	141	168	170	168	161	170	168	168	168	168	168	168	168	168	168	168

Finally, we applied the second model to the stowage plans found by the metaheuristic, and a feasible handling sequence was always found within seconds.

The metaheuristic has shown a remarkable consistency using several parameterizations. As soon as the right number of containers is in stowage, a stable stowage is reached relatively quickly. In the following table, for some instances, we show the performances of the metaheuristic with a selection of temperatures and probabilities for choosing operator O_1 . We have also run the algorithm in a Simulated Annealing (SA) style. Thus, for each set of iterations, the temperature is decreased from 200 to 0.1, at a constant rate. (see Table 3).

The main conclusion that can be drawn from these tests is that the metaheuristic is consistent for several parameterization. Moreover, accuracy decreases, with zero probability of choosing O_1 . The SA approach is slightly outperformed by the single temperature one.

6.3. Results on real-world instances

We also conducted an experimentation on real-world instances, shown in Table 4. For each instance, we report the total number of TEU, the number of 20-foot containers, and the number of stops. Each instance tried matches a specific voyage. We added other containers to the instance, which could be packed in that specific time frame and which were not included in the voyage, but trucked (non-stowed). The reason is usually rejection because, for example, the additional containers would create an extra stop, hard to handle for the creation of the stowage plan, or possible over-stowage.

In Fig. 7, we show the utilization of the barge for each instance in each transport leg.

The results show possible savings and increased utilization of the VM. Potentially, more containers could be included if available. Furthermore, these experiments highlight the limitation of the current procedures. In fact, it is typical for stowage planners to compartmentalize the bays of the ship based on the destination, and to create stacks of containers destined for the same location (Li et al., 2018).

Compartmentalization is of course necessary for manual planning. It may prove to be very difficult for a human planner to mix container stacks and find a balance. However, this has evident drawbacks. For example, assume that for the first stop of the rotation we have a large portion of the export cargo, and this is placed in the ship's aft. When this is unloaded, and not enough cargo is loaded back, imbalance may occur. That is why, we often found in the data-set that the most loaded trips were the ones with fewer stops (1 or 2). In the instances proposed here, more stops were made. However, those trips were organized such that the first stops were mostly dedicated to unloading export containers and the remaining to loading import ones. Consequently, the stowage could be easily separated into two parts. See, for example, some trends in Fig. 7.

Finally, from the data-set turns out that the majority of trucked containers are import. It is clear that the captain is typically more focused on ensuring good navigation for one full way, export in our case. The loading of import containers in intermediate stops of a multi-port route may result in a stowage plan too complex to handle. This once more emphasizes the limitation of manual planning for complex routes with several stops, and with a mix of import and export containers at every dock.

7. Final remarks

In this paper, we tackled the stowage problem for inland shipping, considering a dry-port transportation system. The literature review showed that the majority of contributions were focused on large maritime vessels. However, most of the models are

Table 4

Results on a set of real-world instances.

Instance #	TEU	# 20-foot	Stops	Planning procedure	MH		
				stowed TEU	stowed TEU	Time	Improvement (%)
1	163	5	5	157	163	2.3	3.82
2	143	11	4	135	143	0.5	5.93
3	144	0	4	136	144	0.8	5.88
4	193	11	4	187	193	1.6	3.21
5	196	14	4	192	196	10.0	2.08

Fig. 7. Utilization of the barge in each transport leg.

recognized as being unusable in practice due to several approximations needed to reduce the solution state space. Moreover, stability measurements are typically approximated due to the possibility of compensating imbalance with ballast water. Quite the opposite is true for barges - the specific ship category under consideration here - which handle hundreds of containers at the most and are subject to strict stability regulations. Barges offer opportunities to develop accurate models, which are also recognizable and readily usable by practitioners.

The problem was first modeled mathematically, and a metaheuristic approach was then also developed. Our methodology turned out to be accurate and efficient versus an MILP solver when applied to a set of randomly generated instances, based on a real-world barge, whose technical features were available to us. The results showed the effectiveness and efficiency of our methodology for the stowage problem in inland shipping for container barges. In terms of complexity, even though the ship considered is a small-size barge, the results also showed the limitation of the MILP solver and the need for optimization algorithms to obtain fast solutions.

Experiments conducted in real-world instances showed the limitation of the current trial-and-error approach and the need for advanced systems to increase capacity utilization. In this regard, the proposed framework is currently being embedded in stowage-controlling software, available to us from our industrial partners. We believe that this project can potentially simplify barge-related stowage problems and, consequently, help reach those environment-related targets widely promoted by governments and port authorities worldwide.

In terms of comparison with maritime stowage plans, from the literature review, it turned out that it is typical to distinguish between Master planning and Slot planning (Parreño et al., 2016). In the former, containers are allocated to locations, and the way the total weight is divided over each location guarantees stability. In the latter, containers are allocated to specific slots and basic stowage rules are considered. In this paper, the two planning phases are combined. Solving an aggregated plan for large maritime vessels would certainly require considerable computational effort; that is why, in our opinion, the approach found in the literature is sensible. Moreover, the necessity of avoiding over-stowage is typical for the barge industry due to the reality of businesses that are often family-owned and whose margins are fairly limited (Blecker et al., 2008). In fact, the presence of transport alternatives, such as truck and rail, makes it rather inconvenient to generate extra handling costs.

Possible extensions for this study might involve adding intermediate stops at other inland terminals or allowing the exchange of containers between terminals. Although these research directions might be interesting to explore from a scientific perspective, they represent obsolete settings that do not promote efficiency and consolidation in current logistics systems (Konings et al., 2013; Roso et al., 2009). Finally, as mentioned in the introduction, barge operators are typically focused on maximizing utilization rather than schedules. This is quite different from maritime container ships, where decisions about stowage are also shared with the terminals due to the need to minimize the length of handling times. For future research, it might be interesting to assess the impact of inland shipping on the logistics of large terminals. This may motivate the development of models where the decisions of terminal and barge operators are shared. In this regard, platforms where barge operators and sea terminals communicate and jointly optimize planning are being developed. For instance, Nextlogic is a newly integrated system in the Port of Rotterdam, which supports information sharing and optimizes the rotation of the barge in real time, based on the current status of the port (Port of Rotterdam, 2019). It would be worthwhile investigating whether decisions on stowage plans could also be integrated in such a system, so that, for example, stowage plans could be more robust in terms of adaptation to barge rotation.

Acknowledgments

I would like to thank the industrial partners who provided relevant data and information for this study, particularly Desiré Savelkoul (Autena Marine) and Michel van Dijk (Inland Terminal Veghel). I am grateful to Wayne Roest and Eelco Timmerman for

collecting useful information on the real-world case. Finally, I would like to thank the Center for Information Technology of the University of Groningen for their support and for providing access to the Peregrine high performance computing cluster.

Appendix A. Linearization of non-linear constraints

Consider the following constraints:

$$kg_{max-j} \geq \frac{\overline{KG}_0 W_0 + \sum_{xyz} \sum_i \delta_{xyz,j}^i W_i kg_{i,j}}{W_0 + w_j} \quad \forall j \in N \quad (51)$$

The inequality can be rewritten as:

$$kg_{max-j} W_0 + kg_{max-j} w_j - \overline{KG}_0 W_0 \geq \sum_{xyz} \sum_i \delta_{xyz,j}^i W_i kg_{i,j} \quad \forall j \in N \quad (52)$$

Let us focus, for example, on the non-linear multiplication $kg_{max-j} w_j$, which can be rewritten as:

$$kg_{max-j} w_j = kg_{max-j} \sum_{xyz} \sum_i \delta_{xyz,j}^i W_i \quad (53)$$

Let us define an auxiliary variable $z_j^i = kg_{max-j} \sum_{xyz} \delta_{xyz,j}^i$. As kg_{max-j} is bounded by a known value that, for simplicity, we call M , the following constraints hold:

$$z_j^i \leq M \sum_{xyz} \delta_{xyz,j}^i \quad (54)$$

$$z_j^i \leq kg_{max-j} \quad (55)$$

$$z_j^i \geq kg_{max-j} - \left(1 - \sum_{xyz} \delta_{xyz,j}^i\right) M \quad (56)$$

$$z_j^i \geq 0 \quad (57)$$

Hence (53) can be rewritten as:

$$kg_{max-j} w_j = \sum_i z_j^i W_i \quad (58)$$

Appendix B. “VM” features

Table 5–7

Table 5
Main specifications of the “VM”.

Specification	Value
Length	105,00 meters
Length between perpendicular	100,00 meters
Width	9,51 meters
Bays	6
Rows	3
Max # Tiers	3
Max TEU	108
Maximum freight capacity	1.999,66 tons
Lightship weight	634 tons
KG_0	1,78 meters
H^f	0,52 meters

Table 6
Averaged KG_{max} values for the “VM”. Weight expressed in Tons.

Weight Range	KG_{max} - 2 tiers		KG_{max} - 3 tiers	
	Standard Containers only	With High cube	Standard Containers only	With High cube
$W \leq 800$	3.97	3.93	3.75	3.65
$800 < W \leq 1000$	3.73	3.69	3.54	3.45
$1000 < W \leq 1200$	3.60	3.56	3.43	3.35
$1200 < W \leq 1400$	3.49	3.48	3.35	3.29
$1400 < W < 2000$	2.82	2.80	2.66	2.56

Table 7
Averaged KM values for the “VM”. Weight expressed in Tons.

Weight Range	KM Value
$W \leq 800$	6.94
$800 < W \leq 1000$	5.10
$1000 < W \leq 1200$	4.77
$1200 < W \leq 1400$	4.56
$1400 < W < 2000$	4.26

Appendix C. Full model 2

The following model concerns the computation of stability during handling at a terminal k of the rotation. The following constraints construct the handling sequence, based on the initial and final configuration.

$$\delta_{xyz,k}^i = \Delta_{xyz,k}^{i,T} \quad \forall xyz, i \in C \quad (59)$$

$$\delta_{xyz,k-1}^i = \Delta_{xyz,k}^{i,0} \quad \forall xyz, i \in C \quad (60)$$

$$\sum_{xyz} \sum_{t=1}^{T-1} \Delta_{xyz,k}^{i,t} = p_i \quad \forall i \in C^k \quad (61)$$

$$p_i = T - 1 \quad \forall i \in C^k / (\bar{E} \cup \bar{I}) \quad (62)$$

$$\Delta_{xyz,k}^{i,t} \geq \Delta_{xyz,k}^{i,t+1} \quad \forall i \in E, t \in 0 \dots T - 1, xyz \quad (63)$$

$$\Delta_{xyz,k}^{i,t} \leq \Delta_{xyz,k}^{i,t+1} \quad \forall i \in I, t \in 0 \dots T - 1, xyz \quad (64)$$

$$p_i + 1 \leq p_c + (1 - \Delta_{xyz,k}^{c,0})M + (1 - \Delta_{xyh,k}^{i,0})M \quad \forall i \in \bar{E} \cap \mathcal{T}, c \in \bar{E} \cap \mathcal{T}: c \neq i \\ x \in \mathcal{X}^T, y, z < |Z|, h > z \quad (65)$$

$$p_i + 1 \leq p_c + (1 - \Delta_{xyz,k}^{c,T})M + (1 - \Delta_{xyh,k}^{i,T})M \quad \forall i \in \bar{I} \cap \mathcal{T}, c \in \bar{I} \cap \mathcal{T}: c \neq i \\ x \in \mathcal{X}^T, y, z < |Z|, h > z \quad (66)$$

$$p_i + 1 \leq p_c + (1 - \Delta_{xyz,k}^{c,0})M + (1 - \Delta_{xyh,k}^{i,0} - \Delta_{x+1,yh,k}^{i,0} - \Delta_{x-1,yh,k}^{i,0})M \quad \forall i \in \bar{E}, c \in \bar{E} \cap \mathcal{F}: c \neq i, x \in \mathcal{X}^F \\ y, z < |Z|, h > z \quad (67)$$

$$p_i + 1 \leq p_c + (1 - \Delta_{xyz,k}^{c,T})M + (1 - \Delta_{xyh,k}^{i,T} - \Delta_{x+1,yh,k}^{i,T} - \Delta_{x-1,yh,k}^{i,T})M \quad \forall i \in \bar{I}, c \in \bar{I} \cap \mathcal{F}: c \neq i, x \in \mathcal{X}^F \\ y, z < |Z|, h > z \quad (68)$$

$$\sum_{i \in \mathcal{F}} \Delta_{xyz,k}^{i,t} + 0.5 \sum_{i \in \mathcal{T}} \Delta_{(x+1)yz,k}^{i,t} + 0.5 \sum_{i \in \mathcal{T}} \Delta_{(x-1)yz,k}^{i,t} \leq 1 \quad \forall t \in 1 \dots T - 1, x \in \mathcal{X}^F, yz \quad (69)$$

$$p_i \neq p_c \quad \forall i \in \bar{E}, c \in \bar{E}: c \neq i \quad (70)$$

$$p_i \neq p_c \quad \forall i \in \bar{I}, c \in \bar{I}: c \neq i \quad (71)$$

$$p_i \neq T - 1 - p_c \quad \forall i \in \bar{E}, c \in \bar{I} \quad (72)$$

The next constraints compute for each handling time t , the total weight, KG_{max} and KM values. KG_i is the KG of a container i based

on the given allocation, therefore, it is an input. Whereas $kg_{i,t}$ is the KG value of a container i in period t .

$$w_t = \sum_{xyz} \sum_{i \in C} \Delta_{xyz,k}^{i,t} W_i \quad \forall t \in 1 \dots T-1 \quad (73)$$

$$kg_{i,t} = Kg_i \sum_{xyz} \Delta_{xyz,k}^{i,t} \quad \forall i \in C, t \in 1 \dots T-1 \quad (74)$$

$$\sum_{xy} \Delta_{xy,z,k}^{i,t} \leq \sum_{h=z}^{|Z|} l_t^h \quad \forall i \in C, z > 1, t \in 1 \dots T-1 \quad (75)$$

$$\sum_{z=2}^{|Z|} l_t^z = 1 \quad \forall t \in 1 \dots T-1 \quad (76)$$

$$\sum_{xyz} T_i \Delta_{xyz,k}^{i,t} \leq s_t^2 \quad \forall i \in C, t \in 1 \dots T-1 \quad (77)$$

$$s_t^1 + s_t^2 = 1 \quad \forall t \in 1 \dots T-1 \quad (78)$$

$$w_t \geq \min_r - (1 - u_t^r)M \quad \forall t \in 1 \dots T-1, r \in R \quad (79)$$

$$w_t \leq \max_r + (1 - u_t^r)M \quad \forall t \in 1 \dots T-1, r \in R \quad (80)$$

$$kg_{\max-t} \leq KG_{\max}^{zcr} + (1 - l_t^z)M + (1 - s_t^c)M + (1 - u_t^r)M \\ \forall z > 1, c \in \{1, 2\}, r \in R, t \in 1 \dots T-1 \quad (81)$$

$$km_t \leq KM^r + (1 - u_t^r)M \quad \forall r \in R, t \in 1 \dots T-1 \quad (82)$$

$$km_t \geq KM^r - (1 - u_t^r)M \quad \forall r \in R, t \in 1 \dots T-1 \quad (83)$$

With the following constraints, we constrain KG values, angles of list and trim.

$$GM_t = km_t - \frac{\overline{KG}_0 W_0 + \sum_{xyz} \sum_{i \in C} W_i kg_{i,t}}{W_0 + w_t} \quad \forall t \in 1 \dots T-1 \quad (84)$$

$$kg_{\max-t} \geq \frac{\overline{KG}_0 W_0 + \sum_{xyz} \sum_{i \in C} W_i kg_{i,t}}{W_0 + w_t} \quad \forall t \in 1 \dots T-1 \quad (85)$$

$$\eta \geq \frac{\sum_{xyz} \sum_{i \in C} W_i \Delta_{xyz,k}^{i,t} (dY_y - dY_0)}{(W_0 + w_t) GM_t} \quad \forall t \in 1 \dots T-1 \quad (86)$$

$$\gamma \geq \frac{\sum_{xyz} \sum_{i \in C} W_i \Delta_{xyz,k}^{i,t} (dX_x - dX_0)}{MCT/L} \quad \forall t \in 1 \dots T-1 \quad (87)$$

Appendix D. Pseudocodes

Algorithm 1. Greedy algorithm for initial solution

```

1: Sort set C by: docks, weight and increasing size
2: Select export container leg,  $l = 0$ 
3: for all x, y
4:    $tag_{export}[x][y] = -1$ 
5: end for
6:  $i = 0, x = 0, y = 0$ 
7: while  $i < |\mathcal{E}|$ 
8:   if can be inserted in xy then
9:     insert i
10:    if  $j > tag_{export}[x][y]: D_{ij} = 1$ ; then
11:       $tag_{export}[x][y] = j$ 
12:    end if
13:     $i \leftarrow i + 1$ 

```

```

14:   else if  $x < X$  then
15:      $x \leftarrow x + 1$ 
16:   else if  $x = X$  &  $y < Y$  then
17:      $x = 0, y \leftarrow y + 1$ 
18:   else
19:      $i \leftarrow i + 1, x = 0, y = 0$ 
20:   end if
21: end while
22: Select import container leg,  $l = m$ 
23: forall  $x, y$  do
24:    $tag_{import}[x][y] = bigM$ 
25: end for
26:  $i = 0, x = 0, y = 0$ 
27: while  $i < |I|$  do
28:   if  $j \geq tag_{export}[x][y]: D_{ij} = 1$  &  $i$  can be inserted in  $x, y$  then
29:     insert  $i$ 
30:     if  $j < tag_{import}[x][y]: D_{ij} = 1$  then
31:        $tag_{import}[x][y] = j$ 
32:     end if
33:      $i \leftarrow i + 1$ 
34:   else if  $x < X$  then
35:      $x \leftarrow x + 1$ 
36:   else if  $x = X$  &  $y < Y$  then
37:      $x = 0, y \leftarrow y + 1$ 
38:   else
39:      $i \leftarrow i + 1, x = 0, y = 0$ 
40:   end if
41: end while
42: Compute  $KG, KG_{max}, LIST, TRIM$ 

```

Algorithm 2. Metaheuristic: for conciseness, for the operator container exchange, we describe only the case for export containers

```

1:  $k = 0$ 
2: while  $k < maxIterations$ 
3:   Select operator
4:   if Stack exchange is selected
5:     Select randomly  $xy$ 
6:     Select  $x'y': x' \neq x$  or  $y' \neq y$ 
7:     Swap columns
8:     update configurations for legs  $l = 2...m - 1$ 
9:   else if Container exchange is selected
10:    Select randomly container  $i$ 
11:    if  $i \in \mathcal{E}$ 
12:      counter = 0, focus on leg  $l = 0$ 
13:      while counter < limit
14:        Select  $xy: x(i) \neq x$  or  $y(i) \neq y$ 
15:        Move or swap
16:        if Change is not possible
17:          counter  $\leftarrow$  counter + 1
18:        else
19:          update configurations for legs  $l = 1...m - 1$ 
20:        end if
21:      end while
22:    end if
23:  else if  $k = value$ 
24:    Remove one container
25:  else
26:    Random reshuffle
27: end if
28: Compute  $KG, KG_{max}, LIST, TRIM$  and new cost
29: Accept or Reject New Solution
30: if New Solution satisfies  $KG, LIST, TRIM$ 
31:   Stop
32: else
33:    $k \leftarrow k + 1$ 
34: end if
35: end while

```

Appendix E. Scheme of the solution framework

Fig. 8.

Fig. 8. Flowchart of the solution framework.

Appendix F. Supplementary material

Supplementary data associated with this article can be found, in the online version, at <https://doi.org/10.1016/j.tre.2019.09.008>.

References

- Ambrosino, D., Sciomachen, A., Tanfani, E., 2004. Stowing a containership: the master bay plan problem. *Transport. Res. Part A: Policy Pract.* 38, 81–99.
- Ambrosino, D., Sciomachen, A., Tanfani, E., 2006. A decomposition heuristics for the container ship stowage problem. *J. Heurist.* 12, 211–233.
- Avriel, M., Penn, M., Shpirer, N., Witteboon, S., 1998. Stowage planning for container ships to reduce the number of shifts. *Ann. Oper. Res.* 76, 55–71.
- Barrass, B., 2000. *Ship Stability: Notes and Examples*. Elsevier.
- Barrass, B., Derrett, C.D., 2011. *Ship Stability for Masters and Mates*. Elsevier.
- Blecker, T., Kersten, W., Gertz, C., 2008. *Management in Logistics Networks and Nodes: Concepts, Technology and Applications*, vol. 8. Erich Schmidt Verlag GmbH & Co KG.
- Crainic, T.G., Dell'Olmo, P., Ricciardi, N., Sgalambro, A., 2015. Modeling dry-port-based freight distribution planning. *Transport. Res. Part C: Emerg. Technol.* 55, 518–534.

- Delgado, A., 2013. Models and Algorithms for Container Vessel Stowage Optimization. Ph.D. thesis. IT University Copenhagen.
- Delgado, A., Jensen, R.M., Janstrup, K., Rose, T.H., Andersen, K.H., 2012. A constraint programming model for fast optimal stowage of container vessel bays. *Eur. J. Oper. Res.* 220, 251–261.
- Ding, D., Chou, M.C., 2015. Stowage planning for container ships: a heuristic algorithm to reduce the number of shifts. *Eur. J. Oper. Res.* 246, 242–249.
- Dkhil, H., Yassine, A., Chabchoub, H., 2018. Multi-objective optimization of the integrated problem of location assignment and straddle carrier scheduling in maritime container terminal at import. *J. Oper. Res. Soc.* 69, 247–269.
- Dubrovsky, O., Levitin, G., Penn, M., 2002. A genetic algorithm with a compact solution encoding for the container ship stowage problem. *J. Heurist.* 8, 585–599.
- El Yaagoubi, A., Alaoui, A.E.H., Boukachour, J., 2016. Multiobjective container barge transport problem, in: *Logistics Operations. In: Management (GOL)*, 2016 3rd International Conference on. IEEE, pp. 1–6.
- Fazi, S., Fransoo, J.C., van Woensel, T., 2015. A decision support system tool for the transportation by barge of import containers: a case study. *Decis. Supp. Syst.* 79, 33–45.
- Fazi, S., Roodbergen, K.J., 2018. Effects of demurrage and detention regimes on dry-port-based inland container transport. *Transport. Res. Part C: Emerg. Technol.* 89, 1–18.
- Gumus, M., Kaminsky, P., Tiemroth, E., Ayik, M., 2008. A multi-stage decomposition heuristic for the container stowage problem. In: *Proceedings of the 2008 MSOM Conference*.
- Imai, A., Sasaki, K., Nishimura, E., Papadimitriou, S., 2006. Multi-objective simultaneous stowage and load planning for a container ship with container rehandle in yard stacks. *Eur. J. Oper. Res.* 171, 373–389.
- Iris, C., Christensen, J., Pacino, D., Ropke, S., 2018. Flexible ship loading problem with transfer vehicle assignment and scheduling. *Transport. Res. Part B: Methodol.* 111, 113–134.
- Iris, C., Pacino, D., 2015. A survey on the ship loading problem. In: *International Conference on Computational Logistics*. Springer, pp. 238–251.
- Kang, J.G., Kim, Y.D., 2002. Stowage planning in maritime container transportation. *J. Oper. Res. Soc.* 53, 415–426.
- Kim, K.H., Kang, J.S., Ryu, K.R., 2004. A beam search algorithm for the load sequencing of outbound containers in port container terminals. *OR Spectrum* 26, 93–116.
- Kirkpatrick, S., Gelatt, C.D., Vecchi, M., 1983. Optimization by simulated annealing. *Science* 220, 671–680.
- Konings, R., Kreutzberger, E., Maraš, V., 2013. Major considerations in developing a hub-and-spoke network to improve the cost performance of container barge transport in the hinterland: the case of the port of Rotterdam. *J. Transp. Geogr.* 29, 63–73.
- Lee, D.H., Jin, J.G., Chen, J.H., 2012. Terminal and yard allocation problem for a container transshipment hub with multiple terminals. *Transport. Res. Part E: Logist. Transport. Rev.* 48, 516–528.
- Li, J., Zhang, Y., Ji, S., 2017. A two-phase approach for inland container ship stowage on full route on yangtze river. In: *Transportation Information and Safety (ICTIS)*, 2017 4th International Conference on. IEEE, pp. 799–805.
- Li, J., Zhang, Y., Ma, J., Ji, S., 2018. Multi-port stowage planning for inland container liner shipping considering weight uncertainties. *IEEE Access* 6, 66468–66480.
- Maritime New Zealand, 2006. Barge Stability Guidelines. <<https://www.maritimenz.govt.nz/commercial/safety/vessel-stability/documents/barge-stability-guidelines.pdf>> (accessed January 21st, 2019).
- Meisel, F., Wichmann, M., 2010. Container sequencing for quay cranes with internal reshuffles. *OR Spectrum* 32, 569–591.
- Monaco, M.F., Sammarra, M., Sorrentino, G., 2014. The terminal-oriented ship stowage planning problem. *Eur. J. Oper. Res.* 239, 256–265.
- Pacino, D., Delgado, A., Jensen, R.M., Bebbington, T., 2011. Fast generation of near-optimal plans for eco-efficient stowage of large container vessels. In: *International Conference on Computational Logistics*. Springer, pp. 286–301.
- Pacino, D., Delgado, A., Jensen, R.M., Bebbington, T., 2012. An accurate model for seaworthy container vessel stowage planning with ballast tanks. In: *International Conference on Computational Logistics*. Springer, pp. 17–32.
- Papanikolaou, A., 2018. A Holistic Approach to Ship Design. *Optimisation of Ship Design and Operation for Life Cycle*, vol. 1 Springer.
- Parreño, F., Pacino, D., Alvarez-Valdes, R., 2016. A grasp algorithm for the container stowage slot planning problem. *Transport. Res. Part E: Logist. Transport. Rev.* 94, 141–157.
- Pielage, B.J., Konings, R., Schuylenburg, M.v., 2007. Barge hub terminals: a perspective for more efficient hinterland container transport for the Port Rotterdam. Technical Report.
- Port of Rotterdam, 2015. Bayernhafen Aschaffenburg and Nuremberg terminals join InlandLinks. <<https://www.portofrotterdam.com/en/news-and-press-releases/bayernhafen-aschaffenburg-and-nuremberg-terminals-join-inlandlinks>> (accessed February 10th, 2019).
- Port of Rotterdam, 2017. Mandatory container weighing in port of Rotterdam. <<https://www.portofrotterdam.com/en/doing-business/logistics/cargo/containers/mandatory-container-weighing-in-port-of-rotterdam>> (accessed January 21st, 2019).
- Port of Rotterdam, 2018. Optimising inland container shipping chain. <<https://www.portofrotterdam.com/en/doing-business/logistics/connections/intermodal-transportation/inland-shipping/optimising-inland>> (accessed February 10th, 2019).
- Port of Rotterdam, 2019. Ketenoptimalisatie containerbinnenvaart. <<https://www.nextlogic.nl/>> (accessed July 3rd, 2019).
- Roberti, R., Pacino, D., 2018. A decomposition method for finding optimal container stowage plans. *Transport. Sci.* 52, 1444–1462.
- Roso, V., Woxenius, J., Lumsden, K., 2009. The dry port concept: connecting container seaports with the hinterland. *J. Transp. Geogr.* 17, 338–345.
- Schepler, X., Balev, S., Michel, S., Sanlaville, É., 2017. Global planning in a multi-terminal and multi-modal maritime container port. *Transport. Res. Part E: Logist. Transport. Rev.* 100, 38–62.
- Sciomachen, A., Tanfani, E., 2007. A 3D-BPP approach for optimising stowage plans and terminal productivity. *Eur. J. Oper. Res.* 183, 1433–1446.
- Talbi, E.G., 2009. *Metaheuristics: From Design to Implementation*. Wiley.
- The Naval Arch, 2017. Longitudinal Strength of Ships - an Introduction. <https://thenavalarch.com/longitudinal-strength-ships-introduction/> (accessed September 3rd, 2019).
- Wilson, I.D., Roach, P.A., 1999. Principles of combinatorial optimization applied to container-ship stowage planning. *J. Heurist.* 5, 403–418.
- Zhang, Z., Lee, C.Y., 2016. Multiobjective approaches for the ship stowage planning problem considering ship stability and container rehandles. *IEEE Trans. Syst., Man, Cybernet.: Syst.* 46, 1374–1389.
- Zuidwijk, R.A., Veenstra, A.W., 2014. The value of information in container transport. *Transport. Sci.* 49, 675–685.