Stowage Planning and Storage Space Assignment of Containers in Port Yards

Catarina Junqueira; Aníbal Tavares de Azevedo†; Takaaki Ohishi‡

Abstract

The efficiency of a port terminal is essential toallow the increasing flow of containers in a global supply chain. In this paper, we propose the integration of the problem of storage space assignment to containers in the port yard with the ship stowage planning problem, considering several ports in the container ship's route. Therefore, we present an adaptation of a method that has been successfully employed in the resolution of the problem of the storage plan: the rules representation. For each port is chosen one rule which defines how the sequence of loading and unloading and handling of containers will occur and aims to reduce the number of unnecessary movements. The evaluation of one specific sequence of rules for n ports is made through a simulation. After the simulation the number of movements will be available and will be associated to this sequence. To test various sequences and search for the best in terms of movements, a genetic algorithm will employ the simulation as an evaluation of its individuals. In addition, the rules representation have the advantage of using a very compact representation that ensures the generation of feasible solutions. The results obtained with numerical examples show that with low computational time it has been possible to obtain good sequences of feasible movements.

Keywords: Stowage Plan of a Container ship; Container Handling Problem at the Port Yard; Rules Representation, Genetic Algorithm.

1 Introduction

The use of containers in the shipping market began nearly six decades. Before that, all cargo was handled manually in ports. The long loading periods, loss and damage of the cargo could amount up to 50 percent of the total cost of shipping maritime cargo [Fitzgerald, 1986].

With the introduction of containers, loading and unloading goods was done in a few hours, without having to touch them individually [Fitzgerald, 1986]. The standardization allowed the containers to be stacked efficiently in vessels, trains, trucks and cranes worldwide. According to research undertaken by Bernhofen et al., 2013 the containerization explains an increase of 320 percent in bilateral trade in industrial countries after five years of its adoption, and an increase of 790 percent after 20 years. And represents a development in global trade better than trade policy liberalization.

Nowadays, container terminals play a significant role in global maritime transport. According to Murty et al., 2005, a container terminal is the place where container ships dock on berths and unload inbound containers (empty or filled with cargo) and load outbound containers. Therefore, the terminal's efficiency is essential to increase the flow of containers in a global supply chain.

Hence, in this paper we study two main problems in a container terminal: the storage space assignment of export containers in port yards and the ship stowage planning problem. Thus, we propose the adaptation of a method that has been successfully employed in solving the stowage plan problem: the representation by rules [Azevedo et al., 2011].

In this adaptation we present a set of new rules for the problem of storage space assignment of export containers in the yard. Another contribution is the joint solution of stowage planning problem with the storage space assignment of export containers problem. For loading and unloading the ship, we...
use the rules proposed by [Azevedo et al., 2011], in a two-dimensional manner. All rules are integrated and employed in a simulation program based in a genetic algorithm (GA), which defines the set of rules that provides the least number of re-handle movements of the containers along the N ports that a ship travels. Re-handles are considered unproductive movements, and must be minimized by the terminal.

[Lee and Lee, 2010] studied an important part of the ship-loading process, the problem of retrieving containers, which is equivalent to the storage space assignment of export containers from a yard. The authors presented a three-phase heuristic to solve for an optimized working plan for a crane to retrieve all the containers from a given yard according to a given order, aiming to minimize the number of container movements, as well as the crane's working time. (Kang et al., 2006) propose a method based on a simulated annealing search to derive a good stacking strategy for containers with uncertain weight information, with the purpose of reducing the number of container re-handlings at the time of loading. Also, [Caserta et al., 2012] describe and analyze the blocks relocation problem (BRP) in a two-dimensional stacking area, suggesting two different binary-integer formulations and a heuristic rule. [Zeng and Yang, 2009] address the scheduling problem for loading or unloading containers, both in the yard as well as the ship, by using a simulation optimization method that also schedules and dispatches various kinds of equipment simultaneously.

The stowage planning has been studied since the 70's ([Wilson and Roach, 2000], [Zeng et al., 2010]). In 2000, it was proven by [Avriel et al., 2000] that the problem of stowage planning belongs to the class of NP-Complete problems by showing its relations to the coloring of circle graphs problem. For this reason, due to the computational complexity of the optimization problem, heuristic methods are predominantly used by the literature. According to [Zeng et al., 2010] the majority of the studies refer to the loading of the ship, and [Wilson and Roach, 2000] divide the methods developed to solve the problem in five major areas, which are: simulation based on probability, heuristics, mathematical modeling, expert systems based on rules and decision support systems. The following are some of these works.

[Ariel et al., 1998] developed a heuristic called "Suspensory Heuristic", that dynamically allocate space for the containers, to solve the problem of stowage planning in order to minimize the number of shifting, without considering stability and several other constraints. [Wilson and Roach, 2000] outline a system for generating solutions to the problem. It progressively refines the arrangement of containers within the cargo-space of a container ship until each container is specifically allocated to a stowage location by allowing the problem to be divided into sub-problems: a generalized placement strategy and a specialized placement procedure.

Focusing on stability adjustment using container weights, [Zeng et al., 2010] develop a fully automated system for stowage planning of large container-ships up to 7000 containers. [Tierney et al., 2014] solved the Capacitated k-Shift Problem (CkSP) that was an open problem that changed the stacks from uncapacitated to capacitated and it was introduced by [Avriel et al., 2000]. This is done by providing an algorithm that for any choice of the number of stacks and stack capacities solves the problem in polynomial time. Finally, [Monaco et al., 2014] propose a binary integer and a two-step heuristic algorithm based on tabu search to solve a set of 60 cases of the stowage problem. In all cases the heuristic algorithm gave better solutions than the binary integer model.

 Meanwhile, concerned about the stowage and associated loading plans, [Imai et al., 2006] seek to satisfy ship stability measurements, while minimizing the number of container re-handles regarding the vessel’s loading operations. The problem is formulated as a multi-objective integer programming. Thus, to solve the model, the authors develop a heuristic algorithm based on genetic algorithm.

In this paper, the goal by using the representation by rules is to reduce the number of re-handles during the mentioned operations, without having to employ a binary model that is limited to small instances.

The following sections are organized as: Section 2 gives a detailed description of the problems. Section 3 introduces the methodology proposed, and in Section 4 the results are presented. Lastly, Section 5 present some concluding remarks.

## 2 Problems description

In a terminal, containers can be distinguished by their destination or type. As for the type, a container can be refrigerated, empty, dangerous, out of standard among others. Such containers typically have a specific storage location. Here we study only standardized containers. As for the destination, a container can be classified as ‘export’, ‘import’ or ‘transshipment’. For a more detailed description, see [Kim and Park, 2003]

The port yard is where the containers are stored temporarily until they are loaded on a ship, truck
or train. Frequently stacks are separated into areas for import, export, special and empty containers [Steenken et al., 2004]. This is done because import containers arrive predictably in large batches at yard, but depart one by one in an unpredictable order when they are claimed, and export containers depart predictably but arrive in a random order [Chen and Lu, 2012]. Consequently, the way in which each type of container is operated differs according to its kind, according to [Chen and Lu, 2012].

Regarding to the vessels, after arriving at the port, they are assigned to a berth where is initiated the unloading process of import and transshipment containers. These containers are removed from vessels via quay cranes, which put them in internal trucks (trucks belonging to the terminal) and are lead to the yard, where they are removed from the trucks are and placed in stacks, using the yard cranes. After being stored in the yard, these containers wait for an external truck or a train to pick them up. In the opposite direction, import containers arrive at the port through external trucks or trains and are stored in stacks in the yard for subsequent load on a vessel.

Known the flow of containers at the terminal, the container handling problem and the stowage plan have the following definitions.

• Stowage Planning Problem: Loading and unloading containers between a ship and a berth via a quay crane;

• Storage Space Assignment of Containers Problem: Loading and unloading containers in the port yard (boarding onto the ships, trucks or trains).

The stowage planning specifies the location of each container on the ship and also determines their sequence of loading and unloading. The key objective of stowage planning is to minimize the number of container movements [Dubrovsky et al., 2002], but this problem involves different objectives, such as to optimize the available space and prevent damage to the goods, the container-ship, its crew and its equipment [Ambrosino et al., 2004], also, minimization of the berthing time is a possible objective of the stowage plan [?].

In [?] a more formal definition of the stowage plan problem is given: “consists in determining how to stow a set \( C \) of \( n \) containers of different type into a set \( S \) of \( m \) available locations of a container-ship, with respect to some structural and operational constraints, related to both the containers and the ship, while minimizing the total stowage time, which is given by the time required for loading all containers on board plus the shifting cost due to the removal of containers”.

The elaboration of the stowage plan is related to the cellular structure that the container ship has, as given in Fig. 1.

The cellular structure of the ship is such that a container can only be accessed only by a particular’s cell top. Thus, to remove a container two cases may occur:

• There are no containers immediately above or if there are other containers, they should also be unloaded on the current port.

• There are containers immediately above and these should be unloaded only on forward ports. They are moved to permit the retrieval of the container that must be unloaded in the current port, but must be returned to the vessel. The withdrawal movement and return of a container for a ship is said rehandle movement.

The second situation can occur frequently and result in a larger ship berthing time . To avoid such disorder is necessary to develop the stowage plan so that the decision in a port does not entail in many relocation movements in the next ports to be visited.

As proven by [Avriel et al., 2000], the problem of minimizing the number of rehandlings belongs to the class of NP-Complete problems, which justifies the use of heuristics and meta-heuristics.
Directly related to the stowage plan, the problem the loading and unloading of containers in the port yard, is better detailed in Fig. 2 in which right side we have highlighted the problem we want to solve in the yard.

The container handling problem of a yard stack in a predefined order, in which rehandlings are necessary was also proven to be NP-complete by [Caserta et al., 2012].

The problem loading and unloading containers in the port yard can take advantage of the same knowledge used to solve the stowage plan problem, as there are potential characteristics similar when the organization of containers is obligated to be in stacks, for example, the restriction that a container can only be accessed through the top.

By reducing the number of unnecessary movements while handling containers in the yard and on the ship, a port terminal can get many benefits, such as minimizing the container transfer between the yard and your target means of transport (ship, truck or train) and minimization of the berthing time ([Preston and Kozan, 2001], [Han et al., 2008] and [Bazzazi et al., 2009]), which are major indicators of excellence for a port terminal.

Besides that, the optimization of operations for just one stage (problem) will not increase the overall port efficiency, because further and not optimized stages will behave like blockage activities.

In this context, taking into account that the mathematical models derived from such problems are of great complexity even for small instances, the purpose of this paper is to apply an alternative resolution method to the joint optimization problem of the stowage planning and storage space assignment of containers in port yards. This alternative method consists of a genetic algorithm-based simulation program that employs rules representation, where: the rules define, in each port, how will occur the loading and unloading operations, and the genetic algorithm defines the sequence of rules that provides the least number of rearrangements of containers along the N ports.

So, our objective in this study is to reduce the number of re-handling during loading and unloading of containers for a given stowage plan and its corresponding movement in the yard. The innovation of this study consists of the application for the first time of an alternative to the use of models with binary variables. Such models have serious limitations in real problems implementation, such as the PCCTP ([Avriel et al., 1998]). The importance of modeling and joint optimization of the two problems is found in the sense that any improvement obtained in one of the individual processes ends up being lost if there is a bottleneck in the chain.

However, the literature does not often study how the decision taken on a terminal affects the following terminals, thereby ignoring possible consequences along the chain which in this case is the journey of a ship. In [Azevedo et al., 2011] and [Azevedo et al., 2012] this point was studied considering only the stowage plan. In this paper, we propose to study the effect of the decision taken in the following ports
both in the stowage plan and in the yard, covering thus possible chain bottlenecks.

The determination of the position in which a container is allocated in the port yard is treated in several ways in the literature, as can be seen in [Carlo et al., 2014]. However, the literature related to retrieval of a yard container for loading on vessels is not extensive, as highlighted in section 5.5 of [Carlo et al., 2014].

One of these works are from [Ji et al., 2015], which relates the re-handling problem to the stowage plan for container ships and yards by proposing three different strategies using genetic algorithm, under the preconditions of a known stowage plan and multi-quay crane parallel operations, and [Lee et al., 2006], who study a strategy called consignment, which groups unloaded containers according to their destination vessel in some dedicated storage area, objecting to reduce the reshuffling. They also handle the traffic congestion by proposing two heuristics based on an MIP model. The first is a sequential method while the second is a column generation method.

In [Han et al., 2008] the work of [Lee et al., 2006] is extended to determine the locations to store the incoming containers as well. To solve the yard allocation problem, an iterative improvement method is developed, in which a tabu search based heuristic algorithm is used to generate an initial yard template, while in [Lee et al., 2006] the yard template is given and input to the model.

In this paper, by integrating the two problems, it is expected that the results represent the chain more realistically than the existing papers today. The practical contribution shall be given by the reduction in the amount of information needed to represent decision-making process even without an algebraic mathematical model, but using a simulation and representation by rules scheme. In addition, there is the possibility of using the knowledge of the terminal decision maker in the form of rules.

The following describes the methodology used in this study to treat container stowage plan and the handling problem in the port yard.

3 Methodology

As was seen in Fig. 1 and Fig. 2 the organization of the containers in a cellular structure, both at the container-ship as in the yard allows us to use a matrices to represent its state. Therefore, it becomes possible to work with this two problems together.

In this way, we have adapted a method based on representation by rules that has successfully solved the stowage planning problem in [Azevedo et al., 2011].

The method consists of employing the representation by rules in an simulator integrated with a genetic algorithm that decides the sequence of rules that provides the least number of rearrangements of containers through the N ports on a ship’s travel. This simulator was implemented in Matlab and is detailed in algorithm 1.

The rules state in detail how will occur the movement of containers in each step of the process and are described hereafter.

The use of a method based in rules is justified once the integration of these problems is very hard to solve due to the computational complexity of the mathematical models, which are NP-Complete.

Additionally, examples from literature show that mathematical models are useful only for small instances for both problems. In this sense, in a recent review the authors, who analyzed and classified 120 articles, stressed the importance and difficulties of integrated planning even most of analyzed articles does not show a clear contribution on such task [Bierwirth and Meisel, 2015].

Besides, our method always produces feasible occupation matrices, facilitating and ensuring the achievement of feasible solutions by heuristic methods. Additionally, another advantage is that the skilled personnel prior knowledge can be incorporated in the optimization process under the form of rules.

The same assumptions made for the mathematical models adopted in [Azevedo et al., 2010] and [Caserta et al., 2012] for the stowage plan problem and storage space assignment problem, respectively, had been adopted here. First, let’s enumerate the assumptions that are common in both problems, and in the next sections we present the assumptions for the problems studied, followed by the rules developed to solve each one of them.

(a) All containers have the same size and are only accessible from the top.

(b) The removal order of the containers is known;

(c) There are no ‘holes’. The position (i, j) above a container that is going to be retrieved must be empty and no container can be moved to a position above an empty position.
3.1 The Storage Space Assignment to Containers Problem

The assumptions for the storage space assignment to containers problem are:

(a) The yard has a rectangular format and can be represented by a matrix with rows \( r = 1, 2, \ldots, R \) and columns \( c = 1, 2, \ldots, C \) with maximum capacity of \( R \times C \) containers. As a result, we have that each space to store a container is a coordinate \((i, j)\), where \( i \in \{1, \ldots, R\} \) and \( j \in \{1, \ldots, C\} \). \( C \) and \( R \) maximum values are parameters of the problem. This matrix is called the yard’s occupation matrix. It is worth remembering that for a more realistic representation, line 1 is the bottom line of the matrix and the line \( R \) is the top line. Column 1 is the first column on the left.

(b) In case of relocation, a container must be placed in the same storage area, on top of another container or on the ground;

(c) While the containers are being removed or relocated in the storage area, there are no new containers being placed in this area;

(d) For the containers to be relocated, it is assumed that there is at least one empty space in the storage area, then: \( R \times C \geq N + (C - 1) \), where \( N \) is the number of blocks. Having \( C - 1 \) empty spaces ensures the accessibility of any container.

(e) When the containers were placed in the storage area, the withdrawal sequence was not known and therefore they are randomly stored.

Next section describes the representation by rules for the handling problem.

3.2 The Representation by Rules for the Storage Space Assignment to Containers Problem

To represent container flow through a port yard, an occupation matrix \( P \) is employed and which each element in position \((R, C)\) (row \( R \) and column \( C \)) is an ordered pair \((a, b)\), where the element \( a \) represents the target ship of that container, and \( b \) is the destination port.

Thus, in the example of Fig. 3, the ordered pair \((1,4)\) in position \((1,1)\) is a container that will be loaded on the ship \( 1 \) and unloaded in the port \( 4 \). Similarly, the element on position \((3,1)\), which is the ordered pair \((2,4)\), will be loaded on ship \( 2 \) and unloaded in port \( 4 \), and so on. Consequently, it is clear that the index “\( a \)” represents the withdrawal order of yard containers, since the ship \( 1 \) is the first to dock at the port and to be loaded, followed by the ship \( 2 \), until the last ship.

Observe that since the containers in positions \((1,1)\) and \((2,1)\) are designated to ship \( 1 \) then it will be necessary to remove containers on upper positions which are designated to ship \( 2 \), i.e., \((3,1)\) and \((4,1)\). Therefore, in order to perform the fewest possible movements and considering the items mentioned above were developed 8 withdrawal rules \((W_r)\), which are described as follows.

- **Rule \( W_{r1} \)**: This rule chooses the smallest combination \((R, C)\) to move the container that needs to be relocated. The lower \( C \) is prioritized before the lower \( R \), meaning, the rule runs the yard’s occupancy matrix in search of an empty spot to relocate a container, starting from the bottom line, row by row, from left to right. If a container is at the top of the column is removed with just one movement, without any relocation. After the removal of the all the containers designated for the first ship, the procedure is repeated for the containers designated to the following ships.
• **Rule** $WR_2$: This rule prioritizes the lower $R$ before the lower $C$, meaning, the rule runs the yard's occupancy matrix in search of an empty spot to relocate a container, starting from the bottom line, column by column, from left to right. If a container is at the top of the column is removed with just one movement, without any relocation.

• **Rule** $WR_3$: This rule moves through the matrix in the same way that $WR_1$ rule does, but verifying if the selected position to reallocate a container is immediately above a container that is going to be removed before it. If so, the position is rejected and the container is reassigned to the next available position. If there are no empty positions that fulfills this requirement, the container is relocated in the first "least worst"position. For example, if a container bounded for ship 4 needs to be relocated and the only are empty positions are above containers bounded for ships 2 and 3, the rule will put it on the top of the container bounded for the third ship.

• **Rule** $WR_4$: This rule moves through the matrix in the same way that $WR_2$ rule does, and as rule $WR_3$, verifies if the selected position to reallocate a container is immediately above a container that is going to be removed before it.

• **Rule** $WR_5$: This rule is the reverse of rule $WR_1$. It runs the yard's occupancy matrix in search of an empty spot to relocate a container, starting from the bottom line, row by row, from right to left.

• **Rule** $WR_6$: This rule is the reverse of rule $WR_2$. It runs the yard's occupancy matrix in search of an empty spot to relocate a container, starting from the bottom line, column by column, from right to left.

• **Rule** $WR_7$: This rule moves through the matrix in the same way that $WR_5$ rule does, but verifying if the selected position to reallocate a container is immediately above a container that is going to be removed before it.

• **Rule** $WR_8$: This rule moves through the matrix in the same way that $WR_6$ rule does, but verifying if the selected position to reallocate a container is immediately above a container that is going to be removed before it.

At the end of the use of $WR$ rules is generated a transportation matrix $T$ of dimension $(N-1) \times (N-1)$, with the information of the destination port $j$ which is used for ship loading rules ($LR$). This matrix is superior and triangular, since $T_{ij} = 0$ for every $i \geq j$, as can be seen in Fig. 5.

The transport matrix $T$ has the number of containers originating from port $i$ and destination on port $j$, where: $O$ is the port of origin and $D$ is the destination port.

Next section describes the assumptions for the stowage plan problem and the rules developed to deal with it.

### 3.3 The Stowage Planning Problem

The rules proposed by [Azevedo et al., 2011] and [Azevedo et al., 2010] are used for the stowage planning problem. Consequently, the following assumptions are analogous to it.

(a) The container ship also has a rectangular format that can be represented by a matrix with rows $(r = 1, 2, \ldots, R)$ and columns $(c = 1, 2, \ldots, C)$ with maximum capacity of $R \times C$ containers, and each space to store a container is a coordinate $(i, j)$, where $i \in \{1, \ldots, R\}$ and $j \in \{1, \ldots, C\}$. This matrix is called the ship's occupation matrix.

(b) The ship starts to be loaded in Port 1, where it arrives empty;

(c) The ship visits ports $2, 3, \ldots, N$. In each port $i, i = 1, \ldots, N - 1$ the vessel receives the loading of containers bound for port $i + 1, \ldots, N$. In the last port it unloads all the containers and will be empty.

(d) The container ship can always carry all the containers available in each port and this will never exceed its capacity.
3.4 The Representation by rules for the stowage planning problem

As shown in Fig. 1, once the container ship has a cellular structure, its arrangement state can be represented by a matrix called the ship’s occupation matrix \( B \).

In this case, each \( B_{rc} \) element of the two-dimensional occupation matrix \( B \) represents the state of a \((r, c)\) cell, that is, if \( B_{rc} = 0 \), means that the cell which occupies the line \( r \), column \( c \) is empty and if \( B_{rc} = j \) means that the cell contains a container whose destination is the port \( j \). Therefore, in the example of Fig. 4 below, the element \((1, 3)\) is equal to 6 (remembering that the lines start from the bottom and line 4 is the top line of the stack), meaning that at this location there is a container that is going to be unloaded at the port 6. Equivalently, the element \((2, 3)\) is equal to 3, meaning that the cell \( B_{23} \), contains a container whose destination is the port 3. Then the elements of the array \( B_{rc} \) represent the occupancy of the container ship and are shown in Fig. 4.

The resolution procedure used here treats the stowage plan as a problem where the occupation \( B \) matrix is the arrangement of the containers on the ship before arriving in the port \( p \). This arrangement is then modified in each port by the decision variable \( R_{ej} \), which defines the way (meaning, rules) that the containers will be handled along the ship’s journey.

The resolution of the problems using the representation by rules consists in determining which rule will be used at each port to perform the loading and unloading operations of the ship, in order to minimize the amount of movements in the ports. It is considered as a movement, each loading or unloading operation of a container at a port \( p \). The rules were developed noting that, often, to do the unloading at the port \( p \), it is necessary to do reshuffling operations (unloading followed by loading) of containers whose destination are the ports of \( p+1 \) to \( N \).

This situation occurs when there are containers that occupy a position above another container that is going to be unloaded at the port \( p \). As well as to remove a container \( j \) from the yard which is located below another ship designed for the ship \( j+1 \) to \( N \).

See, for example, that in the occupation matrix \( B \) of Fig. 4, to unload the containers designated to port 3 it will be necessary to unload, in order, the containers contained in the positions \((3,3)\), \((2,3)\) and \((3,4)\). Note that the container that occupies the position \((3,3)\) is the port 4, but it needs to be relocated in the port 3 so the container that is in the position \((2,3)\) can be unloaded. Thus, in order to reduce the number of reshuffles, when making the loading of containers in a given port \( j \), it must be taken into account the containers that already are on the ship, because they were loaded in previous ports (ports 1 to \( j-1 \)) designated for the ports \( j+1 \) to \( N \) port. Because of that it sought to establish rules for loading and unloading containers at each port that takes into account this relationship.

First, let's enumerate the loading rules \( L_r \):

- **Rule \( L_{r1} \):** This rule fills matrix \( B \) row by row, from left to right, starting from the bottom line in a manner that the containers with the farthest destination are placed first, on the lowest part of the stacks. The application of this rule considering the matrix \( T \) of Fig. 5 and that the ship is in port 1, will result in the matrix \( B \) of Fig. 6(a).
• **Rule** $Lr_2$: This rule fills matrix $B$ column by column, from left to right, from the bottom to the top, in a manner that the containers with the farthest destination are placed on the lowest part of the stacks. The application of this rule considering the matrix $T$ of Fig. 5 and that the ship is in port 1, will result in the matrix $B$ of Fig. 6(b).

• **Rule** $Lr_3$: This rule is the reverse of $Lr_1$. It fills matrix $B$ row by row, from right to left, starting from the bottom line in a manner that the containers with the farthest destination are placed first, on the lowest part of the stacks. The application of this rule considering the matrix $T$ of Fig. 5 and that the ship is in port 1, will result in the matrix $B$ of Fig. 6(c).

• **Rule** $Lr_4$: This rule is the reverse of $Lr_2$. It fills matrix $B$ column by column, from right to left, from the bottom to the top, in a manner that the containers with the farthest destination are placed on the lowest part of the stacks. The application of this rule considering the matrix $T$ of Fig. 5 and that the ship is in port 1, will result in the matrix $B$ of Fig. 6(d).

![Figure 6: Occupation matrix $B$ in port one, after the application of the loading rules](image)

Finally, the unloading rules $Ur$ are:

• **Rule** $Ur_1$: Suppose that the container ship arrived at port $p$. This rule will only remove the containers whose destination is port $p$, and all the ones that are blocking the stacks.

• **Rule** $Ur_2$: This rule imposes that the container ship must unload every container when arriving at a specific port $p$, in a manner that it allows a complete rearrangement of every stack.

Next is shown the way in which the rules for yard and the ship are integrated so they can be used by the simulation program.

### 3.5 Integrating Stowage Planning with Handling Problem

For the rules to be used by the simulation program they need to be combined into a set of decision rules, called $Re_j$.

Therefore, as described in the previously sections, it has been created eight rules for yard withdrawal ($Wr_1$, $Wr_2$, $Wr_3$, $Wr_4$, $Wr_5$, $Wr_6$, $Wr_7$ and $Wr_8$), four for loading the ship ($Lr_1$, $Lr_2$, $Lr_3$, $Lr_4$) and two for unloading it ($Ur_1$, $Ur_2$), which combined generate 64 decision rules $Re_j$. Fig. 7 then leaves this idea clearer.

The process of evaluating a solution $S$ combines the evaluation of the rules with the simulation of the ship’s state after the application of each rule over the ports covered.
In the simulation, there is no analytical evaluation of the objective function. This is done by the genetic algorithm. Hence, the simulation always starts in the port 1 with the number of movements equals to zero, and then runs until the ship arrives at the last port.

With the rules in hand, it is checked in which port the ship is. If it is at the first port there is no unload operation, once the vessel arrives empty. Therefore, the yard containers destined the that vessel are removed and loaded onto the ship. Finished all the first port operations, the rule Re_j is extracted from the second gene of the same individual S, and is applied in the yard of the second port and in the updated occupation matrix B. From the second port to the second last port, the sequence of operations is: unload the ship - yard withdrawal - ship loading. Obviously, the containers that are unloaded are only those destined to that port, as well as those retrieved from the yard and on loaded into the ship are only containers destined for that ship. In the last port only the unload operation is performed and the ship becomes empty again.

Algorithm 1 Simulation scheme

```
1: p ← 1, nmov ← 0
2: initialize(yard, B, T)
3: while (p < N) do
4:   [rt, rc, rd] = ExtractRules(s(p))
5:   if (p > 1) then
6:     [aux, B] = unload(rd, B, p)
7:     nmov ← nmov + aux
8:   end if
9:   if (p > N - 1) then
10:  [aux] = withdraw(rt, patio)
11:     nmov ← nmov + aux
12:   [aux, B] = load(rc, B, T)
13:     nmov ← nmov + aux
14: end if
15: p ← p + 1
16: return nmov
17: end while
```

So, the first step of the simulation is to extract the rule Re_j of the first gene from the first solution S contained in A. This function translates the Re_j rule into the Wr_j, Lr_j and Ur_j rules, as shown in Fig. 7.

After each operation the number of movements performed is given. The number of movements performed throughout the entire ship’s journey, considering the withdrawal, loading and unloading operations, is called f_k (fitness function of the k individual). The simulation is done for all individuals of the initial population. Algorithm 1 brings the simulation program scheme just described.
The symbols and functions used in the algorithm are described in Table 1.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p$</td>
<td>Counter variable, indicates the current port simulation;</td>
</tr>
<tr>
<td>yard</td>
<td>Cell array sized $np \times 1$ ($np =$ number of ports), where each cell contains the yard of a port in the ship's travel;</td>
</tr>
<tr>
<td>$B$</td>
<td>Occupation matrix, indicates the condition of the ship in each port $i$;</td>
</tr>
<tr>
<td>$T$</td>
<td>Transport matrix, stores data of the containers to be loaded and unloaded in the ship's travel;</td>
</tr>
<tr>
<td>$N$</td>
<td>Total number of ports</td>
</tr>
<tr>
<td>$S$</td>
<td>Vector whose element $s(i)$ contains the rule to be applied to the $i$ port and modify the matrix $B$ accordingly;</td>
</tr>
<tr>
<td>ExtractRules</td>
<td>Function that defines the correspondence between the rule $k$, contained in $s(i)$ with the rules of unloading, loading and withdraw to be stored in variables $rd$, $rc$ and $rt$, respectively;</td>
</tr>
<tr>
<td>nmov</td>
<td>Number of movements performed to load or unload the ship and the patio over the $N$ ports;</td>
</tr>
<tr>
<td>$rd$</td>
<td>Variable that contains the name of the ship's unloading rule to be applied;</td>
</tr>
<tr>
<td>$rc$</td>
<td>Variable that contains the name of the ship's loading rule to be applied;</td>
</tr>
<tr>
<td>$rt$</td>
<td>Variable that contains the name of the yard's withdraw rule to be applied;</td>
</tr>
<tr>
<td>initialize</td>
<td>Function that fills the matrix $B$ with zero values;</td>
</tr>
<tr>
<td>withdraw</td>
<td>Function that applies, in the yard of port $i$, the withdrawal rule contained in $rt$, and returns the number of movements made;</td>
</tr>
<tr>
<td>unload</td>
<td>Function that applies to unloading rule contained in $rd$ in the matrix $B$ in port $i$, and returns the number of movements made, $B$ and $T$ updated;</td>
</tr>
<tr>
<td>load</td>
<td>Function that applies the loading rule contained in $RC$, in the matrix $B$ in $i$ port and returns the number of movements made, $B$ and $T$ updated;</td>
</tr>
</tbody>
</table>

Table 1: symbols and functions used in the algorithm

\[
\begin{bmatrix}
P1 \\
P2 \\
P3 \\
P4 \\
\end{bmatrix}
\begin{bmatrix}
31 \\
2 \\
15 \\
60 \\
\end{bmatrix}
= S
\]

Figure 8: Chromosome codification. Source: Adapted from [Azevedo et al., 2010]

The symbols and functions used in the algorithm are described in Table 1.

### 3.6 A Genetic Algorithm

The simulation program integrated to a genetic algorithm allows us to evaluate for a given sequence of rules which will be the number of movements performed on the ship and on the yard over $N$ ports.

The concept of genetic algorithms was introduced in 1975 by [Holland, 1975], which is a search heuristic resembled in the process of natural selection, where the population has a random evolution, and that uses techniques inspired by natural evolution, such as inheritance, mutation, selection and crossover. Algorithm brings the general structure of a genetic algorithm according to [Michalewicz, 1996].

According to [Azevedo et al., 2010], genetic algorithms try to balance two seemingly conflicting goals: the use of the best solutions and the exploration of the search space. For this reason, the genetic algorithm can be applied to combinatorial problems, whose size prevents the solution by exact methods.

At initialization is generated a population formed by a set of random individuals. Each individual of this population has a genetic code, called a chromosome, which is the possible solutions of the problem. The coding used to represent the chromosome is made by a vector $S$, whose element $s_j$'s value is equal to $k$ and indicates which decision rule is going to be used on port $j$. 
Algorithm 2 General structure of a genetic algorithm.

1: \begin{algorithm}
2: \textbf{begin}
3: \hspace*{1em} \(t \leftarrow 0, \text{nmov} \leftarrow 0\)
4: \hspace*{1em} \text{initialize} \(A(t)\)
5: \hspace*{1em} \text{evaluate} \(A(t)\)
6: \hspace*{1em} \textbf{while} \text{termination condition} \textbf{do do}
7: \hspace*{2em} \textbf{begin}
8: \hspace*{3em} \(t \leftarrow t + 1\)
9: \hspace*{3em} \text{select} \(A(t)\) \text{from} \(A(t - 1)\)
10: \hspace*{3em} \text{alter} \(A(t)\)
11: \hspace*{3em} \text{evaluate} \(A(t)\)
12: \hspace*{2em} \textbf{end while}
13: \textbf{end}
\end{algorithm}

<table>
<thead>
<tr>
<th>Port 1</th>
<th>Port 2</th>
<th>Port 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solution 1: 31</td>
<td>Solution 2: 2</td>
<td>Solution 3: 64</td>
</tr>
<tr>
<td>Solution 4: 7</td>
<td>Solution 5: 48</td>
<td></td>
</tr>
<tr>
<td>Solution 1: 11</td>
<td>Solution 2: 1</td>
<td>Solution 3: 13</td>
</tr>
<tr>
<td>Solution 4: 21</td>
<td>Solution 5: 13</td>
<td></td>
</tr>
<tr>
<td>Solution 1: 2</td>
<td>Solution 2: 20</td>
<td>Solution 3: 57</td>
</tr>
<tr>
<td>Solution 4: 1</td>
<td>Solution 5: 12</td>
<td></td>
</tr>
</tbody>
</table>

Figure 9: Representation of the b best solutions in a population of 5 individuals throughout a matrix. Source: Adapted from [Azevedo et al., 2010]
Regarding to the transportation matrices, these can be of three types ([Avriel et al., 1998]): 1-Mixed, 2-Long Distance and 3-Short.

In the matrix type 2, the containers travel through many ports before being unloaded, staying on average a relatively long period of time on board the ship. In the meantime, in the matrix type 3, the containers travel through few ports before being unloaded and are on averaged a short period of time on board the ship. In the matrix type 1 are mixed elements of the matrices type 2 and 3, that is, there are containers that travel through several ports before being unloading and containers that travel through a few ports.

For comparison purposes, all scenarios were simulated for a second time, but with flat yards. The advantage of a flat yard is that it is possible to have access to all the containers with only one movement.

This is an important study in order to verify the need for solving the two problems together. In this experiment it is possible to infer how the consideration of the yard in the problem may affect the arrangement of the containers on the ship. Thus, if the number of movements grow significantly when not only the stowage planning problem is considered, but also the storage space assignment of containers problem in the yard, then both should be considered together.

Table 2 shows the results obtained for each test scenario, where the first column \(I\) indicates the test scenario number, \(N\) denotes the number of ports traveled by the ship in each stage which varies from 3 to 10 ports, \(M\) indicates the type of transportation matrix. \(N_{\text{min}}\) is the minimum number of movements to be performed with the containers in that scenario and \(OF_1\) is the number of movements performed by the simulator with normal yards (objective 1), and \(OF_2\) is the number of movements performed by the simulator with the flat yards (objective 2). Next we have \(Avg_1\) is the average of the movements performed by container \((\text{number of containers} \div \text{number of movements performed})\) for the objective 1 and \(Avg_2\) for the objective 2. Finally, \(T_1\) and \(T_2\) are the computational time in hours for the objective 1 and 2, respectively.

<table>
<thead>
<tr>
<th>(I)</th>
<th>(N)</th>
<th>(M)</th>
<th>(N_{\text{min}})</th>
<th>(OF_1)</th>
<th>(OF_2)</th>
<th>(Avg_1)</th>
<th>(Avg_2)</th>
<th>(T_1)</th>
<th>(T_2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td>2820</td>
<td>4518</td>
<td>2194</td>
<td>1.60</td>
<td>1.28</td>
<td>16.97</td>
<td>0.06</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3428</td>
<td>5053</td>
<td>3889</td>
<td>1.47</td>
<td>1.13</td>
<td>16.21</td>
<td>0.06</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2719</td>
<td>4367</td>
<td>3291</td>
<td>1.61</td>
<td>1.21</td>
<td>15.95</td>
<td>0.07</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>1</td>
<td>3636</td>
<td>4804</td>
<td>3911</td>
<td>1.32</td>
<td>1.07</td>
<td>23.29</td>
<td>0.10</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>2</td>
<td>4557</td>
<td>8208</td>
<td>5788</td>
<td>1.80</td>
<td>1.27</td>
<td>27.43</td>
<td>0.27</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>3</td>
<td>4890</td>
<td>7972</td>
<td>5510</td>
<td>1.63</td>
<td>1.12</td>
<td>26.93</td>
<td>0.18</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>1</td>
<td>7575</td>
<td>14620</td>
<td>9057</td>
<td>1.93</td>
<td>1.19</td>
<td>43.37</td>
<td>0.46</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>2</td>
<td>7250</td>
<td>16221</td>
<td>10743</td>
<td>2.24</td>
<td>1.48</td>
<td>45.63</td>
<td>0.46</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>3</td>
<td>7533</td>
<td>13566</td>
<td>7985</td>
<td>1.80</td>
<td>1.06</td>
<td>46.37</td>
<td>0.35</td>
</tr>
</tbody>
</table>

Table 2: Results of the Simulations - Comparison with Flat Yard

The most important observation regarding the results refers to the increased complexity of the problem when the yard is included in the simulation. In [Azevedo et al., 2011], considering only the stowage planning were obtained feasible results for 30 ports in less than 8 minutes. Including the withdrawal operation of the yard containers, scenarios with 10 ports took longer than 40 hours to be completed.

When we assumed that the yard is flat, it is clear that the complexity of the problem is reduced considerably, causing the number of moves to be much closer to the optimum. The computational time spent processing scenarios with flat yards was also reduced drastically.

This can be justified because, unlike the ship, which starts out empty and since the beginning of his journey receives containers following some rule, in the yard, containers arrive in an unknown order and are arranged randomly. This feature makes the yard initial arrangement much more difficult to handle than the ship.

As a result, it is clear that the fact that the arrangement of the containers in the yard is random justifies the greater time required for a withdrawal rule process a yard in comparison to the time that an unloading or loading rule need to process a ship.

However, despite the high processing time, all test scenarios have generated achievable results and the average number of movements per container was below 2, with the exception of scenario 8.

The following section presents the conclusions of these papers.
5 Conclusion

This paper have presented the adaptation of the methodology based on representation by rules, successfully employed in the stowage planning problem in [Azevedo et al., 2011], for the storage space assignment to containers problem integrated with the ship’s stowage planning problem, in order to reduce the number of movements during such operations.

As an additional contribution of this paper were developed and tested rules for withdrawal of the yard containers, that were later included in the simulation program described here.

By comparing the results with the what was achieved by [Azevedo et al., 2011], it was possible to prove the increasing complexity of the problem when the yard is included in the simulation. Showing the importance of joint modeling and optimization of the two problems.

Furthermore, it is believed that it was possible to reduce the amount of information needed to represent decision making process, since the resolution of the examples of this work by a mathematical model would not be feasible in computational time.

Finally, despite the increased complexity of the problem, it was possible to get sequences of feasible movements both in the yard as the ship and reduce the number of unnecessary movements during loading and unloading of container operations for a given stowage plan and its corresponding movement in the yard.

6 Acknowledgments

This research was supported by the Fund to Support Teaching, Research and Extension (FAEPEX) from the University of Campinas (UNICAMP) and by the Foundation for the Support of Research of the State of São Paulo (FAPESP) under the process 2015/24295-5.

Also, this research was conducted during a scholarship financed by CAPES – Brazilian Federal Agency for Support and Evaluation of Graduate Education within the Ministry of Education of Brazil.
References


