Solution of the 3D Stochastic Stowage Planning for Container Ships through Representation by Rules

Anibal Tavares de Azevedo[1], Edilson Fernandes de Arruda[2], Luiz Leduino de Salles Neto[3], Antônio Augusto Chaves[3], Antônio Carlos Moretti[1]

[1]State University of Campinas, Campinas, Brazil
[2]Federal University of Rio de Janeiro, Rio de Janeiro, Brazil
[3]Federal University of São Paulo, São José dos Campos, Brazil
anibal.azevedo@fca.unicamp.br, antonio.moretti@fca.unicamp.br, efarruda@pep.ufrj.br, luiz.leduino@unifesp.br, antonio.chaves@unifesp.br

Abstract

This paper formulates the 3D Stochastic Stowage Planning (3D SSP) problem. The key objective of 3D SSP is to minimize the number of container movements and maximize the ship´s stability considering multiple scenarios. The binary formulation of this problem is described and an alternative formulation, called Representation by Rules, is combined with a Genetic Algorithm and a Participatory Learning System. The robustness of the developed framework is verified in a problem for which the corresponding binary representation demands 405,450,000 variables.

Keywords: Stochastic Stowage Planning, Container Ship, Representation by Rules, Genetic Algorithm, Participatory Learning System.

1. Introduction

Today, international sea freight container transportation and container terminals play a key role in the global transportation network. According to [21], over 60% of the world's deep-sea general cargo is transported in containers, and the routes between some countries are containerized up to 90%.

The improvement of the operational efficiency of container terminals is essential to handle the increasing flow of containers that has occurred over the last years. Reducing the time ships must spend in port makes container seaports more competitive because they can offer lower rates for loading and discharging. Therefore, an essential competitive advantage is the reduction of the time in port of the container ships, and of the costs of the transshipment process itself. The optimization of seaport operations problems should be defined and methods of solution proposed for such tasks.

According to [13, 21], the seaport container terminal operations may be divided into five main problems:
(1)Berth allocation: the output of this problem is the scheduling of the ships in each berth by considering the minimal distance considered to be safe between two ships and two cannot share the same berth at the same period of time;
(2) Stowage planning: this problem consists of determining how to organize the containers in a ship in order to minimize the number of movements necessary to unload and load the container ship;
(3) Crane Split: the container ship time for unloading and loading depends on the scheduling of Quay cranes for each vessel section. One important constraint is to not allow Quay cranes to pass each other, since their movement are limited to a common rail;
(4) Quayside transport: this problem defines which machines will be used and their trajectory from ship to port yard to transport containers from ship to landside or vice-versa;
(5) Landside transport: This problem deals with unloading and loading containers efficiently off from onto trucks or trains to maximize the flow through the port yard.

This paper focuses on Problem (2). Stowage planning is related to the cellular structure of container ships, as shown in Figure 1.

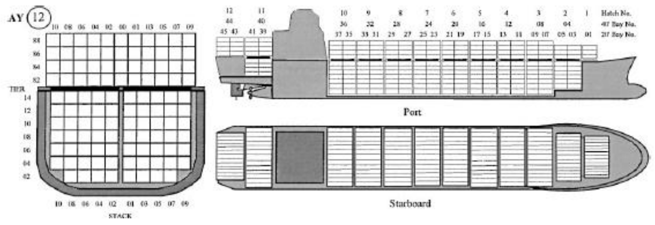


Fig. 1: Container ship cellular structure (From: [21]).

This structure means containers may only be reached by removing any containers stacked on top of them in a column. There are, thus, two unloading cases:

(i) Containers to be unloaded at a given port are at the top of the stack.

(ii) Containers to be unloaded at a given port are blocked by one or more containers that are to remain aboard the container ship. These must be unloaded and reloaded after all containers in the column for that port have been unloaded. This movement of unloading and loading blocking containers is called re-handling and must be minimized to improve the port efficiency.

The more such blockage occurs, the longer the ship must stay in port. So, stowage planning is the key to minimizing re-handling movements. In [5] it was proved that the 2D stowage plan is NP-Complete, which justified the development of a series of heuristics and meta-heuristics to obtain good solutions for this problem.

In [4], the authors proposed the Suspensory Heuristic Procedure, which avoids the binary model problem by observing only the container ship arrangement and the containers that must be loaded for each specific port.

In [10], Dubrovsky developed a Genetic Algorithm with compact solution encoding as well as instability constraints as a penalty function. The proposed approach was tested in instances with 11 ports and a container ship with a 1000 TEU capacity.

Wilson and Roach in [23, 24] developed a methodology that solves stowage planning in two steps: the first step uses a branch-bound to solve the problem of assigning generalized containers to a bay; the second step consists in using a Tabu search to assign each container a specific position in each bay. This approach takes 2 hours to obtain a solution for a 688 TEU container ship and 4 destinations.

Ambrosino [1, 2] also introduced and tested the notion of stability by considering that each container had a weight in the model. Ambrosino also tested ant colony optimization (ACO) in a real problem for the SECH Terminal of Genoa, Italy, and a container ship whose capacity was 1800 TEUs for instances with 2 or 3 destinations, 4920 TEUs for 4 destinations, and 5510 TEUs for 5 destinations.

Imai et al. [15] presented a multi-criteria optimization method for the ship stowage problem by considering two conflicting objectives: the ship´s stability and the number of container re-handles. Computational tests were carried out with a proposed Genetic Algorithm for instances of 504 and 2016 TEUs, and 3 and 4 destinations, respectively.

In [19], Sciomachen and Tanfani formulated the problem as a three-dimensional bin packing problem and presented a heuristic solution method which was based on this relation. Objectives were to minimize the total loading time as well as to efficiently use the quay equipment. The approach was validated by using real test cases from the port of Genoa (Italy), namely: a container ship with a 198-TEU capacity, containers with different weights, one or two cranes, and 2 or 3 destinations.

In [12], Fan et al. (2010) carried out Stowage Planning with a tradeoff between crane workload balance and ship stability solving instances in which the container ship had a 5000-TEU capacity, 5 quay cranes, 5 types of containers, and 5 destination ports.

None of these approaches treated problems with more than 5 destination ports or uncertainty about the destination port of each container. Container destination information is not available until the ship arrives at the port. This information is essentially stochastic and in the literature, the papers treated this information as deterministic one. The objective of this paper is to solve a stochastic version of the stowage plan optimization for a large number of port destinations.

For such a complex problem, this paper proposes a three-level solution to minimize the unnecessary container movements and also ship instability. In the 1st level, the meta-heuristic manipulates the encoded solutions. In the 2nd, a specific demand scenario is selected. In 3rd, the solutions are evaluated by a decoding algorithm that simulates the loading and unloading of a container ship at each port. By using this decoding procedure, the size of the search space can be reduced and human knowledge may be properly incorporated. This encoding and decoding is called Representation by Rules.

This paper is organized as follows. Section 2 presents the mathematical model for Stowage Planning. Section 3 explains the encoding, using meta-heuristic methods, and their advantages. Section 4 describes the Genetic Algorithm. Section 5 describes how the Participatory Learning System is coupled with Representation by Rules in order to deal with the stochastic number of containers. Section 6 presents and discusses the computational results and Section 7 presents the conclusions and possible future work.

2. Mathematical Model

The following assumptions have been made for the sake of simplicity, without compromising the solution's general application.

(a) The container ship has a rectangular format and can be represented by a matrix with rows ($r = 1, 2,\dots, R$), columns ($c = 1, 2, \dots, C$) and bays ($d = 1, 2,\dots, D$) with maximum capacity of $R \times C \times D$ containers.

(b) All containers have the same size and weight.

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

(d) The ship visits ports $2, 3,\dots, N$ such that the container ship will be empty in the last port, because the ship performs a circular route where port N, in fact, represents Port 1.

(e) In each port $i = 1, 2,\dots, N,$ the container ship can be loaded with containers whose destination are ports $i+1, \dots, N.$

(f) The container ship can always carry all the containers available in each port and this will never exceed its capacity.

In addition to conditions (a)-(f), the number of containers that must be loaded at a certain port is given by a transportation matrix T of dimension *(N-1)×(N-1)*, whose element T_{ij} represents the number of containers from port i that must be transported to the destination port j. This matrix is an upper triangular matrix, since $T_{ij}=0$ for every $i \geq j$. The T_{ij} variable is essentially stochastic and the number of containers is not known in advance, which leads the model to use $\hat{T}_{ij}$. Without knowing the probability and corresponding distribution function of $\hat{T}_{ij}$, a different approach must be employed, in which s multiple scenarios with a specific number of containers for each destination port are given in a corresponding transportation matrix T_{ij}^{s} [11].

The mathematical model uses linear programming with binary variables for 3D Stochastic Stowage Planning (3D SSP) as given by (1)-(6) below.

$$Min\ f(x) = \alpha\phi_1^s(x) + \beta\phi_2^s(y)$$
$$i = 1,\cdots,N-1,\ j = i+1,\cdots,N; \quad (1)$$

S.a:

$$\sum_{v=i+1}^{j}\sum_{r=1}^{R}\sum_{c=1}^{C}\sum_{d=1}^{D} x_{ijv}^{s}(r,c,d) - \sum_{k=1}^{i-1}\sum_{r=1}^{R}\sum_{c=1}^{C}\sum_{d=1}^{D} x_{kji}^{s}(r,c,d) = T_{ij}^{S}$$
$$i = 1,\cdots,N-1,\ j = i+1,\cdots,N; \quad (2)$$

$$\sum_{k=1}^{i}\sum_{j=i+1}^{N}\sum_{v=i+1}^{j} x_{kjv}^{s}(r,c,d) = y_i^s(r,c,d),$$
$$i = 1,\cdots,N-1, r = 1,\cdots,R;$$
$$c = 1,\cdots,C;\ d = 1,\cdots,D; \quad (3)$$

$$y_i^s(r,c,d) - y_i^s(r+1,c,d) \geq 0,$$
$$i = 1,\cdots,N-1, r = 1,\cdots,R-1;$$
$$c = 1,\cdots,C;\ d = 1,\cdots,D; \quad (4)$$

$$\sum_{i=1}^{j-1}\sum_{p=j}^{N} x_{ipj}^{s}(r,c,d) + \sum_{i=1}^{j-1}\sum_{p=j+1}^{N}\sum_{v=j+1}^{p} x_{ipv}^{s}(r+1,c,d) \leq 1$$
$$j = 2,\cdots,N, r = 1,\cdots,R-1;$$
$$c = 1,\cdots,C;\ d = 1,\cdots,D; \quad (5)$$

$$x_{ijv}^{s}(r,c,d) = 0 \text{ or } 1;$$
$$y_i^s(r,c,d) = 0 \text{ or } 1; \quad (6)$$

where the binary variable $x_{ijv}^{s}(r,c,d)$ is defined as follows: if, in scenario s, in port i, the compartment *(r,c,d)* has a container whose destination is port j and this container was moved in port v, then the variable assumes value 1; otherwise value 0 is assumed. The term compartment *(r,c,d)* represents row r, column c for the container ship bay d. Similarly, variable $y_i^s(r,c,d)$ is defined as follows: if, in scenario s, in port i, the compartment *(r,c,d)* has a container; then the variable assumes value 1; otherwise value 0 is assumed.

Constraint (2) is related to the container conservation flow. In other words, for each scenario s, the total number of containers in the ship in a port i must be equal to the number of containers that have been loaded in all ports $p = 1\ 2,\ldots,\ i$ minus the total number of containers unloaded in all ports $p = 1, 2, \ldots, i$. Constraint (3) requires that in each scenario s, each compartment *(r,c,d)* of the container ship is always occupied by at most one container. Constraint (4) is related to the physical storage of the containers in the ship, and it imposes that, in each scenario s, for each container in row $r+1$, there be another container in the row r for all r = 1*,…, R-1*. Constraint (5) defines how, in a scenario s, a container can be unloaded from the ship in port j by requiring that, if a container occupies the position *(r,c,d)* at port j, and it will be unloaded, then, there are no containers above or the containers above have already been unloaded at previous ports.

The objective function given by Eq. (1) is composed of two terms for each scenario s: the first term is a function of the number of containers moved, $\phi_1^s(x)$, and the second is the sum of instability measures for the container ship configuration in each port, $\phi_2^s(y)$. The two criteria are combined by values given by each weight $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in a manner that forms a bi-objective optimization framework.

The term $\phi_1^s(x)$ assumes that for all ports, the container movement cost is the same and is equal to one which may be translated as Eq. (7).

$$\phi_1^s(x) = \sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\sum_{v=i+1}^{j-1}\sum_{r=1}^{R}\sum_{c=1}^{C}\sum_{d=1}^{D} x_{ijv}^{s}(r,c,d) \quad (7)$$

Term $\phi_2^s(y)$ refers to the container ship´s stability and assumes that every container has the same mass and is equal to one. This term measures the distance between the mass and the geometric center of the container ship for each bay in every port in each scenario s as described by Eq. (8).

$$\phi_2^s(y) = \sum_{i=1}^{N}\left(\sum_{d=1}^{D}\left(\Delta xcm_{di}^{s}\right)^2 + \sum_{d=1}^{D}\left(\Delta zcm_{di}^{s}\right)^2\right) \quad (8)$$

where:

$\Delta zcm_{di}^{s} = zcm_{di}^{s} - R/2$ and

$\Delta xcm_{di}^{s} = xcm_{di}^{s} - C/2$;

$$zcm_{di}^{s} = \frac{\left(\sum_{r=1}^{R}\sum_{c=1}^{C}\left(y_i^s(r,c,d) \bullet (r-0.5)\right)\right)}{\left(\sum_{r=1}^{R}\sum_{c=1}^{C} y_i^s(r,c,d)\right)},$$

$$xcm_{di}^{s} = \frac{\left(\sum_{c=1}^{C}\sum_{r=1}^{R}\left(y_i^s(r,c,d) \bullet (c-0.5)\right)\right)}{\left(\sum_{r=1}^{R}\sum_{c=1}^{C} y_i^s(r,c,d)\right)}.$$

The resulting problem has binary variables, is multi-objective, non-linear, stochastic and large-scale for real instances. This mathematical model is the first that extends the 2D Stowage Planning (2D SP) problem proposed by Avriel and Penn [3] to 3D SSP. In [5], Avriel and Penn showed that the 2D SP is NP-Complete; so, the 3D SSP is also. Another drawback of the 2D or 3D model for real problems is related to solution encoding, which demands a large number of binary variables to represent a solution. Thus, they may only be used for small instances in practice. For example, a 3D SSP solution that produces a complete stowage plan through *N* ports for a problem instance with a container ship with a *(R $\times$ C$\times$ D$\times$ S)* bay dimension demands an amount *(R $\times$ C $\times$ D $\times$ N^3 $\times$ S)* of $x_{ijv}^s(r,c,d)$ variables and *(R $\times$ C $\times$ D $\times$ N)* of $y_i^s(r,c,d)$ variables. That means that with D = 5, R = 6, C = 50, N = 30 and S = 10 there will be 405,000,000 $x_{ijv}^s(r,c,d)$ variables and 450,000 $y_i^s(r,c,d)$ variables to represent just one solution. This fact justifies not only the use of heuristics for the 3D SSP, but also a different way to represent the solution.

The next Section shows how to represent a solution without using a binary model that leads to a large number of variables. The Section describes a special encoding/decoding approach which combines rules that describe how to load and unload a container ship, and a simulation procedure. This encoding will be called Representation by Rules.

3. Representation by Rules

The container ship stowage planning problem consists in determining optimal loading and unloading movements and this problem is as difficult as playing the famous game Tetris. The reason for this starts with the uncertainty of how many containers there are and what their destinations will be. This information only becomes available at each future port that container ship docks at. Also, both problems have in common the division of the space as a grid as shown in Fig. 1 and 2.

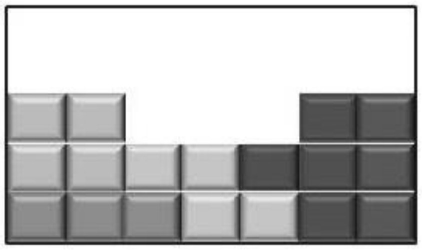


Fig. 2: Tetris pieces in an arrangement that considers space as a grid.

In the container ship problem some physical constraints should be obeyed like in Tetris. One of them is that two pieces (or containers) could never occupy the same space at the same time as shown is Fig. 2.

Actually, fulfilling the constraints is just one part of the problem. It is also interesting to find a set of actions that minimizes the objective function that could be the number of movements performed or the instability measure of containers arrangement.

At the beginning of the container ship's trip, one simple action in the first port can greatly affect the arrangement of the container ship in the next ports, resulting in several necessary container movements. The same happens with Tetris, in which a set of actions at the beginning of the game can greatly affect the pieces' arrangement after a period of time.

Stowage planning and Tetris share the same type of problems: stochastic information, physical constraints, and actions which, although seemingly the best at one point in time, may have disastrous results later on.

The analogy of 3D SSP with Tetris is proper since the developed approach leads to a change of view of how to solve the problem. No Tetris player tries to imagine a mathematical model with binary decision variables to make a decision in seconds. Instead, a good player has a collection of rules to be performed according to the pieces' arrangement. These rules are obtained by observing some heuristic principle. Their validity and adequacy can be inferred after a series of game simulations.

In the port, the selection of container ship loading or unloading movements might follow the same pattern of thinking: an experienced operator could determine when a rule to fill the ship was proper or not. In other words, the operators could have a series of rules at their disposal. To select one or another, operators would use their cumulative knowledge, acquired after seeing many configurations and their corresponding results. So, multiple scenarios are important to select a robust sequence of rules that can handle different arrangements and situations. For both problems, the key ideas are:

- Build heuristic procedures, called rules, which must obey or are based on some physical properties, constraint or objective function.
- Test a sequence of rules with a simulation to provide information about how good or bad this sequence is in terms of one or more measures.

In 3D SSP, the matrix *B* represents the container ship arrangement, since each element of *B* is represented by $B_{drc,}$. It describes whether there is a container whose destination is port *p* in the cell located in Row *r*, Column *c* and Bay *d*, if $B_{drc} = p$; if B_{drc} is empty, then B_{drc} is *0*. To better illustrate this, a B matrix where D = 3 and R = C = 2 is shown in Fig. 3.

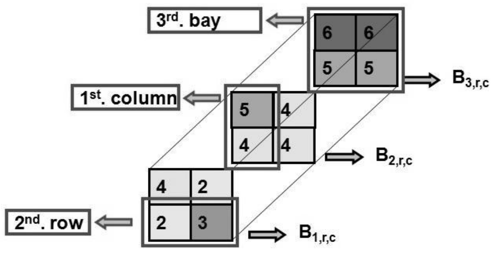


Fig. 3: State matrix B representing the container arrangement in a 12-container container ship to 6 ports.

In Fig. 3, Row 2 represents the bottom of the container ship and Row 1 represents the top of the container ship. Thus, element (1,1,1) is equal to 4, which means that this cell is occupied by a container whose destination is port 4. Using the same criteria, element (3,2,2) is equal to 5 and it means that this cell is occupied by a container whose destination is port 5.

Supposing that matrix *B* in Fig. 3 represents the container ship in port 2, in order to unload this container ship, it is necessary to move the containers located in cells (1,2,1) and (1,1,2). However, observe that the container in cell (1,2,1) can only be unloaded if the container located in cell (1,1,1) is unloaded too; even though the destination of this container is port 4. The goals of the 3D SSP are: minimize the number of movements of this kind by an adequate arrangement of the container ship bay at every port for every scenario, and; minimize the instability measure of the containers' arrangement.

The Representation by Rules approach treats the 3D SSP as a problem where matrix *B* represents the container ship arrangement before arriving in port *p*. It will be modified at each port by deciding how to perform the unloading and loading operations according to the corresponding unloading and a loading rule, respectively. Note that matrix B is equivalent to Constraint (2) that connects decisions from port to port. The choice of an unloading or loading rule for port 2 can indirectly influence the container ship arrangement in port 4.

In the next sections, the loading and unloading rules proposed to solve the 3D SSP will be presented. Also, a description of how the loading and unloading rules affect the container ship arrangement will be presented.

3.1 Loading Rules

Loading Rule LR1: This rule fills matrix B row by row, from left to right, starting from the bottom row for each bay in a manner that the containers with the farthest destination are placed on the lowest rows and each bay is filled before the next. Fig. 4 shows the application of the loading rule for the container ship at port 1.

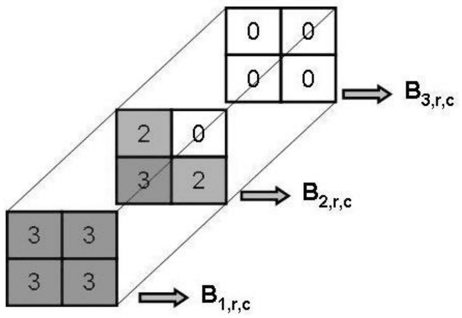


Fig. 4: State matrix B representing the container arrangement after applying **LR1**.

Loading Rule LR2: This rule fills matrix B row by row, from left to right, starting from the first bay and filling only one row per bay in a manner that the containers with the farthest destination are placed on the lowest rows and distributed among the bays.

Loading Rule LR3: This rule is the reverse of **LR1** which means matrix B is filled row by row, from right to left, starting from the bottom row for each bay in a manner that the containers with the farthest destination are placed on the lowest rows and each bay is filled before the next is begun.

Loading Rule LR4: This rule is the reverse of **LR2** in the sense that it fills matrix B row by row, from right to left, one row per bay starting from the first bay until it reaches the last in a manner that the containers with the farthest destination are placed on the lowest rows and distributed among the bays.

Loading Rule LR5: This rule fills matrix B row by row from left to right with containers destined for the nearest port, starting from the first bay and continuing until the number of elements θ_p in a column is reached. The value θ_p is computed by Eq. (9).

$$\theta_p = \left\lceil \frac{\sum_{i=1}^{p} \sum_{j=p+1}^{N} T_{ij}}{D \times C} \right\rceil \tag{9}$$

Loading Rule LR6: This rule is the reverse of **LR5** in the sense that it fills matrix B row by row from right to left with containers destined for the nearest port, starting from the first bay and continuing until the number of elements θ_p in a column is reached. The value θ_p is also computed by Eq. (9).

3.2 Unloading Rules

Unloading Rule UR1: Suppose that the container ship arrives at port *p*. This rule will only remove the containers whose destination is port *p*, and all the ones that are blocking the stacks.
Unloading Rule UR2: This rule states 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.
A more detailed description about these rules can be found in [6].

3.3 Combining loading and unloading rules

In order to avoid the need of using two values to indicate which loading and unloading rules will be used for every port, it is possible to simplify the encoding by defining the various combinations of the loading and unloading rules. The specific combination of loading and unloading rules for port *p* is defined as a rule.

To better illustrate the rule concept, the six loading rules (LR) and the two unloading rules (UR) described before can be combined to generate twelve new rules. Table 1 illustrates all the rules created as a result of these LR and UR combinations.

Table 1: Rules produced by the combination of loading and unloading rules.

Loading Rules	Unloading Rules	Rule
LR1	UR1	1
	UR2	2
LR2	UR1	3
	UR2	4
LR3	UR1	5
	UR2	6
LR4	UR1	7
	UR2	8
LR5	UR1	9
	UR2	10
LR6	UR1	11
	UR2	12

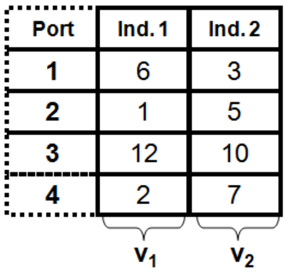
Table 1 shows the various combinations of LR and UR. For example, Rule 2 is a combination of the unloading rule **UR2** and the loading rule **LR1**. This encoding allows the representation of a solution for the 3D SSP that employs a vector whose number of elements is equal to the number of ports.

To adopt this encoding process it is necessary to define a translation scheme which simulates the unloading and loading actions proposed by the rules. This translation extracts the unloading and loading rules that should be applied for the container ship arrangement in every port. Afterwards, the scheme gives the number of containers moved in every port and total instability measure. A more detailed description about this translation scheme can be found in [7].

4. Genetic Algorithm

The genetic algorithm uses a population of individuals represented by: $A(t) = \{A_1^t, ..., A_n^t\}$ for each generation (iteration) *t*, in which each individual represents a vector of rules. In the computational implementation adopted here, the population is stored in a matrix *A(t)* and each line A_i^t represents this vector of rules. Each vector A_i^t is evaluated according to the number of movements and to the instability measure in various scenarios. Then, *fitness*, a measure of how successful this individual is in the problem, is calculated using Eq. (1). *Fitness* is calculated for the entire population and is based on this new population that combines the most capable individuals that will form generation *t+1*. During the formation of the new population, some individuals from generation *t* are submitted to a transformation process by genetic operators in order to form new sequence of rules. These transformations include unary operators m_i (mutation), that allow the creation of new rules by small changes in the individual attributes (m_i: $A_i \rightarrow A_i$), and superior order transformation c_j (crossover) that produces new individuals by combining one or more individuals (c_j: $A_j \times ... \times A_k \rightarrow A_j$). This process is carried out until a previous specified maximum number of generations is reached [14, 17].

Each genetic algorithm individual is associated to a set of rules by using a vector *v* with, for example, 4 elements. The values inside the elements correspond to each combination of unloading and loading rules (1 to 12) applied to each port (1 to 4). The set of rules for various individuals is stored in a matrix and each column represents an individual, as shown in Fig. 5. In Fig. 5, the first column contains a vector $\mathbf{v_1}$ with 4 elements, in which each value corresponds to Table 1.



Port	Ind. 1	Ind. 2
1	6	3
2	1	5
3	12	10
4	2	7

$\mathbf{v_1}$
$\mathbf{v_2}$

Fig. 5**:** The genetic algorithm individual (column vector). Population matrix A has two individuals (vectors) $\mathbf{v_1}$ and $\mathbf{v_2}$.

Once the individual is defined, the population which consists of *numpop* individuals stored in a matrix A of dimension $10 \times numpop$ can also be defined. For every instance, the value of *numpop* equals 100 was adopted. Since each column of matrix *A* represents individuals/solutions, every element *A(i,j) = k* defines what rule *k* (*k* equals 1 to 12) in port *i* will be used, if the individual *j* is chosen. For example, *A(1,1) = 6* means that the individual/solution 1 in port 1 is to apply Rule 6, which means unloading the container ship with **UR2** and loading using **LR3**.

The complete framework integration of genetic algorithm, rules, scenarios and simulation is shown in Fig. 6.

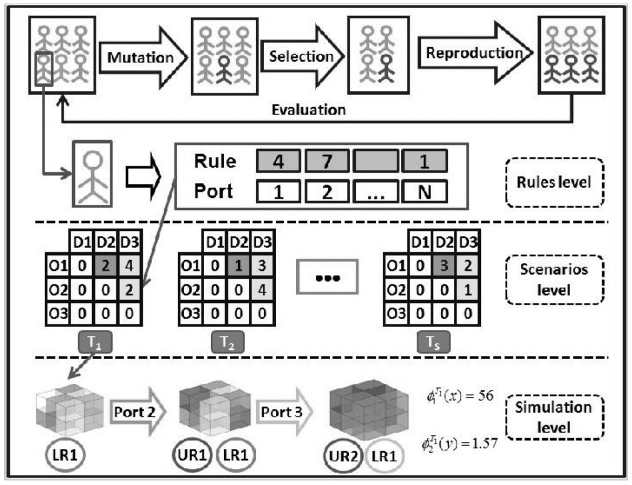


Fig. 6: Combining the Genetic Algorithm with Representation by Rules, Scenarios and Simulation.

The most important property of the developed framework is that a solution will be feasible no matter the scenario that was selected. This is possible only because rules don't specify the precise place which each container should be, but give general guide rules for the container ship arrangement.

5. Participatory Learning System

In various problems, learning is seen as a process where, as new information becomes available, the initial beliefs framework is revised. In this case, the knowledge about the actual learning object is related to the learning process. This kind of system is named Participatory Learning System (PLS), because a set of beliefs and theories will affect how the information is accepted and processed, and the observations will affect the set of beliefs. This kind of framework is better described in [22].

The application of PLS in SSP consists in considering a collection of C scenarios with an index *k = 1,...,C*. Let V_k be the value associated with the trust in scenario *k*, and $V_k \in [0,1]$. Assume the knowledge is acquired through a series of vectors of observations: *D(1), D(2), ..., D(S),* and each vector has *C* elements. In the SSP problem context, $D_k(j) \in [0,1]$ represents the *k*-th trust value for the solution of a known scenario *C* when the *j*-th observation related to an unknown scenario *S* occurs.

The *D* vectors are employed to learn about the *V* values. The learning process is participative if the utility of each observation *D(j)* contributes to the acceptance of an up-to-date estimation of V´s as valid observations. In this sense, a mathematical formulation to update the index associated with the set of beliefs is given by Eq. (10).

$$V(j+1) = V(j) + \lambda\rho_j(D(j) - V(j)) \qquad (10)$$

where: *V(j+1)*, *V(j)* e *D(j)* are the vectors with *C* new values of trust in the beliefs, the old values of trust in the beliefs and the most recent observation, respectively. Furthermore, $\lambda \in [0,1]$ is the learning rate and ρ_j corresponds to the level of compatibility between the observed data (*D(k)*) and the set of beliefs (V_k) in a manner that $\rho_j \in [0,1]$.

The compatibility measure ρ_j can be viewed as a function of the most recent set of beliefs and current observations. The used function for ρ_j is given by Eq. (11).

$$\rho_j = \frac{1}{c}\sum_{k=1}^{c} | D_k(j) - V_k(j) | \qquad (11)$$

A second improvement in the PLS is the insertion of an arousal mechanism which monitors the confidence in the set of beliefs. The higher the arousal rate is, the confidence of the set of beliefs is leaving the system more open to learn with conflicting observations and to change current beliefs. Let $a_j \in [0,1]$ be the arousal index. The higher value of a_j enables more criticism of the current set of beliefs. Mathematically it corresponds to Eq. (12).

$$a_{j+1} = a_j + \gamma(\rho_{j+1} - a_j) \qquad (12)$$

The arousal index is combined with the participatory learning system through Eq. (13).

$$V(j+1) = V(j) + \lambda(\rho_j)^{1-a_j}(D(j) - V(j)) \qquad (13)$$

The Equations (12) and (13) are combined to form a participatory learning system with two parts: one is responsible for learning from observation and the other considers different beliefs.

6. Computational Results

To test the proposed framework, 15 different types of instances were automatically and randomly generated. The instances were classified according to the number of ports and the transportation matrix type. Each instance was associated to a specific transportation matrix T in a way that the container ship capacity would not be exceeded. This means every transportation matrix would be feasible if Eq. (14) held.

$$\sum_{i=1}^{p} \sum_{j=p+1}^{N} T_{ij} \le D \times R \times C, \ \forall p = 1, ..., N \qquad (14)$$

According to Avriel et al., three types of transportation matrix can be created: 1 - Mixed, 2 - Long, 3- Short [4]. This classification for the transportation matrix expresses how long a container must be carried by the container ship from port to port. The Short Transportation Matrix indicates that the majority of the containers will remain for a small number of ports until unloading. The Long Transportation Matrix indicates that most containers will remain onboard the container ship from port to port before being unloaded at one of the last ports. The Mixed is created in a manner that mixes Short and Long instance characteristics. There are different types of containerships that can carry between 500 (Handysize type) to 10.000 (New Panamax type), and even 18.270 (Ultra Large Container Vessel type - Mærsk Mc-Kinney Møller) containers. The ship dimensions adopted for the instances presented in this article were D = 5, R = 6, C = 50 which corresponds to a Supramax ship type that can carry 1500 containers. All instances are available at the following site:

https://sites.google.com/site/3dcontainershipproject/home

One important test of the mathematical model, given by Eq. (1)-(8), is to verify if the solutions that come from it are "immune" to the stochastic values of $\hat{T}_{ij}$. For this purpose, two sets of scenarios were created: a training and a validation set. The training set was composed of 10 different scenarios for each one of the 15 instances and with equal probability of occurrence. The validation set was composed of 50 different scenarios for each one of the 15 instances and with equal probability of occurrence. The steps to evaluate the mathematical model were as follows:

(i)Select one scenario from the training set and obtains the best solution for it according to the model (1)-(8). The solution for each scenario **Ci** was obtained using just one transportation matrix T_{ci} (**C1-C10** solutions) or all ten transportation matrices corresponding to ten scenarios from the training set (**M1-10** solution) using the model (1)-(8).

(ii)Evaluate the solutions obtained in step (i) in the validation set.

(iii)Compute the mean performance of each solution from the training set in the validation set.

Table 2: Performance results for the Genetic Algorithm in the validation set.

Table 2 presents the performance of solutions that came from 10 scenarios of the training set (**C1, ...,C10**) and solution for the model (1)-(8) that observed all ten scenarios (**M1-10**) applied in 50 unknown scenarios that formed the validation set. Due to space constraints, only the performance of **C1**, **C10** and **M1-10** solutions are shown in Table 2.

Table 2: Results for the Genetic Algorithm in the validation set.

N	**M**	**α=1, β=0**			**α=0, β=1**		
		C1	**C10**	**M1-10**	**C1**	**C10**	**M1-10**
10	**1**	12.40	0.25	104.1	0.00	4.80	1.02
	2	47.08	6.64	8.86	3.17	2.21	1.07
	3	2.07	2.90	815.5	361.0	47.26	0.05
15	**1**	6.35	17.25	0.00	30.48	54.51	2.35
	2	13.23	15.36	88.47	41.20	2.28	2.16
	3	1.50	0.00	168.0	96.06	24.47	0.83
20	**1**	6.47	0.00	41.47	89.77	44.14	1.80
	2	13.98	0.91	15.77	6.08	0.00	0.00
	3	0.95	1.51	55.23	130.1	79.24	0.00
25	**1**	13.24	10.02	135.7	60.27	9.22	4.82
	2	20.42	4.67	6.65	5.45	0.80	3.21
	3	0.67	1.39	149.7	93.85	19.74	0.74
30	**1**	7.30	13.58	49.78	111.2	30.25	7.54
	2	0.00	5.44	3.69	2.04	0.00	19.43
	3	0.00	1.01	81.66	24.03	0.00	0.63
Mean		**9.71**	**5.40**	**3.04**	**114.9**	**70.32**	**21.26**

In Table 2, the column **C1** presents the performance of the best solution obtained from known scenario 1. The values in Table 2 express the percentage deviation from the best solution found for the unknown scenarios.

There are two set of parameters in Table 2: **(α, β)=(1, 0)** and **(α, β)=(0, 1)**. The first only minimizes the number of movements and the second only the instability measure, respectively. The best mean value (last Table line) was obtained with **M1-10** for both sets.

Table 3: Different PLS in the validation set: **(α, β)=(1, 0)**.

N	**M**	**α=1, β=0**				
		M1-	**SAP**	**SAP**	**SAP**	**SAP**
10	**1**	1.02	0.17	0.08	0.08	0.08
	2	1.07	1.62	1.61	1.61	1.61
	3	0.05	0.07	0.07	0.07	0.07
15	**1**	2.35	0.08	0.08	0.08	0.08
	2	2.16	1.14	0.61	0.61	0.61
	3	0.83	0.19	0.31	0.31	0.31
20	**1**	1.80	0.08	0.41	0.34	0.34
	2	0.00	0.64	0.11	0.11	0.11
	3	0.00	0.17	0.15	0.15	0.15
25	**1**	4.82	0.56	0.56	0.56	0.56
	2	3.21	0.29	0.58	0.58	0.58
	3	0.74	0.41	0.26	0.26	0.26
30	**1**	7.54	0.22	0.22	0.21	0.21
	2	19.43	0.38	0.38	0.16	0.16
	3	0.63	0.29	0.24	0.24	0.24
Mean		**9.71**	**0.42**	**0.37**	**0.37**	**0.35**

Tables 3 and 4 present the **PLS** performance for different parameter values. All versions of **PLS** produced much better values than the solutions obtained for a fixed scenario (**C1**, …, **C10**, **M1-10**). The results strongly suggest that **PLS** is robust in terms of its parameters, because a modification in parameter values hardly changed the performance of **PLS**. For **(α, β)** = **(1, 0)** the **PLS** solution was just 5% of the **M1-10** function value. For **(α, β)** = **(0, 1)** the **PLS** solution was only 10% of the **M1-10** function value. A possible explanation for these results is that **PLS** dynamically chose which one of the eleven solutions (**C1, ...,C10** and **M1-10**) should be used.

Table 4: Different PLS in the validation set: **(α, β)**=**(0, 1)**.

N	**M**	**α=0, β=1**				
		M1-10	**SAP $\lambda\rho_j$=0.1**	**SAP ρ λ=0.1**	**SAP ρ λ=0.1 γ=0.1**	**SAP ρ λ=0.9 γ=0.1**
10	**1**	4.8	21.1	18.4	18.4	18.4
	2	2.2	3.0	3.0	2.8	2.8
	3	47.2	15.1	15.1	17.8	17.8
15	**1**	54.5	13.4	12.7	5.4	5.4
	2	2.2	3.3	3.3	3.0	3.0
	3	24.4	5.1	5.1	13.4	13.4
20	**1**	44.1	13.7	14.5	37.0	37.0
	2	0.0	8.0	12.0	11.5	11.5
	3	79.2	1.2	3.1	6.1	6.1
25	**1**	9.2	7.7	3.1	4.2	4.2
	2	0.8	7.1	6.2	6.2	6.2
	3	19.7	16.4	22.1	33.7	33.7
30	**1**	30.2	9.9	8.8	14.0	14.0
	2	0.0	8.6	15.6	17.6	17.6
	3	0.0	6.6	6.8	5.6	5.6
Mean		**21.2**	**9.3**	**9.9**	**13.1**	**13.1**

7. Conclusions and Future Works

This article presented a framework for solving the Stochastic Stowage Planning Problem. The resulting problem had binary variables, was multi-objective, non-linear, stochastic and large-scale for real instances. To solve such a problem, an alternative representation, named Representation by Rules, was employed and combined with a genetic algorithm, a simulation procedure and a participatory learning system. This new approach can deal with a problem for which the corresponding binary representation demands 405,450,000 variables. The use of the participatory learning system is important; because it improves the quality of solutions obtained and greatly outperforms the solution given by a method based on the evaluation of multiple scenarios. In future studies, other learning-based methods will be tested and compared, such as the Q-Learning method. Also, instead of using just a genetic algorithm with different values of weights for each objective, a multiobjective optimization method like NPGA or NSGA II will be employed and its performance tested.

8. References

[1] D. Ambrosino, A. Schiomachen, E. Tanfani, "A decomposition heuristics for the container ship stowage problem", *J. Heuristics*, v.12, p. 211-233, 2006.

[2] D. Ambrosino, D. Anghinolfi, M. Paolucci, A. Sciomachen, "An Experimental Comparison of Different Heuristics for the Master Bay Plan Problem", *Lecture Notes in Computer Science*, Volume 6049, pp. 314-325, 2010.

[3] M. Avriel, M. Penn, "Container ship stowage problem", *Computers and Industrial Engineering*, v. 5, p. 271-274, 1993.

[4] M. Avriel, M. Penn, N. Shpirer, S. Wittenboon, "Stowage planning for container ships to reduce the number of shifts", *Annals of Operations Research*, v. 76, p. 55-71, 1998.

[5] M. Avriel, M. Penn, N. Shpirer, "Container ship stowage problem: complexity and connection to the coloring of circle graphs", *Discrete Applied Mathematics*, v. 103, p. 271-279, 2000.

[6] A.T. Azevedo, C.M. Ribeiro, G.J. Sena, A.A. Chaves, L.L. Salles, A.C. Moretti, "Solving the 3D Container Ship Loading Planning Problem by Representation by Rules and Beam Search", in ICORES'12, p. 132-141, 2012.

[7] A.T. Azevedo, C.M. Ribeiro, G.J. Sena, A.A. Chaves, L.L. Salles, A.C. Moretti, "Solving the 3D Container Ship Loading Planning Problem by Representation by Rules and Meta-heuristics", to appear in *International Journal of Data Analysis Techniques and Strategies – Special issue on: "Optimization and Simulation Real-life Scenarios"*, 2013.

[8] C. Blum, A. Roli, "Metaheuristics in combinatorial optimization overview and conceptual comparison", *ACM Computing Surveys*, v. 35, n. 3, p. 268 -308, 2003.

[9] R.C. Botter, M. A. Brinati, "Stowage Container Planing: a model for getting optimal solution." In: *International Conference of Computer Applications in the Automation of Shipyard Operation and Ship Design*, Rio de Janeiro, p. 217-229, 1991.

[10] Dubrovsky, O.; Levitin, G., Penn, M. (2002) 'A Genetic Algorithm with a Compact Solution Encoding for the Containership Stowage Problem', *Journal of Heuristics*, v. 8, p. 585-599.

[11] J. Dupacova, "Stochastic programming: Approximation via Scenarios", 1996.

[12] L. Fan, M.Y.H. Low, H.S. Ying, H.W. Jing, Z. Min, W.C. Aye, "Stowage Planning of Large Containership with tradeoff between Crane Workload Balance and Ship Stability", *Proceedings of the International MultiConference of Engineers and Computers Scientists*, Vol III, pp. 1-7, 2010.

[13] Y. Guan, K-H. Yang, Z. Zhou, "The Crane Scheduling: models and solution approaches", *Annals of Operations Research*, 203, p. 119-139, 2013.

[14] J. H. Holland, “Adaptation in natural and artificial systems”, The University of Michigan Press, 1975.

[15] A. Imai, K. Sasaki, E. Nishimura, E., S. Papadimitriou, “Multi-objetive simultaneous stowage and loading planning for a container ship with container rehandle in yard stacks”, *European Journal of Operational Research*, 171, pp. 373-389, 2006.

[16] D.H. Lee, H.Q.,Wang, L. Miao, “Quay crane scheduling with non-interference constraints in port container terminals”, *Transportation Research E, 44*, p. 124–135, 2008.

[17] Z. Michalewicz, “Genetic Algorithms + Data Structures = Evolution Programs”, 3rd edition, Springer-Verlag, 1996.

[18] E. Nishimura, A., Imai, S. Papadimitriou, “Berth allocation planning in the public berth system by genetic algorithms”, *European Journal of Ooperational Research*, 131, p. 282-292, 2001.

[19] A. Sciomachen, E. Tanfani, “A 3D-BPP approach for optimizing stowage plans and terminal productivity”, *European Journal of Operational Research*, v.183, p. 1433–1446, 2007.

[20] R. Stahlbock, S. Voss, “Operations research at container terminals: a literature update”, *OR Spectrum*, 30 (1), p. 1–52, 2008.

[21] D. Steenken, S. Voss, R. Stahlbock, “Container terminal operation and operations research - a classification and literature review”, *OR Spectrum*, 26 (1), p. 3–49, 2004.

[22] R. R. Yager, “A model of Participatory Learning”, *IEEE Transactions on Systems, Man and Cybernetics*, vol. 20, no. 5, p. 1229-1234, 1990.

[23] I. Wilson, P. Roach, “Container stowage planning: a methodology for generating computerised solutions”, *Journal of the Operational Research Society*, v. 51, p. 1248-1255, 2000.

[24] I. Wilson, P.A. Roach, “Principles of combinatorial optimization applied to container-ship stowage planning”, *Journal of Heuristics*, v. 5, p. 403-418, 1999.

[25] Y. Zhu, A. Lim, “Crane scheduling with non-crossing constraint”, *Journal of the Operational Research Society, 57*(12), p. 1464–1471, 2006.