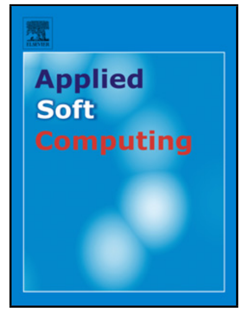
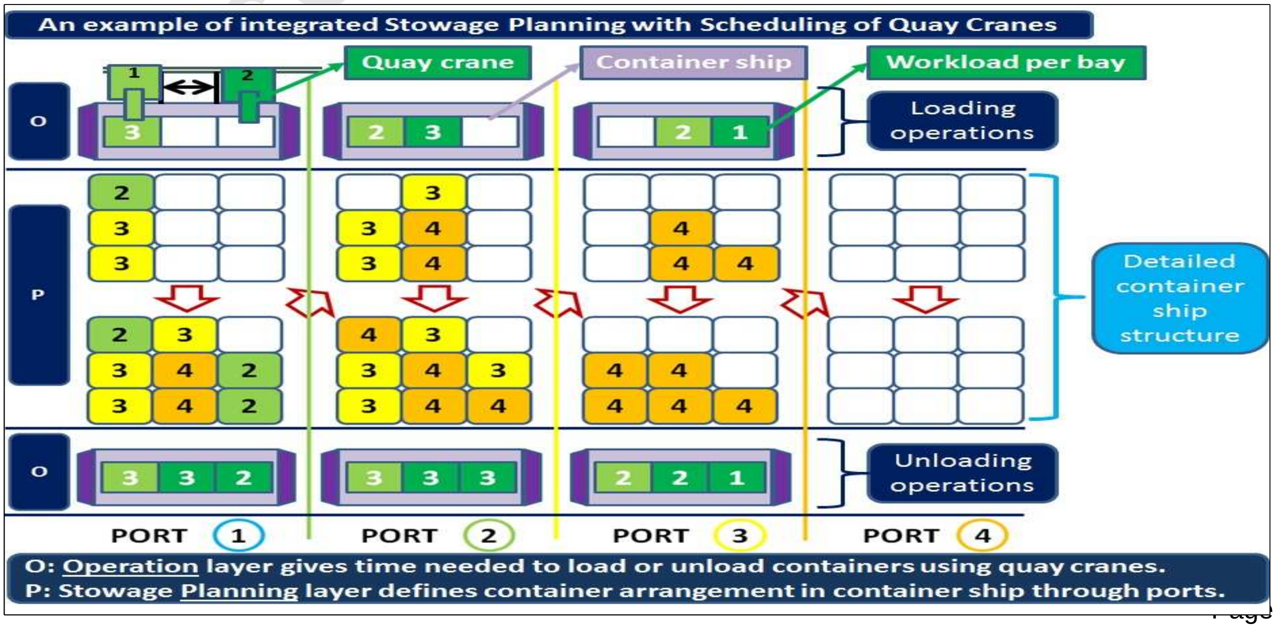
Accepted Manuscript



Title: Solving the 3D Stowage Planning Problem integrated with the Quay Crane Scheduling Problem by Representation by Rules and Genetic Algorithm

Author: Anibal Tavares de Azevedo Luiz Leduino de Salles Neto Antônio Augusto Chaves Antônio Carlos Moretti

PII:	S1568-4946(18)30012-7
DOI:	https://doi.org/doi:10.1016/j.asoc.2018.01.006
Reference:	ASOC 4653
To appear in:	*Applied Soft Computing*
Received date:	22-11-2015
Revised date:	23-12-2017
Accepted date:	9-1-2018



An example of integrated Stowage Planning with Scheduling of Quay Cranes
Quay crane
Container ship
Workload per bay
O
P
O
Loading operations
Detailed container ship structure
Unloading operations
PORT 1
PORT 2
PORT 3
PORT 4
O: Operation layer gives time needed to load or unload containers using quay cranes.
P: Stowage Planning layer defines container arrangement in container ship through ports.

HIGHLIGHTS

This paper showed a new approach to solve the 3D Stowage Planning for Container ship (3D SP) problem integrated with the Scheduling of quay cranes (SQC) problem. It has three modeling advantages:

• It shows how the 3D SP will affect other problems related to port operation, like the SQC problem.

• It provides better accuracy in estimating total time, which can save the charterer charges for the extra use of the vessel. In the instances studied, the solution from integrated approach provided solutions with a 45.82% higher total time spent, on average, and prevented an underestimation of necessary time for ship travel.

• In 40% of the instances, the Integrated 3D SP and SQP problem helped to avoid a misleading analysis, where the adoption of good practices for 3D SP produce a worse total time to unload and load the ship.

• The developed approach enables a series of analysis that partially explains the advantages on use bigger container ships and justifies a long term tendency of continuous increasing on container ship size.

Solving the 3D Stowage Planning Problem integrated with the Quay Crane Scheduling Problem by Representation by Rules and Genetic Algorithm

Anibal Tavares de Azevedo[a], Luiz Leduino de Salles Neto[b], Antônio Augusto Chaves[b], Antônio Carlos Moretti[a]

[a] *University of Campinas, R. Pedro Zaccaria, 1300 - Caixa Postal 1068 - CEP 13484-350 - Limeira - São Paulo, Brazil, anibal.azevedo@fca.unicamp.br, moretti@ime.unicamp.br, ++55 19 3701-6662.*

[b] *Federal University of São Paulo, Rua Talim, 330 - São José dos Campos - São Paulo -CEP 12231-280, Brazil, luiz.leduino@unifesp.br, antonio.chaves@unifesp.br, +55 12 3309-9500.*

Abstract

The operational efficiency of a port depends on proper container movement planning, called "stowage planning", especially because unloading and loading container ships demands time, and this has a cost. Thus, the optimization of operations through stages is important to avoid blockage activities. This paper proposes a framework for solving the 3D Stowage Planning (3D SP) problem for Container ships integrated with the Scheduling of Quay Cranes (SQC) problem. 3D SP and SQC problems are interrelated and combinatorial, justifying the applications of meta-heuristics like a genetic algorithm combined with Simulation and Representation by Rules. The robustness of the developed approach is attested in problems with 30 ports, 1500 TEUs ship or 15 ports and 22,000 TEUs ship and two quay cranes. These studies showed that the addition of the SQC problem leads to a 45.82% increase in load/unload time for the 3D SP problem solution, on average. This could help the charterer to avoid paying charges to the shipowner due to its an extra unplanned use of the vessel. Additionally, the developed methodology also helps to explain a long term phenomena of continuous increasing in container ship capacity since 1950's.

Keywords: Stowage Planning, Quay Crane Scheduling, Genetic Algorithm, Representation by Rules

1. Introduction

According to [1], 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. The problem of optimizing seaport operations should be defined and methods of solution proposed for such tasks. In this sense, the classification made by [1, 2] is useful, since seaport container terminal operations are 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 minimum safety distance between two ships, given that two ships 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 movements 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. Land-side transport: This problem deals with unloading and loading containers efficiently off from/onto trucks or trains to maximize the flow through the port yard.

The division of port operations into five problems may be helpful to develop mathematical models, but the final objective is to find a solution that encompasses all operations. For example; from problem (2), it could be said that the number of movements in Stowage Planning is just an estimation of necessary time to unload or load the ship, because it depends on the availability of cranes and quayside transport. So, a more precise model should couple Stowage Planning and Quay Crane Scheduling problems.

Recent literature shows a tendency to integrate the five aforementioned problems. For example, some articles integrate berth allocation and quay crane scheduling [3, 4, 5, 6, 7]; others integrate the allocation of berths and yard operation planning [8]; and some integrate empty container allocation in the yard with vehicle routing [9].

This recent tendency is justified by the obvious fact that the optimization of operations for just one stage does not increase overall port efficiency, because further and non optimized stages behave as blockage.

The integration of Berth allocation and Crane Split is also a problem as mentioned by [10], who analyzed and classified 120 articles. In the following commentary, they concluded: "Nevertheless, to obtain more reliable estimates, BAP, QCAP, and QSCP must be solved jointly, which is referred to as integrated seaside operations planning. A deep integration means to solve a monolithic model where the interdependencies of the involved problem-individual decisions are considered on the background of the merged set of constraints. Solving a monolithic model can deliver the best overall solution but is usually extremely difficult due to the huge complexity of the merged problem."

Indeed, the integration of port problems is very difficult due to the computational complexity of the mathematical model related to each stage which is NP-Hard. In fact, a port may be seen as a complex system as observed by [11]: "Many complex systems such as manufacturing, supply chain, and container terminals are too complex to be modeled analytically. Discrete event simulation has been a useful tool for evaluating the performance of such systems. However, simulation can only evaluate a given design, not provide more optimization functions. Therefore, the integration of simulation and optimization is needed."

Stowage Planning studies have totally ignored quay crane scheduling or have dealt superficially with it by making simple assumptions such as the use of only one quay crane [12, 13, 14, 15, 16, 17, 18, 19]. In fact, none of the Stowage Planning papers in the literature have considered the important features of quay crane scheduling described in [6].

Our study employs a simulation to represent logical and physical constraints instead of using a monolithic mathematical model with binary variables and algebraic equations. This means the use of an alternative approach called Simulation-Optimization. A very detailed review about it is given by [20] and a taxonomy is given on [21]. The paper [22] gives a detailed practical introduction to this approach. Simulation-Optimization is simply the combination of simulation, that describes the complicated logical operations of one system, with optimization, that cares about how to choose the decision variables values in order to minimize costs. According to the classification proposed by [21], the approach developed in this paper could be classified as MH, i.e., an optimization algorithm that employs a simulation to evaluate each solution. Additionally, our study applies a new innovative approach on representing decision variables that substantially reduces the search space.

The decision variable in the developed Simulation-Optimization approach describes the action that could be made in a specific activity in generic terms. For example, instead of creating binary variables and corresponding equations that control the position of each container in a ship, it is better to create a generic procedure of how to unload or load containers into a ship and couple the procedure with a simulation. This manner of representing decisions has been successfully applied in 3D SP and is described in [17], which we will extend to SQC in this paper.

The paper contributions are:

- Solve for the first time the integrated problem of Stowage planning with a detailed Quay Crane Scheduling which actual monolithic mathematical models with binary variables and algebraic equations are not capable to solve or even model;
- This was done by extending and employing a new simulation - optimization methodology which was successful to obtain quality solutions for the Stowage planning problem;
- The solution for integrated problem avoids underestimation on time necessary to do unloading and loading operations which could leads to solutions for the Stowage Planning Problem with a higher operation costs;
- Finally, the developed approach enables a series of analysis that partially explains the advantages on use bigger container ships and justifies a long term tendency of continuous increasing on container ship size.

This paper is organized as follows. Section 2 presents the model features for problems 3D SP and SQC. Section 3 explains the representation by rules and how it is combined with a Genetic Algorithm. Sections 4, 5, and 5 presents and discusses the computational results. Finally, Section 7 presents the conclusions and Section 8 possible future work.

2. A Definition of the Problem

2.1. The Stowage Planning Model

A solution for 3D SP should produce a stowage plan that is strongly related to the cellular structure of container ships, as shown in Figure 1.

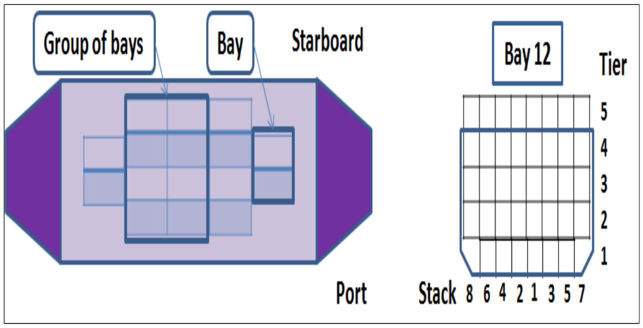


Fig. 1. The cellular structure of a container ship: top view and a cross-sectional view of a bay.

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:

- Containers to be unloaded at a given port are at the top of the stack.
- 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 port efficiency.

The more such blockage occurs, the longer the ship must stay in the port. So, Stowage Planning (SP) is the key to minimizing re-handling movements. In [23] it was proved that the 2D stowage plan is NP-Hard, which justified the development of a series of heuristics and meta-heuristics to obtain good solutions for this problem, as may be seen in approaches from the literature [24, 12, 16, 19, 15, 14, 13, 25]. The container's position in the stack affects ship and stack stability. In the literature, there are two different manners to deal with ship stability in the mathematical model: as an objective function [26, 14], or as a constraint [27, 16, 15, 18, 25]. Stability concerns are related to ship features, like metacentric height and trim [14], or the limits of height and weight in a stack [16, 25], and vertical equilibrium [28]. A recent review provided a classification of previous articles [18].

The mathematical model assumptions adopted in [17] has been adopted here and 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 ($r = 1, 2, \cdots, R$), columns ($c = 1, 2, \cdots, C$) and bays ($d = 1, 2, \cdots, D$) with maximum capacity of $R \times C \times D$ containers. An irregular format may be achieved by simply adding constraints which represent imaginary containers that occupy the same spaces during the whole voyage [19].

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

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

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

(e) For each Port i, $i = 1, 2, \cdots, N$, such that the container ship can be loaded with containers whose destination are ports $i + 1, \cdots, N$.

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

The Appendix D presents the corresponding mathematical model formulation. The Appendix G presents a small numerical example to illustrate some model features related with number of movements and stability.

2.2. Quay Crane Scheduling Model

Quay Crane Scheduling is directly related to stowage planning, detailed in Figure 2.

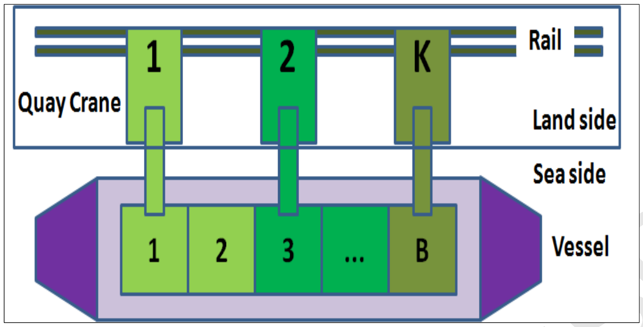


Fig. 2. Quay Crane Scheduling Problem.

The Scheduling of Quay Cranes (SQC) problem should consider constraints related to equipment characteristics. For example, quay crane assignment through time must not allow the crossing of cranes, since they are mounted on the same rail. One formulation of this problem, with non-crossing constraints, was shown as NP-Hard in paper [29]. This motivated the development of a genetic algorithm by the same authors. A lot of authors in the literature adopted the same procedure of modeling the quay crane problem to solve the models by meta-heuristics [2, 30, 31, 32, 33].

The SQC problem resembles that of parallel machines, but it has additional features that make it more complex as observed by [34]. In this study, the following assumptions were made in terms of objective function and constraints.

(a) Instead of dividing the container ship subject into arbitrary sections as done in [31], each container ship bay was considered. This enabled the direct use of the stowage planning solution as information of the total workload for each bay in the container ship.

(b) It was assumed that the total workload was the total number of containers and each container stored in position (r, c, d) in Port i would take $\psi_{1i}(r, c, d)$ units of time to be unloaded and $\psi_{2i}(r, c, d)$ to be loaded into the container ship.

(c) Each bay could only hold one crane at a time.

(d) Once one quay crane started, the service of loading or unloading a bay would not stop until it was finished (this is no preemption assumption).

(e) When a crane changed its operation to another bay, it would take a constant time to reach its destination which was equal to move three containers. Thus, it was assumed that this repositioning movement demanded three units of time.

(f) All cranes had the same constant service operation rate of one movement per time unit.

(g) One crane could not cross another, since they were mounted on the same rail.

(h) A minimal distance between cranes had to be observed. This meant that sometimes one crane would not be allowed to move or start its service until a second crane had finished its work and moved to a new position that ensured a minimal distance of one bay between two cranes.

The last two assumptions are what make the problem more complex than the parallel machines problem [3].

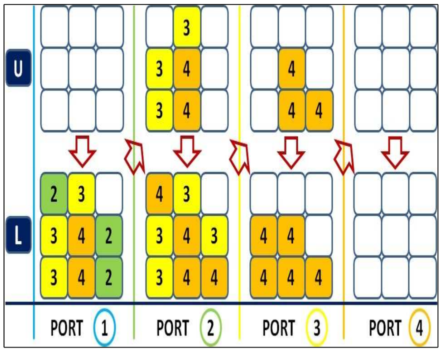
The Appendix E presents the corresponding mathematical model formulation. The Appendix H presents a small numerical example to illustrate some model features related with the impact of non-crossing constraint impact on quay cranes movement and the necessary total time to perform tasks along a container ship.

2.3. Integrating Stowage Planning with Quay Crane Scheduling

The problems modeled in subsections 2.1 and 2.2 are not independent. Once a container ship arrives at or leaves a port it is necessary to perform unloading and loading container movements according to previous Stowage Plan. The 3D SP model tries to produce container ship arrangement that minimizes the number of movements. As shown in Figure 3, this is a hard problem since the arrangement in one port could greatly affect the arrangement in future ports. The element i, j in Figure 3a determines the quantity of containers that should be transported from origin Port i (row O_i) to others j destination ports (D_js columns on row O_i). Figure 3b shows the container ship arrangement in each port after unloading (U) and loading (L) operations.

	$D1$	$D2$	$D3$	$D4$
$O1$	0	3	3	2
$O2$	0	0	2	2
$O3$	0	0	0	2
$O4$	0	0	0	0

(a) Information about the number of containers that must be transported through ports.



(b) Container ship arrangement after unloading and loading operations.

Fig. 3. One solution for the Stowage Planning Problem (not optimal).

The real problem, however, is more complex since the number of movements is just an estimation of the time necessary to unload or load ship using Quay Cranes. This is done by coupling the 3D SP and SQC problems through transforming the value of the variable that controls container position into total workload for unloading and loading operations. In mathematical terms, it leads to the introduction of non-linear constraints that can be linearized with proper techniques. This, however, makes the search for an optimal solution a challenge for exact methods. Figure 4 shows the physical meaning of coupling both models.

In Figure 4, once a container ship arrangement is given, it is possible to translate the total number of movements in each bay into total workload per bay. For example, suppose one solution suggests loading the container ship with containers $3, 3, 2$ in Bay 1, containers $3, 4, 4$ in Bay 2, and containers $2, 2$ in Bay 3 as shown in Figure 4. This information comes from the first model which is responsible for planning the best arrangement considering for loading and unloading through all ports. This first phase could be seen as a planning phase (P). In a second phase it is necessary to put in details about the necessary port operations for such planning. This means using

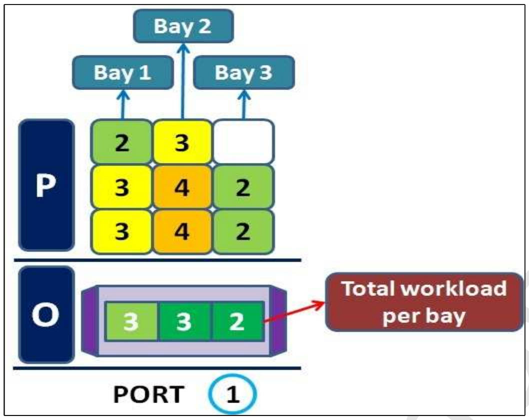


Fig. 4. Coupling Stowage Planning and Quay Cranes Operation problems.

the information of how many containers there are in each bay, from the first phase, to compute the total workload per bay. Once this is done, the second model will try to determine the best policy of quay crane operation without making any prohibited movement. This phase may be seen as an operation phase (O) and will use the following values per bay: $3, 3, 2$.

The concepts described in Figure 4 can be applied to the planning model given in Figure 3b to produce Figure 5, that illustrates how planning (P) and operation (O) phases are interrelated.

In Figure 5, the estimation of movements to unload and load in the stowage planning phase (P) will be replaced by total workload for quay crane operation simulation (O). For such simulation, the workload for each bay in each port should be computed from the container ship arrangement, enabling a more accurate estimation of time to perform a proposed stowage plan. This means, however, that a 3D SP problem with 30 ports must solve SQC 60 times.

The Appendix F presents the corresponding mathematical model formulation. The Appendix I presents a small numerical example to illustrate the problems in trying to find optimal solution through the integrated model.

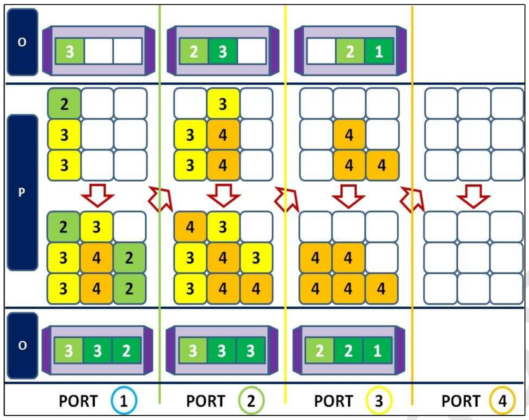


Fig. 5. The interrelation of Planning (P) and Operation (O) for the problem presented in 3b.

3. Solution Technique

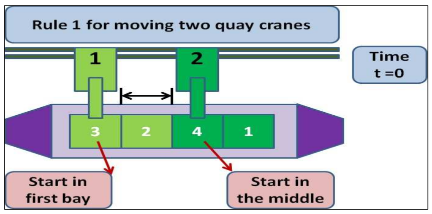
This problem requires more than the integration of simulation and optimization methods in order to deal with decision-making in complex systems, such as ports. For such a purpose, the Zadeh's incompatibility Principle is very useful [35]: "Stated informally, the essence of the principle is that as the complexity of a system increases, our ability to make precise and yet significant statements about its behavior diminishes until a threshold is reached beyond which precision and significance (or relevance) become almost mutually exclusive characteristics."

The approach adopted presents generic procedures, as had been successfully applied in 3D SP (detailed in [17]). These procedures will be extended to SQC and 3D SP integrated with SQC as detailed in subsections 3.1 and 3.2, respectively.

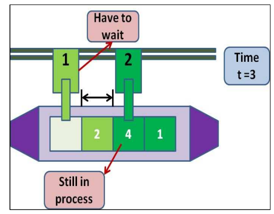
3.1. Representation by rules for Quay Crane Scheduling

After determining the container ship arrangement by solving 3D SP, through the use of unloading and loading rules detailed in Appendix A, the next step is to determine how much time is necessary to perform the operation. For such a task, a new second set of four rules related to quay crane scheduling are considered. To illustrate this second set of rules, Figure

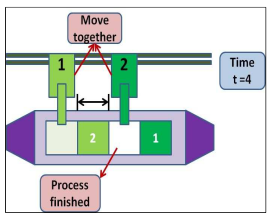
6 shows one of the rules that was employed in this paper. The time to move cranes was not counted in Figure 6 and the movement of one container was set at one unit of time.



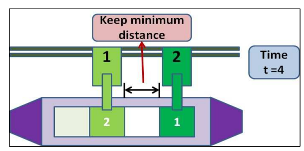
(a) Beginning



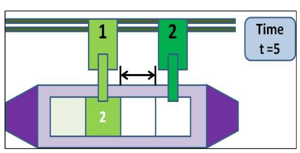
(b) Distance



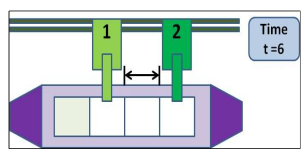
(c) Synchronism



(d) Distance



(e) End Crane 1



(f) Total Time

Fig. 6. A detailed description of operation via Rule 1 for 2 quay cranes.

The first rule for quay crane operations assigns one crane to the container ship bow and another one to the middle of the ship as shown in Figure 6a. All rules must lead to coordinated crane operations which respect the minimum security distance as shown in 6d. Sometimes this constraint implies in an increment in total time necessary to perform unloading or loading operations as shown in Figures 6b and 6c. Figures 6e and 6f show that the total time to process all the workload is the maximum time between crane one and crane two to perform all operations.

The second rule for quay crane operations is similar to the first rule, except that it assigns the first quay crane to the container ship stern and the second to the middle of the ship. The difference between the crane operations for Rules 1 and 2 can be seen by comparing Figures 6a-6f with Figures 7a-7d, respectively.

The first and second rules always allocate quay cranes at the same beginning position without observing the total container ship workload. The third and fourth rules differ from the first and second ones by doing the initial allocation of the second quay crane according to an equation that considers the "center of mass" of the total workload for the container ship. This difference is illustrated in Figures 7e-7f and 7g-7h.

Although, all four rules for quay cranes used in this study were developed for the operation of two quay cranes, the approach could be generalized to any number of quay cranes as detailed in Appendix B.

3.2. Combining Stowage Planning rules and Quay Crane Scheduling rules

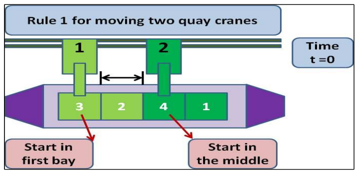
The efficiency of combining rules for loading and unloading container ships and for quay crane operation can be measured through a two-stage simulation. The planning stage (P) consists of solving 3D SP via the application of a first set of rules that produces container ship arrangement and two values: the total number of movements, and the instability measure as described in [17]. Then, the next stage, the operation phase (O), converts the container ship arrangement into workload for each container ship bay and then in total time for quay cranes to perform all operations according to a second set of rules: quay crane scheduling rules. Finally, the shipś total mooring time depends on the selection of rules to be applied in each stage from a set of 64 possible combinations. Table 1 describes all possible combinations. The symbol # can be 1, 2, 3, or 4. Each rule number represents the combination of three decisions among the two unloading (UR), eight loading (LR) and four quay crane (QC) rules. This means a complete solution should determine which rule will be applied in each port from a set of 64 possible combinations.

3.3. The Genetic Algorithm

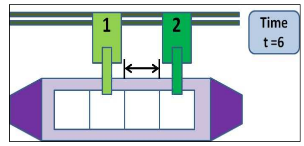
The genetic algorithm, parameters, and the same strategies of [17] were adopted in this work with changes made with the respect to the integration of 3D SP and SQC, as described in Figure 4. A detailed description of these parameters and its values is given on Appendix J. Additionally, the strategy of using simulation instead of binary variables with algebraic equations was adopted and modified to solve SQC. The simulation, as shown in Algorithm 1, is the evaluation scheme used for each genetic algorithm individual.

Algorithm 1 The evaluation algorithm encloses functions that emulate the simulation of 3D SP and SQC.

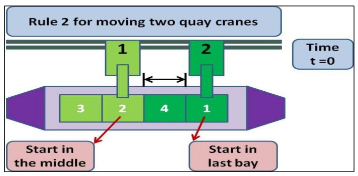
```
procedure EVALUATE
    p ← 1, tmov ← 0, instab ← 0.0
    Initialize(B, T)
    while p < N do
        [rc, rd, rq] = extractRules(s(p))          ▷ Translation using Table 1.

        if (p > 1) then
            [B, T] = unloading(rd, B, p);           ▷ Planning Phase(P)
            instab ← instab+ calcDXDZ(B);
            tmov ← tmov + TotalTime(rq, B);         ▷ Operation Phase(O)
            B ← B2;
        end if

        if (p < N − 1) then
            [B, T] = loading(rc, B, T, p);          ▷ Planning Phase(P)
            instab ← instab+ calcDXDZ(B)
            tmov ← tmov + TotalTime(rq, B);         ▷ Operation Phase(O)
            B ← B2;
        end if
        p ← p + 1;
    end while
    return α × tmov + β × instab;
end procedure
```



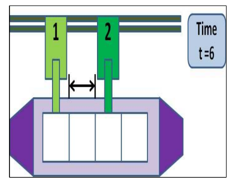
(a) Beginning of quay crane - Rule 1



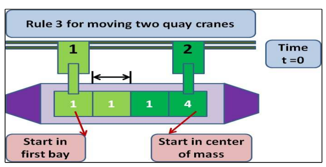
(b) Total Time



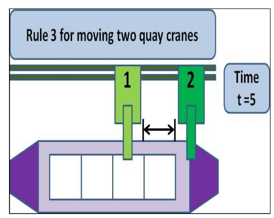
(c) Beginning of quay crane - Rule 2



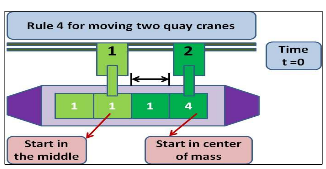
(d) Total Time



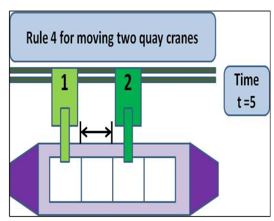
(e) Beginning of quay crane - Rule 3



(f) Total Time



(g) Beginning of quay crane - Rule 4



(h) Total Time

Fig. 7. A detailed description of operation via rules 1, 2, 3, and 4 for 2 quay cranes.

The symbols used in Algorithm 1 have the following meaning: *tmov* is a variable that represents the total time to perform quay crane operations (unloading or loading) according to the rule described in function name *rq*; and container distribution in ship spaces is represented by *B* matrix. The *instab* variable measures the instability of the container ship arrangement

Table 1. All rules produced by the combination of loading, unloading, and quay cranes movement rules.

Quay Rule	QC#							
Loading Rule	LR1		LR2		LR3		LR4	
Unloading Rule	UR1	UR2	UR1	UR2	UR1	UR2	UR1	UR2
Rule	1	2	3	4	5	6	7	8
Quay Rule	QC#							
Loading Rule	LR5		LR6		LR7		LR8	
Unloading Rule	UR1	UR2	UR1	UR2	UR1	UR2	UR1	UR2
Rule	9	10	11	12	13	14	15	16

after unloading and loading operations. The *unloading* and *loading* functions (using *rd* and *rc* rules, respectively) modify the container ship arrangement after unloading and loading operations, respectively. All eight loading rules and two unloading rules are detailed in Appendix A. The function *calcDXDZ* computes the instability measure as a distance between geometric center and the center of mass of the ship in accordance with the computation details given by the small example from [17]. The function *TotalTime* computes the total time necessary for a quay crane to perform an operation of unloading or loading the ship, considering the minimal distance between quay cranes as one bay as recommended by [36, 37], employing one rule *rq* of the four quay crane rules (see Appendix B for quay crane Rule 1 details). Finally, the symbols α and β are real numbers in interval $[0, 1]$ and represent the importance given to minimize the total time to unload/load ship or to the container ship arrangement instability measure, respectively.

Each genetic algorithm individual is a vector of integers for which each element is an integer number whose range varies from 1 to 64. Each vector element represents which rule will be applied in each port. Algorithm 1 is the evaluation of each individual. The genetic operators adopted are: OX crossover operator, and a mutation operator that modifies a certain percentage of bits from all uniform randomly chosen individuals for a given population. The operators and parameters adopted are detailed in [17].

4. Computational Results

The same set of instances used in [17] had been employed here in order to verify the performance of the developed framework. The ship dimensions adopted for the instances presented in this article are $D = 5$ (number of bays), $R = 6$, $C = 50$. All instances are available at the following site: `https://drive.google.com/open?id=0B4zUGKja09uEVndXZWsteG81S1k`.

In Tables 2 and 3, the column index I corresponds to instance number; N corresponds to how many ports the container ship has to pass through; the column index M refers to the type of transportation matrix (1 - Mixed, 2 - Long, 3- Short; there are more details on Appendix C); $FO1$ is the total time to perform all crane movements (see subsection 2.2 for constraints and parameter details); $FO2$ is the total measure of instability; $T(s)$ is the computational time spent in seconds to obtain the solution; $FO1 - SP$ is an estimation of the total time to perform all crane movements by considering only Stowage Planning constraints; one container movement takes one unit of time and time is simply divided by the number of cranes, which in this case is two. This estimation merely illustrates what happens with the solution from an optimization model that basically ignores, or that makes a simple assumption about quay crane operation. The value $FO1 - SP$ is presented just for cases where $FO1$ is minimized. The results presented in Table 2 and 3 were obtained with a program created in a Matlab 7.0, a machine with a 1.66 GHz Core Duo Intel Processor, RAM memory of 2 GB, and Windows Vista Operational System with Service Pack 2. The genetic algorithm (more details about parameters on Appendix J) was executed 5 times to illustrate the potential of the developed approach.

Tables 2 and 3 present the worst and best solutions found by merely minimizing the total workload time and by merely minimizing the instability measure, respectively.

The results presented in Tables 2 and 3 show that the Genetic Algorithm combined with the Representation by Rules approach was able to provide solutions in less than nine minutes for instances with 30 ports and a container ship with 5 bays, 6 rows, and 50 columns. In this case, each solution had $40,545,000$ binary variables only in the planning phase.

The difference between $FO1$ and $FO1 - SP$ is an important issue since, for all instances, $FO1 - SP$ underestimated the necessary time to unload and load the ship through all ports. $FO1$ was at least 18.36% higher than $FO1 - SP$ for Instance 3, and at most 84.92% higher than $FO1 - SP$ for

Table 2. Results obtained for GA when minimizing only total time.

I	N	M	FO1		FO2		T(s)		FO1-SP	
			Worst	Best	Worst	Best	Worst	Best	Worst	Best
1	10	1	5468	5359	776.11	621.87	222.95	206.00	3779	3896
2	10	2	3565	3448	650.49	715.86	186.79	199.73	2417	2131
3	10	3	10579	10397	418.42	554.81	218.32	226.01	8938	8555
4	15	1	7841	7460	415.64	436.76	347.34	429.65	5753	5301
5	15	2	4788	4588	658.87	642.27	300.37	298.93	2980	2694
6	15	3	16095	15729	640.24	372.60	429.42	336.57	12680	12870
7	20	1	8420	8050	337.20	334.08	494.63	451.82	5513	5787
8	20	2	5437	5066	473.00	561.12	414.09	395.05	2945	2897
9	20	3	20510	20288	574.33	665.96	462.93	427.99	16777	16441
10	25	1	9403	9110	256.66	274.98	658.27	718.28	5999	6145
11	25	2	4938	4679	596.59	455.26	607.69	571.34	3069	3055
12	25	3	28060	27806	329.34	349.80	636.99	631.45	22604	22329
13	30	1	9410	9126	287.76	279.09	621.03	667.37	6479	6355
14	30	2	4918	4709	248.77	225.97	559.20	656.13	2687	2875
15	30	3	34399	33819	423.69	281.54	732.75	707.35	27911	28137

instance 8, with a mean deviation of 45.82%. This means that Stowage planning models without careful and detailed features from the Quay Cranes Scheduling problem will produce inaccurate unloading and loading times that could lead the charterer to pay demurrage, i.e., which are charges that the charterer pays to the shipowner for the extra use of the vessel.

The Table 4 shows the possible savings to be obtained by computing the difference of total time for a model with proper quay crane modeling ($FO1 - Best$) and without it ($FO1 - SP - Best$) from Table 2. Each container unloaded/loaded after the time estimated by the model without quay crane data is multiplied by US$ 110.00. This value is a mean since the cost varies according to container type and how long the container stays in port. More details are available at [38].

Table 4 shows the savings could be anywhere from US$ $144,870.00$ (instance 2) to US$ $625,020.00$ (instance 15) per container ship trip.

Worse than the money wasted, is the misleading analysis that is carried out by using models without integrating 3D SP and SQC. For example, in Instance 1 in Table 2, the best solution had a value of $5,359$ and the worst

Table 3. Results obtained for GA when minimizing only the container ship arrangement instability measure.

I	N	M	FO1		FO2		T(s)	
			Worst	Best	Worst	Best	Worst	Best
1	10	1	7384	8258	296.33	296.00	205.92	221.99
2	10	2	4988	6856	297.10	296.67	259.74	191.19
3	10	3	12588	12596	295.71	294.37	219.40	257.75
4	15	1	14175	14207	202.61	202.10	476.96	368.63
5	15	2	8636	6633	207.63	206.49	283.00	293.62
6	15	3	16654	17341	210.58	208.92	329.30	346.73
7	20	1	13920	13533	155.60	154.75	432.06	552.70
8	20	2	10609	9084	158.82	157.56	434.95	427.61
9	20	3	21224	21891	509.77	487.35	539.97	405.29
10	25	1	14601	14666	127.57	126.54	691.01	555.37
11	25	2	8590	9383	209.86	206.82	554.53	471.54
12	25	3	30941	28680	162.92	154.31	576.99	611.04
13	30	1	15967	15153	109.12	107.51	628.86	623.65
14	30	2	7992	8627	107.34	107.24	551.69	547.36
15	30	3	35190	36805	180.72	163.12	700.12	696.65

Table 4. Savings obtained minimizing total time considering quay crane operations ($FO1$) instead of ($FO1 - SP$).

Instance	Saving(US$)	Instance	Saving(US$)	Instance	Saving(US$)
1	160,930	6	314,490	11	178,640
2	144,870	7	248,930	12	602,470
3	202,620	8	238,590	13	304,810
4	237,490	9	423,170	14	201,740
5	208,340	10	326,150	15	625,020

had value of $5,468$ according to $FO1$. But, according to $FO1-SP$ the best solution (found observing $FO1$) was $3,896$ and the worst solution was $3,779$. So, $FO1-SP$ does not recommend the same solution that $FO1$ does. In other words, it leads to a faulty loading and unloading sequence. This is not an isolated case. It happened in 40% of the instances: instances 6, 7, 10, 14, and 15 in Table 2. The next section explains in detail how these differences are related to SQC features.

Results show that optimization methodology must consider the trade-off between minimization of ship stay time and the stability measure in the determination of a proper arrangement of containers in a ship. In one option, the decision maker has to decide which combination of weights produces solutions that constitute a better trade-off in terms of stability and ship stay time. The weights represent the decision makers preferences in each instance. However, this approach may suffer with the effect of not reaching certain points on Pareto front [39]. Another option is to simultaneously consider stowage planning and stability measures, using the Pareto curve as detailed in [40].

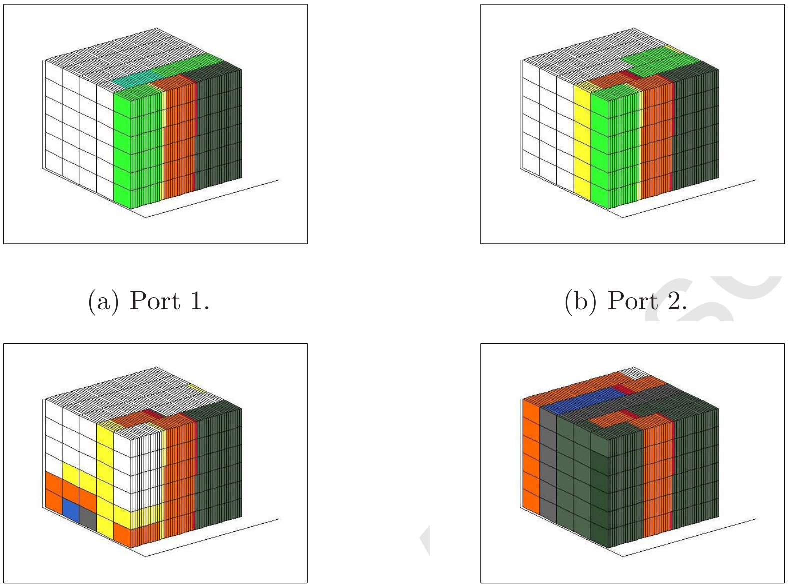
5. A Detailed Analysis of Results

The purpose of Figure 8 is to illustrate how the container ship arrangement is connected to quay crane scheduling in the first four ports for the best solution found in Instance 1 from Table 2. Figure 9 shows the corresponding quay crane scheduling for arrangements in Figure 8.

Figure 8a shows the container ship arrangement after departing from Port 1 and Figure 9a shows its consequent quay crane scheduling for unloading service in Port 2. All other subfigures have the same relationship. In the first ports, the containers are organized so that very few bays require unloading service. This happens because the objective function minimizes the total time spent, but completely ignores any stability measurement.

Figure 9a better presents the reason why $FO1-SP$ is an underestimation of unloading and loading total time of $FO1$. The ship has a total time of 112 units of unloading service time, but this could not be shared by the two quay cranes because the containers were not adequately distributed among the bays.

Another possible situation that also leads to underestimation of unloading and loading total time of $FO1$ is presented in Figure 9b. In this case, the two cranes could operate in parallel, but it does not imply in the division of



(a) Port 1. (b) Port 2.

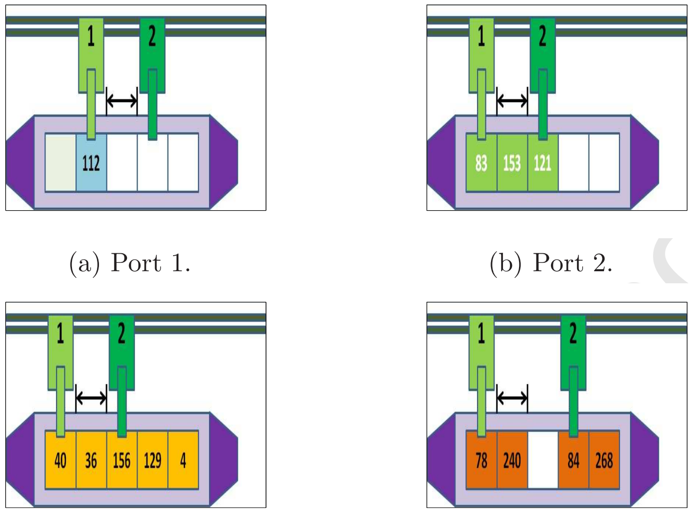
(c) Port 3. (d) Port 4.

Fig. 8. Detailed Container ship arrangement after loading operations in the first four ports for the best solution found with the total time minimization objective function.

total time by two. Quay crane 1 will finish the bay 1 at time 83, but cannot move to Bay 2 until quay crane 2 finishes its job at time 121.00. So, the ship unloading operation, instead of taking $\frac{357}{2} = 178.5$ units of time, will need $121 + 153 = 274$ units of time, meaning $FO1$ will be 53.93% higher than $FO1 - SP$.

Figures 10 and 11 also illustrate the best solution found for instance 1, but considering a completely different parameter setting, $\alpha = 0$ and $\beta = 1$. This setting tries to minimize only the instability measure for the container ship arrangement. The values of number of units of time, and the instability measure are found in Table 3.

Figure 10a shows the container ship arrangement after departing from Port 1. This container ship arrangement will produce a corresponding quay crane schedule to unload containers at Port 2 as shown in Figure 11a. All other subfigures of Figures 10 and 11 have the same relationship. Figure 10 also represents a more stable container ship arrangement than the one presented in Figure 8. The more stable arrangement of Figure 10 was ob-



(a) Port 1. (b) Port 2.

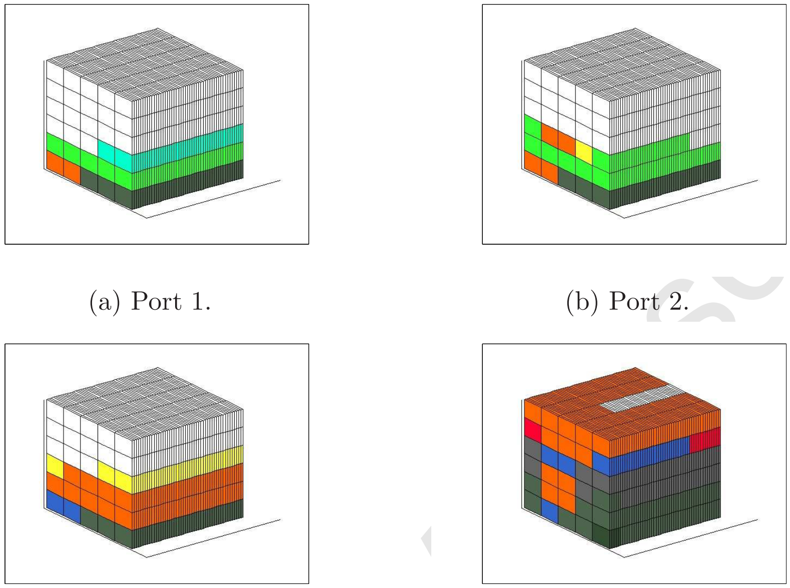
(c) Port 3. (d) Port 4.

Fig. 9. Detailed quay crane scheduling before unloading operations in the first four ports for the best solution found with the total time minimization objective function.

tained by ignoring the corresponding increment in time necessary to perform unloading and loading movements. This resulted in an extremely time consuming quay crane operation as shown in Figure 11. This is most evident, for example, in Port 5 where it may be seen by comparing Figures 9d and 11d. Also, the best solution for instance 1 when instability is minimized, as shown in Table 3, has a 54.09% higher value in total time than the best solution found when total time was minimized as shown in Table 2.

Another set of results reinforces the fact that minimization of instability does not imply in the minimization of number of movements. For example, in Instance 1 from Table 3, the worst solution in terms of FO2, had a significantly lower value of FO1 $(7,384)$ than the best solution $(8,258)$. This happened for 66% of the instances: Instances 1, 2, 3, 4, 6, 9, 10, 11, 14, and 16 from Table 3.

The most important conclusion from the results is that by using representation by rules, it was possible to quickly build the Stowage Plan and study its impact on port quay crane operation.



(a) Port 1. (b) Port 2.

(c) Port 3. (d) Port 4.

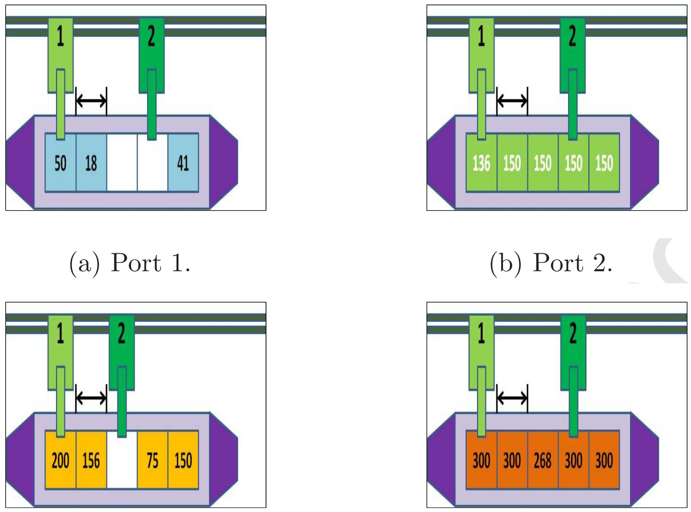
Fig. 10. Detailed Container ship arrangement in the first four ports for the best solution found with the instability minimization objective function.

6. Explaining an Economical Phenomena

During the last years the container ship experienced an increase in its size [41]. The Figure 12 illustrates the evolution of ships dimensions through years until 2013. The early container ship's at 1950's had a maximum capacity of 800 TEUs. During 1980's the maximum capacity had a great improvement from $3,000$ TEUs to $5,000$ TEUs. But, the 2000's had the most impressive evolution since in thirteen years (from 2000 to 2013) the capacity increased from $6,000$ to $18,000$ TEUs.

Nowadays exists ships with a capacity around $22,000$ TEU capacity, and there are plans now to produce container ships with a $27,000$ to $30,000$ TEU capacity [43]. The reason why ship's size is still increasing is because it allows the integration of routes reducing the number of necessary ships to carry goods [41, 43].

To verify the impact of continuous increasing of container ship's size in computational effort to find a quality solution and the necessary time to perform unloading and loading operations, a completely new set of instances had



(a) Port 1. (b) Port 2.

(c) Port 3. (d) Port 4.

Fig. 11. Detailed quay crane scheduling in the first four ports for the best solution found with the instability minimization objective function.

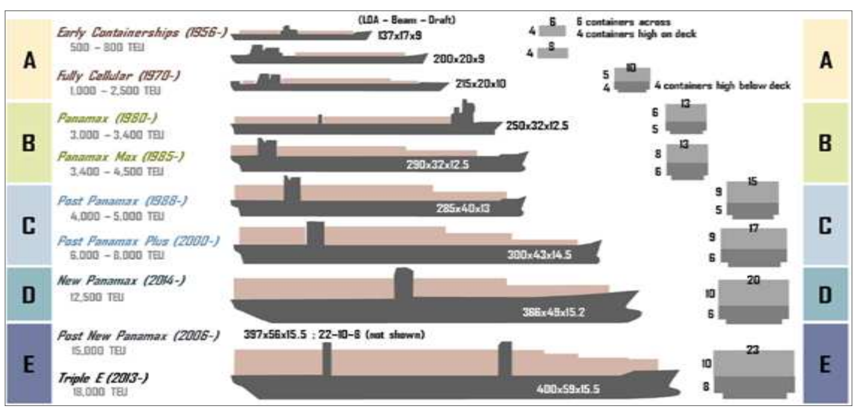


Fig. 12. Container ship's size evolution from 1956 until 2013. Source: [42].

been created and employed. This set is based on a hypothetical integration of three existing routes, as presented on Figure 13, which could be served by only one large container ship with a high carrier capacity [44].

The ship dimensions adopted for the instances presented in this article are

Fig. 13. Container ship routes through southeast Asia countries. Source: [44].

$D = 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60$ (number of bays), $R = 6$, $C = 25$, $N = 5, 10, 15$ (number of ports). All instances are available at the following site:

`https://drive.google.com/open?id=1attXqQZewMWr545-HxFZOAk5gSKwI7M-`.

This set enables an analysis of what happens in terms of the necessary time to unload and load a container ship due to an increment in the container ship capacity and the number of total ports.

In this sense, each subplot presented in Figure 14 shows what happens for a specific container ship size when the operation planning consider 5, 10 and 15 ports. All results considered that the transportation matrix is type 1 (mixed).

For example, the first subplot in Figure 14 was generated considering a container ship with a total capacity of $11,250$ TEUs ("Dimension $11,250$"). Horizontal axis is about number of ports, and vertical axis points the corresponding time to perform unloading and loading operations. The subplot shows an almost linear relation between an increase in number of ports and an increase in necessary time to perform operations. Although, for some container ships size there is not a linear increasing law. This happens, for example, with a ship with a total capacity of $13,500$, $16,875$, $18,000$, $19,125$, $20,250$, $21,375$, and $22,500$ TEUs.

Each subplot presented in Figure 15 shows what happens for a specific container ship size when the operation planning consider 5, 10 and 15 ports. All results considered that the transportation matrix is type 2 (long).

Some subplots in Figure 15 shows a different behavior when compared

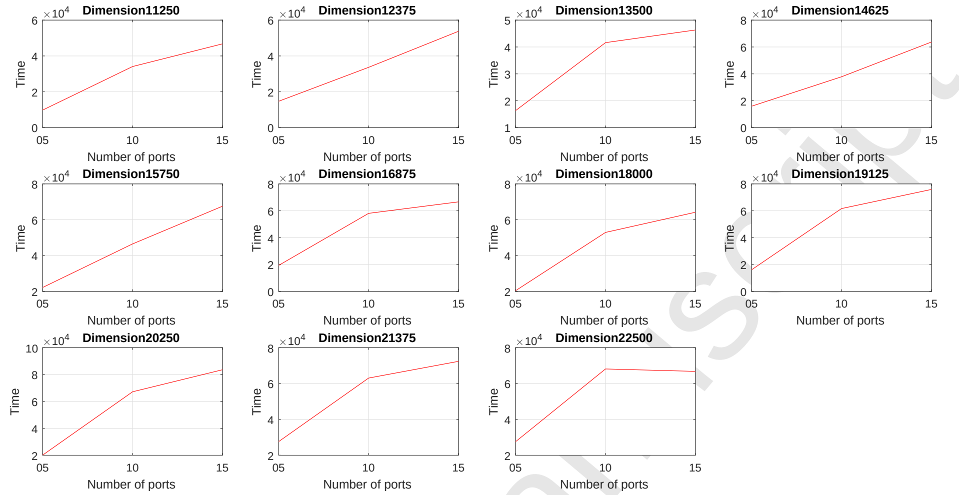


Fig. 14. Improvement of Container ship size and number of ports and its impact on necessary time to perform operations considering mixed transportation matrix.

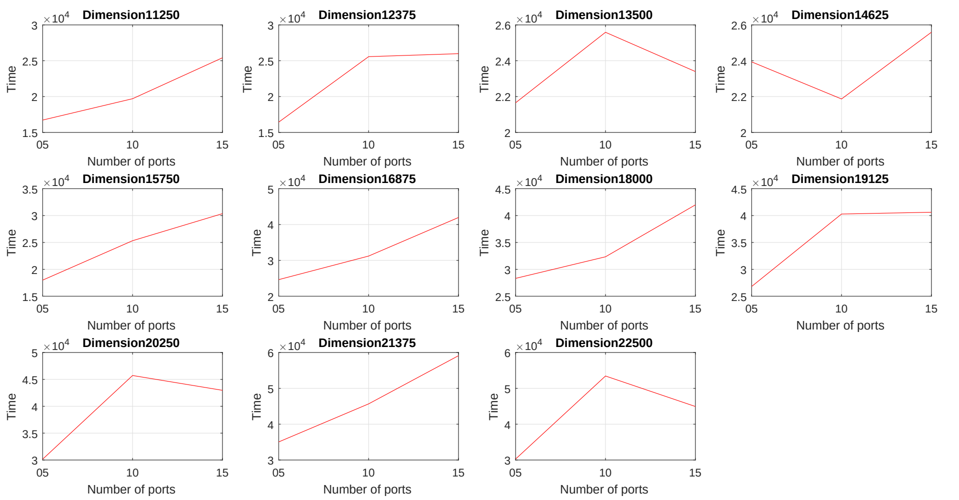


Fig. 15. Improvement of Container ship size and number of ports and its impact on necessary time to perform operations considering long transportation matrix.

to the ones presented in Figure 14. For example, in subplots corresponding to Container ships with dimension of 13, 500, 20, 250, and 22, 500 TEUs, the necessary time to perform unloading and loading operations for 15 ports is lower than for 10 ports. This is interesting feature since the improvement of Container ships size does not necessarily results in higher times to perform operations, and could results in more efficient operations. This partially explains the recent race for Container ships size increase, since the time to perform unloading and loading operations in a bigger ship that performs a longer route will not increase more than linear. In some cases a reduction in this time could even happen.

For the transportation matrix is type 3 (short), each subplot presented in Figure 16 shows what happens for a specific container ship size when the operation planning consider 5, 10 and 15 ports.

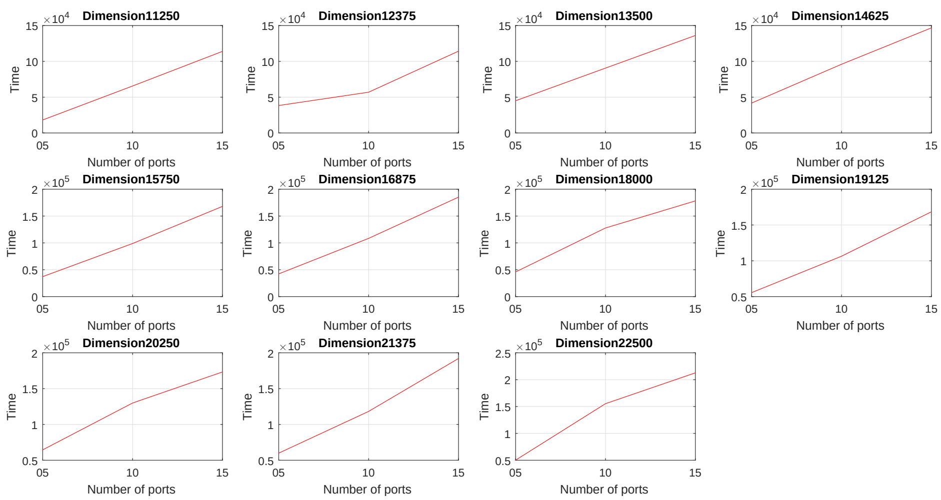


Fig. 16. Improvement of Container ship size and number of ports and its impact on necessary time to perform operations considering short transportation matrix.

The same pattern of linear increment in time to perform operation when number of ports are increased, that appeared in some subplots from Figure 14, appears in all subplots in Figure 15.

Finally, the complete data employed to create the Figures 14, 15, and 16 could be found at the Appendix K.

7. Conclusions

This paper showed a new approach to solving the 3D Stowage Planning Problem for Container ship (3D SP) integrated with the Scheduling of quay cranes (SQC) problem. It has four modeling advantages:

- It shows how the 3D SP will affect other problems related to port operation, like the SQC problem.
- It provides better accuracy in estimating total time, which can save the charterer charges for the the extra use of the vessel. In the instances studied, the solution from the integrated approach provided solutions with a 45.82% higher total time spent, on average, and prevented an underestimation of necessary time for ship travel.
- In 40% of the instances, the Integrated 3D SP and SQP problem helped to avoid a misleading analysis, where the adoption of good practices for 3D SP produced a worse total time to unload and load the ship.
- The developed approach showed that a linear increment in container ship size could produce a linear or lesser grow on time to perform unloading and loading operations. This partially explains the advantages on use bigger container ships and justifies a long term tendency of continuous increasing on container ship size.

The aforementioned advantages were attained without much computational effort. The reason for this is that the representation by rules encoding approach saves considerable computational time. For example, large-scale instances with a 30-port horizon were solved in less than 12 minutes, showing that this approach may eventually have practical commercial value. Another advantage is that the integrated decision process makes it possible to observe the detailed impact of different options of Stowage Planning in terms of the total time necessary to perform the total workload by quay cranes.

Different or more complex rules would be needed to ensure other stability features and to address specific unloading/loading constraints, or even quay crane special features at a particular port. Actually, this should not be a problem, since the proposed approach can deal with these practical situations by merely changing the set of rules.

8. Future Works

A promising idea for future work is to codify the previous approaches proposed for the 3D SP or interview crane operators to produce rules from crew experience and observe their impact on total time necessary to complete the total workload. Future work could also address this integrated problem with methods that deal with the Pareto-optimal frontier in order to choose the solutions.

Since the port yard has the same stack structure that appears on a container ship, one possible future work is to adapt the developed methodology to determine the yard cargo arrangement and operation of equipment like gantry cranes.

Another possibility of future work is to couple container ship loading and unloading operations with retrieving and storing cargo into port yard by determining a proper control on routing vehicles like AGVs or trucks.

Finally, a promising research branch is to consider different types of containers observing its size (20 feet, 40 feet, 40 feet extended), format (open, flat, and others) or cargo type (dry, explosive, reefer, and others), and its corresponding constraints of allocation.

9. Compliance with Ethical Standards

- Funding: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 2010/51274-5 and 2015/24295-5.

- Conflict of Interest: The authors declare that they have no conflict of interest.

- The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

References

References

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

[2] Y. Guan, K.-H. Yang, Z. Zhou, The crane scheduling: models and solution approaches, Annals of Operations Research 203 (1) (2013) 119–139.

[3] C. Bierwirth, F. Meisel, A survey of berth allocation and quay crane scheduling problems in container terminals, European Journal of Operational Research 202 (3) (2010) 615 – 627.

[4] D. Chang, Z. Jiang, W. Yan, J. He, Integrating berth allocation and quay crane assignments, Transportation Research Part E 46 (6) (2010) 975–990.

[5] A. Imai, H. Chen, E. Nishimura, S. Papadimitriou, The simultaneous berth and quay crane allocation problem, Transportation Research Part E 44 (5) (2008) 900–920.

[6] F. Meisel, C. Bierwirth, Heuristics for the integration of crane productivity in the berth allocation problem, Transportation Research Part E: Logistics and Transportation Review 45 (1) (2009) 196–209.

[7] C. Yang, X. Wang, Z. Li, An optimization approach for coupling problem of berth allocation problem and quay crane assignment in container terminal, Computers & Industrial Engineering 63 (1) (2012) 243–253.

[8] M. P. M. Hendriks, E. Lefeber, J. T. Udding, Simultaneous berth allocation and yard planning at tactical level, OR Spectrum 35 (2) (2013) 441–456.

[9] K. Braekers, A. Caris, G. Janssens, Integrated planning of loaded and empty container movements, OR Spectrum 35 (2) (2013) 457–478.

[10] C. Bierwirth, F. Meisel, A follow-up survey of berth allocation and quay crane scheduling problems in container terminals, European Journal of Operational Research 244 (3) (2015) 675–689.

[11] Q. Zeng, Z. Yang, Integrating simulation and optimization to schedule loading operations in container terminals, Computers & Operations Research 36 (6) (2009) 1935–1944.

[12] M. Avriel, M. Penn, S. Wittenboon, Stowage planning for container ships to reduce the number of shifts, Annals of Operations Research 76 (1) (1998) 55–71.

[13] I. Wilson, P. Roach, Container stowage planning: a methodology for generating computerised solutions, Journal of the Operational Research Society 51 (11) (2000) 1248–1255.

[14] A. Imai, K. Sasaki, E. Nishimura, 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 (2) (2006) 373–389.

[15] O. Dubrovsky, G. Levitin, M. Penn, A genetic algorithm with a compact solution encoding for the containership stowage problem, Journal of Heuristics 6 (2002) 585–599.

[16] A. Delgado, R. M. Jensen, K. Janstrup, T. H. Rose, K. H. Andersen, A constraint programming model for fast optimal stowage of container vessel bays, European Journal of Operational Research 220 (1) (2012) 251–261.

[17] A. Azevedo, C. Ribeiro, G. Sena, A. Chaves, L. Leduino, A. Moretti, Solving the 3d container ship loading planning problem by representation by rules and meta-heuristics, International Journal of Data Analysis Techniques and Strategies: Special Issue on Optimisation and Simulation in Realistic Scenarios 6 (3) (2014) 228–260.

[18] M. Monaco, M. Samarra, G. Sorrentino, The terminal-oriented ship stowage planning problem, European Journal of Operational Research 239 (1) (2014) 256–265.

[19] D. Ding, M. Chou, Stowage planning for container ships: A heuristic algorithm to reduce the number of shifts, European Journal of Operational Research 246 (1) (2015) 242 – 249.

[20] S. Amaran, N. Sahinidis, B. Sharda, S. Bury, Simulation optimization: a review of algorithms and applications, 4OR 12 (4) (2014) 301–333.

[21] G. Figueira, B. Almada-Lobo, Hybrid simulationoptimization methods: A taxonomy and discussion, Simulation Modelling Practice and Theory 46 (2014) 118–134, simulation-Optimization of Complex Systems: Methods and Applications.

[22] J. April, F. Glover, J. Kelly, M. Laguna, Simulation-based optimization: Practical introduction to simulation optimization, in: Proceedings of the 35th Conference on Winter Simulation: Driving Innovation, WSC '03, Winter Simulation Conference, 2003, pp. 71–78.

[23] M. Avriel, M. Penn, N. Shpirer, Container ship stowage problem: complexity and connection to the coloring of circle graphs, Discrete Applied Mathematics 103 (1-3) (2000) 271–279.

[24] 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 6049 (2010) 314–325.

[25] D. Pacino, R. Jensen, Constraint-based local search for container stowage slot planning, in: In: Proceedings of the International MultiConference of Engineers and Computers Scientists II, 2012, pp. 1–6.

[26] L. Fan, M. Low, H. Ying, H. Jing, Z. Min, W. Aye, Stowage planning of large containership with tradeoff between crane workload balance and ship stability, in: In: Proceedings of the International MultiConference of Engineers and Computers Scientists III, 2010, pp. 1–7.

[27] D. Ambrosino, A. Sciomachen, E. Tanfani, A decomposition heuristics for the container ship stowage problem, Journal of Heuristics 12 (3) (2006) 211–233.

[28] L. Cruz-Reyes, P. Hernández H., P. Melin, H. J. Fraire H., J. Mar O., Constructive algorithm for a benchmark in ship stowage planning, in: O. Castillo, P. Melin, J. Kacprzyk (Eds.), Recent Advances on Hybrid Intelligent Systems, Vol. 451 of Studies in Computational Intelligence, Springer Berlin Heidelberg, 2013, pp. 393–408.

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

[30] H. Javanshir, S. S. Ganji, Yard crane scheduling in port container terminals using genetic algorithm, Journal of Industrial Engineering International 6 (11) (2010) 39–50.

[31] D. Lee, D. Wang, L. Miao, Quay crane scheduling with non-interference constraints in port container terminals, Transportation Research E 44 (1) (2008) 124–135.

[32] P. Legato, R. Mazza, R. Trunfio, Simulation-based optimization for the quay crane scheduling problem, in: WSC '08 Proceedings of the 40th Conference on Winter Simulation, 2008, pp. 2717–2725.

[33] K. Mak, D. Sun, Scheduling yard cranes in a container terminal using a new genetic approach, Engineering Letters 17 (4) (2009) 274–280.

[34] K. Kim, Y.-M. Park, A crane scheduling method for port container terminals, European Journal of Operational Research 156 (3) (2004) 752 – 768.

[35] K. Dompere, Fuzziness and Foundations of Exact and Inexact Sciences, Vol. 290 of Series in Fuzziness and Soft Computing, Springer, 2012.

[36] C. Bierwirth, F. Meisel, A fast heuristic for quay crane scheduling with interference constraints, J. of Scheduling 12 (4) (2009) 345–360.

[37] J. Chen, D.-H. Leeb, M. Gohc, An effective mathematical formulation for the unidirectional cluster-based quay crane scheduling problem, European Journal of Operational Research 232 (1) (2014) 198–208.

[38] H. S. L. Services, Import demurrage: Freetime and tariffs (2016). URL http://www.hamburgsud-line.com/hsdg/en/hsdg/regionalinformation/southamerica/brazil/surcharges_7/demurrage_1/demurrage_1.jsp

[39] M. Caramia, P. Dell'Olmo, Multi-objective Management in Freight Logistics: Increasing Capacity, Service Level and Safety with Optimization Algorithms, Springer London, London, 2008, Ch. Multi-objective Optimization, pp. 11–36.

[40] E. J. Arajo, A. A. Chaves, L. L. de Salles Neto, A. T. de Azevedo, Pareto clustering search applied for 3d container ship loading plan problem, Expert Systems with Applications 44 (2016) 50–57.

[41] J.-P. Rodrigue, The geography of transport systems - the largest available containership (2017).
URL https://people.hofstra.edu/geotrans/eng/ch3en/conc3en/largestcontainerships.html

[42] E. Altena, Optimization of modal shift and container (re-)positioning at maersk line (2013).
URL http://repository.tudelft.nl/assets/uuid:66811c74-3c70-4b88-ab02-72f2fe86839d/Report_final_altena.pdf

[43] J.-P. Rodrigue, The geography of transport systems - containerships evolution (2017).
URL https://people.hofstra.edu/geotrans/eng/ch3en/conc3en/containerships.html

[44] C. from China, Shipping routes from china (2017).
URL https://cargofromchina.com/shipping-routes/

[45] C. Derrett, B. Barrass, Ship Stability for Masters and Mates, 5th Edition, Butterworth-Heinemann, 1999.
URL https://www.123library.org/ebook/isbn/9780080480886/

[46] A. Diabat, E. Theodorou, An integrated quay crane assignment and scheduling problem, Computers & Industrial Engineering 73 (2014) 115123.

[47] Y.-M. Fu, A. Diabat, I.-T. Tsai, A multi-vessel quay crane assignment and scheduling problem: Formulation and heuristic solution approach, Expert Systems with Applications 41 (15) (2014) 6959–6965.

[48] J. Bisschop, AIMMS - Optimization Modeling, AIMMS B.V., 2017.
URL https://download.aimms.com/aimms/download/manuals/AIMMS3_OM.pdf

Appendices

A. A Detailed Description of Loading and Unloading Rules Description

This appendix describes in detail the eight loading and two unloading rules called in the evaluation of Algorithm 1 in lines 15 and 8, respectively. The loading rules vary in terms of:

- Filling the ship by forming and positioning vertical stacks among bays or bay by bay;
- Forming flooring layers in the ship basement among bays or bay by bay.

The first six rules are detailed in [17], but Rules 7 and 8 are new and an original contribution of this work. The detailed description of the rules are:

- LR1: This rule fills the ship 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. Figure A-1 shows the application of the loading rule for the container ship at Port 1. The zero numbers indicate empty space. Positive numbers indicates spaces occupied by containers whose destination port is the corresponding positive number.
- LR2: This rule fills the ship 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.
- LR3: This rule is the reverse of LR1, which means the ship 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.
- LR4: This rule is the reverse of LR2 in the sense that it fills the ship 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.

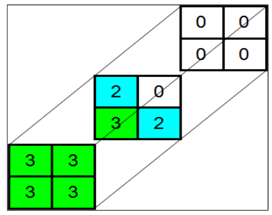


Fig. A-1. The container arrangement in ship after applying LR1.

- LR5: This rule fills the ship, 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 has been reached. The value θ_p is computed by Eq. (A-1).

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

  Then another bay is filled in a manner that the containers with the nearest destination are placed first to form the stacks.

- LR6: This rule is the reverse of LR5 in the sense that it fills the ship, 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 has been reached. The value θ_p is also computed by Eq. (A-1).

- LR7: All previous rules start from the ship bow. This rule fills the ship by alternating between filling from the middle to bow and from middle to stern. Additionally, it fills the ship from left to the right, first filling the lower part of each stack of each line with loads whose destination is farthest away.

- LR8: This rule is similar to LR7, but it begins it fills from middle to stern and from bow to middle.

There are only two unloading rules:

- UL1: Remove all containers to allow a complete rearrangement in containers and avoid future re-handles or;
- UL2: Remove only the containers that have currently arrived at the destination port or the containers that are blocking the former ones.

B. An example of the function related to Quay Crane Rule 1

This Appendix presents the m-code function that computes the total time necessary to perform two quay cranes operations in container ship bays.

```
function [ tmax ] = fqc1( vtbay )
% Inputs:
% (1) vtbay - a vector of total workload per bay.
%
% Parameters:
% (1) mdist - minimal distance between quay cranes
% (2) esp   - actual space between quay cranes
% (3) nbay  - number of bays (= length(vtbay))
% (4) nquay - number of quay cranes(qcs)-(suppose 2)
% (5) tf1   - total time for qc1 finish all services
% (6) tf2   - total time for qc2 finish all services
% (7) i1    - index for bay served with qc1
% (8) i2    - index for bay served with qc2
% (9) tmove - time to move a qc to another bay:
%           - considered 3 x (time to load) or
%           - (time to unload) one unit from ship
%
% Outputs:
% (1) tmax - total time need
%

% Initialization of values.
nbay  = length(vtbay); % number of ship bays.
tf1   = 0;   % initial total time of qc1 operation.
tf2   = 0;   % initial total time of qc2 operation.
i1    = 1;   % Starting qc 1 in the prow.
i2    = min(round(nbay/2)+1,nbay); %qc2 at middle|stern
    .
s1    = i2 - 1;
```

```
s2     = nbay;
esp    = i2 - i1 - 1; % space between qcs.
mdist  = 1;   % minimum distance between qcs.
tmove = 3;   % time to move qc from one bay to another.
% Initial position for both cranes.
tf1 = tf1 + vtbay(i1); % To create a time line for qc1.
tf2 = tf2 + vtbay(i2); % To create a time line for qc2.

% Three main cases:
% (1) Qcs last bay: no more dist. and time-line check.
% (2) With or without minimal distance.
% (3) Processing of qc1 operation (tf1 < tf2)
%     or qc 2 (tf2 < tf1) or both (tf1 == tf2).

% Loop to load or unload all the ship.
while ((i1 <= s1)||(i2 <= s2))

   % Case (3): qcs are not in the last operation bay.
   if ((i1 < s1)&&(i2 < s2))
     % Case (2.A): min distance ok, update qc positions.
     if (esp > mdist)
       % Case (3.A): qc1 operation (tf1 < tf2).
       if (tf1 < tf2)
         esp = esp - 1;
         i1 = i1 + 1;
         tf1 = tf1 + vtbay(i1) + tmove;
       % Case (3.B): qc2 operation (tf2 < tf1).
       elseif (tf2 < tf1)
         esp = esp + 1;
         i2 = i2 + 1;
         tf2 = tf2 + vtbay(i2) + tmove;
       % Case (3.C): qc1 & 2 operation (tf2 == tf1).
       else
         i1 = i1 + 1;
         tf1 = tf1 + vtbay(i1) + tmove;
         i2 = i2 + 1;
         tf2 = tf2 + vtbay(i2) + tmove;
```

```
        end

      % Case (2.B): minimal distance preserved or not.
      % Depends on what qc opr. (esp == mdist).
      elseif (esp == mdist)
        % Case (3.A): qc2 operation (tf2 < tf1).
        if (tf2 < tf1)
          esp = esp + 1;
          i2 = i2 + 1;
          tf2 = tf2 + vtbay(i2) + tmove;
        % Case (3.B): qc1 candidate for opr.(tf1 < tf2),
        % but it results in distance violation !!
        % Syncronization of qc1 & 2 operations.
        elseif (tf1 < tf2)
          tf1 = tf2; % synchronization of qcs movement!!
          i1 = i1 + 1;
          i2 = i2 + 1;
          tf1 = tf1 + vtbay(i1) + tmove;
          tf2 = tf2 + vtbay(i2) + tmove;
        % Case (3.C): qc1 & 2 opr. (tf2 == tf1).
        else
          i1 = i1 + 1;
          tf1 = tf1 + vtbay(i1) + tmove;
          i2 = i2 + 1;
          tf2 = tf2 + vtbay(i2) + tmove;
        end
      % Case (2.C): min. dist. not ok. Move only qc2.
      else
        tf1 = tf2; % synchronization of qcs movement!!
      end % end of if for minimal distance checking.

    % Case (1):At least one qc operates in its last bay.
    % Space between qcs is no longer a problem, then it
    % is not updated anymore.
    else
      % Case (1.A): qc1 on the last bay - move only qc2.
      if ((i1 >= s1)&&(i2 < s2))
        i2 = i2 + 1;
```

```
          tf2 = tf2 + vtbay(i2) + tmove;
        end
        % Case (1.B): qc2 on the last bay: move only qc1.
        if ((i1 < s1)&&(i2 >= s2))
          i1 = i1 + 1;
          tf1 = tf1 + vtbay(i1) + tmove;
        end
        % Case (1.C): qc1 & 2 are on last bay: both move.
        if ((i1 >= s1)&&(i2 >= s2))
          i1 = i1 + 1;
          i2 = i2 + 1;
        end
      end    % end of if for last bay operation check.

end    % end of for the loop of using all qc movement.

% Total processing time is the max. time of processing
% all tasks designated for qc1 & 2.
tmax = max(tf1, tf2);
```

The previous code could be generalized for more than two quay canes by observing the following conditions:

1. Suppose there are two quay cranes: qc_1 and qc_2. The position of each quay crane i is tracked by a corresponding variable tf_i.
2. The lowest value on index i indicates which is the leftmost quay crane, i.e., qc_1.
3. Suppose the two cranes qc_1 and qc_2 are moving to the right, because, according to [37], when quay cranes move in the same direction, the optimal solution can still be found.
4. The distance between cranes should be at least equal or higher than a minimal distance $mdist$.
5. A quay crane cannot overpass another quay crane.
6. From these suppositions, the leftmost quay crane's, qc_1, move depends on the rightmost quay crane's position, qc_2. This may be expressed in the following condition: $tf_1 + mdist \leq tf_2$.

7. This results in the following: qc_2 is free to move, but the position of qc_1 is determined by the position of qc_2.

The previous conditions lead to the fact that a quay crane rule may be applied to N quay cranes by applying the following recursive pairwise computation:

1. Check distance between the rightmost quay crane and its immediate predecessor, i.e, the positions of qc_N and qc_{N-1}.

2. Once the position of qc_{N-1} has been computed, the same previous check can be carried out for qc_{N-i} and $qc_{N-(i+1)}$, $i = 1$.

3. This sequence can be applied recursively, i.e., $\forall i = 2, \cdots, N-2$.

C. Generating Instances

As suggested by [12], it is possible to create three types of transportation matrix: 1 - Mixed, 2 - Long and 3 - Short. This classification describes how long the majority of containers must be carried by the container ship for a specific transportation matrix (instance). 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 for most of the containers will remain on board the container ship. The Mixed is created in a manner that mixes Short and Long instance characteristics. The generation of all instances follows the procedure described in [12], which ensure ship will carry all containers available in ports, but with the modification proposed in [17] has been added, however, to consider the available space of 3D container ship:

$$\sum_{i=1}^{k} \sum_{j=k+1}^{N} T_{ij} \leq R \times C \times D, \forall i = 1, \cdots, N-1, \tag{C-1}$$

where T_{ij} is an element in line i and column j from the transportation matrix T; N is the number of ports; R is the number of the rows; C is the number of columns; and D is the number of bays.

For mixed matrices the distance matrix T_{ij} is random such that Eq. (C-1) is satisfied and, additionally, each distance matrix element T_{ij} is subject

	$D1$	$D2$	$D3$	$D4$
$O1$	0	2	1	2
$O2$	0	0	2	0
$O3$	0	0	0	0
$O4$	0	0	0	0

(a) Mixed distance.

	$D1$	$D2$	$D3$	$D4$
$O1$	0	0	0	8
$O2$	0	0	0	1
$O3$	0	0	0	0
$O4$	0	0	0	0

(b) Long distance.

	$D1$	$D2$	$D3$	$D4$
$O1$	0	3	0	0
$O2$	0	0	9	0
$O3$	0	0	0	0
$O4$	0	0	0	0

(c) Short distance.

Fig. C-1. Small examples of three types of distance matrix.

to $T_{ij} \leq 0.2 \times R \times C \times D$. This constraint prevents the number of matrix elements equal to zero from being large.

For long distance matrices, the constraint $T_{ij} \leq 0.2 \times R \times C \times D$ is replaced by $T_{ij} \leq ((j-i)/(N-1))^2$ to prevent near-diagonal elements from having large values.

Finally, the short distance matrices consider only Eq. (C-1) and a special order to fill the distance matrix in a manner such that near-diagonal matrices become somewhat large.

Examples of the three matrix types are illustrated in Fig. C-1 using the same parameters of Fig. 3: $D = 1$, $R = 3$, $C = 3$, and $N = 4$.

D. Stowage Planning Mathematical Model

The mathematical model in terms of linear programming with binary variables for the 3D SP is given by (D-1)-(D-8).

$$\min f\left(x_{ijv}(r,c,d), y_i(r,c,d)\right) = \alpha\phi_1\left(x_{ijv}(r,c,d)\right) + \beta\phi_2\left(y_i(r,c,d)\right) \quad \text{(D-1)}$$

subject to

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

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

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

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

$$x_{ijv}(r,c,d) = 0 \;\; or \;\; 1$$
$$i,j,v = 1,\cdots,N; r = 1,\cdots,R; c = 1,\cdots,C; d = 1,\cdots,D. \quad \text{(D-6)}$$

where: the binary variable $x_{ijv}(r,c,d)$ is defined as follows: if, 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 $y_i(r,c,d)$ is defined as follows: if, in port i, the compartment (r,c,d) has a container; then the variable assumes value 1; otherwise value 0 is assumed. The objective function (D-1) is composed of two terms: the first is the total cost of moving a container and, the second is the sum of instability measures for the container ship configuration in each port. It is assumed that, for all ports, the container movement costs the same and is equal to one. Constraints (D-2) express the total number of containers that will be shipped from port i to port j. Constraints (D-3) require that each compartment (r,c,d) of the container ship is always occupied by at most one container. Constraints (D-4) are related to the physical storage of the containers in the ship, and it imposes that, for each container in row $r+1$, there be another container in the row r for all $r = 1,\cdots,R-1$. Constraints (D-5) define how 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. Finally, Constraints (D-6) defines the variables nature.

The two terms which compose the objective function (Eq. (D-1)) define two optimization criteria: the first term is a function of the number of containers moved, $\phi_1(x)$, and the second depends on how the container ship is organized in each port, $\phi_2(y)$. The two criteria are combined by values given by weights α and β in a manner that forms a bi-objective optimization framework.

The term $\phi_1(x_{ijv}(r,c,d))$ assumes that for all ports, the container movement cost is the same and is equal to one which may be translated as Eq. (D-7).

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

The term $\phi_2(y_i(r,c,d))$ refers to the container ship's transverse stability and assumes that every container has the same mass and is equal to one. This term is to control the container ship transverse stability before leave the port which means after all loading movements had been performed as described by Eq. (D-8).

$$\phi_2(y_i(r,c,d)) = \sum_{i=1}^{N}(-\Delta GM_i + \Delta L_i) \tag{D-8}$$

where:

$$\Delta GM_i = \left(\sum_{r=1}^{R}\left(\sum_{d=1}^{D}\sum_{c=1}^{C} y_i(r,c,d)\right)\cdot(GY_{ship} - r + 0.5)\right),$$

$$\Delta L_i = hp_i + hn_i,$$

$$\left(\sum_{c=1}^{C}\left(\sum_{d=1}^{D}\sum_{r=1}^{R} y_i(r,c,d)\cdot(GX_{ship} - c + 0.5)\right)\right) = hp_i - hn_i,$$

$$hp_i, hn_i \geq 0$$

where: the values GY_{ship} and GX_{ship} represent vertical and horizontal coordinates of gravity center of the ship, respectively. The variables ΔGM_i represent the variance in metacentric height GM in each port i after loading all containers. Since GM is the distance between the centre of gravity of a ship and its metacentre, as much metacentric height is increased with a $\Delta GMi > 0$, it turns more difficult the ship to overturn. The variables ΔLi also helps with the reduction of angle of list after loading all containers. Angle list measures the vessel leaning to either port or starboard. More discussion about metacentric height increasing and reduction of angle of list and corresponding objective functions could be seen at [14, 45].

One model feature is that the number of containers that must be loaded at a certain port is given by a transportation matrix T of dimension $(N-1) \times (N-1)$, 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 $i \geq j$.

Another feature is that the container ship has a rectangular format and can be represented by a matrix with rows $(r = 1, 2, \cdots, R)$, columns $(c = 1, 2, \cdots, C)$ and bays $(d = 1, 2, \cdots, D)$ with maximum capacity of $R \times C \times D$ containers. Irregular formats could be achieved by simply adding constraints which represents imaginary containers that occupies same spaces during the whole voyage [19].

Finally, this model presents an original contribution to the literature since it precisely computes the number of movements, instead of doing an estimation as done in [14], and the computation of stability issues is done by using an objective function linear, instead of using non-linear one as done in [17].

E. Quay Crane Scheduling Problem

The mathematical model in terms of linear programming with binary variables for the Scheduling Quay Cranes (SQC) had been developed to avoid indexing in time that could be advantageous as observed by [46, 47] and used by [36, 37]. The model is given by (E-1)-(E-8).

$$\min \tau \tag{E-1}$$

subject to

$$\sum_{k=1}^{K} w_{j,k} = 1, \qquad j \in \Omega_B \tag{E-2}$$
$$t_{1,k} \geq a_1 w_{1,k}, \qquad k \in \Omega_K, j \in \Omega_B \tag{E-3}$$
$$t_{j+1,k} \geq t_{j,k} + a_{j+1} w_{j+1,k}, \qquad k \in \Omega_K, j \in \Omega_{B-1} \tag{E-4}$$
$$w_{j+1,k} + w_{j,k+1} = z_{j,k}, \quad j \in \Omega_{B-1}, k \in \Omega_{K-1} \tag{E-5}$$
$$t_{j+1,k} \geq t_{j,k+1} + a_{j+1} \left(z_{j,k} - 1 \right), \qquad j \in \Omega_B, k \in \Omega_{K-1} \tag{E-6}$$
$$\tau \geq t_{j,k}, \qquad j \in \Omega_B, k \in \Omega_K \tag{E-7}$$
$$z_{j,k} \geq 0, \; w_{j,k} = 0 \; or \; 1,$$
$$a_d = f(x_{ijk}(r,c,d), \; y_i(r,c,d)), \qquad j \in \Omega_B, k \in \Omega_K. \tag{E-8}$$

From the given model the following decisions variables had been considered:

- The binary variable $w_{j,k}$ assumes value 1, if the crane k is allocated to bay or bay group j;
- The integer auxiliary variable $z_{j,k}$ helps to prevent quay cranes overpass each other;
- The real variable $t_{j,k}$ represents the total time necessary to perform operations for all k cranes until bay or group of bays j;

The objective function is the minimization of the time necessary to finish the necessary services on the container ship, and this could be unloading or loading movements. This is represented by the minimization of the finish time of the last work unit, i.e., the makespan. Constraint (E-2) determines that each group of bays j should be attend by only one crane k. Constraints (E-3) and (E-4) helps in the computation of total time necessary to perform operations till bay j. Constraint (E-5) defines auxiliary variable that helps to identify possible points where quay cranes could overpass each other. Constraint (E-6) compute total processing time by avoiding quay crane overpass. Constraints (E-7) gives the makespan. Finally, Constraints (E-8) defines the variables nature. For instance, once a crane starts to work on a section, it has to finish the workload in this section before it moves to other sections.

F. Integrated Mathematical Model for 3D SP and SQC

The SQC problem defined by Eq. (E-1)-(E-8) could be coupled with the 3D SP Eq. (D-1)-(D-8), but by solving the SQC problem two times at every port i: once after ship perform unloading and after ship loading containers. Also, indexation should be adjusted to be compatible with the one used in 3D SP which means index i is for port and s is kind of operation performed (1 for unloading and 2 for loading). It was also assumed, for simplicity, that every group of bays b corresponds exactly to one bay d. Finally, some additional constraints connect the decisions made in 3D SP and correspondent unloading and loading effort through ports. These assumptions lead to the model given by Eq. (F-1)-(F-10).

$$\min \sum_{i=1}^{N} \sum_{s=1}^{S} \tau_{i,s} \tag{F-1}$$

subject to

$$\sum_{k=1}^{K} w_{i,s,j,k} = 1,$$
$$i = 1, \cdots, N, s = 1, \cdots, S, j = 1, \cdots, D \qquad \text{(F-2)}$$
$$t_{i,s,1,k} \geq a_{i,s,1} w_{i,s,1,k},$$
$$i = 1, \cdots, N, s = 1, \cdots, S, k = 1, \cdots, K, j = 1, \cdots, D \qquad \text{(F-3)}$$
$$t_{i,s,j+1,k} \geq t_{i,s,j,k} + a_{i,s,j+1} w_{i,s,j+1,k},$$
$$i = 1, \cdots, N, s = 1, \cdots, S, k = 1, \cdots, K, j = 1, \cdots, D-1 \qquad \text{(F-4)}$$
$$w_{i,s,j+1,k} + w_{i,s,j,k+1} = z_{i,s,j,k},$$
$$i = 1, \cdots, N, s = 1, \cdots, S, j = 1, \cdots, D-1, k = 1, \cdots, K-1 \qquad \text{(F-5)}$$
$$t_{i,s,j+1,k} \geq t_{i,s,j,k+1} + a_{i,s,j+1} \left(z_{i,s,j,k} - 1 \right),$$
$$i = 1, \cdots, N, s = 1, \cdots, S, j = 1, \cdots, D, k = 1, \cdots, K-1 \qquad \text{(F-6)}$$
$$\tau_{i,s} \geq t_{i,s,j,k},$$
$$i = 1, \cdots, N, s = 1, \cdots, 2, j = 1, \cdots, D, k = 1, \cdots, K \qquad \text{(F-7)}$$
$$z_{i,s,j,k} \geq 0, w_{j,k} = 0 \ \ or \ \ 1. \qquad \text{(F-8)}$$
$$\sum_{j=1}^{N} \sum_{k=1}^{i-1} \sum_{r=1}^{R} \sum_{c=1}^{C} \psi_{i,1}(r,c,d) x_{kji}(r,c,d) = a_{i,1,d}$$
$$i = 1, \cdots, N, d = 1, \cdots, D \qquad \text{(F-9)}$$
$$\sum_{j=i+1}^{N} \sum_{v=j}^{N} \sum_{r=1}^{R} \sum_{c=1}^{C} \psi_{i,2}(r,c,d) x_{ijv}(r,c,d) = a_{i,2,d}$$
$$i = 1, \cdots, N, d = 1, \cdots, D \qquad \text{(F-10)}$$

The constraints (F-2)-(F-8) are just the SQC model adapted to give the minimal time of loading and unloading to 3D SP through feasible quay cranes scheduling. Constraints (F-9)-(F-10) couple the decision of how to perform container ship arrangement ($x_{i,j,v}(r,c,d)$) through ports with the total workload per bay employed in unloading ($a_{i,1,d}$) and loading cranes scheduling service ($a_{i,2,d}$). Finally, Eq. (F-1) should replace the term $\phi_1(x)$ in objective function (Eq. (D-7)) since the term is just an estimation of how much time will be spent by ship through ports.

Although, this integrated mathematical model has constraints (F-3), (F-4) and (F-6) as nonlinear ones. Constraints (F-3) and (F-4) is a product of $a_{i,s,d}$, which is a sum of $x_{kji}(r,c,d)$ variables multiplied by a real number $\psi_{i,1}(r,c,d)$ and $w_{i,s,j+1,k}$ binary variables. To avoid such problem, it is necessary to apply some integer programming techniques, in particular, the elimination of products of variables as described in [48]. Let $w_{i,s,j+1,k}$ be a binary variable, and $a_{i,s,j}$ be a continuous variable for which $0 \leq a_{i,s,j} \leq a_{i,s,j}^{max}$ holds. Now, a new continuous variable, $q_{i,s,j,k}$, is introduced to replace the product $q_{i,s,j+1,k} = a_{i,s,j} w_{i,s,j+1,k}$. The following constraints must be added to force $q_{i,s,j+1,k}$ to take value of $a_{i,s,j} w_{i,s,j+1,k}$.

$$q_{i,s,j+1,k} \leq a_{i,s,j}^{max} w_{i,s,j+1,k} \tag{F-11}$$

$$q_{i,s,j+1,k} \leq a_{i,s,j} \tag{F-12}$$

$$q_{i,s,j+1,k} \geq a_{i,s,j} - a_{i,s,j}^{max}(1 - w_{i,s,j+1,k}) \tag{F-13}$$

$$q_{i,s,j+1,k} \geq 0 \tag{F-14}$$

Additionally, constraints (F-3) and (F-4) should be replaced by (F-15) and (F-16).

$$t_{i,s,1,k} \geq q_{i,s,1,k}, \quad i = 1, \cdots, N, s = 1, \cdots, 2, k = 1, \cdots, K, j = 1, \cdots, D \tag{F-15}$$

$$t_{i,s,j+1,k} \geq t_{i,s,j,k} + q_{i,s,j+1,k}, \quad i = 1, \cdots, N, s = 1, \cdots, 2, k = 1, \cdots, K, j = 1, \cdots, D-1 \tag{F-16}$$

The same technique is applied to (F-6) constraint by defining a new continuous variable, $u_{i,s,j,k}$ to replace the product $u_{i,s,j,k} = a_{i,s,j} z_{i,s,j,k}$.

$$u_{i,s,j,k} \leq a_{i,s,j}^{max} z_{i,s,j,k} \tag{F-17}$$

$$u_{i,s,j,k} \leq a_{i,s,j} \tag{F-18}$$

$$u_{i,s,j,k} \geq a_{i,s,j} - a_{i,s,j}^{max}(1 - z_{i,s,j,k}) \tag{F-19}$$

$$u_{i,s,j,k} \geq 0 \tag{F-20}$$

Finally, constraint (F-6) is replaced by (F-21).

$$t_{i,s,j+1,k} \geq t_{i,s,j,k+1} + u_{i,s,j,k} - a_{i,s,j+1}, \quad i = 1, \cdots, N, s = 1, \cdots, 2, k = 1, \cdots, K, j = 1, \cdots, D-1 \tag{F-21}$$

The mathematical model for 3D SP integrated with SQC is formed by Eqs. (D-1)-(D-8), (F-1),(F-2), (F-5),(F-7)- (F-10), and (F-11)-(F-21).

Table G-1. Transportation information for a five port example.

	P2	P3	P4	P5
P1	2	5	0	0
P2	0	2	3	1
P3	0	0	2	2
P4	0	0	0	1

G. Small numerical example for 3D SP Model

The mathematical model has been applied in a small example just to illustrate how container ship arrangement could be affected by stability measures and the proposed model had been successful to provide more stable ship arrangements through ports.

The numerical example consists on a container ship with dimensions $R = 4$, $C = 4$ and $D = 1$. The number of ports is $N = 5$, and in each port there are $K = 2$ quay cranes available for unloading and loading operations. It was also considered, for simplicity, each column as bay. Each element $T_{i,j}$ from transportation matrix gives the number of containers that should be loaded in port i which destination is port j. The transportation matrix used in this example is shown in Table G-1.

The mathematical model had been solved using GUSEK (the GLPK Integer Optimizer v4.55 with all cuts enabled) and the corresponding model has 176 constraints, and 400 binary variables. The integer solution had been found in 0.4 seconds in a laptop with Intel Core i7-4500U CPU and 1.80 GHz of RAM.

When the model is set to minimize only the total number of container movements ($\alpha = 1$ and $\beta = 0$), the resulting container ship arrangement is given in Figure G-1. A different container ship arrangement is produced for only stability measures maximization ($\alpha = 0$ and $\beta = 1$) as shown in Figure G-2. Each square in both figures has a pair of numbers (i, j) representing a space of the ship occupied by a container, or 0 representing an empty space. The first number of the pair gives the container loading port information and the second gives the container unloading port information. The number of movements per port for both solutions is presented in Table G-2 which leads to a total cost of US$7200 (since each movement costs US$200). The article [17] provides a detailed procedure on how to compute stability measure.

Figure G-2 shows a more stable container ship arrangement specially in ports 2 and 3 when compared with the arrangement in Figure G-1, but

Table G-2. Number of unloading and loading movements per port.

Port	1	2	3	4	5	Total
loading	7	6	4	1	0	18
unloading	0	2	7	5	4	18

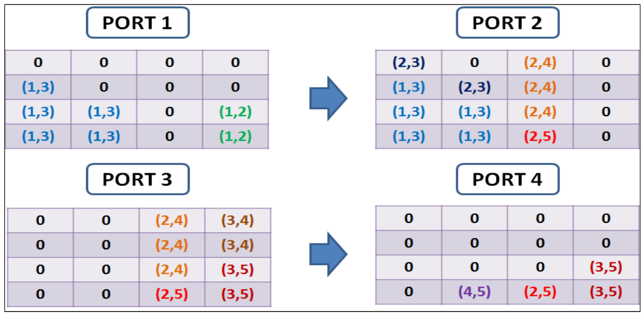


Fig. G-1. The container ship arrangement for minimization of $\phi_1(\cdot)$.

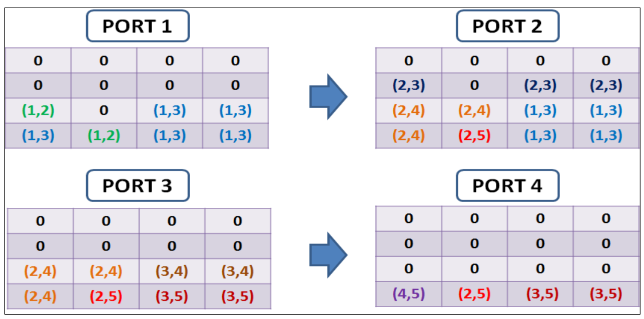


Fig. G-2. The container ship arrangement for minimization of $\phi_2(\cdot)$.

without an additional number of container movements.

H. Small numerical example for SQC Model

The mathematical model (as given on Appendix E) has been applied in a small example just to illustrate how quay crane scheduling could be affected by the constraint of one crane could not cross another.

The numerical example consists on a container ship with number of bays equals to $B = 9$. The number of quay cranes available to perform operations

Table H-1. Time to process for a nine bay container ship.

Bay	1	2	3	4	5	6	7	8	9
Time	22	46	8	70	10	38	40	16	22

Table H-2. Scheduling of quay cranes through bays of the container ship.

Bay	1	2	3	70	5	6	7	8	9
QC2	0	0	0	70	80	0	120	0	132
QC1	22	68	76	0	0	118	0	136	0

is $K = 2$. It was also considered, for simplicity, no time is spent on moving quay cranes and minimal distance between them could be zero. Each element on vector represents a bay of the container ship and gives the total workload in terms of amount of time necessary to process a task (unloading or loading). The vector used in this example is shown in Table H-1 and comes from the article [6].

The mathematical model had been solved using GUSEK (the GLPK Integer Optimizer v4.55 with all cuts enabled) and the corresponding model has 62 constraints, and 18 binary variables. The integer solution had been found in not significant time in a laptop with Intel Core i7-4500U CPU and 1.80 GHz of RAM.

The optimal scheduling found is presented in Table H-2 which leads to a makespan of 136 units of time. Each element corresponds to the instant a task on a bay had been finished for a corresponding quay crane. Figure H-1 shows the corresponding quay crane movement and the constraint of quay crane 1 could overpass quay crane 2 is fulfilled.

Furthermore, Figure H-1 shows two moments in scheduling that the overpass constraints leads to an increasing in the quay crane 1 processing time. The first is when quay crane 1 finished process on bay 3 at the time 76, but have to wait quay crane 2 finish its work on bay 5 at time 80. As a consequence only after time 80, quay crane 1 and 2 could move to bay 6 and 7, respectively. The processing on these two bays leads to second moment and quay crane 1 had to wait two additional units of time to move to bay 8.

The allocation of quay cranes found with the proposed model is the same found by [6] with only a difference in makespan caused by considering zero time to perform a quay crane movement. Although, the proposed model on Appendix F is more compact.

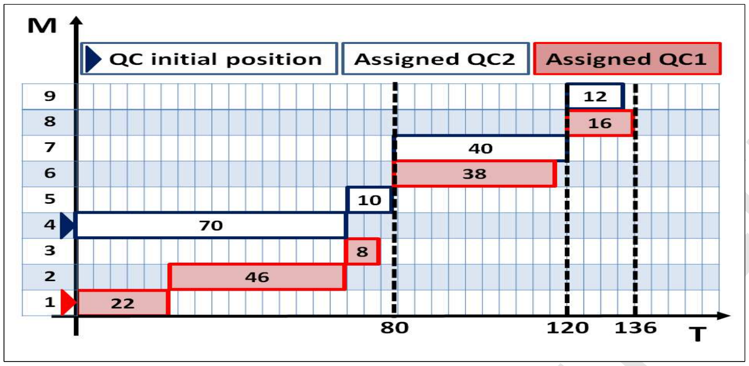


Fig. H-1. Optimal QC schedule with makespan equals to 136.

Table J-1. GA Parameters and Values.

Parameter	Value
Population Size	10
Number of generations	1000
Crossover probability	0.8
Mutation probability	0.15

I. Small numerical example for Integrated 3D SP and SQC Model

Finally, tests had been carried for Integrated Mathematical Model for 3D SP and SQC which means the inclusion of two quay cranes to perform unloading and loading operations in each port. The data is the same given on D.

The Integrated model (as given on Appendix F) with 796 rows and 1290 columns (960 binary variables) demanded more than 11 hours without returning an integer optimal solution (the process stopped with a GAP of 23.9%).

J. Genetic Algorithm Parameters and Corresponding Values

The adopted parameters for Genetic Algorithm and its corresponding values are given in Table J-1.

The crossover employed is the crossover OX. The mutation is simply the random selection of a vector element and changes it to a random integer number in the interval $[1, N]$, where N is the number of ports.

Table K-1. Results obtained for GA when minimizing only total time.

I	S	N	M	FO1	T(s)
1	11,250	05	1	9,792	273.23
2			2	16,724	353.36
3			3	18,147	266.66
4		10	1	34,090	776.52
5			2	19,703	910.71
6			3	65,577	915.01
7		15	1	46,741	1,471.31
8			2	25,413	900.49
9			3	113,948	1,276.74

Finally, tests had been carried for Integrated Mathematical Model for 3D SP and SQC which means the inclusion of two quay cranes to perform unloading and loading operations in each port. The data is the same given on Appendix D.

K. Detailed results from the new data set

In Table K-1, the column index I corresponds to instance number; S corresponds to container ship total capacity; N corresponds to how many ports the container ship has to pass through; the column index M refers to the type of transportation matrix (1 - Mixed, 2 - Long, 3- Short; there are more details on Appendix C); $FO1$ is the total time to perform all crane movements (see subsection 2.2 for constraints and parameter details); $T(s)$ is the computational time spent in seconds to obtain the solution.

The results presented from Table K-1 to Table K-11 were obtained with a program created in a Matlab 7.0, a machine with a 1.66 GHz Core Duo Intel Processor, RAM memory of 2 GB, and Windows Vista Operational System with Service Pack 2. The genetic algorithm (more details about parameters on Appendix J) was executed 5 times to illustrate the potential of the developed approach.

The crossover employed is the crossover OX. The mutation is simply the random selection of a vector element and changes it to a random integer number in the interval $[1, N]$, where N is the number of ports.

Finally, tests had been carried for Integrated Mathematical Model for 3D SP and SQC which means the inclusion of two quay cranes to perform unloading and loading operations in each port.

Table K-2. Results obtained for GA when minimizing only total time.

I	S	N	M	FO1	T(s)
10	12,375	05	1	14,724	287.88
11			2	16,440	340.59
12			3	38,337	448.42
13		10	1	33,646	783.76
14			2	25,571	842.21
15			3	56,915	848.20
16		15	1	53,757	1,151.08
17			2	25,985	983.77
18			3	114,232	1,355.01

Table K-3. Results obtained for GA when minimizing only total time.

I	S	N	M	FO1	T(s)
19	13500	05	1	16,333	305.02
20			2	21,655	399.52
21			3	45,001	433.35
22		10	1	41,613	795.61
23			2	25,588	776.74
24			3	90,586	1,002.22
25		15	1	46,318	1,179.21
26			2	23,397	985.71
27			3	136,163	1,651.72

Table K-4. Results obtained for GA when minimizing only total time.

I	S	N	M	FO1	T(s)
28	14625	05	1	15,943	334.15
29			2	23,936	405.13
30			3	41,769	493.24
31		10	1	37,831	865.11
32			2	21,867	767.56
33			3	95,953	1,114.09
34		15	1	63,692	1,309.32
35			2	25,591	1,170.09
36			3	146,765	1,812.58

Table K-5. Results obtained for GA when minimizing only total time.

I	S	N	M	FO1	T(s)
37	15750	05	1	22,214	367.77
38			2	17,995	314.25
39			3	37,206	437.42
40		10	1	46,509	3,591.25
41			2	25,325	819.16
42			3	98,872	1,218.65
43		15	1	67,543	1,438.74
44			2	30,361	1,500.57
45			3	167,953	1,881.28

Table K-6. Results obtained for GA when minimizing only total time.

I	S	N	M	FO1	T(s)
46	16875	05	1	19,389	440.81
47			2	24,629	432.70
48			3	42,310	615.95
49		10	1	58,090	1,108.71
50			2	31,206	870.19
51			3	108,407	1,247.28
52		15	1	66,632	1,794.58
53			2	41,967	1,254.07
54			3	185,187	2,137.30

Table K-7. Results obtained for GA when minimizing only total time.

I	S	N	M	FO1	T(s)
55	18000	05	1	20,417	390.65
56			2	28,319	501.97
57			3	45,898	2,509.26
58		10	1	52,922	1,153.40
59			2	32,334	907.81
60			3	127,759	1,495.79
61		15	1	64,176	1,495.48
62			2	41,993	2,049.09
63			3	178,191	2,729.01

Table K-8. Results obtained for GA when minimizing only total time.

I	S	N	M	FO1	T(s)
64	19125	05	1	16,130	359.14
65			2	26,812	370.56
66			3	55,575	646.24
67		10	1	61,650	1,118.45
68			2	40,288	402.46
69			3	106,395	1,324.42
70		15	1	75,876	1,668.47
71			2	40,621	1,450.72
72			3	168,456	1,958.39

Table K-9. Results obtained for GA when minimizing only total time.

I	S	N	M	FO1	T(s)
73	20250	05	1	20,101	153.20
74			2	30,161	557.02
75			3	64,474	269.33
76		10	1	67,197	1,302.42
77			2	45,722	1,045.01
78			3	129,923	1,586.77
79		15	1	83,589	1,742.81
80			2	42,966	1,823.37
81			3	173,287	2,353.86

Table K-10. Results obtained for GA when minimizing only total time.

I	S	N	M	FO1	T(s)
82	21375	05	1	27,632	508.25
83			2	35,050	642.47
84			3	59,906	624.90
85		10	1	63,059	1,573.77
86			2	45,667	1,167.78
87			3	118,192	1,661.03
88		15	1	72,384	2,096.61
89			2	59,110	1,808.01
90			3	192,097	2,774.22

Table K-11. Results obtained for GA when minimizing only total time.

I	S	N	M	FO1	T(s)
91	22500	05	1	27,564	487.22
92			2	30,278	677.47
93			3	50,194	860.48
94		10	1	68,137	1,456.33
95			2	53,442	1,271.27
96			3	155,321	1,787.77
97		15	1	66,788	2,002.10
98			2	44,935	1,971.58
99			3	212,738	2,701.42