

A MATHEMATICAL MODEL AND SOLUTION METHOD FOR THE BERTH ALLOCATION PROBLEM WITH VARIABLE HANDLING TIME AND CONTINUOUS TIME HORIZON

Bruno Luís Hönigmann Cereser^{1*}, Aurelio Ribeiro Leite de Oliveira² and
Antonio Carlos Moretti³

Received March 4, 2022 / Accepted June 29, 2022

ABSTRACT. In this paper, we present the integration of two problems related to the operations in a port terminal: the Berth Allocation Problem (BAP) integrated with the Machine Assignment Problem. We present a mixed-integer linear programming (MILP) formulation, capable of assigning and scheduling incoming vessels to berthing positions and the assignment of machines for handling the vessels. The machines can be quay cranes, mobile cranes, straddle carriers, forklifts, trucks, and any other machine. The problem aims to minimize the waiting time plus the handling time of the vessels. To solve the problem, we developed a heuristic algorithm, capable of solving a problem instance in seconds. To compare the results, we generate several instance problems based on real data and solve them with our MILP formulation implemented in a solver, our heuristic, and a First In First Out (FIFO) algorithm. The solver was able to find solutions only in small-scale instances, and the heuristic was able to find good solutions for all instances.

Keywords: Berth allocation problem, quay crane allocation, allocation of port machines, optimization of port processes.

1 INTRODUCTION

Maritime shipping is the main transportation modal used in worldwide trade and, reducing the time and cost of such transportation continues to be an important goal. In order to reduce transportation costs, terminal managers seek to increase efficiency, with better management and increasing infrastructure and technologies able to efficiently handle vessels. A single port can han-

*Corresponding author

¹IMECC/UNICAMP, Cidade Universitária, 13083-856 Campinas, SP, Brasil – E-mail: brunolhcreser@gmail.com – <https://orcid.org/0000-0003-0753-9386>

²IMECC/UNICAMP, Cidade Universitária, 13083-856 Campinas, SP, Brasil – E-mail: aurelio@ime.unicamp.br – <https://orcid.org/0000-0002-6471-4710>

³IMECC/UNICAMP, Cidade Universitária, 13083-856 Campinas, SP, Brasil – E-mail: moretti@ime.unicamp.br – <https://orcid.org/0000-0001-5604-1002>

dle several types of goods, such as container cargo, liquid bulk, dry bulk, Breakbulk, and ro-ro, which increase the need for efficient management of logistic activities at modern terminals, therefore, several papers addressed this problem in the literature Crainic & Kim (2007).

Most commercial products, such as notebooks, clothes, and books, are transported in containers for convenience. Containers are easy to handle and offer protection to the goods. The fixed size of the containers is also an advantage. Generally, containers are handled by quay cranes. Liquid bulk refers to crude oil, petrol, fuel oil, vegetable oils, and even wine. Liquid products are often transported in tankers installed on vessels. Pumping equipment, pipelines, and loading arms are some machines used to handle liquid bulk. Dry bulk refers to grain, coal, iron ore, cement, sugar, salt, and sand. They are transported in large quantities to the compartment of the vessel. The dry bulk can be handled by conveyors, ship unloaders, ship loaders, and truck unloaders. Breakbulk refers to paper, wood, bags of cocoa, rolls of steel, and parts of wind turbines; these are all products that can be transported in a container or simply put on a vessel. A crane or forklift truck can easily load or discharge the goods. Ro-ro refers to *roll-on / roll-off*. This name explains how the cargo is discharged and loaded. This concerns cargo that can be driven, which is only done by specially trained drivers. Ro-ro is used for cars, busses, trucks, agricultural vehicles, and cranes Crainic & Kim (2007).

One of the most relevant logistic problems to address the management of ports is the Berth Allocation Problem (BAP), which consists of assigning and scheduling vessels to berthing positions along the quay, with the aim of minimizing the waiting time of the vessels Imai et al. (2003). Since quay cranes are vastly used in terminals for container movement, an inefficient quay crane employment can negatively impact the BAP results, therefore, many papers start to address the Quay Crane Assignment Problem (QCAP), which consists of assigning the quay cranes to vessels, integrated with the BAP Giallombardo et al. (2010) and Imai et al. (2008).

Containers, dry and oil bulks are the majority of cargo handled by maritime shipping. Ship unloaders and pumping equipment can be used for the movement of dry and oil bulk, respectively, therefore, a mathematical model capable of assigning such machines can further improve the management efficiency. In this paper, we propose a mathematical model to integrate the BAP with the machine assignment problem, such as quay cranes, ship unloaders, pumping equipment, and conveyors, among others. The model can use one or more types of machines, since the handling of vessels can require, for instance, quay cranes and trucks. The model also uses a continuous-time horizon and fixed-length berths. We also provide an efficient heuristic, capable of solving large instances of the model in minutes.

The paper is organized as follows. The literature review related to the berth allocation problem is in Section 2. A heuristic to solve the problem is presented and discussed in Section 4. The computational analyses of the models, the heuristic, and numerical experiments are presented in Section 5. The final section concludes the paper.

2 LITERATURE REVIEW

One of the first mathematical models for the BAP was proposed in Imai et al. (1997) where the port berths have a fixed length and are considered fixed points along the quay. In this model, the authors assume a situation named static, where all vessels are assumed to be in the port, this model is referenced as static discrete BAP or static DBAP by the authors. The model aims to minimize the allocation time of the vessels in the berths to provide a better solution than the FIFO technique, also called *first in first served technique* (FIFS). The FIFO technique consists of handling the vessels by the arrival order, this technique is vastly used in commercial ports.

In Imai et al. (2001) the BAP of Imai et al. (1997) is extended to the dynamic version, where all vessels have an arrival time on the planning horizon. This kind of model is called dynamic DBAP by the authors. Both static and dynamic DBAP models assume that all vessels can be allocated in any berth.

Cordeau et al. (2005) formulates the BAP as a vehicle routing problem with time windows (MD-VRPTW) and develops a heuristic based on tabu search to solve instances of the model. The heuristic has a version for the discrete BAP and another for the continuous BAP, where the port berths have a variable length. In the continuous BAP, also called CBAP, the quay is divided into sections of any length and the length of all vessels is expressed in units of the same length as the sections. The berth length can vary with each vessel allocation and is given by the number of units of sections the vessel used for allocation. The authors also compare the heuristic results with solver results.

Another group of models is the Tactical Berth Allocation Problem (TBAP). One of the first works to address this kind of problem is Moorthy & Teo (2006) where their model aims to represent the trade-off between the waiting time of the vessels and the costs of moving containers between the berths and yards. The model is formulated as a rectangular packing problem in a cylinder and uses simulated annealing based on pairs of sequences algorithm to solve the problem. The objective is to maximize the service level and minimize the costs related to the container movements between berths and yards.

Giallombardo et al. (2010) proposed another TBAP mathematical model to integrate the BAP with the quay crane schedule (CS). In this model, the time horizon is discretized into several time partitions of the same length and every variable and parameter is defined with an index related to each time partition. The variables and parameters associated with the vessels and berths are very similar to the variables and parameters of the DBAP. The quay crane schedule (CS) presupposes that a reallocation of the quay cranes can occur at the end of each work shift change and, therefore, the length of each time partition used in this model is given by the work shift duration. This model also considers the yard cost dependent on the allocation berth of each vessel. This model is one of the first to consider the integration between the BAP and the quay crane scheduling, but the partition of the horizon time causes a large increase in the number of variables and consequently increases the memory cost of and processing time to solve instances of the model.

Zhen et al. (2011) proposed an integrated template-planning model for both berthing locations in continuous indexes and yard container stack arrangement. They also provide a heuristic to solve instances of the model. Hendriks et al. (2012) extend the BAP model for a multi-terminal port, which can allocate two connected vessels in different terminals, which generates an inter-terminal container transport. The objectives of the model are balancing the working load of the quay crane between the terminals along the time and minimizing the number of containers transported between terminals.

Hendriks et al. (2013) approach a TBAP for terminals with discrete berths integrated with the allocation of yard spaces. Lee & Jin (2013) discuss a TBAP for feeder terminals to allocate the feeders in discrete locations berths and yard storage assignment for the cargo. Vacca et al. (2013) developed an exact-solution algorithm based on the branch and price technique for the TBAP of Giallombardo et al. (2010). Shang et al. (2016) investigates the integrated berth allocation and quay crane assignment problem in container terminals under data uncertainties. A deterministic model is formulated by considering the setup time of quay cranes, the horizon time is discretized.

He (2016) formulated the problem of integrated berth allocation and quay crane assignment for the trade-off between time-saving and energy-saving as mixed-integer programming, in order to minimize the total departure delay of all vessels and the total handling energy consumption of QCs. In the mathematical model, the berth position of each vessel is a parameter and the QC allocation is made by an allocation variable defined in a discretized time horizon.

Iris et al. (2017) focuses on the integrated berth allocation and quay crane assignment problem in container terminals, with a continuous berth approach, and the quay is divided into small equal-sized sections. The problem considers the decrease in the marginal productivity of quay cranes and the increase in handling time due to deviation from the desired position, and an Adaptive Large Neighborhood Search heuristic has been proposed too. The parameters and variables are defined in a discretized time horizon. Agra & Oliveira (2018) proposed a model based on the relative position formulation to avoid big-M constraints. The model results from a discretization of the time and space variables.

Zheng et al. (2019) studies an integrated berth allocation and quay crane assignment problem, where the quay crane maintenance is involved and establishes an integer linear programming with the objective of minimizing the total turnaround time. The authors also provide a GA-based metaheuristic to solve large instances. Correcher et al. (2019) proposed a new mixed integer linear model for the Berth Allocation Problem and the Quay Crane Assignment Problem, for the continuous BAP version with time-invariant crane assignment, in this model the vessels can be allocated at any quay position, not requiring any quay discretization, but the time horizon is discretized.

Other works model the machine assignment problem apart from the BAP. One of the most recent works is Zhen et al. (2019), which proposed a mixed-integer programming model for the scheduling of quay cranes and yard trucks for unloading operations, they use a particle swarm optimization-based solution method and compare the proposed method with the CPLEX solver

and the genetic algorithm. Bierwirth & Meisel (2009) considers the problem of scheduling quay cranes and presents a revised optimization model for the scheduling of quay cranes and proposes a heuristic solution procedure. They applied a Branch-and-Bound algorithm for searching a subset of above-average quality schedules. Cao et al. (2010) proposes a novel integrated model for yard truck and yard crane scheduling problems for loading operations in container terminals. The problem was formulated as a mixed-integer programming model, and two efficient solution methods are provided.

Chen et al. (2013) studies the interactions between crane handling and truck transportation, where trucks are shared among different ships, which helps to reduce empty truck trips in the terminal area. The problem is formulated as a constraint programming model and a three-stage algorithm is provided. At the first stage, crane schedules are generated by a heuristic method. At the second stage, the multiple-truck routing problem is solved based on the precedence relations of the transportation tasks derived from the first stage. At the last stage, a complete solution is constructed by using a disjunctive graph. The authors claimed that the three procedures facilitated and obtained the search for good solutions. Cao et al. (2010) proposes a novel integrated model for yard truck and yard crane scheduling problems for loading operations in a container terminal. The problem was formulated as a mixed-integer programming model, and two efficient solution methods are provided.

Fu et al. (2014) proposes a new approach to analyze the integrated quay crane assignment and scheduling problem (QCASP). The problem determines the assignment of quay cranes to vessels and the sequence of tasks to be processed by each quay crane simultaneously and accounts for important considerations such as safety margins between quay cranes (QCs), ordering conditions, and vessel priority. The authors provide a genetic algorithm (GA) to solve and validated the integrated problem. Bierwirth & Meisel (2010) and Bierwirth & Meisel (2015) contain a review of publications up to 2014/2015 related to the BAP and machine assignment problem.

The mathematical model proposed in this paper aims to optimize the allocation of vessels integrated with the machine assignment problem needed to load or unload the vessels. In order to find the machine schedule, models like Park & Kim (2003), Imai et al. (2008) and Meisel & Bierwirth (2009) makes an hour-by-hour relocation of all machines and Giallombardo et al. (2010) and Vacca et al. (2013) makes turn-by-turn relocation of all machines, which requires a discretized horizon time. Our model uses novel variables defined in a continuous horizon time and constraints to detect simultaneous vessel allocations in different berths, and with that information, the model is capable of distributing the machines for the handling of the vessels.

3 MATHEMATICAL MODEL

The problem consists in deciding the berth, the position, and the number of machines for each vessel handling, aiming to minimize the total waiting time and handling time. To model the problem, we make the following assumptions.

- Each berth can serve one vessel at a time.

- Vessel handling time is dependent on the number of machines.
- Vessel is served after its arrival.
- Vessel handling must be finished without interruption once they get started.
- Vessel handling has a minimum and a maximum number of machines, and it begins when the vessel is assigned to a berth.
- The number of machines for handling tasks cannot surpass the total available at any given time.
- Each type of machine generates its own service rate based on its number.
- The effective rate of handling of a vessel is given by the slowest rate between the machine types.
- Machines can allocate during the vessel allocation process.

To address the machine assignment, consider the following example of 1, where the container port has berths distributed by sets of three, and each set has disconnected rails.

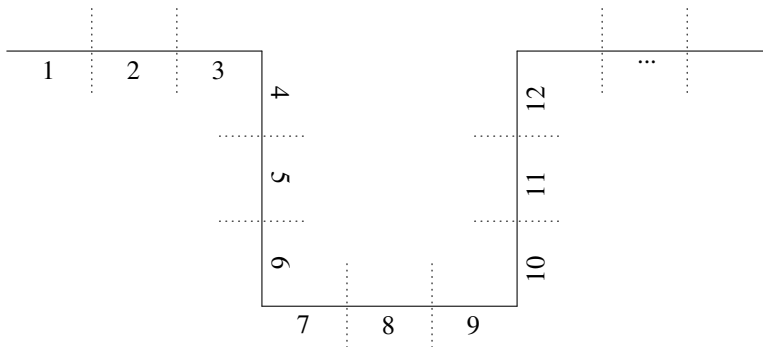


Figure 1 – Example of berths division.

Berths only compete for the usage of quay cranes with the ones in the same rail. To address this issue, we introduce in this work the concept of regions of the quay. We define sets of berths that compete for the usage of a given type of machine. Each set has a number of available machines. In our example, for the quay crane, we have the sets given by 2.

The sets generated by different types of machines are not necessarily equal. In our example, the sets of 2 are not the same for trucks. Since trucks have free mobility between berths, we have a single set, which contains all berths and has the total number of trucks available.

With the concept of berth sets, we present the mathematical model created to optimize the handling of vessels on a terminal with multiple types of machines. There is a presentation of the sets,

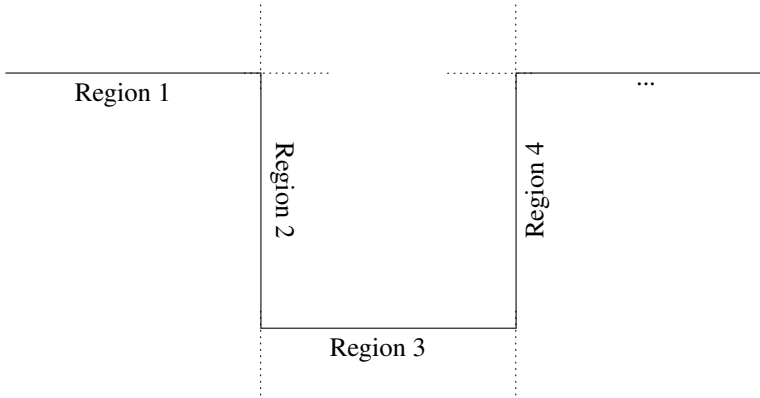


Figure 2 – Example of terminal/quay regions.

parameters, and variables before the mathematical model, and the explanation of the objective function and every constraint is presented after the mathematical model.

Sets

- B : set of berths.
- B^r : subset of berths in region r .
- B_k^r : subset of all k -combinations of B^r elements.
- N : set of vessels.
- N_k : set of all k -combinations of N elements, without repetition.
- C_{N_k} : is the Cartesian product $N_k \times N_k$.
- O : set of allocation order.
- P : set of machine type.
- R_α : set of regions of machine type $\alpha \in P$.

Variables

- $x_{i,o,k} = \begin{cases} 1, & \text{if the vessel } i \text{ is the } o\text{-th handling on berth } k \\ 0, & \text{otherwise.} \end{cases}$
- T_i : allocation time of vessel i in units of time.
- t_i : handling duration of vessel i in units of time.

- $s_{i,j} = \begin{cases} 1, & \text{if the end of vessel } i \text{ handling occurs before the beginning of} \\ & \text{the vessel } j \text{ handling} \\ 0, & \text{otherwise.} \end{cases}$
- $w_{i,j} = \begin{cases} 1, & \text{if the beginning of vessel } i \text{ handling occurs before the end of} \\ & \text{the vessel } j \text{ handling} \\ 0, & \text{otherwise.} \end{cases}$
- $m_{i,\alpha}$: number of α type machines allocated for the handling of vessel i in units.

Parameters

- β_i and β'_i : costs waiting and handling time of vessel i , respectively.
- δ_α : handling rate of one α type machine in cargo volume per unit of time.
- $U_{i,\theta}^\alpha$: handling time difference of vessel i between using $\theta - 1$ and θ machines of type α in units of time.
- a_i and b_i : arrival and maximum departure time of vessel i in units of time, respectively.
- $c_{i,\alpha}$ and $d_{i,\alpha}$: minimum and maximum number of machines type α to handling vessel i in units, respectively.
- M'_α : number of machine type α available in region r in units.
- M_α : total number of machine type α in units.
- Q_i : load of vessel i in units of cargo.
- H : large enough constant.

Mathematical Model

$$\min \sum_{i \in N} \beta_i(T_i - a_i) + \beta'_i t_i \tag{1a}$$

$$\text{s.t. } t_i \geq (m_{i,\alpha} - \theta + 1) \left(\frac{Q_i}{j\delta_\alpha} - \frac{Q_i}{(\theta - 1)\delta_\alpha} \right) + \frac{Q_i}{(\theta - 1)\delta_\alpha} \tag{1b}$$

$$\forall i \in N, \forall \alpha \in P, \forall \theta \in \{2, 3, \dots, M'_\alpha\}, r \in R_\alpha$$

$$T_i \geq a_i \quad \forall i \in N \tag{1c}$$

$$T_i + t_i \leq b_i \quad \forall i \in N \tag{1d}$$

$$\sum_{i \in N} x_{i,o,k} \leq 1 \quad \forall o \in O, \forall k \in B \tag{1e}$$

$$\sum_{k \in B} \sum_{o \in O} x_{i,o,k} = 1 \quad \forall i \in N \tag{1f}$$

$$\sum_{i \in N} x_{i,o,k} \leq \sum_{i \in N} x_{i,o-1,k} \quad \forall o \in O \setminus \{1\}, \forall k \in B \tag{1g}$$

$$T_i - H(2 - x_{i,o,k} - x_{j,o-1,k}) \geq T_j + t_j \tag{1h}$$

$$\forall i \in N, \forall j \in N, \forall o \in O \setminus \{1\}, \forall k \in B$$

$$T_i + t_i - H(1 - s_{i,j}) \leq T_j \quad \forall i \in N, \forall j \in N \tag{1i}$$

$$T_i + H(1 - w_{i,j}) \geq T_j + t_j \quad \forall i \in N, \forall j \in N \tag{1j}$$

$$\sum_{i \in \bar{N}} m_{i,\alpha} \leq ((k^2 + 1) - \sum_{v \in C_{\bar{N}}} (w_v + s_v) - \sum_{b \in \bar{B}} \sum_{i \in \bar{N}} \sum_{o \in O} x_{i,o,b}) M'_\alpha \tag{1k}$$

$$\forall \alpha \in P, \forall \bar{N} \in N_k, \forall \bar{B} \in B_k^r, \forall r \in R_\alpha, \forall k \in \{1, 2, \dots, |\bar{B}|\}$$

$$m_{i,\alpha} \geq c_{i,\alpha} \quad \forall i \in N, \forall \alpha \in P \tag{1l}$$

$$m_{i,\alpha} \leq d_{i,\alpha} \quad \forall i \in N, \forall \alpha \in P \tag{1m}$$

$$x_{i,o,k} \in \{0, 1\} \quad \forall i \in N, o \in O, \forall k \in B \tag{1n}$$

$$T_i \geq 0 \quad \forall i \in N \tag{1o}$$

$$s_{i,j}, w_{i,j} \in \{0, 1\} \quad \forall i \in N, \forall j \in N \tag{1p}$$

$$m_{i,\alpha} \in \{0, 1, 2, 3, 4, \dots\} \quad \forall i \in N, \forall \alpha \in P. \tag{1q}$$

The Function 1a is the objective function that represents the weighted sum of waiting and handling time of the vessels. The Constraints 1b represent the handling time calculation of each vessel. The handling time of the vessel i is given by $\min_{\alpha \in P} \left\{ \frac{Q_i}{m_{i,\alpha} \delta_\alpha} \right\}$. The handling time generate by the usage of j machines is given by (2).

$$t_j = \frac{Q_i}{j\delta_\alpha} \tag{2}$$

Since (2) is a non-linear equation, a set of linear equations is used to represent the handling time of each vessel. To avoid non-linear equations, the model uses line equations defined by the values of the given non-linear equation in integer points, as illustrated in Figure 3.

$$y = \left(\frac{Q_i}{j\delta_\alpha} - \frac{Q_i}{(j-1)\delta_\alpha} \right) (x - j + 1) + \frac{Q_i}{(j-1)\delta_\alpha} \tag{3}$$

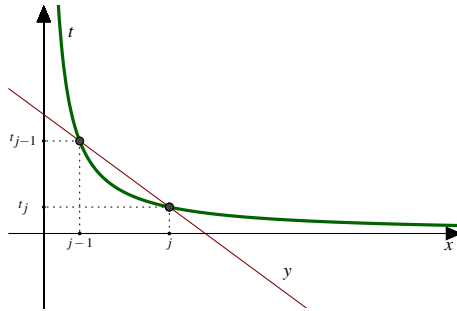


Figure 3 – Linear approach to represent exponential functions in integer points.

The red line in Figure 3 represents the function $\frac{Q_i}{m_{i,\alpha}\delta_\alpha}$, Therefore, the value of y is greater than $t = \frac{Q_i}{m_{i,\alpha}\delta_\alpha}$ when $j - 1 < m_{i,\alpha} < j$ for a machine type α and the vessel i . The number of machines is an integer value, therefore, $y \leq t$ for any vessel i and any machine type α .

The Constraints 1c ensure the allocation time of each vessel occurs before the arrival time. The Constraints 1d ensure the end of the handling before the maximum departure time. The Constraints 1e and 1f ensure one allocation in one berth for the vessels. The Constraints 1g ensure the allocation position of each vessel. The Constraints 1h realize the calculation of each vessel allocation time. The Constraints 1i and 1j compute the case where the vessels have simultaneous handling in different berths. For example, given the region \hat{r} with two berths, which is $B^{\hat{r}} = \{k_1, k_2\}$, therefore, the sets C_2 , N_2 and $B_2^{\hat{r}}$ are given by:

- $B_2^{\hat{r}} = \{k_1, k_2\}$.
- $N_2 = \{\{i, j\}\} \quad \forall i \in N, j \in N, i \neq j$.
- $C_2 = \{(i, j)\} \quad \forall i \in N, j \in N, i \neq j$.

The Constraints 1k will have the form of Constraints 4.

$$m_{i,\alpha} + m_{j,\alpha} \leq (5 - w_{i,j} - s_{i,j} - \sum_{o \in O} x_{i,o,k_1} - \sum_{o \in O} x_{j,o,k_2}) M_\alpha^{\hat{r}} \quad \forall i \in N, \forall j \in N, \forall \alpha \in P. \tag{4}$$

If $w_{i,j} = 1$ and $s_{i,j} = 1$ the handling of vessels i and j are simultaneous, and if $\sum_{o \in O} x_{i,o,k_1} = 1$, $\sum_{o \in O} x_{j,o,k_2} = 1$ the handling of vessels i and j occurs in berths of the same region. In this case, the machines of this region will be divided between the vessels i and j . When \hat{r} has three berths, which is $B^{\hat{r}} = \{k_1, k_2, k_3\}$, therefore, the sets C_2 , C_3 , N_2 , N_3 , $B_2^{\hat{r}}$ and $B_3^{\hat{r}}$ are given by:

- $B_2^{\hat{r}} = \{\{k_1, k_2\}, \{k_1, k_3\}, \{k_2, k_3\}\}$.
- $B_3^{\hat{r}} = \{\{k_1, k_2\}\}$.
- $N_2 = \{\{i, j\}\} \quad \forall i \in N, j \in N, i \neq j$.
- $N_3 = \{\{i, j, p\}\} \quad \forall i \in N, j \in N, p \in N, i \neq j \neq p$.
- $C_2 = \{(i, j)\} \quad \forall i \in N, j \in N, i \neq j$.
- $C_3 = \{(i, j), (i, p), (j, p)\} \quad \forall i \in N, j \in N, p \in N, i \neq j \neq p$.

The Constraints 1k will have the form of Constraints 5, 6, 7 and 8.

$$m_{i,\alpha} + m_{j,\alpha} \leq (5 - w_{i,j} - s_{i,j} - \sum_{o \in O} x_{i,o,k_1} - \sum_{o \in O} x_{j,o,k_2}) M_{\alpha}^{\hat{r}} \quad \forall i \in N, \forall j \in N, \forall \alpha \in P, \quad (5)$$

$$m_{i,\alpha} + m_{j,\alpha} \leq (5 - w_{i,j} - s_{i,j} - \sum_{o \in O} x_{i,o,k_2} - \sum_{o \in O} x_{j,o,k_3}) M_{\alpha}^{\hat{r}} \quad \forall i \in N, \forall j \in N, \forall \alpha \in P, \quad (6)$$

$$m_{i,\alpha} + m_{j,\alpha} \leq (5 - w_{i,j} - s_{i,j} - \sum_{o \in O} x_{i,o,k_1} - \sum_{o \in O} x_{j,o,k_3}) M_{\alpha}^{\hat{r}} \quad \forall i \in N, \forall j \in N, \forall \alpha \in P, \quad (7)$$

$$m_{i,\alpha} + m_{j,\alpha} + m_{p,\alpha} \leq \left(10 - w_{i,j} - s_{i,j} - w_{i,p} - s_{i,p} - w_{j,p} - s_{j,p} - \sum_{o \in O} x_{i,o,k_1} - \sum_{o \in O} x_{j,o,k_2} - \sum_{o \in O} x_{q,o,k_3} \right) M_{\alpha}^{\hat{r}} \quad \forall i \in N, \forall j \in N, \forall p \in N, \forall \alpha \in P. \quad (8)$$

The Constraints 5, 6, 7 and 8 verify if the handling of two and three vessels occurs at the same time in berths of the same region.

The Constraints 1j compute the values of $s_{i,j}$ and $w_{i,j}$. The Constraints 1k, 1l and 1m limit the number of machines. The Constraints 1n, (1o), (1p) and (1q) ensure that the allocation variables are binary and time variables are non-negative continuous.

The U^{α} matrix, without any penalties for handling a vessel with more than one machine of the same type, is given by:

$$U_{i,\theta}^{\alpha} = \begin{cases} \frac{Q_i}{\delta_{\alpha}} & \text{if } \theta = 1 \\ \frac{Q_i}{\theta \delta_{\alpha}} - \frac{Q_i}{(\theta-1)\delta_{\alpha}} & \text{se } \theta \geq 2. \end{cases}$$

When the mathematical model is used with one type of machine, for example, an instance with only quay cranes, we can prove that the variable related to the number of machines can be implemented as continuous and the solution will be an integer. The Figure 1 enunciates and a proof follows.

Theorem 1. *When $|P| = 1$ in an instance of the Mathematical Model 1, with b_i large enough for all $i \in N$. Then there is an optimal solution of that instance with $m_i \in \mathbb{Z}$, for all $i \in N$, using Constraints 1q written as:*

$$m_i \in \mathbb{R}^+ \quad \forall i \in N. \quad (9)$$

Proof. As $|P| = 1$, to simplify the notation, the index α will be omitted, and we also consider a single region, which is $R = 1$, cases with $R \geq 2$ are similar. Given the objective function $\min \sum_{i \in N} \beta_i T_i + \beta'_i t_i$, with $\beta > 0$ and $\beta' > 0$, that is, the constant part of the objective function given by $\sum_{i \in N} \beta_i a_i$ will be also omitted to facilitate the proof. Let an instance solution of the Mathematical Model 1 given by:

$$x_{i,o,k}^*, T_i^*, s_{i,j}^*, w_{i,j}^*, m_i^*, \text{ where } i \in N, j \in N, o \in O, k \in B. \tag{10}$$

As 10 it is a solution of an instance, then it satisfies Constraints 1b - 1q.

Given $m_i^* = m_i^I + m_i^R$, where $m_i^I \in \mathbb{Z}$ and $m_i^R \in [0, 1)$, with $i \in N$, $N^R = \{i \in N | m_i^R > 0\}$, where $k \in B$, suppose $N^R \neq \emptyset$ and the subsets D_g , such that, given $i, j \in D_g$, with $g \in \{1, 2, 3, \dots\}$, then $s_{i,j} = w_{i,j} = 1$, besides that $x_{i,o_1,k_1} = 1$ and $x_{j,o_2,k_2} = 1$ with $o_1, o_2 \in O$ and $k_1, k_2 \in B$ where $k_1 \neq k_2$, that is, D_g is a set of vessels dividing machines.

Such as Solution 10 satisfy Constraints 1l and 1m, then:

$$c_i < m^* < d_i \quad i \in N^R. \tag{11}$$

Of Equation 11 follow that:

$$0 < m_i^I - c_i \leq m_i^R \leq d_i - m_i^I < 1 \quad i \in N^R. \tag{12}$$

Defining:

$$\hat{m}_i = m_i^I \quad i \in N^R \tag{13}$$

Initially, taking $t'_i = t_i^*$ where $i \in N$.

1. Let $m_g^a = M - \sum_{i \in D_g} \hat{m}_i \quad g \in \{1, 2, 3, \dots\}$.
 2. With the definition of m_g^a , define the subset $G' = \{g \in \{1, 2, 3, \dots\} : m_g^a > 0\}$.
 3. Taking the number of machines from each ship as \hat{m}_i , where $i \in N$ defined on Equation 13, update the value of handling time
- $$t'_i = \max_{j \in \{2, 3, \dots, |M|\}} \left\{ t'_i + m_i^R \left(\frac{Q_i}{(j-1)\delta} - \frac{Q_i}{j\delta} \right) \right\} \quad j \in \{2, 3, \dots, |M|\}, i \in N.$$
4. Given $g \in G'$ and $i \in g$, define $\hat{f}_i^g = \sum_{j \in N} (\beta_j \hat{T}_j + \beta'_j \hat{t}_j)$ where:

$$\hat{t}_v = \begin{cases} t_v & v \in N \setminus \{i\} \\ t'_v - \left(\frac{Q_v}{(j-1)\delta} - \frac{Q_v}{j\delta} \right) & \text{if } j \in \{2, 3, \dots, |M|\}, \text{ if } v = i. \end{cases}$$

$$\hat{T}_v = \begin{cases} T_v & \text{if } x_{i,o,k} = x_{v,o',k} = 1, \text{ with } o' \leq o \\ \max(\hat{T}_u + \hat{t}_u, a_j) & \text{if } x_{j,o,k} = x_{u,o-1,k} = 1. \end{cases}$$

5. Calculate $(\hat{g}, \hat{i}) = \operatorname{argmin}_{g \in G'} \left\{ \operatorname{argmin}_{i \in D_g} \{ \hat{f}_i^g \} \right\}$, $e \hat{m}_i \leftarrow \hat{m}_i + 1$ and $m_g^a \leftarrow m_g^a - 1$.
6. Calculate $\hat{f} = \sum_{i \in N} \beta_i \hat{T}_i + \beta'_i \hat{t}_i = \min_{g \in G'} \left\{ \min_{i \in D_g} \{ \hat{f}_i^g \} \right\}$, we have $\hat{f} \leq f$, with the new solution, given by $x_{i,o,k}^*$, \hat{T}_i , $s_{i,j}^*$, $w_{i,j}^*$, \hat{m}_i , where $i \in N$, $j \in N$, $o \in O$, $k \in B$.

Repeat Steps 1 - 6 until $G' = \emptyset$. Note that the solution on Step 6 always satisfy the Constraints 1l and 1m. \square

The theorem hypothesis of a large enough maximum departure time can be achieved using fines or financial personality for departure delays.

4 SOLUTION METHOD

This section contains the description of the main heuristics used to solve the mathematical model. The heuristic first finds an upper time limit (T), this time limit will be used to determine the set of vessels that will be assigned to a berth in the iteration. A set of vessels (S) are assigned to berths in each iteration, the set S is given by all not assigned vessels that have an arrival time less than T . The set S can contain more vessels than the total number of available berths, in that case, we prioritize vessels with less load. After allocating the vessels of S , the assignment of machines to handle the vessels is made with an auxiliary MILP problem, this MILP is called at most $|S|$ times, with fewer vessels and limiting the number of machines every time, in an effort to find better local solutions. After assigning all vessels of S , the value of T is updated and a new iteration is initiated. To facilitate understanding of some data checks, reallocation and re-indexations will be skipped in the Algorithm 1.

The sets, variables, and parameters of the algorithm are:

- N is the set of Vessels;
- $S_b \leftarrow \emptyset$ is the vessel schedule of Berth $b \in B$
- $m_{i,\alpha}$ is the number of machines type α allocate to handling the vessel i .
- $T_i; t_i; z$ are the allocation time, handling time, objective function value, respectively. Where $i \in N$.
- $T_i^*; t_i^*; z^*; m_{i,\alpha}^*$ are the best allocation time, handling time, objective function value and machine allocation, respectively. Where $i \in N$.
- BT_b is the opening time of berth $b \in B$;
- T is the initial time reference;
- $mB_{b,\alpha}$ is the number of machines type α in the berth b , where $b \in B$ and $\alpha \in P$;
- Initialize the values of $T_i^*; z^*; mB_{b,\alpha}$ as zero;

Algorithm 1 Algorithm to Solve the Berth and Machine Allocation Problem

- 1: Initialize the values of T_i^* ; z^* ; $mB_{b,\alpha}$ as zero;
- 2: **while** $N \neq \emptyset$ **do**
- 3: $S \leftarrow \{i : a_i < T + tol, i \in N\}$;
- 4: Sort S by the value of Q in ascendant order;
- 5: **if** $|S| > |B|$ **then**
- 6: remove the last $|S| - |B|$ elements of S ;
- 7: **end if**
- 8: $T_i^* \leftarrow T$ where $i \in S$;
- 9: Find t_i^* ; z^* ; $m_{i,\alpha}^*$ for $i \in S$ by solving the Mathematical Model 14;

$$\min \sum_{i \in S} t_i \tag{14a}$$

$$s.a. t_i \geq (m_{i,\alpha} - \theta)U_{i,\theta}^\alpha + \sum_{j=1}^{\theta} U_{i,j}^\alpha \tag{14b}$$

$$\forall i \in S, \forall \alpha \in P, \forall \theta \in \{2, 3, \dots, M_\alpha\}$$

$$\sum_{k \in B} m_{i,\alpha} \leq M_\alpha \quad \forall i \in S, \alpha \in P \tag{14c}$$

$$m_{i,\alpha}^{\min} \leq m_{i,\alpha} \leq m_{i,\alpha}^{\max} \quad \forall i \in S, \alpha \in P \tag{14d}$$

$$m_{i,\alpha} \in \{0, 1, 2, 3, \dots\} \quad \forall i \in S, \alpha \in P. \tag{14e}$$

- 10: Sort B according to the value of BT in descendant order;
- 11: $S' = S$;
- 12: **for** $b \in B$ **do**
- 13: $T \leftarrow BT_b$;
- 14: $T_i \leftarrow T$ for $i \in S$;
- 15: Find $t_i; z; m_{i,\alpha}$ for $i \in S, \alpha \in P$ by solving the Mathematical Model 14
- 16: Calculate the value of z ;
- 17: **if** $z < z^*$ **then**
- 18: $z^* \leftarrow z$;
- 19: $T_i^* \leftarrow T_i; t_i^* \leftarrow t_i, m_{i,\alpha}^* \leftarrow m_{i,\alpha}$; where $\alpha \in P$ and $i \in S$;
- 20: **end if**
- 21: $M_\alpha \leftarrow M_\alpha - mB_{b,\alpha}$ for all $\alpha \in P$;
- 22: $S_b \leftarrow S_b \cup \{i\}$ where i is the first element of S ;
- 23: remove the first element of S ;
- 24: **end for**
- 25: $BT_b \leftarrow T_j^* + t_j^*$ for all $b \in B$, where j is the last element of S_b ;
- 26: $mB_{b,\alpha} \leftarrow m_{j,\alpha}$ for all $b \in B$ and $\alpha \in P$, where j is the last element of S_b ;
- 27: $N \leftarrow N \setminus S'$
- 28: **end while**

To show an example of how the Algorithm 1 works, consider Figure 4.

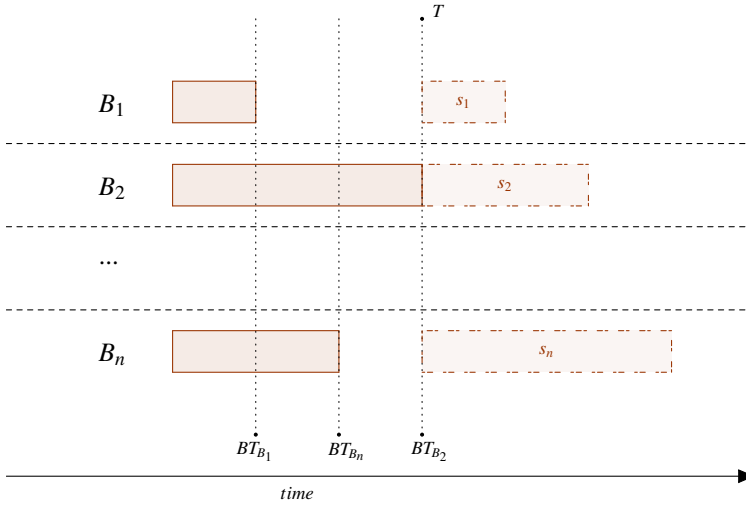


Figure 4 – Example of Heuristic 1 - Iteration 1.

Figure 4 shows three berths, B_1 , B_2 and B_n , therefore, $B = \{B_1, B_2, B_n\}$. Each rectangle represents a vessel and its length represents the handling time. Solid rectangles are vessels treated in previous iterations, and dotted rectangles are vessels treated in the current iteration. Each berth became available in the time BT_{B_i} , which is given by the end of the handling of previous iterations.

In the Step 9 the values of $m_{i,\alpha}^*$ are determined. With $m_{i,\alpha}^*$ the values of T_i^* ; t_i^* and z^* are calculated. In this step all berths are available at the time $T = \max_{b \in B} BT_b$ and, in this time position, every machine is available to work at any berth, therefore, it is possible to allocate any number of machines as long as it satisfies the vessels and berths limitations. The set of vessels treated in this iteration are $S = \{s_1, s_2, s_n\}$. In Step 10 the set S is ordered by berth time BT_b in descendant order, which is $S = \{s_2, s_n, s_1\}$. The Loop 12 begins with $T = \max_{b \in B} BT_b$, in our example, $T = BT_{B_2}$. In this iteration, the vessel in B_2 has its position fixed, that is, the values of allocation time, handling time and number of machines are given by $T_{s_2}^*$, $t_{s_2}^*$ and $m_{s_2,\alpha}^*$, respectively, and s_2 is removed from the set S and insert as the last element of S_{B_2} . For the next step, the value of T returns to the previous position, which is $T = BT_{B_n}$. Figure 5 shows the second iteration of Loop 12.

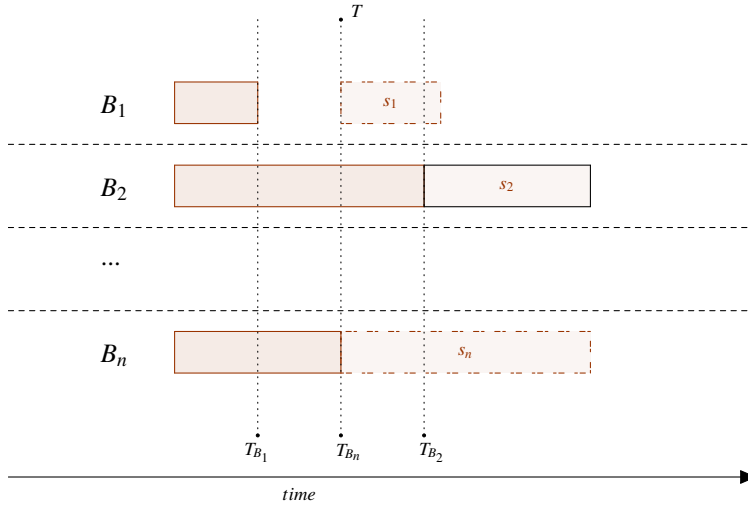


Figure 5 – Example of Heuristic 1 - Iteration 2.

In the second iteration of Loop 12, we have $T = BT_{B_n}$ and $S = \{s_n, s_1\}$. We added the constraint limiting the machines of vessels s_1 and s_n to the sum of machines in B_1 plus the machines in B_n plus the idle machines, which is $m_{s_1,\alpha} + m_{s_n,\alpha} \leq mB_{B_1,\alpha} + mB_{B_n,\alpha} + (|M_\alpha| - mB_{B_2})$ for all $\alpha \in P$ to the problem. In Step 15 we select the number of machines that generates the best objective function with the new constraint. If $z < z^*$ then $z^* \leftarrow z$, $T_{s_1}^* \leftarrow T_{s_1}$, $t_{s_1}^* \leftarrow t_{s_1}$, $m_{s_1,\alpha}^* \leftarrow m_{s_1,\alpha}^*$, $T_{s_n}^* \leftarrow T_{s_n}$, $t_{s_n}^* \leftarrow t_{s_n}$ and $m_{s_n,\alpha}^* \leftarrow m_{s_n,\alpha}^*$. If $z \geq z^*$ or the problem is infeasible, nothing changes for $T_{s_1}^*, t_{s_1}^*, T_{s_n}^*, t_{s_n}^*, m_{s_1,\alpha}^*$ and $m_{s_n,\alpha}^*$. The vessel s_n is fixed with the values of $T_{s_n}^*, t_{s_n}^*$ and $m_{s_n,\alpha}^*$. In our example, the value of $z < z^*$ and values of s_1 and s_n are updated, s_n is removed from set S and insert as the last element of S_{B_n} which can be seen in Figure 6. In the iteration 3, $T = BT_{B_1}$ and $S = \{s_1\}$. Figure 6 shows the beginning of the iteration 3.

Adding the constraint limiting the machines of vessel s_1 to the sum of machines in B_1 plus the idle machines, that is $m_{s_1,\alpha} \leq mB_{B_1,\alpha} + (|M_\alpha| - mB_{B_2,\alpha} - mB_{B_n,\alpha})$ for $\alpha \in P$. With that constraint and the initial berth time set to $T = T_{B_1}$ we solve the model of Step 15 with the new constraint. In our example, $z > z^*$, therefore, $z^*, T_{s_1}^*, t_{s_1}^*, m_{s_1,\alpha}^*, \alpha \in P$, does not change. The value T_{s_1} is fixed as $T_{s_1} = BT_{B_n}$, s_1 is removed from S and insert as the last element of S_{B_1} and the Loop 12 ends. Figure 7 shows the final allocation of s_1, s_2 and s_n .

After finding the values for each element of S , the values of BT, mB, p , and T are updated. On the next iteration of the main loop, T will be used to select the new group of vessels. The handling time value can change as the number of machines changes.

5 TESTS AND RESULTS

The model, heuristic and FIFO has been implemented in MATLAB using the commercial solver CPLEX 12.3 when needed and tested on the same set of instances. Instances generated to validate

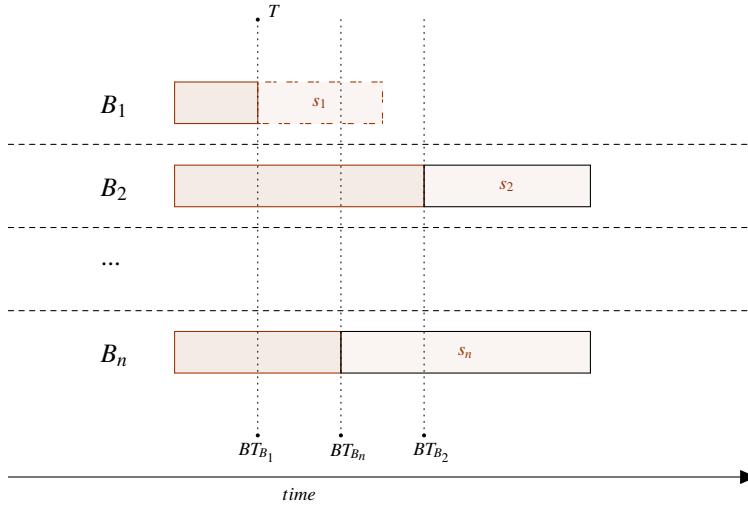


Figure 6 – Example of Heuristic 1 - Iteration 3.

our models are based on real data, from a small terminal with two berths, a few vessels, and machines to a large multi-terminal with more than a hundred berths, vessels, and machines. Table 1 brings the data interval of the instances of Table 2.

Table 1 – Parameters interval.

Parameter	interval
Number of berths	[2; 125]
Number of vessels	[4; 600]
Types of machines	[1; 3]
M_α	[2; 300]
a_i	[1; 100]
Q_i	[1; 100000]
$c_{i,\alpha}$	[1; 2]
$d_{i,\alpha}$	[2; 10]
δ_α	[3000; 9000]
β	4
β'	1

Experiments have been run with a time limit of 1 hour for instances with less than 4 berths and 3 hours otherwise. The solver was able to find solutions only in some small-scale instances, but not all. As the number of berths and vessels increases, the solver was only able to solve instances close to trivial, where the arrival of vessels is sparse. In cases with more than 4 berths, the solver was not able to find a solution in any instance.

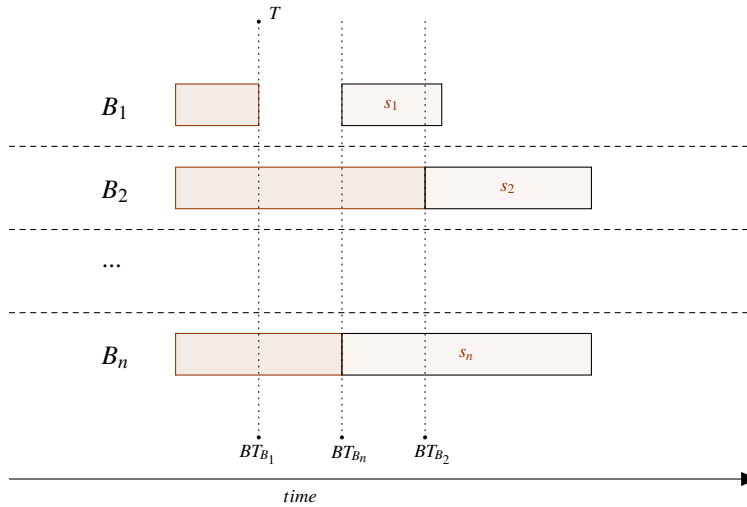


Figure 7 – Example of Heuristic 1 - Final.

Table 2 brings the results of some instances. In Table 2 the name of each case is composed by: [number of berths] + B + [number of vessels] + N + [quantity of machine Type 1] + [quantity of machine Type 2] + [quantity of machine Type 3].

Table 2 – Tests and results.

Case	Solver			Heuristic		FIFO	Solver vs Heu	Heu vs FIFO
	O.F.	gap	time	O.F.	time	O.F.		
2B4N43	21,4	0%	0	22,9	1	39,8	-7%	-74%
2B4N44	16,5	0%	0	16,5	1	16,5	0%	0%
2B4N52	119,4	0%	0	125,2	1	125,2	-5%	0%
2B4N54	10,5	0%	0	10,5	2	11,1	0%	-6%
2B4N63	43,6	0%	0	43,6	1	43,6	0%	0%
2B5N27	141,5	0%	1	145,6	2	194,4	-3%	-33%
2B5N32	181,1	0%	1	181,1	1	227,0	0%	-25%
2B5N86	40,0	0%	0	43,7	2	45,9	-9%	-5%
2B6N43	81,9	0%	1	88,8	2	94,7	-8%	-7%
2B6N53	72,0	0%	1	76,7	3	86,3	-7%	-13%
2B6N66	72,5	0%	0	72,5	2	105,4	0%	-45%
2B6N75	42,9	0%	1	46,3	3	46,3	-8%	0%
2B8N33	534,8	0%	422	564,9	1	643,8	-6%	-14%
2B4N272	81,8	0%	0	85,8	1	147,1	-5%	-71%
2B4N375	86,9	0%	0	86,9	1	92,7	0%	-7%
2B4N432	131,7	0%	0	131,7	1	138,7	0%	-5%
2B4N678	100,6	0%	0	101,0	1	101,0	0%	0%
2B4N714	242,4	0%	0	242,3	2	296,1	0%	-22%
2B5N354	53,8	0%	0	53,8	2	63,7	0%	-18%
2B5N363	76,1	0%	1	80,0	3	80,0	-5%	0%
2B5N772	78,9	0%	0	84,0	2	84,0	-6%	0%
2B5N837	56,6	0%	0	58,6	2	148,2	-4%	-153%
2B6N235	143,5	0%	1	154,7	2	251,9	-8%	-63%
2B6N628	221,0	0%	1	231,0	2	282,9	-5%	-22%
2B7N256	509,3	0%	9	509,2	3	657,5	0%	-29%
2B7N346	130,1	0%	6	137,3	3	165,7	-6%	-21%
2B8N237	73,0	0%	1	73,0	5	74,6	0%	-2%
2B8N428	592,3	0%	68	592,5	4	742,5	0%	-25%

Table 2 – Tests and results.

<i>Case</i>	<i>Solver</i>			<i>Heuristic</i>		<i>FIFO</i>	<i>Solver vs Heu</i>	<i>Heu vs FIFO</i>
	<i>O.F.</i>	<i>gap</i>	<i>time</i>	<i>O.F.</i>	<i>time</i>	<i>O.F.</i>		
2B9N476	215,1	0%	416	215,3	5	319,3	0%	-48%
3B4N26	56,7	0%	0	56,7	1	56,7	0%	0%
3B4N36	27,8	0%	0	28,5	1	39,6	-3%	-39%
3B4N55	8,8	0%	0	8,8	2	20,4	0%	-132%
3B4N66	31,1	0%	0	31,1	2	32,5	0%	-5%
3B4N72	89,1	0%	0	89,1	2	160,3	0%	-80%
3B4N73	56,9	0%	0	64,1	1	64,1	-13%	0%
3B4N86	5,6	0%	0	5,6	3	15,3	0%	-173%
3B5N34	20,8	0%	1	23,7	2	35,5	-14%	-50%
3B5N35	84,6	0%	7	87,3	2	91,1	-3%	-4%
3B5N45	62,8	0%	1	71,2	2	104,9	-13%	-47%
3B5N57	73,8	0%	11	76,7	2	95,3	-4%	-24%
3B6N22	216,4	0%	95	228,7	3	228,7	-6%	0%
3B6N37	94,4	0%	11	96,0	1	116,5	-2%	-21%
3B6N55	67,6	0%	16	74,1	3	74,1	-10%	0%
3B7N43	133,0	0%	11	134,8	3	182,6	-1%	-35%
3B7N45	110,5	0%	205	132,2	4	258,9	-20%	-96%
3B8N43	256,5	0%	42	284,3	3	297,6	-11%	-5%
3B9N33	143,6	84%	3600	163,7	3	163,7	-14%	0%
3B9N34	311,8	82%	3600	366,3	4	373,9	-17%	-2%
3B10N33	181,1	82%	3600	200,7	4	223,1	-11%	-11%
3B4N433	25,9	0%	1	25,9	1	27,7	0%	-7%
3B4N725	81,4	0%	0	86,2	3	112,0	-6%	-30%
3B4N832	44,0	0%	0	46,8	4	59,2	-6%	-27%
3B5N478	42,2	0%	1	45,2	2	53,9	-7%	-19%
3B6N457	151,5	0%	4	172,3	2	205,7	-14%	-19%
3B8N735	272,0	0%	968	288,6	6	288,6	-6%	0%
10B35N2834				71,2	173	71,2		0%
9B55N2923				423,3	512	449,2		-6%
8B70N1726				23,2	102	26,7		-15%
7B80N1520				104,9	214	104,9		0%
7B40N1221				11,8	168	15,4		-30%
6B65N1120				30,0	47	39,5		-32%
5B60N1207				9646,4	22	12477,5		-29%
5B50N1518				2941,7	41	3851,2		-31%
4B50N1806				36,8	45	39,4		-7%
8B50N30				5117,4	257	8864,1		-73%
16B50N47				628,2	309	628,2		0%
20B60N5445				692,5	394	801,3		-16%
30B100N6745				563,4	3545	654,4		-16%
30B100N7550				3687,6	550	6045,8		-64%
30B200N6099				602,0	2088	602,0		0%
60B300N150120				945,1	2722	1057,9		-12%
125B600N300250				1810,6	4649	2080,8		-15%

In Table 2 the column *Case* bring the instance name, all columns of *Solver*, *Heuristic* and *FIFO* brings the respective results. The *O.F.* columns are the value of the objective function, *gap* is the difference between the primal and the dual function obtained by the respective strategy. The *time* columns are the processing time in seconds, the column *Solver vs Heu* is the comparison between solver and heuristic solutions and *Heu vs FIFO* is the comparison between heuristic and FIFO solutions.

A negative value in column *Solver vs Heu* indicates a better solution of the solver over the heuristic by the respective percentage and a negative value the otherwise, same for the column *Heu*

vs *FIFO*. The *FIFO* implementation needed only a few seconds for finding solutions even in large-scale instances, therefore, the processing time was omitted in Table 2.

The instances from *2B4N54* to *3B8N735* have up to three berths, nine vessels, and 3 types of machines. In those instances, the heuristic obtained solutions close to the optimal obtained by the solver. In 20 of 55 instances, about 36%, the solutions of the heuristic and solver are the same. All heuristic solutions are at most 20% worse than the solver solution. The instance *3B7N45* has the worst result, with 20% higher value than the solver solution. In all instances, the heuristic solution is better or equal to the *FIFO* solution.

The instances from *10B35N2834* to *4B50N1806* have from 4 to 10 berths, from 35 to 80 vessels and 2 types of machines. The solver was unable at finding a feasible solution after three hours of processing. The heuristic obtained solutions better or equal than *FIFO* in at most one minute, only two instances have the same solutions and the others have solutions up to 32% better.

The instances from *8B50N30* to *125B600N300250* have from 8 to 125 berths, from 50 to 600 vessels and 2 types of machines. The solver was unable at finding solutions after three hours of processing. The heuristic was able to obtain better or equal solutions than *FIFO* in at most one and half hours, only two instances have the same solutions and the others have solutions up to 73% better.

Most of the optimal results of the small-scale instances prioritize the allocation of vessels with low handling time, which are related to the choice of $\beta \geq \beta'$. Choosing $\beta \geq \beta'$ prioritizes the allocation time over handling time. If two vessels are waiting in a berth queue, the sum of the allocation time of both vessels will be minimal by choosing to allocate the vessel with the smaller handling time first, which leads to a better objective function. Choosing $\beta \geq \beta'$ also leads to a better occupation of the berths, since we want to allocate the vessels early as possible, it is better to split the number of machines into different berths to be possible to allocate the vessels early. The usage of $\beta \geq \beta'$ seems more adequate, since choosing the $\beta' \geq \beta$ can lead to the concentration of the machines in a few berths to reduce the handling time of the vessels. The insight obtained from the small-scale instances was used in the heuristic. The heuristic prioritizes the allocation of vessels with less handling time in the Step 10.

6 CONCLUSIONS

A mixed-integer programming formulation has been presented to address the integration of the berth allocation problem with the machine assignment problem. The model has been validated on instances based on real data using a solver. These tests show that the problem is hardly solvable even in small instances. The proposed theorem can be used to reduce the computational complexity in some instances. As the number of berths and vessels increase, the solver was unable to find solutions, even with the application of the theorem, due to the high computational complexity. A heuristic algorithm to efficiently solve instances of the problem has also been presented, the heuristic is able to provide good feasible solutions in minutes.

The solver and heuristic solutions had similar results in the small instances. In 38 out of 55 cases, about 69%, the heuristic was worse than the optimal solution for small problems (2 and 3 berths, not more than 10 ships). The solver results are, on average, 4.96% lower. In larger instances, the solver was unable at finding solutions and the heuristic obtains solutions around 15% better than the FIFO strategy. The heuristic was able to solve instances with more than 100 berths, 500 vessels, and 250 machines in one and half hours. These dimensions are similar to the biggest terminals in the world.

The data can be found at <https://data.mendeley.com/datasets/kdr7cn53k4>.

References

- AGRA A & OLIVEIRA M. 2018. MIP approaches for the integrated berth allocation and quay crane assignment and scheduling problem. *European Journal of Operational Research*, **264**(1): 138–148.
- BIERWIRTH C & MEISEL F. 2009. A fast heuristic for quay crane scheduling with interference constraints. *Journal of Scheduling*, **12**(4): 345–360.
- BIERWIRTH C & MEISEL F. 2010. A survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, **202**(3): 615–627.
- BIERWIRTH C & MEISEL F. 2015. A follow-up survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, **244**(3): 675–689.
- CAO JX, LEE DH, CHEN JH & SHI Q. 2010. The integrated yard truck and yard crane scheduling problem: Benders' decomposition-based methods. *Transportation Research Part E: Logistics and Transportation Review*, **46**(3): 344–353.
- CHEN L, LANGEVIN A & LU Z. 2013. Integrated scheduling of crane handling and truck transportation in a maritime container terminal. *European Journal of Operational Research*, **225**(1): 142–152.
- CORDEAU JF, LAPORTE G, LEGATO P & MOCCIA L. 2005. Models and tabu search heuristics for the berth-allocation problem. *Transportation science*, **39**(4): 526–538.
- CORRECHER JF, ALVAREZ-VALDES R & TAMARIT JM. 2019. New exact methods for the time-invariant berth allocation and quay crane assignment problem. *European Journal of Operational Research*, **275**(1): 80–92.
- CRAINIC TG & KIM KH. 2007. Intermodal transportation. *Handbooks in operations research and management science*, **14**: 467–537.
- FU YM, DIABAT A & TSAI IT. 2014. A multi-vessel quay crane assignment and scheduling problem: Formulation and heuristic solution approach. *Expert Systems with Applications*, **41**(15): 6959–6965.

- GIALLOMBARDO G, MOCCIA L, SALANI M & VACCA I. 2010. Modeling and solving the tactical berth allocation problem. *Transportation Research Part B: Methodological*, **44**(2): 232–245.
- HE J. 2016. Berth allocation and quay crane assignment in a container terminal for the trade-off between time-saving and energy-saving. *Advanced Engineering Informatics*, **30**(3): 390–405.
- HENDRIKS M, ARMBRUSTER D, LAUMANN S, LEFEBER E & UDDING J. 2012. Strategic allocation of cyclically calling vessels for multi-terminal container operators. *Flexible Services and Manufacturing Journal*, **24**(3): 248–273.
- HENDRIKS M, LEFEBER E & UDDING J. 2013. Simultaneous berth allocation and yard planning at tactical level. *OR spectrum*, **35**(2): 441–456.
- IMAI A, CHEN HC, NISHIMURA E & PAPADIMITRIOU S. 2008. The simultaneous berth and quay crane allocation problem. *Transportation Research Part E: Logistics and Transportation Review*, **44**(5): 900–920.
- IMAI A, NAGAIWA K & TAT CW. 1997. Efficient planning of berth allocation for container terminals in Asia. *Journal of Advanced Transportation*, **31**(1): 75–94.
- IMAI A, NISHIMURA E & PAPADIMITRIOU S. 2001. The dynamic berth allocation problem for a container port. *Transportation Research Part B: Methodological*, **35**(4): 401–417.
- IMAI A, NISHIMURA E & PAPADIMITRIOU S. 2003. Berth allocation with service priority. *Transportation Research Part B: Methodological*, **37**(5): 437–457.
- IRIS Ç, PACINO D & ROPKE S. 2017. Improved formulations and an adaptive large neighborhood search heuristic for the integrated berth allocation and quay crane assignment problem. *Transportation Research Part E: Logistics and Transportation Review*, **105**: 123–147.
- LEE DH & JIN JG. 2013. Feeder vessel management at container transshipment terminals. *Transportation Research Part E: Logistics and Transportation Review*, **49**(1): 201–216.
- MEISEL F & BIERWIRTH C. 2009. Heuristics for the integration of crane productivity in the berth allocation problem. *Transportation Research Part E: Logistics and Transportation Review*, **45**(1): 196–209.
- MOORTHY R & TEO CP. 2006. Berth management in container terminal: the template design problem. *OR spectrum*, **28**(4): 495–518.
- PARK YM & KIM KH. 2003. A scheduling method for berth and quay cranes. *OR spectrum*, **25**(1): 1–23.
- SHANG XT, CAO JX & REN J. 2016. A robust optimization approach to the integrated berth allocation and quay crane assignment problem. *Transportation Research Part E: Logistics and Transportation Review*, **94**: 44–65.

VACCA I, SALANI M & BIERLAIRE M. 2013. An exact algorithm for the integrated planning of berth allocation and quay crane assignment. *Transportation Science*, **47**(2): 148–161.

ZHEN L, CHEW EP & LEE LH. 2011. An integrated model for berth template and yard template planning in transshipment hubs. *Transportation Science*, **45**(4): 483–504.

ZHEN L, YU S, WANG S & SUN Z. 2019. Scheduling quay cranes and yard trucks for unloading operations in container ports. *Annals of Operations Research*, **273**(1): 455–478.

ZHENG F, LI Y, CHU F, LIU M & XU Y. 2019. Integrated berth allocation and quay crane assignment with maintenance activities. *International Journal of Production Research*, **57**(11): 3478–3503.

How to cite

CERESER BLH, OLIVEIRA ARL AND MORETTI AC. 2022. A Mathematical Model and Solution Method for the Berth Allocation Problem with Variable Handling Time and Continuous Time Horizon. *Pesquisa Operacional*, **42**: e261709. doi: 10.1590/0101-7438.2022.042.00261709.