Pesquisa Operacional (2021) 41: e239994 p.1-28
doi: 10.1590/0101-7438.2021.041.00239994

Printed version ISSN 0101-7438 / Online version ISSN 1678-5142
www.scielo.br/pope
ARTICLES

A MATHEMATICAL MODEL AND GENECTIC ALGORITHM SOLUTION METHODS FOR THE BERTH ALLOCATION PROBLEM WITH SEVERAL TYPES OF MACHINES

Bruno Luís Hönigmann Cereser[1*], Luiz Leduíno de Salles Neto[2],
Antonio Carlos Moretti[3] and Aurelio Ribeiro Leite de Oliveira[4]

Received June 23, 2020 / Accepted February 6, 2021

ABSTRACT. Maritime shipping is vital to worldwide commerce. Due to the high flow in ports throughout the world, the efficient allocation of vessels in berths has become a problem. A new mathematical model and several algorithms are proposed in this paper to planning the allocation of the vessels in berths and the allocation of resources to the service of each vessel. Those resources, in general, are machines to load or unload vessels. The mathematical model was implemented on Cplex and can solve small scale instances, due to its high complexity. To solve larger instances, a genetic algorithm-based metaheuristic, a first-in first-out heuristic, and a machine allocation algorithm are also proposed in this paper. The model and the algorithms produce very useful and interesting results. Comparing, the results produced by the GA are, on average, 94% better than the results of the Cplex and 26% better than the results of FIFO.

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

1 INTRODUCTION

More than 80% of world trade is carried out by maritime transportation, and the shipping of all these goods is only possible by loading them at least one port of origin and unloading them at least one port of destination. Due to the high volume of trade, it is common for a port to have a queue of vessels to be serviced, causing delays for all waiting vessels, which can cause losses to the port, to its clients, and even to the country.

*Corresponding author

[1] IMECC, Unicamp, Campinas, São Paulo, Brazil – E-mail: brunolhcereser@gmail.com – https://orcid.org/0000-0003-0753-9386

[2] ICT, Unifesp, São José dos Campos, São Paulo, Brazil – E-mail: luiz.leduino@unifesp.br – https://orcid.org/0000-0001-8938-5370

[3] IMECC, Unicamp, Campinas, São Paulo, Brazil – E-mail: moretti@ime.unicamp.br – https://orcid.org/0000-0001-5604-1002

[4] IMECC, Unicamp, Campinas, São Paulo, Brazil – E-mail: aurelio@ime.unicamp.br – https://orcid.org/0000-0002-6471-4710

Good planning can reduce the occurrences and the length of the queues on the terminals of the port. This planning should determine where and when each vessel will be serviced.

The duration of the service of the vessel depends on the number of resources allocated. Therefore any planning should also inform the number of resources that will be allocated to the service of each vessel. The resources, in general, are machines like quay cranes, unloaders, trucks, conveyor belts, among others. More than one type of resource can be necessary, like quay cranes and trucks or unloaders and conveyor belts, depending on the operations of the port.

In this paper we propose a new mathematical model to the berth allocation problem, considering berths with a predetermined length. The mathematical model has the objective to minimize the waiting time and the service time of the vessels. The waiting time is the difference between the arrival time and the allocation time of the vessel to begins its service, and the service time is the duration of the service. The premise used on the problem is that the service is made by a chain of machines, and the service time is determined by the level with the slowest rate of that chain.

The mathematical model was implemented and solved by the CPLEX and several instances were generated to validate the model and the algorithms. Despite the high complexity, the model was able to generate results in several small-scale instances. The small scale instances generated useful results and insights. A Genetic Algorithm and a fist-in first-out heuristic were proposed to solve medium and large-scale instances. The GA and the FIFO uses a heuristic to determine the resource allocation and the service time of each vessel.

Section 1 of this paper brings an introduction to the motivation and the main results of this work. Section 2 brings a literature review, with a survey of important results related to this work. Section 3 brings the description of the problem studied in this work. Section 4 brings the mathematical model proposed to the berth allocation model with several types of machines to servicing the vessel and a comparison of the proposed model with other models in the literature. Section 6 brings the description of the algorithms used to find solutions to the problem of this work. Section 5 brings the algorithms used to solve the proposed mathematical model. Section 6 brings an analysis of the results of several instances of the problem. Section 7 brings the conclusion.

2 LITERATURE REVIEW

Since the port terminal is getting busier caused by the intense worldwide trade growth, the efficiency of the port terminal is gaining importance and that makes the amount of works in this area significantly grow. One important issue related to the efficiency of terminals is the berth allocation problem, which consists in allocate the vessel that arrives at the port on berths along the quay.

One of the earliest models for the berth allocation problem (BAP) can be found in Imai et al. (1997) where the berths have fixed length and are considered discrete points along the quay. Their model assumes a situation they named static, where the vessels to be served on the time horizon are already in port, this model is referenced as static Discrete Berth Allocation Problem, or static BAPD, by the authors. They called Discrete referring to the fixed-length berth that generates discrete berths. This model aims to minimize the vessels mooring time to give a better solution than the first-in first-out (FIFO), also called first-in fist-served (FIFS), a strategy that is used in most commercial ports.

In Imai et al. (2001) they extend the BAP in Imai et al. (1997) to the dynamic version where all vessels that have an arrival time on the planning horizon are considered in the model, this model is referenced as dynamic BAPD by the authors. The BAPD model assumes that all vessels can be allocated in any berth, which is not usual in practice. They also develop a heuristic method based on a subgradient method with the Lagrangian relaxation to find solutions for the dynamic BAPD. Nishimura et al. (2001) and Xu et al. (2012) further extended the dynamic version of the BAPD limiting the berth where each vessel can moor and Nishimura et al. (2001) also solve the problem with a genetic algorithm (GA).

Cordeau et al. (2005) formulated the BAP as a Multi-Depot vehicle-Routing Problem with Time Windows (MDVRPTW) and developed two versions of a heuristic beased on tabu search. The heuristic has a version for the discrete and another one for the continuous BAP, the continuous version of BAP will be described further in this literature review. The results obtained by the heuristics were compared with the results obtained by the Cplex with the same instances in their work.

There is another approach for the BAP that consist in partitioning the quay in several parts of any length unit and every vessel have their length expressed in that same unit. This property is called continuous location, and the BAP model that uses this type of quay is referred to as Continuous Berth Allocation Problem (BAPC). In this type of model, the allocation of vessels is planned based on how many units of quay they need for mooring. In this approach, the berths lengths can vary with each service. If the model uses the two described kinds of berths, it is called hybrid. That type of BAP model is inspired by the cutting-stock problem, where the vessels are the commodities and the length of the quay (expressed in partitions) is the space available to pack the commodities. Lim (1998) is the first to proposed a BAP model with this approach. The model proposed by him considers that a vessel remains in the same position all service time, and his model aims to minimize the maximum quay space used at any time.

In Li et al. (1998) a vessel can be relocated during its service. Park & Kim (2003) propose a subgradient Lagrangian relaxation heuristic to solve the BAPC. The BAPC model proposed in Imai et al. (2005) have different service time for each position of the quay. They also develop a heuristic to solve the instances of the BAPC model proposed and, using a previous heuristic to solve the same instances of BAPD, they argue that the solutions obtained from the BAPC and BAPD models with the same instances have equivalent solutions. They also argue that real instances of BAP always can be solved using models that have discrete berths.

The last group of problems related to the BAP is the tactical berth allocation problem (TBAP). The first work to address this problem is Moorthy & Teo (2006) where their model aims to represent the trade-off between the waiting time for vessels and the cost of moving containers between berth and yard. They model this as a rectangle-packing problem on a cylinder and use a sequence pair based simulated annealing algorithm to solve the problem. The two objectives of their model are maximizing the service level (defined as the number of vessels served in two hours) and minimize the costs related to the movement of containers between berth and yard.

Giallombardo et al. (2010) proposed another TBAP model, which integrates the BAP with the quay crane schedule (QC). In such models, the variables and parameters associated with the vessel and berth are very similar to the BAPD except for a new index associated with the discretization of the time horizon in several time partitions, the model also has variables and parameters related to the QC allocation and schedule. The QC schedule assumes that relocation can occur every work shift change, so the time partition on this model is the length of the work shift time. Additionally to the cost associated with QCs, they consider the yard cost that depends on the berthing location in the objective function. This model is one of the first to consider the integration between the BAP and the port machine schedule, but the time partition usually causes a large increase in the number of variables and consequently demands more memory and processing time to solve instances of this model.

Vacca et al. (2013) developed an exact-solution algorithm based on branch and price technique for the TBAP of Giallombardo et al. (2010). Iris et al. (2017) focuses on the integrated berth allocation and quay crane assignment problem in container terminals based on a continuous berth, discretized in small equal-sized sections, they consider the decrease in the marginal productivity of quay cranes and the increase in handling time due to deviation from the desired position an Adaptive Large Neighborhood Search heuristic has proposed too. The parameters and variables, use a discretized time horizon.

Pereira et al. (2018) compare the simulated annealing (SA) and genetic algorithm (GA) metaheuristics applied to the berth allocation problem (BAP) of a port container terminal. They also compare the OX, CX, PMX, and HX crossovers operators that can be used in the GA.

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 BACAP version with time-invariant crane assignment, in this new model the vessels can be moored at any position on the quay, not requiring any quay discretization, but the time horizon is discretized.

For more bibliographical information related to BAP, Bierwirth & Meisel (2010) and Bierwirth & Meisel (2015) contain a large and detailed review. Imai (2015) also contains a good literature review.

The mathematical model proposed here is inspired by models described in Imai et al. (2001) and Cordeau et al. (2005) and has been developed to plan the allocation of the full chain of machines relevant to a vessel's servicing. To achieve this, the mathematical model does not make use of time intervals, just like hour-by-hour (Park & Kim (2003) and Meisel & Bierwirth (2009)) or turn-by-turn (Giallombardo et al. (2010) and Vacca et al. (2013)), but considering that the operation of machines on the vessel begins with the mooring of that vessel and ends with its departure. In this way, variables defined in time intervals are not needed; only allocation variables are needed for each vessel. Additionally, the model allocates machine-by-machine, allowing for the fact that machines of the same type may work at different rates.

Comparing our mathematical model with the mathematical model proposed in Imai et al. (2008), which aims to model a similar problem, some major differences are worth mentioning. Imai et al. (2008) works with quay crane machines, while the proposed model works with any type of machine. The number of quay cranes that should serve each vessel is fixed in Imai et al. (2008), while in our model the number of machines is variable, limited by a minimum and a maximum value for each machine type. The service time of each vessel needs to be provided as a parameter to their model meanwhile in the proposed model the service time is a variable that depends on how many machines are allocated to serve each vessel. Also, the mathematical model of Imai et al. (2008) is not optimized or implemented in any mixed-integer programming software. It is solved by a metaheuristic based on the genetic algorithm while our mathematical model is implemented in the IBM CPLEX and tested for several small instances and a specific heuristic and a genetic algorithm metaheuristic are provided to solve medium and large-scale instances.

Also, we cannot find a mathematical model for the BAP and QC allocation with continuous-time variables. All mathematical models found work discretizing the time horizon to define their parameters and variables.

3 PROBLEM DESCRIPTION

The problem consists of deciding the order, where, and the number of resources allocated to the service of each vessel to minimize the total waiting and handling time. If there are idle berths, the vessels can be allocated without waiting time. If every berth is handling vessels when a vessel arrives, that vessel needs to wait in a queue. The time horizon is the length of time considered on the problem. When the handle of a vessel is initiated, it needs to be performed until the end. That means every vessel is handled one time by one berth.

Each vessel has a maximum departure time and the servicing of each vessel needs to be completed before the departure time. The time between the arrival of a vessel and its maximum departure time is called the time window. The objective of the problem is to determine when and where each vessel is allocated to minimize the total waiting time plus the service time.

When a vessel is allocated to a berth, it needs to be handled by port machines such as quay cranes, drain machines, ship loaders, ship unloaders, trucks, conveyor belts, container handlers, tractors, among others. In many cases, more than one type of machine is needed to handle a vessel, for instance, a quay that load vessels with containers need quay cranes and trucks to load the vessels because the truck brings the container to the quay crane and the quay crane load the container on the vessel. Another example is a quay that unloads solid bulk, a vessel needs ship unloaders and conveyor belts to be unloaded. The ship unloader removes the solid bulk of the cargo hold of the vessel and puts it on the conveyor belt. We can also have a container handler to withdraw the containers of the trucks or tractors to stack the solid bulk in the cargo hold of the vessel.

We have a chain of machines to do the service of the vessel. Each level of the chain has several machines of the same type. The machines do not necessarily need to be in the berth area, but they need to be handling cargo of the same vessel, like trucks to move containers from the allocation berth of the vessel to the yard area. Each type of machine or each level of the machine chain will have some rate of operation based on the number of machines and the individual rate of that type of machine.

Also, if a level of the machine chain has a total rate lesser than the total rate of the remaining levels, that level will limit the rate of the entire operation. For instance, if we have two trucks and just one quay crane, where each truck can bring thirty containers in one hour and the quay crane can handle forty containers in one hour, therefore the rate of the entire operation can not be more than forty containers per hour. That is, we have the total operation rate given by the minimum operation rate of the chain of machines.

Each vessel has a maximum and a minimal number of machines of each type that can handle the vessel at the same time. The vessel can only moor when enough number of machines of each type is available to handle it. Each machine allocated to handle a vessel begins to operate when the vessels moor on the berth.

Some types of machines have restrictions of overlapping, as the quay cranes that move on the same conveyor belt. Therefore the order of some machines can not change. Giallombardo et al. (2010) approach a similar problem and use a discrete-time horizon to manage the allocation of quay cranes. In general, they use a discrete-time horizon in small units of some length of time and define the allocation variable of the quay cranes. When the quay crane is allocated to a vessel on some segment of time, the respective variable assumes the value one. We manage to model this problem with a continuous-time horizon as it is shown in Section 4.

4 MATHEMATICAL MODEL FOR A MULTI-MACHINE TERMINAL

The mathematical model for terminals that works with multiple types of machinery, and allocates vessels in berths and machines in vessels. The mathematical model was developed with a generics machine, type $\alpha \in P$ and can be used for any type of machine including quay cranes, trucks, drain machines, pipelines, container handlers, among others. Some types of machines could generate specific constraints.

Sets

- B: Set of berths. Index k.
- N: Set of vessels. Index i, i'.
- O: Set of order. Index j.
- P: Set of machine types. Index α.
- M_α: Set of α type machines, where $\alpha \in P$. Index c.

Variables

- $x_{i,j,k}$: Equals 1 if the vessel i is the j-th vessel in berth k, 0 otherwise.
- T_i: Allocation time of vessel i. Can also be described as the mooring time of vessel i.
- t_i: Duration of servicing of vessel i.
- $y^{\alpha}_{i,j,k,c}$: Equals 1 if the vessel i is the j-th vessel being serviced by machine c type α in berth k, otherwise 0.
- $R^{\alpha}_{i,c}$: Allocation time of the machine c type α in vessel i.
- $w^{\alpha}_{i,c}$: Operation time of the machine c type α on vessel i.

Parameters

- β and γ: Weight of T_i and t_i in the objective function, respectively.
- a_i and b_i: Arrival time and the maximum departure time of each vessel in the port, respectively.
- u_B: Last berth of the set B.
- u_{α}: Last machine of the M_{α} set of machine type α.
- m^{α}_i and n^{α}_i: Minimum and Maximum numbers of type α machines needed to serve vessel i, respectively.
- Q_i: Load of vessel i.
- r^{α}_c: Efficiency rate of machine c type α.
- H: Very large number, used in the modeling technique.

Mathematical Model

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

$$\text{s.t. } w^{\alpha}_{i,c} \geq t_i - H \left(1 - \sum_{j \in O} \sum_{k \in B} y^{\alpha}_{i,j,k,c} \right) \quad \forall i \in N, \forall c \in M_{\alpha}, \alpha \in P \tag{1b}$$

$$w^{\alpha}_{i,c} \leq H \left(\sum_{j \in O} \sum_{k \in B} y^{\alpha}_{i,j,k,c} \right) \quad \forall i \in N, \forall c \in M_{\alpha}, \alpha \in P \tag{1c}$$

$$\sum_{c \in M_{\alpha}} w^{\alpha}_{i,c} r^{\alpha}_c \geq Q_i \quad \forall i \in N, \alpha \in P \tag{1d}$$

$$w^{\alpha}_{i,c} \leq t_i \quad \forall i \in N, \forall c \in M_{\alpha}, \alpha \in P \tag{1e}$$

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

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

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

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

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

$$T_i + H(2 - x_{i,j,k} - x_{i',j-1,k}) \geq T_{i'} + t_{i'} \qquad \forall i \in N, \forall i' \in N, \forall j \in O \setminus \{1\}, \forall k \in B \tag{1k}$$

$$\sum_{j \in O} \sum_{k \in B} \sum_{c \in M_\alpha} y^{\alpha}_{i,j,k,c} \geq m^{\alpha}_{i} \quad \forall i \in N, \forall \alpha \in P \tag{1l}$$

$$\sum_{j \in O} \sum_{k \in B} \sum_{c \in M_\alpha} y^{\alpha}_{i,j,k,c} \leq n^{\alpha}_{i} \quad \forall i \in N, \forall \alpha \in P \tag{1m}$$

$$\sum_{i \in N} \sum_{k \in B} y^{\alpha}_{i,j,k,c} \leq 1 \quad \forall j \in O, c \in M_\alpha, \forall \alpha \in P \tag{1n}$$

$$\sum_{j \in O} y^{\alpha}_{i,j,k,c} \leq \sum_{j \in O} x_{i,j,k} \quad \forall i \in N, k \in B, \forall c \in M_\alpha, \forall \alpha \in P \tag{1o}$$

$$R^{\alpha}_{i,c} + H\left(2 - \sum_{k \in B} y^{\alpha}_{i,j,k,c} - \sum_{k \in B} y^{\alpha}_{i',j-1,k,c}\right) \geq T_{i'} + t_{i'} \qquad \forall i \in N, \forall i' \in N, \forall j \in O \setminus \{1\}, \forall c \in M_\alpha, \forall \alpha \in P \tag{1p}$$

$$R^{\alpha}_{i,c} \leq T_i + H\left(1 - \sum_{j \in O} \sum_{k \in B} y^{\alpha}_{i,j,k,c}\right) \quad \forall i \in N, \forall c \in M_\alpha, \forall \alpha \in P \tag{1q}$$

$$R^{\alpha}_{i,c} \geq T_i - H\left(1 - \sum_{j \in O} \sum_{k \in B} y^{\alpha}_{i,j,k,c}\right) \quad \forall i \in N, \forall c \in M_\alpha, \forall \alpha \in P \tag{1r}$$

$$\sum_{j \in O} y^{\alpha}_{i,j,1,c} \geq \sum_{j \in O} y^{\alpha}_{i,j,1,c+1} \quad \forall i \in N, c \in M_\alpha \setminus \{u_\alpha\}, \forall \alpha \in V \tag{1s}$$

$$\sum_{j \in O} y^{\alpha}_{i,j,u_B,c+1} \leq \sum_{j \in O} y^{\alpha}_{i,j,u_B,c} \quad \forall i \in N, c \in M_\alpha \setminus \{u_\alpha\}, \forall \alpha \in V \tag{1t}$$

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

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

$$t_i \geq 0 \quad \forall i \in N \tag{1w}$$

$$w^{\alpha}_{i,c} \geq 0 \quad \forall i \in N, \forall c \in M_\alpha, \forall \alpha \in P \tag{1x}$$

$$y^{\alpha}_{i,j,k,c} \in \{0,1\} \quad \forall i \in N, \forall j \in O, \forall k \in B, \forall c \in M_\alpha, \forall \alpha \in P \tag{1y}$$

$$R^{\alpha}_{i,c} \geq 0 \quad \forall i \in N, \forall c \in M_\alpha, \forall \alpha \in P. \tag{1z}$$

Where (1a) is the objective function that is the minimization of the waiting time plus the service time of all vessels. The parameters β and γ was added due to the different importance of waiting time and the service time may have. For example, if the service is given much importance, the solution tends to be allocating the maximum number of machines in few berths to rush the service of the vessels, while a high importance of the waiting time tends to occupy more berths at the same time. Constraints (1b), (1c) and (1d) represent the calculation of the servicing time for each type of machine. If the machine c is allocated to the vessel i then $\sum_{j\in O}\sum_{k\in B} y^{\alpha}_{i,j,k,c} = 1$ and the Constraints (1b) and (1c) became $w^{\alpha}_{i,c} \geq t_i$ and $w^{\alpha}_{i,c} \leq H$ respectively, which makes the operation time of the machine c equals the service time of the vessel. If the machine c is not allocated to the vessel i then $\sum_{j\in O}\sum_{k\in B} y^{\alpha}_{i,j,k,c} = 0$ and the Constraints (1b) and (1c) become $w^{\alpha}_{i,c} \geq -H$ and $w^{\alpha}_{i,c} \leq 0$ respectively, which does not interfere on the value of the service time of the machine c. Since the minimum number of machines of any type should be at least one, $w^{\alpha}_{i,c} > 0$ for at least one $c \in M_{\alpha}$, for all $\alpha \in P$.

The Constraints (1d) calculates the operation time generated by the machines of the same type. Constraints (1e) represent the calculation of the service time for each vessel. The Constraints (1e) combined with (1b), (1c) and (1d) makes the service time of the vessel equals to the slowest time generate by the allocated machines of a same type.

Constraints (1f) ensures that the docking time of each vessel is greater than its arrival hour. Constraints (1g) ensures that the vessel has completed its service before reaches its maximum departure time. Constraints (1h) ensures at most one vessel in each order position for each berth. Constraints (1i) ensures that each vessel is serviced once and in one berth. Constraints (1j) ensures the sequence of the order for each berth.

Constraints (1k) compute the mooring time of each vessel. If the vessel i is allocated immediately after vessel i' in berth k, then the Constraints (1k) is given by $T_i \geq T_p + t_p$. Otherwise, the Constraints (1k) is given by $T_i + H \geq T_p + t_p$ and the constraint is redundant. Constraints (1l) and (1m) ensures the minimum and maximum number of machines for each vessel. Constraints (1n) ensures a single order in the sequence of each machine. Constraints (1o) ensures that each machine serves a vessel at most once in one berth.

Constraints (1p), (1q) and (1r) compute the time spent by each machine on each vessel. If the machine c are allocated to the vessel j cames imediatly after vessel p then $\sum_{k\in B} y^{\alpha}_{i,j,k,c} = 1$ and $\sum_{k\in B} y^{\alpha}_{p,j-1,k,c} = 1$ and the Constraints (1p) is given by $R^{\alpha}_{i,c} \geq T_p + t_p$. Otherwise, the Constraints (1p) is given by $R^{\alpha}_{i,c} + H \geq T_p + t_p$ which is redundant. If the machine c is allocated to the vessel i then the Constraints (1q) and (1r) are given by $R^{\alpha}_{i,c} \leq T_i$ and $R^{\alpha}_{i,c} \geq T_i$, respectively, which makes $R^{\alpha}_{i,c} = T_i$. Otherwise the Constraints (1q) and (1r) are given by $R^{\alpha}_{i,c} \leq T_i + H$ and $R^{\alpha}_{i,c} \geq T_i - H$, respectively, which are redundant. Constraints (1s) and (1t) ensures no overlap of the machines, that constraint is not necessarily applied to every machine type.

Constraints (1u), (1v), (1w), (1x), (1y) and (1z) ensures the allocation variables to be binary and time variables to be non-negative reals.

The constraints (1s) and (1t) restrict this model to three or fewer berths sharing machines; with four or more berths, the constraint to ensure no overlap will be much more complicated to elaborate. However, this constraint can easily be contoured when using heuristics, metaheuristics, or even grouping three adjacent berths at a time.

4.1 Comparison with Some Existing Mathematical Models

As mentioned before, the mathematical model proposed in Imai et al. (2008) and the proposed model has some major differences that make the proposed model efficient and innovative.

The proposed model of Imai et al. (2008) aims to optimize the vessel allocation and the quay crane allocation scheduling. To do that they add constraints of the CAP to the BAP model of Imai et al. (2001). They named that model as B&CAP and the objective function is the minimization of the total service (or handling) time plus the waiting time of vessels. Each vessel has a parameter called F_i defined as the number of cranes needed by vessel i that means the service of each vessel needs a predetermined number of cranes. Our proposed model uses a variable $y^{\alpha}_{i,j,k,c}$ where the $\sum\limits_{j\in O}\sum\limits_{k\in B}\sum\limits_{c\in M_\alpha} y^{\alpha}_{i,j,k,c}$ represents the number of machine type α used to the service of the vessel i. That means the number of each machine type, including quay cranes, will be determined by the model and should be between a minimum, given by m^{α}_{i}, and a maximum, given by n^{α}_{i}, for each vessel and each machine type.

As the number of quay cranes of Imai et al. (2008) model for each vessel is invariant, they also have a parameter for the service duration of each vessel, given by $C_{I,k}$ defined as service (or handling) time of vessel i at berth k. Although seemingly the service time no longer depends on the allocation berth of the vessel, the authors do not specify any kind of change or the reason to maintain the berth index on this parameter. As the number of machines is a variable in our proposed model, the service time is also a variable defined as t_i the service time of vessel i that depends on how many machines of each type are allocated to the serve the vessel.

The quay crane schedule in Imai et al. (2008) is made by transfers between adjacent berths controlled by variable $m_{kk'ii'}$ define as the number of cranes transferred from berth $i^{'}$ after serving the $i^{'}$–th vessel to berth k (the next to berth $k^{'}$) after serving the i-th vessel. In our proposed model the machine schedule is controlled by $y^{\alpha}_{i,j,k,c}$, which is set to 1 if the vessel i is the j-th vessel being served by machine c type α in berth k and 0 otherwise.

Finally, the mathematical model of Imai et al. (2008) is not optimized or implemented in any mixed-integer programming software, it is solved by a metaheuristic based on the genetic algorithm. Our mathematical model is solved using the IBM CPLEX and tested for several small-scale instances. Also, a specific heuristic and a genetic algorithm-based metaheuristic are provided to solve medium and large-scale instances.

Another model that aims to optimize the vessel allocation and the quay crane allocation, scheduling is proposed in Giallombardo et al. (2010) where they propose a mixed-integer quadratic programming formulation for the berth allocation problem with quay crane allocation which they called tactical berth allocation problem. The proposed mathematical model is a mixed-integer linear programming formulation.

In their model, the quay crane machine is allowed and it does not allow any other machine type. The handling of the BAP variable and parameters are similar to the BAP of Imai et al. (2001), also similar to our proposed model. The quay crane schedule is made by a time step allocation variable defined as λ_i^p which is 1 if vessel i is served by the profile p, and 0 otherwise. Every vessel has a set of feasible quay crane profile. The time step used is the working turn shifts, which means the quay cranes can only reallocate at the end of a work shift (and before the next work shift), regardless of the allocation of the vessels.

Our proposed model considers that a machine works in the vessel until its service ends. With that assumption, we do not need the time step index on the time variables and parameters. The time-related variables are continuous and the machines are treated individually with the variable $y^{\alpha}_{i,j,k,c}$. The individual treatment of each machine brings some advantages such as a distinct work rate of each quay crane (or any machine).

5 ALGORITHMS TO SOLVE THE BERTH AND MACHINES ALLOCATION PROBLEM

To solve the mathematical model (1) we develop a First-In-First-Out heuristic and a genetic algorithm, both described in this section.

5.1 Genetic Algorithm

The genetic algorithm described in Algorithm (1) follows the steps and structure of Glover & Kochenberger (2003).

Algorithm 1 Genetic Algorithm to Allocate Vessels and Machines in Berths

```
Regulates the number of berths;
Definition of the initial population;
for i = 1,2,...,MaxIt do
    Generation of children based on OX permutation;
    Generation of children based on PMX permutation;
    Generation of Machine Allocation for each child;
    Calculation of the mooring and service time;
    Mutation;
    Selection of individuals that will compose the next population;
    Selection of the best individual;
end for
```

The algorithm (1) first verifies if the numbers of berths are consistent with the number of machines if the number of machines of a determined type is less than the number of berths, we need to reduce the number of berths to the same amount of the machine type in question. After that validation, the algorithm defines the initial population, which is generated by a heuristic that aims to achieve good individuals, and the rest is generated randomly. A detailed description of our GA elements is given in this section.

Solution Representation

The representation of a solution is an ordered vector with the numbers 1 to N representing the vessels and $|B|-1$ positions with zero representing the berth, where $|N|$ is the number of vessels and $|B|$ the number of berths. A position with zero on the vector indicates a change of berth. The Vector 2 shows an example.

$$[\underbrace{5\ \ 3\ \ 6\ \ 9}_{\text{Berth 1}}\ \ 0\ \ \underbrace{2\ \ 1\ \ 7}_{\text{Berth 2}}\ \ 0\ \ \underbrace{4\ \ 8\ \ 10}_{\text{Berth 3}}] \tag{2}$$

On the Vector 2 we have 3 berths and 10 vessels. The Berth 1 services the vessels 5, 3, 6, and 9 in that order. The Berth 2 services the vessels 2, 1, and 7 in that order, and the Berth 3 services the vessels 4, 8, and 10 in that order. Each solution has a machine allocation, a fitness function, a mooring time, and a service time associated. The tests use a population size of 25 since each solution requires a considerable amount of memory to be stored.

Machine Allocation, Mooring Time and Sevice Time

A machine allocation of a solution is a matrix $A : |P| \times |N|$, where $A_{i,j}$ is the number of machines type i allocated to serve the vessel j. The Matrix 3 shows an example of machine allocation.

$$A = \begin{bmatrix} 1 & 3 & 2 & 1 & 1 & 4 & 2 & 1 & 2 & 3 \\ 2 & 5 & 3 & 1 & 2 & 6 & 4 & 2 & 4 & 3 \end{bmatrix}. \tag{3}$$

Each column j of the Matrix 3 is the machine allocation of the vessel j, therefore, vessel 2 is served by 3 machines of type 1 and 5 machines of type 2. To generate the machine allocation for each vessel allocation we use the Heuristic given by the Algorithm 2.

The estimated service time of the vessels is calculated by equally divided the machines between the berths, if the number is not an integer, the number is regulated to an integer. With this machine allocation, we calculate the service time of each vessel, considering the berth allocated to its service. To allocate machines on Step 20 we use a mathematical model. Given a vessel schedule, first, we take a subset $V \in N$ of vessels that are serviced at the same time in different berths. We define the problem of finding the machine allocation and the service time of the vessels in V as a mathematical model described in Model (4).

Algorithm 2 Algorithm to Allocate Machines in a Vessel Schedule

```
$S_b$ is the vessel schedule of Berth $b \in B$;
$N$ is the set of Vessels;
$T_i; t_i; m_{i,\alpha}; z$ are the morring time, service time, machine number, objective function value, respectively. Where $i \in V$ and $\alpha \in P$.
$T_i^*; t_i^*; m_{i,\alpha}^*; z^*$ are the best morring time, service time, machine number, objective function value, respectively. Where $i \in V$ and $\alpha \in P$.
$BT_b$ is the opening time of berth $b \in B$;
$p_b \leftarrow 1$ for $b \in B$, is the position of the schedule of berth $b$;
$T \leftarrow \min_{b \in B}(a_{p_b} + t_{p_b})$, is the initial time reference;
$mB_{b,\alpha}$ is the number of machines type $\alpha$ in the berth $b$, where $b \in B$ and $\alpha \in P$;
$n \leftarrow 0$;
Initialize the values of $T_i^*; m_{i,\alpha}^*; z^*; mB_{b,\alpha}$ as zero;
$t_i \leftarrow$ estimate service time of the vessel $i \in N$;
while $n < |N|$ do
    for $b \in B$ do
        if $p_b > |S_b|$ then
            Remove the berth $b$ of set $B$;
        end if
    end for
    $I \leftarrow \{b : BT_b < T, b \in B\}$ ;
    $S \leftarrow \{S_b(p_b) : b \in I\}$;
    Find $T_i^*; t_i^*; m_{i,\alpha}^*; z^*$ for $i \in S$ and $\alpha \in P$ solving the Model (4);
    Ordered $I$ and $S$ according to the value of $BT$ in descendent order;
    for $i \in I$ do
        $T \leftarrow BT_i + t_{S(i)}$;
        $n_{S(i),\alpha} \leftarrow mB_{i,\alpha}$;
        Find $T_j; t_j; m_{j,\alpha}; z$ for $j \in S$ and $\alpha \in P$ solving the Model (4) with the new machine constraint ;
        if $z < z^*$ then
            $z^* \leftarrow z$;
            $BT_i \leftarrow T_{S(i)} + t_{S(i)}$; $mB_{i,\alpha} \leftarrow m_{S(i),\alpha}$ $T_j^* \leftarrow T_j$; $t_j^* \leftarrow t_j$; where $j \in S$ and $\alpha \in P$.
        end if
        Remove $S(i)$ from $S$;
    end for
    $T \leftarrow \max_{b \in B}\{T^*_{S_b(p_b)} + t^*_{S_b(p_b)}\}$;
    $BT_b \leftarrow T^*_{S_b(p_b)} + t^*_{S_b(p_b)}$ for all $b \in B$;
    $mB_b \leftarrow m^*_{S_b(p_b),\alpha}$ for all $b \in B, \alpha \in P$;
    $p_b \leftarrow p_b + 1$ for all $b \in I$;
    $n \leftarrow n + |I|$;
end while
```

$$z = min \sum_{i \in V} \sum_{i \in N} \beta T_i + \gamma t_i \tag{4a}$$

$$s.a.\ t_i \geq \sum_{\alpha \in M} \frac{Q_i}{r_\alpha m_{i,\alpha}} \qquad \forall i \in V \tag{4b}$$

$$T_i \geq BT \qquad \forall i \in V \tag{4c}$$

$$T_i \geq a_i \qquad \forall i \in V \tag{4d}$$

$$\sum_{k \in B} m_{i,\alpha} \leq M_\alpha \qquad \forall i \in V, \alpha \in M \tag{4e}$$

$$o_{i,\alpha} \leq m_{i,\alpha} \leq n_{i,\alpha} \qquad \forall i \in V, \alpha \in M \tag{4f}$$

$$m_{i,\alpha} \in \{0,1,2,3...\} \qquad \forall i \in V, \alpha \in M. \tag{4g}$$

where T_i is the mooring time of vessel i, t_i is the service time of the vessel i, BT is the opening time of the berths, a_i is the arrival time of vessel i, Q_i is the amount of cargo of the vessel i, r_α is the rate of a machine of type α, $m_{i,\alpha}$ is the quantity of α type machines allocated to serve the vessel i, M_α is the total amount of type α machines available, $o_{i,\alpha}$ is the minimum number of machines type α needed to serve the vessel i and $n_{i,\alpha}$ is the maximum number of machines type α that can serve the vessel i.

To show how the Algorithm 2 works, consider Figure 1.

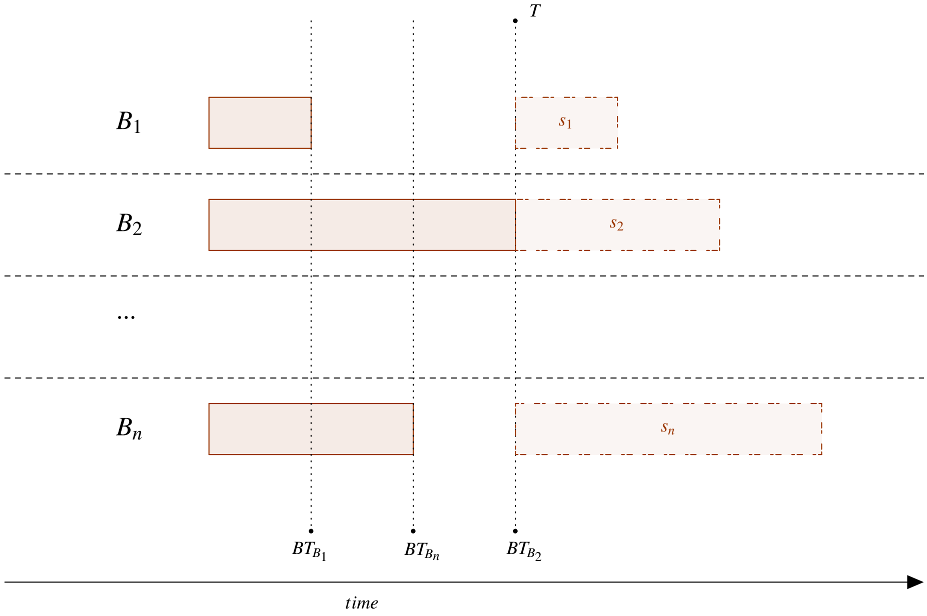


Figure 1 – Example of Heuristic 2 - Iteration 1.

The Figure 1 shows three berths, B_1, B_2 and B_n. Each rectangle represents a vessel and its length represents the service time. The solid rectangles are vessels treated in previous iterations and the dotted rectangles represent vessels treated in the actual iteration. Each berth became available in the time BT_{B_i}, which is given by the end of the service of the vessels of the previous iterations. The Step 18 of the Algorithm 2 results in $I = \{B_1, B_2, B_n\}$.

The optimization of Step 20 determines the values of $T_i^*; t_i^*; m_{i,\alpha}^*; z^*$. In this optimization all berths are available at the time $T = max_{b\in B} BT_b$, in this time position, every machine is available to work in any berth, therefore, the optimization consider the standard machine constrains, which are the limitation on machine number due to vessels and berths limitations. The set vessels treated in this iteration are $S = \{s_1, s_2, s_n\}$. On the Step 21 the set S is ordered by berth time BT_b in the descendant order, which is $S = \{s_2, s_n, s_1\}$. The Step 22 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 and it is removed from the set S, that is, the values of mooring time, service time and number of machines are given by $T_{B_2}^*, t_{B_2}^*$ and $m_{B_2,\alpha}^*$, respectively. On the next step the value of T returns to the previous position, which is $T = BT_{B_n}$. The Figure 2 shows the second iteration of the Loop 22.

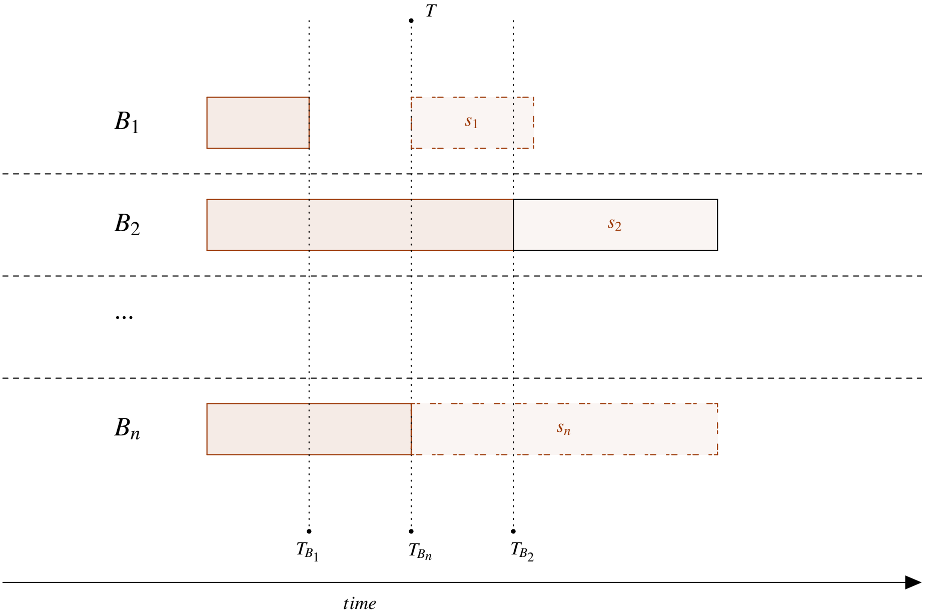


Figure 2 – Example of Heuristic 2 - Iteration 2.

On the second iteration of the Loop 22, we have $T = BT_{B_n}$ and $S = \{s_n, s_1\}$. Adding 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 $m_{s_1,\alpha} + m_{s_n,\alpha} \leq mB_{B_1,\alpha} + mB_{B_n,\alpha} + (|M_\alpha| - \sum_{b\in B} mB_b)$ for all machine type α. With that constraint and the initial berth time set to $BT = T_{B_n}$, the model of Step 20 is solved. If the objective function value z, is better than z^*, that is $z < z^*$ then $z^* \leftarrow z$, $T_{s_1}^* \leftarrow T_{s_1}$, $t_{s_1}^* \leftarrow t_{s_1}$, $m_{s_1,\alpha}^* \leftarrow m_{s_1,\alpha}$, $T_{s_n}^* \leftarrow T_{s_n}$, $t_{s_n}^* \leftarrow t_{s_n}$, $m_{s_n,\alpha}^* \leftarrow m_{s_n,\alpha}$, for all $\alpha \in P$. If $z \geq z^*$

or the problem is infeasible, nothing changes for $T^*_{s_1}, t^*_{s_1}, m^*_{s_1,\alpha}, T^*_{s_n}, t^*_{s_n}, m^*_{s_n,\alpha}$, for all $\alpha \in P$. The vessel s_n is fixed with the values of $T^*_{s_n}, t^*_{s_n}$ and $m^*_{s_n,\alpha}$, for all $\alpha \in P$. In our example, the value of z is greater than the value of z^*, therefore, the vessel s_n is fixed with $T_{s_n} = BT_{B_2}$, which can be seen in Figure 3.

In Iteration 3, $T = BT_{B_1}$ and $S = \{s_1\}$. The Figure 3 shows the beginning of the iteration 3.

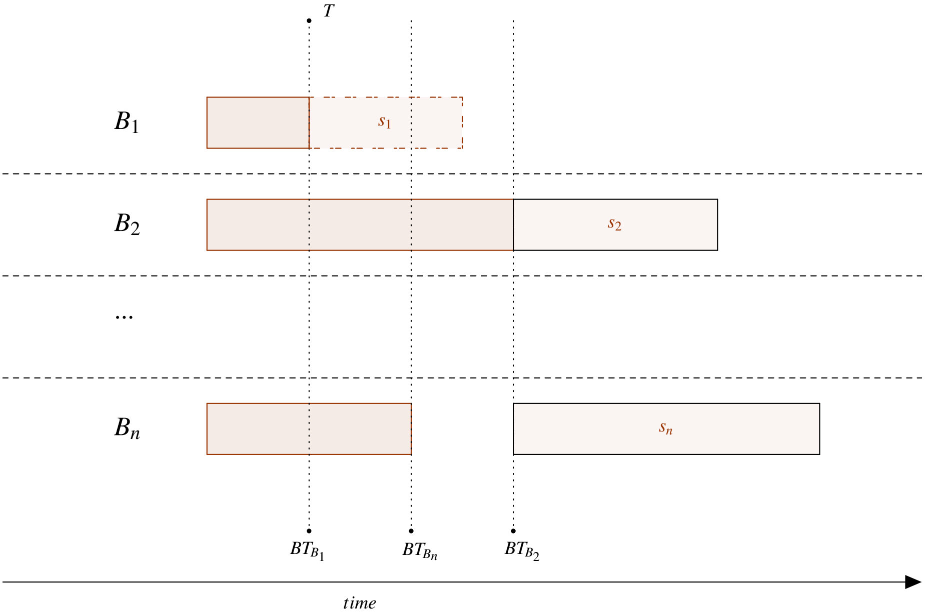


Figure 3 – Example of Heuristic 2 - Iteration 3.

Adding the constraint limiting the machines of vessel s_1 to the sum of machines in B_1 plus the idle machines, that is $m_{s_1,\alpha} \leq mB_{B_1,\alpha} + (|M_\alpha| - \sum_{b \in B} mB_b)$ for all machine type α. With that constraint and the initial berth time set to $BT = T_{B_1}$, the model of Step 20 is solved. In our example, $z < z^*$, therefore, $z^* \leftarrow z$, $T^*_{s_n} \leftarrow T_{s_n}$, $t^*_{s_n} \leftarrow t_{s_n}$, $m^*_{s_n,\alpha} \leftarrow m_{s_n,\alpha}$. The value T_{s_1} is fixed as $T_{s_1} = BT_{B_1}$, s_1 is remove from S and the Loop 22 ends. The Figure 4 shows the final allocation of s_1, s_2 and s_n.

After the allocation of the machines in all vessels 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 service time value can change as the number of machines changes. To simplify the Algorithm 2 some special cases tests were omitted, one example is in Step 18, if the vessels have sparse arrival time and tend to have low service time, the Step 18 can find an empty set before the vessels queue of all berths ends, in this case, the value $T = \min_{b \in B}(a_{p_b} + t_{p_b})$ will generate a nonempty set.

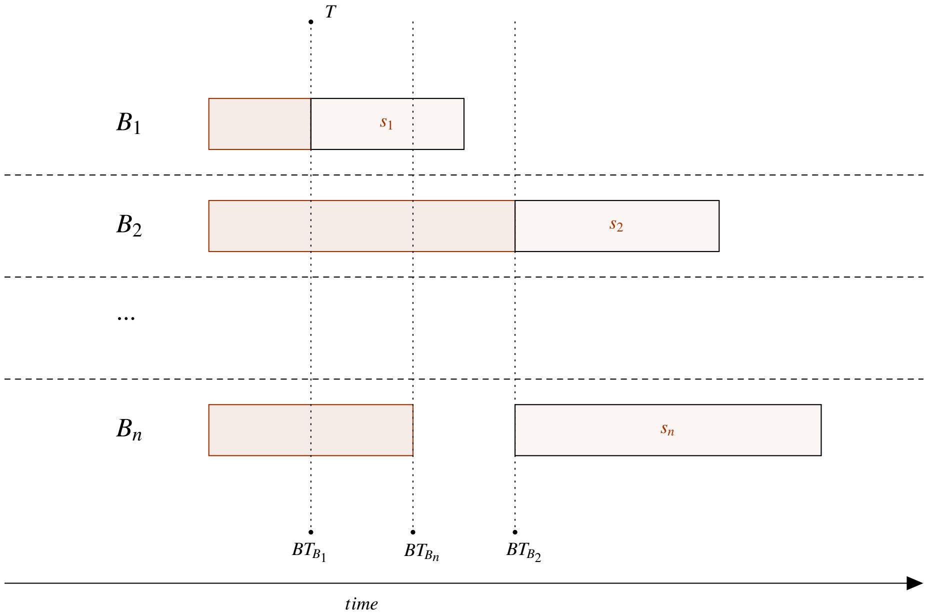


Figure 4 – Example of Heuristic 2 - Final.

Fitness Function

The value of the fitness of a solution is given by Equation 5.

$$fit_i = \frac{1}{\sum_{i \in N} \beta(T_i - a_i) + \gamma t_i + 1} \tag{5}$$

Where $fit_i \in (0,1]$, $\beta = 4$ and $\gamma = 1$.

Permutation

To improve the efficiency, two permutation heuristics are used, the *Partially-mapped Crossover* and the *Order Crossover*.

The permutation *Partial-Mapped Crossover* (PMX) can be described in three steps:

1. Randomly selects two points of the $Father_1$ and two points of $Father_2$.
2. Trade values between positions of $Father_1$ making the interior of the segment of the $Father_1$ equals the interior of the segment of the $Father_2$.
3. Exchange the segment of $Father_1$ with a segment of $Father_2$.

For example, consider the Solutions 6a and 6b. The segment is defined by positions 5 to 8.

$$Father_1 = \left[\begin{array}{cccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 2 & 1 & 5 & 4 & |7 & 8 & 9 & 3| & 6 & 10 \end{array}\right]. \tag{6a}$$

$$Father_2 = \left[\begin{array}{cccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 1 & 5 & 4 & 6 & |10 & 2 & 8 & 7| & 3 & 9 \end{array}\right]. \tag{6b}$$

To make the segment of $Father_1$ equal to the segment of $Father_2$, we exchange the value of position 7 with the value of position 1 and the value of position 8 with the value of position 10. The Solutions 7b and 7a shows the result of the exchanges.

$$Father_1 = \left[\begin{array}{cccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 9 & 1 & 5 & 4 & |7 & 8 & 2 & 10| & 6 & 3 \end{array}\right]. \tag{7a}$$

$$Father_2 = \left[\begin{array}{cccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 1 & 5 & 4 & 6 & |10 & 2 & 8 & 7| & 3 & 9 \end{array}\right]. \tag{7b}$$

With equal values in the segment interior, we exchange the segments to generate new solutions. The Solutions 7c and 7d shows the final solutions generated by PMX.

$$Son_1 = \left[\begin{array}{cccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 9 & 1 & 5 & 4 & |10 & 2 & 8 & 7| & 6 & 3 \end{array}\right]. \tag{7c}$$

$$Son_2 = \left[\begin{array}{cccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 1 & 5 & 4 & 6 & |7 & 8 & 2 & 10| & 3 & 9 \end{array}\right]. \tag{7d}$$

The permutation *Order Crossover* (OX) can be summarized in three steps:

1. Randomly selects two points to generate a segment in each Father.
2. Separate the segments of $Father_1$ and $Father_2$ to apply them on Son_1 and Son_2 respect.
3. Complete the Son_1 with elements of $Father_2$ which are not in Son_1, in the order, they appear in the $Father_2$. Do the same with Son_2.

For example, using the Solutions 6a and 6b and the segment defined by the position 5 to 8. We separate the segments of Solutions 6a and 6b. The Solutions 8a and 8b shows the segments.

$$Son_1 = \left[\begin{array}{cccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ & & & & |7 & 8 & 9 & 3| & & \end{array}\right]. \tag{8a}$$

$$Son_2 = \begin{array}{c} \begin{array}{cccccccccc} 1&2&3&4&5&6&7&8&9&10 \end{array} \\ \left[\begin{array}{cccccccccc} &&&& |10 & 2 & 8 & 7| && \end{array}\right] \end{array}. \tag{8b}$$

The missing values of 8a will be filled by values of 6b in the order that they appear and the missing values of 8b will be filled by values of 6a in the order that they appear. The solutions 8c and 8d shows the final solutions of OX.

$$Son_1 = \begin{array}{c} \begin{array}{cccccccccc} 1&2&3&4&5&6&7&8&9&10 \end{array} \\ \left[\begin{array}{cccccccccc} 1 & 5 & 4 & 6 & |7 & 8 & 9 & 3| & 10 & 2 \end{array}\right] \end{array}. \tag{8c}$$

$$Son_2 = \begin{array}{c} \begin{array}{cccccccccc} 1&2&3&4&5&6&7&8&9&10 \end{array} \\ \left[\begin{array}{cccccccccc} 9 & 1 & 5 & 4 & |10 & 2 & 8 & 7| & 6 & 3 \end{array}\right] \end{array}. \tag{8d}$$

Mutation

The mutation is the swap between two positions of a solution.

Selection

We use two types of selection, a portion of the solutions are selected by Tournament with a size of two, and the remaining by Elitism.

5.2 FIFO strategy

We compare the results of our model with the first-in-first-out strategy. The FIFO strategy consists of equally divided the machines between the berths, if the number is not an integer, the number is regulated to an integer by round up on the firsts berths until the division of the remaining machines on the remaining berths is an integer. With this machine allocation, we can create a matrix $|B| \times |N|$ with the service time of each vessel on each berth.

The vessels are ordered according to the arrival time and are allocated on the first available berth. Therefore the FIFO algorithm first ordered the vessel vector by arrival time, then go through this vector allocating the vessel in the first available berth, updating the berth time and the vessel parameters.s

6 NUMERICAL TESTS

To validate the proposed mathematical model, we implemented it in the IBM OPL^{TM} software, which has a Cplex library capable of solving linear integer problems. The instances tested here are small, medium, and large-scale. The Cplex was capable of solving only the small-scale instances, in the large and medium-scale instances, the Cplex does not even find a feasible solution. In those cases, the GA metaheuristic and a FIFO heuristic were used.

The data of each instance are randomly generated, a control is included to generate a nontrivial instance. The small scale instances have the arrival time of each vessel between 1 and 10, the medium and large scale instances have the arrival time of each vessel between 1 and 100. The load is a number between 1 and 100,000. The maximum number of machines that can operate in a berth is a number between 1 and 5. The rate of each machine type is a number between 3000 and 9000. Every instance is nontrivial, which means, the best solution is not as allocated every machine in a single berth and service the vessels as they arrive. We use $\beta = 4$ and $\gamma = 1$.

All test results were carefully analyzed and no inconsistencies were found. The data, the result, and codes of each test instance on the Cplex, GA, and FIFO algorithms can be found at Cereser (2018). Each instance was optimized on the GA metaheuristic only once and the result was collected.

The Table 1 brings the results of some of those instances. The name of each instance is given by *[number of berths]* + B + *[number of vessels]* + N +*[quantity of machine Type 1*] + *[quantity of machine Type 2*] + *[quantity of machine Type 3*].

In the Table 1 the column **Instance** is the name of the instance, **Cplex** is the result reached by the Cplex algorithm, **GAP** is the percentage gap of the Cplex solution, **Cplex Time** is the computational time of the Cplex algorithm, **FIFO** is the result achieved by the FIFO algorithm, **GA** is the result of a single run of the Genetic Algorithm (2), **GA Time** is the computational time of the Genetic Algorithm, **GA / Cplex** is the comparison between the GA and the Cplex solutions, given by $\frac{(GA\ solution - Cplex\ solution)}{GA\ solution}$ and **GA / FIFO** is the comparison between the GA and the FIFO solutions given by $\frac{(GA\ solution - FIFO\ solution)}{GA\ solution}$. The lower values of **GA / Cplex** and **GA / FIFO** are better to the GA.

The small-scale instances with three or fewer berths have a maximum optimization time of 900 seconds (or 15 minutes) in the Cplex, the GA, and the FIFO. In some instances, the post-optimization processes make the computational time greater than 900 seconds. The FIFO time was omitted since the algorithm took less than one second on small scale instances and up to two seconds on the largest case of the table.

Most of the Cplex solutions appear to be very early feasible solutions, which are indicated by the reach of the maximum optimization time, the high gap, and can be transformed into solutions with better objective function with some swaps between vessel positions. In cases with more than three berths, the Cplex was unable to find a feasible solution with more than one hour of optimization time. The genetic algorithm was executed a single time per instance and found solutions equal or better than the Cplex in all instances of Table (1). More than a hundred instances were generated to test the Mathematical Model, the GA, and the FIFO algorithm, and only in a few small scale instances, which are close to trivial, the Cplex was able to find a better solution than the GA.

Table 1 – Tests and results.

Instance	Cplex	GAP	Cplex Time	FIFO	GA	GA Time	GA / Cplex	GA / FIFO
2B7N73	227,1	87%	900	231,8	204,4	338	-11%	-13%
2B8N33	611,4	88%	900	599,8	426,6	426	-43%	-41%
2B8N55	218,8	89%	900	161,8	99,2	335	-120%	-63%
2B8N64	288,7	90%	906	168,0	168,0	295	-72%	0%
2B10N46	394,9	94%	904	147,8	138,1	484	-186%	-7%
2B4N432	131,7	0%	76	138,7	131,7	215	0%	-5%
2B6N628	221,0	0%	183	282,9	221,0	562	0%	-28%
2B7N254	168,0	82%	900	230,1	158,9	290	-6%	-45%
2B7N256	509,2	0%	678	657,5	509,2	371	0%	-29%
2B7N346	133,0	74%	900	124,3	85,4	434	-56%	-46%
2B8N237	73,0	0%	266	74,6	73,0	492	0%	-2%
2B8N428	884,8	86%	901	742,5	592,5	565	-49%	-25%
2B9N476	721,2	94%	901	319,3	223,6	707	-223%	-43%
2B9N815	1319,3	96%	916	804,8	505,1	1118	-161%	-59%
3B7N37	100,9	82%	911	81,0	67,2	438	-50%	-21%
3B7N43	156,6	54%	905	182,6	136,4	448	-15%	-34%
3B7N47	266,7	92%	903	112,3	93,7	92	-184%	-20%
3B7N85	123,1	84%	901	76,2	73,7	277	-67%	-3%
3B8N55	317,8	91%	902	151,7	108,0	520	-194%	-39%
3B9N33	583,6	97%	902	163,7	163,7	307	-256%	0%
3B9N34	866,2	94%	902	373,9	372,6	421	-132%	0%
3B10N33	618,5	95%	909	223,1	193,6	498	-219%	-15%
3B6N457	198,8	79%	911	171,8	137,4	273	-45%	-25%
3B6N536	89,8	81%	904	61,3	46,8	281	-92%	-31%
3B7N645	338,1	94%	905	125,1	125,1	350	-170%	0%
3B8N735	502,4	90%	917	288,6	288,4	296	-74%	0%
7B30N30	-	-	-	8864,1	5746,7	1838	-	-54%
16B50N47	-	-	-	628,2	620,8	2295	-	-1%
20B60N54	-	-	-	801,3	774,8	2727	-	-3%
30B100N6745	-	-	-	654,4	293,6	1457	-	-123%
30B100N75	-	-	-	6045,8	5624,7	2241	-	-7%
30B200N6099	-	-	-	602,0	602,0	1586	-	0%
50B200N125105	-	-	-	24314,0	23058,0	6916	-	-5%
60B300N150120	-	-	-	1057,9	631,5	2288	-	-68%
125B600N300250	-	-	-	2080,8	1192,8	4533	-	-74%

The GA and the FIFO aim to distribute the vessels between all berths, which tends to generate good results. The solutions found by GA are at least equal to the FIFO solutions, since some individuals of the initial population are generated by the FIFO algorithm but, in most instances, the GA was able to found a better solution. The GA solutions are up to 250% better than the Cplex solutions and up to 120% better than the FIFO solutions. The average gain of GA is around 94% over Cplex and 26% over FIFO.

The instances with more than three berths are made based on the dimensions of real terminals. In those instances, the average save of the GA is 37% when compared to the FIFO strategy.

The largest statistics of a port that we could found have approximately 125 berths, 600 vessels, and 300 machines, which is the base of the largest instance of this paper. The GA was able to find a solution that is 74% better than the FIFO solution in approximately one hour and fifteen minutes.

Analysis of a Solution

To present a solution to the GA, we will analyze the results of *3B8N55*. The Tables 2 and 3 brings the input data of the instance.

Table 2 – $3B8N55$ General and Machines Data.

Number of Vessels	8
Number of Berths	3
Large Number	300
Machine Type	2
Rate of Machine 1	4656,5
Number of Machines 1	5
Minimum Machine 1 per Vessel	1
Maximum Machine 1 per Vessel	3
Rate of Machine 2	7984,25
Number of Machines 2	5
Minimum Machine 2 per Vessel	1
Maximum Machine 2 per Vessel	4

In this instance, every vessel has the same limitation of minimum and maximum machines. But, in some cases, that can change due to the length of the vessel, the length of the berth, lack of space, among others. A deadline time of 100 in this instance is very large, which increases the number of possible allocations. To estimate the service time of each vessel, as described in Section 5, the number of machines is equally divided between the berths, in this case, is 1.6 of each. To make the value more accurate, we can round up the value on the first berths until the remaining number of machines are multiple of the remaining number of berths. This leads to the machine numbers of 2, 2, and 1, respectively from berth 1 to 3.

Table 3 – $3B8N55$ Vessel Data.

Vessel	Arrival Time	Dead Line Time	Load
s_1	9	100	15312
s_2	7	100	20095
s_3	7	100	91641
s_4	2	100	8856
s_5	8	100	21530
s_6	8	100	97494
s_7	3	100	72479
s_8	6	100	97728

The rate generated by machine 1 on berths 1 and 2 are 9313, and the rate generated by machine 2 is 15968.5, from that, the estimated rate of service on berths 1 and 2 are 9313 units of load per unit of time. In the same way, the estimated rate of service on berths 3 is 4656.5. Therefore, the estimated service time of vessel s_1 on berths 1 and 2 is 1.64, and in berth 3 is 3.29. Those values are used to generate the group of vessels of an iteration, described in Section 5.

The Table (4) shows the GA solution found by Algorithm (3).

Table 4 – AG Solution of $3B8N55$.

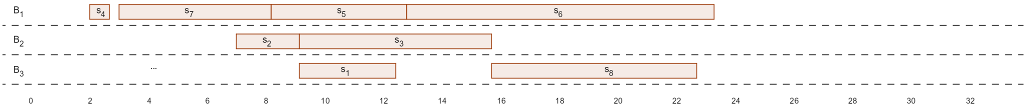
Berth 1	Vessel	s_4	s_7	s_5	s_6
	Time	2.0	3.0	8.2	12.8
	Service	0.6	5.2	4.6	10.5
	Machine 1	3	3	1	2
	Machine 2	4	3	1	3

Berth 2	Vessel	s_2	s_3
	Time	7.0	9.2
	Service	2.2	6.6
	Machine 1	2	3
	Machine 2	2	2

Berth 3	Vessel	s_1	s_8
	Time	9.2	15.8
	Service	3.2	7.0
	Machine 1	1	3
	Machine 2	2	2

Figure 5 was generate with the data of the Table 4.

Figure 5 – Representation of AG Solution of $3B8N55$



The vessel allocation found by the GA has an objective function of 108.0 and a better occupation rate of berths and machines. Four vessels are allocated on Berth 1, two on Berth 2 e two on Berth 3.

- At the time 2, the vessel s_4 arrives and it is immediately allocated in Berth 1 with the maximum amount of machines, which generates a service time of 0.6.

- At the time 3, the vessel s_7 arrives and it is immediately allocated in Berth 1 with the three machines of each type which generates a service time of 5.2.

- At the time 6, the vessel s_8 arrives, at this time the berths 2 and 3 are idle. The vessel s_8 is the first to go to the queue.

- At the time 7, the vessels s_2 and s_3 arrives, the berths 2 and 3 are idle and the vessel s_2 is immediately allocated in Berth 2 with two machines of each type. The vessel s_3 is sent to the queue, which is given by $\{s_8, s_3\}$.

- At the time 8, the vessels s_5 and s_6 arrives. Both of them are sent to the queue, which is given by $\{s_8, s_3, s_5, s_6\}$.

- At the time 8.2, the Berth 1 ends the service of the vessel s_7 and the vessel s_5 is allocated in Berth 1 with one machine of each type, generating a service time of 6.6. The queue is given by $\{s_8, s_3, s_6\}$.

- At the time 9, the vessels s_1 arrives and is sent to the queue, which is given by $\{s_8, s_3, s_6, s_1\}$.

- At the time 9.2, the Berth 2 ends the service of the vessel s_2 and the vessel s_3 is allocated in Berth 2 with 2 machines of each type, and the vessel s_1 is allocated in the Berth 3 with one machine type 1 and two machines type 2. The queue is given by $\{s_8, s_6\}$.

- At the time 12.4, the Berth 3 ends the service of the vessel s_1.

- At the time 12.8, the Berth 1 ends the service the vessel s_5, and the vessel s_6 is allocated with 2 machines of type 1 and 3 machines of type 2, which generates a service time of 10.5. The queue is given by $\{s_8\}$. At this time, every machine is occupied, which forces berth 3 to be idle with a nonempty queue of vessels.

- At the time 15.8, the Berth 2 ends the service of the vessel s_3, and the vessel s_8 are allocated on Berth 3, with three machines type 1 and two machines type 2. The queue is empty.

- At the time 22.8, the service of vessel s_8 ends.
- At the time 23.3, the service of vessel s_6 ends and the process is complete.

If the vessel s_8 was allocated in Berth 3 at 12.8, then it will be served with one and two machines, respectively, which generates a service time of 20. The vessel s_6 has the same allocation time, but with one machine of each type, which generates a service time of 20.9. Despite the lower allocation time of the vessel s_8, this solution has an objective function of 119.8, which is worst than the actual solution.

7 CONCLUSIONS

The model and the algorithms are designed to allocate vessels in berths and resources to service vessels. Most of the models of the literature do not allocate resources to determine the service time of each vessel and some models only allocate specific machines like quay cranes, normally not integrated with the allocation of vessels in berths.

The mathematical model was successfully validated by the instances and is effective to allocate vessels in berths and machines in vessels. Due to the complexity of the mathematical model, it was necessary to develop algorithms to solve larger instances. The algorithms mainly consist of a genetic algorithm, a first in first out heuristic, and a heuristic to allocate machines. The model and the algorithms are very sophisticated and produce important results and insights for the problem.

In many instances, one berth with many machines is used to service vessels with low cargo, while berths with fewer machines were servicing vessels with more cargo, which is very interesting since seems to be more natural to allocate more machines in vessels with more cargo and fewer machines in vessels with less cargo. Another important result is the existence of a commitment between early allocation in berths or waits for more idle machines to the service, which could generate some, generally small, waiting times between services. There has also a tendency to allocate vessels with less cargo first and vessels with more cargo last, which is consistent since those vessels have a longer service time, and allocating it first will delay the allocation of the remaining vessels and, in general, will generate a worst objective function.

The Genetic Algorithm proposed here, was run only once per instance, and the solutions found are at least equal to the FIFO solutions but, in most instances, the GA was able to found a better solution. The GA solutions are up to 250% better than the Cplex solutions and up to 120% better than the FIFO solutions. The average gain of GA is around 94% over Cplex and 26% over FIFO. Multiple runs of the GA could lead to betters solutions.

The Genetic Algorithm allows solving larger-scale instances, with more than 125 berths, 600 vessels, and several machines. Instances of this size are close to the largest ports in the world. The mathematical model with the genetic algorithm could be used for the strategic planning of the service of the vessels on terminals. It can be used to assist decisions like the acquisitions of a new machine, the build of new berths or to reduce the occurrence of late fees since the algorithm can be used to estimate the allocation time and the duration of the service of the vessels. The gain of the largest cases presented here is around 20% and can reach 55% compared to the FIFO strategy, which is used on some terminals. Future work is to implement the model and the algorithms in real terminals to evaluate their practical value.

References

[1] 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.

[2] 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.

[3] CERESER BLH. 2018. A Mathematical Model for Optimizing Port Terminals with Several Types of Machines (Tests and Codes). Available at: https://data.mendeley.com/datasets/5226rtjtwc.

[4] 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.

[5] 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.

[6] 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.

[7] GLOVER F & KOCHENBERGER GA. 2003. *Handbook of metaheuristics*. Springer Science & Business Media.

[8] IMAI A. 2015. An Extension of the Berth Allocation Problem to the Berth. *Journal of Maritime Researches Vol*, **35**: 59.

[9] 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.

[10] 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.

[11] 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.

[12] Imai A, Sun X, Nishimura E & Papadimitriou S. 2005. Berth allocation in a container port: using a continuous location space approach. *Transportation Research Part B: Methodological*, **39**(3): 199–221.

[13] 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.

[14] Li CL, Cai X & Lee CY. 1998. Scheduling with multiple-job-on-one-processor pattern. *IIE transactions*, **30**(5): 433–445.

[15] Lim A. 1998. The berth planning problem1. *Operations research letters*, **22**(2-3): 105–110.

[16] 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.

[17] Moorthy R & Teo CP. 2006. Berth management in container terminal: the template design problem. *OR spectrum*, **28**(4): 495–518.

[18] Nishimura E, Imai A & Papadimitriou S. 2001. Berth allocation planning in the public berth system by genetic algorithms. *European Journal of Operational Research*, **131**(2): 282–292.

[19] Park YM & Kim KH. 2003. A scheduling method for berth and quay cranes. *OR spectrum*, **25**(1): 1–23.

[20] Pereira ED, Coelho AS, Longaray AA, Machado CMdS & Munhoz PR. 2018. Metaheuristic analysis applied to the berth allocation problem: case study in a port container terminal. *Pesquisa Operacional*, **38**(2): 247–272.

[21] 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.

[22] Xu D, Li CL & Leung JYT. 2012. Berth allocation with time-dependent physical limitations on vessels. *European Journal of Operational Research*, **216**(1): 47–56.

How to cite

CERESER BLH, SALLES NETO LL, MORETTI AC & OLIVEIRA ARL. 2021. A Mathematical Model and Genectic Algorithm Solution Methods for the Berth Allocation Problem with Several Types of Machines. *Pesquisa Operacional*, **41**: e239994. doi: 10.1590/0101-7438.2021.041.00239994.