New exact methods for the time-invariant berth allocation and quay crane assignment problem

Juan F. Correcher[a,*], Ramon Alvarez-Valdes[a], Jose M. Tamarit[a]

[a] *University of Valencia, Department of Statistics and Operations Research, Doctor Moliner 50, 46100 Burjassot, Valencia, Spain*

Abstract

Efficient management of operations in seaport container terminals has become a critical issue, due to the increase in maritime traffic and the strong competition between ports. In this paper we focus on two seaside operational problems: the Berth Allocation Problem and the Quay Crane Assignment Problem, which are considered in an integrated way. For the continuous BACAP problem with time-invariant crane assignment we propose a new mixed integer linear model in which the vessels can be moored at any position on the quay, not requiring any quay discretization. The model is enhanced by adding several families of valid inequalities. The resulting model is able to solve instances with up to 50 vessels and outperforms other recently published proposals. In a second part, the model is extended to include the assignment of specific cranes to each vessel: the BACASP. This assignment ensures that the handling of each vessel can be carried out without disruptions, thus producing solutions that can be applied in practice. We also propose an iterative procedure for the BACASP in which the BACAP model is solved, and whenever its solution is not feasible for the BACASP, specific constraints are added until an optimal solution for the BACASP is found. Additionally, a branch-and-cut algorithm is proposed based on the cuts used in the iterative procedure. The computational study on several classes of test instances shows that problems with up to 40 vessels can be solved to optimality.

Keywords: Combinatorial optimization, Container terminal, Berth allocation, Quay crane assignment, Integer programming.

1. Introduction

Seaports are a key factor in maritime commerce and the global market economy. Most manufactured goods are carried in standardized containers, which are handled by specialized facilities implemented in ports all over the world. This kind of transportation provides great advantages: protection for goods, standardization, easy interchange between different transportation modes, and consequently cost reduction and increased productivity. In 2016, the total volume of containerized trade exceeded 140 million TEUs (Twenty-foot Equivalent Units), and the most important ports, such as Shanghai and Singapore, handled more than 30 million TEUs. Overall, in the decade 2007–2016, the increase in containerized trade volume was more than 40% (UNCTAD, 2017).

[*] Corresponding author.

Email addresses: juan.correcher@uv.es (Juan F. Correcher), ramon.alvarez@uv.es (Ramon Alvarez-Valdes), jose.tamarit@uv.es (Jose M. Tamarit)

The specialized facilities dedicated to these tasks are container terminals. They consist of three main areas: seaside, yard, and landside. In the first area, container ships are moored at the quay and quay cranes perform the loading and unloading operations. Then vehicles move containers to and from the yard, which is used as a buffer between ships and land modes of transportation such as trains or trucks. Finally, the third area is set up to make this interchange effective.

Marine container terminals compete to offer better conditions to their clients, and this involves improving the efficiency of their operations. Waiting times of vessels, terminal operational costs, contractual penalties, and other costs depend on the efficient management of such operations. Their number and complexity is so large that an integrated mathematical programming approach has nowadays become impracticable. Hence, they are commonly grouped by the aforementioned areas (landside, yard, and seaside operations) and tackled separately. General literature overviews of container terminal operations were presented by Stahlbock & Voß (2008), Rashidi & Tsang (2013), and Li et al. (2015).

In the seaside area, the most important operational problem is the Berth Allocation Problem (BAP). It concerns the assignment of a berthing position and a berthing time to every vessel projected to be served within a given planning horizon. In this problem, the handling times of vessels are usually assumed to be fixed and known in advance. In addition to quay space, quay cranes (QCs) are also a scarce resource. Whenever several vessels moor simultaneously at the quay, a Quay Crane Assignment Problem (QCAP) arises. The number of QCs serving a vessel simultaneously is often restricted within a minimum and a maximum number, for technical and contractual reasons.

There is an increasing trend to consider these two problems together, because the number of quay cranes assigned to a vessel determines its handling time. In the combined Berth Allocation and quay Crane Assignment Problem (BACAP), as well as the time and berthing position, a number of cranes is also assigned to each vessel.

Two versions of the BACAP have been tackled in the literature. In the *time-invariant* version, the number of cranes assigned to each vessel remains constant throughout its handling, while in the *variable-in time* version this number can be changed in each period. The variable-in-time version allows a more efficient use of cranes, as those initially assigned to a vessel can be reassigned to newly arrived vessels. This is useful, for example, when the processing of a large vessel is nearly finished and only a few cranes are needed to complete it. The outer cranes may be reassigned to other vessels, thereby accelerating their handling. Terminal operating systems try to achieve this variable-in-time assignment whenever possible; however, this advantage can turn into a big problem when applying the plan to real situations. The solutions may entail more complex crane-to-vessel assignments and thus become more difficult for human operators to handle. Moreover, they can also result in a greater number of crane movements, thereby demanding a more effective and efficient management of operations at the docks, including the organization of workers and machines. As a consequence, variable-in-time plans are more sensitive to contingencies and elements not usually considered in models and thus the real execution may rapidly deviate from the theoretical schedule. In this respect, an especially relevant case is the common assumption that the time required by cranes when moving from one vessel to another is negligible. A large number of movements substantially increases the time lost in setup operations and makes the optimal variable-in-time schedule less and less realistic as it is implemented. For this reason and to prevent other unrealistic schedules, the number of crane movements must be minimized in these models, taking them into account in the objective function. In contrast, time-invariant models may accept several estimates of the handling time for each vessel considering various numbers of cranes and their corresponding setup times. Moreover, as the assignment of each specific crane has to be carried out for each vessel and not for each time period, they lead to solutions with fewer crane movements, so in general the resulting schedules may keep closer to their real execution and thus become more trustworthy in themselves. For the same reason, time-invariant models require fewer variables, which makes these models computationally easier to handle, and therefore the size of the instances that they can solve to optimality is usually larger than in the case of variable-in-time models. From a theoretical perspective, the main disadvantage of time-invariant models is that they nearly always waste crane capacity; in practice, however, this shortcoming turns out to be an interesting alternative to provide conservative schedules useful for guiding real operations.

Although both versions of the problem are interesting in real practice and are receiving increasing attention in the literature, in this paper we limit our study to the time-invariant with continuous quay. The main objective is to conduct a more in-depth study of models and exact methods capable of dealing with this version of the BACAP in realistic situations. In the first half of the paper, a new mixed integer linear model is proposed and several families of valid inequalities are identified and added to enhance the formulation. The resulting model is able to obtain optimal solutions for different classes of instances with up to 50 vessels per week. In the second half, we address the BACASP, the problem in which not a number of cranes but a set of specific cranes must be assigned to each vessel, thus ensuring that the solution is physically realizable. For this problem we have also developed a mixed integer model and two exact algorithms, based on the BACAP model, which can optimally solve instances with up to 40 vessels per week and thus provide more realistic solutions to this berth allocation problem. The exact methods proposed require fewer variables and outperform the state-of-the-art approaches in realistic problem instances. Moreover, they prevent the poor allocation of quay space that may result from using integer variables to represent the continuous berth, which is common practice in the literature.

The rest of this paper is organized as follows. In Section 2, we review the previous work. In Section 3, we describe the problem and the specific assumptions of our approach. In Section 4, we propose a new mixed integer linear programming model for the BACAP. Several constraints are added to reinforce the proposed formulation in Section 5. The model is extended to solve the BACASP in Section 6. In Section 7, we describe the experiments conducted and discuss their results. Finally, in Section 8, we draw some conclusions and indicate future research.

2. Literature review

In this section we review the academic literature on problems that integrate BAP and QCAP, focusing on the continuous quay version of both BACAP and BACASP. Readers interested in the discrete variant can consult recent studies such as those by Hsu (2016), Lalla-Ruiz et al. (2014) and Ursavas (2014). Additionally, for a detailed review of the approaches that address these problems and the different integrated versions, see the surveys by Bierwirth & Meisel (2010, 2015) and Carlo et al. (2015).

2.1. BACAP studies

The first work on the integration of BAP and QCAP was presented by Park & Kim (2003). This pioneering paper formulated an integer model for continuous BACAP and variable-in-time crane assignment. In order to obtain a near-optimal solution they proposed a Lagrangean relaxation of the model and used the subgradient method. Additionally, the specific sets of QCs assigned to each vessel are determined using a dynamic programming procedure that minimizes the number of QC changes among vessels. On the basis of this approach, Zhang et al. (2010) proposed a MILP for a BACASP that takes into account the limited range of movement of each crane.

Other authors have also studied the continuous BACAP with variable-in-time crane assignment. Meisel & Bierwirth (2009) included the possibility of speeding up the arrival of each vessel, incurring a cost proportional to the time advanced, and considered a decreasing marginal crane productivity as the number of cranes serving a vessel increases, due to interferences between them. They proposed a mixed integer linear model and two metaheuristic approaches: a tabu search and a squeaky wheel optimization which obtained good results over previous and newly generated instances. Xuelian & Zhiying (2012) proposed an integer linear model and a decomposition heuristic procedure to assign quay cranes and berthing positions and times. Liang et al. (2012) first applied the BAP model of Kim & Moon (2003) and then assigned a number of QCs to each vessel assuming that it is dependent on its requested departure time. To do so, they developed several heuristics and applied them in a Sequence

Optimized Particle Swarm Optimization framework. Elwany et al. (2013) considered the case in which the water depth is not the same in all berthing positions. They extended the model presented by Meisel & Bierwirth (2009) and proposed a heuristic to construct feasible solutions from ordered lists of vessels used in a Simulated Annealing framework. He (2016) formulated a bi-objective model in which, besides the minimization of delay costs, the minimization of handling energy consumption was also considered. An integrated simulation and optimization method was also developed, in which the simulation was designed for evaluation and a memetic algorithm was designed for searching the solution space. Karam & Eltawil (2016) also solved the BACAP by decomposing it into two subproblems, BAP and QCAP, that are iteratively solved until the complete solution cannot be improved. A similar approach was followed by Han et al. (2015), but including other considerations related to the efficient use of the cranes.

The continuous BACAP with time-invariant crane assignment has also been studied by several authors. Blazewicz et al. (2011) formulated the problem as a moldable task scheduling problem in which the durations of tasks are non-linear functions of the resources assigned to them. They proposed a heuristic algorithm which obtained good average results. Yang et al. (2012) solved the BACAP by developing genetic algorithms that solved the BAP and the QCAP separately and combining them by means of a Nested Loop-based Evolutionary Algorithm (NLEA), which obtained good results compared with Park & Kim (2003). Yang et al. (2013) formulated a MILP for the integrated problem and proposed a Particle Swarm Optimization heuristic. More recently, Türkoğullari et al. (2014) proposed a new integer linear model, discretizing both the planning horizon and the quay length. Iris et al. (2015) proposed generalized set partitioning formulations (GSPP) for the BACAP considering both time-variant and time-invariant QC assignment policies. Computational results show that both formulations performed well with respect to both upper and lower bounds and can provide optimal solutions for small and medium sized instances.

2.2. BACASP studies

The BACASP with continuous dynamic berth allocation and variable-in-time QC-to-vessel assignment was studied by Chang et al. (2010), who proposed a multi-objective mixed integer linear model. The authors also applied a rolling time horizon strategy and proposed a heuristic algorithm to construct feasible solutions and a Hybrid Parallel Genetic Algorithm. Another study in the same vein is Raa et al. (2011), in which a MILP is proposed and tested. Rodriguez-Molins et al. (2014) solved the BACASP using a GRASP metaheuristic algorithm. Recently, Türkoğullari et al. (2016) proposed an integrated integer linear model and an iterative algorithm based on a decomposition scheme for the variable-in-time BACASP. This variant of the problem was also addressed by Agra & Oliveira (2018), who proposed a new model avoiding the use of the Big-M of previous formulations, new valid inequalities and a branch-and-cut algorithm.

The BACASP with time-invariant quay crane assignment was studied by Le et al. (2012), who proposed a multi-objective MILP model and a multi-objective PSO. Türkoğullari et al. (2014) also proposed a MILP and an iterative algorithm based on their BACAP model.

Broadening the scope of the BACASP, Meisel & Bierwirth (2013) addressed, in an integrative manner, the main optimization problems that can be found on the seaside of container terminals (BAP, QCAP and QCSP: Quay Crane Scheduling Problem). They proposed a three-phase framework, and one of these phases involves the solution of a continuous BACAP with variable-in-time crane assignment by means of a MILP similar to the one presented in Meisel & Bierwirth (2009).

In this paper we address the state-of-the-art versions of both the time-invariant BACAP and BACASP presented by Türkoğullari et al. (2014). Unlike that approach, we propose various MILP models, iterative procedures and branch-and-cut algorithms which do not require the use of integer variables to represent quay positions. Consequently, the number of variables involved is dramatically reduced and solutions can be more useful in real situations without compromising the efficiency of the solution process. Moreover, tackling the problem in this way also prevents underuse of the quay space due to reasons derived from the mathematical formulation of the problem. In summary, our contribution aims at reducing the gap between the abstract representation of the problem and its applicability in real situations by proposing tighter formulations which lead to more efficient methods.

3. Description of the problem

The Berth Allocation and quay Crane Assignment Problem (BACAP) is the optimization problem of achieving the least-cost berth plan for a set of vessels calling at the port that have to be moored at a quay with a given number of quay cranes available to serve them. We address the version of the problem in which there is only one continuous quay and in which the number of quay cranes serving a vessel, once assigned, is kept fixed until the end of its last loading/unloading operation. This berth plan can be rendered as a space-time layout (see Fig. 1) in which the vertical axis represents the mooring position at the quay and the horizontal axis represents the time. Each vessel is a rectangle whose length represents its handling time and whose width represents its length. Thus, vessels can be moored along the quay within a given time horizon, while quay cranes can move along the quay to serve the vessels provided that they do not cross each other. According to the scheme proposed by Bierwirth & Meisel (2015), this problem can be classified as $cont \,|\, dyn \,|\, QCAP \,|\, \sum(w_1\, pos + w_2\, wait + w_3\, tard)$.

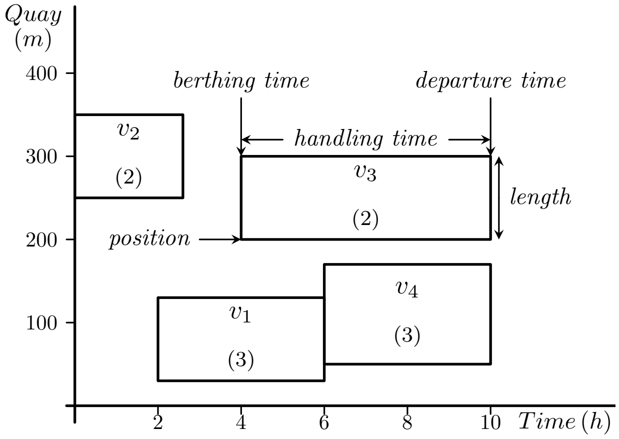


Figure 1: Graphical representation of a berth plan with 4 vessels and a quay of 400 m long with 5 cranes. The number of cranes appears in parentheses.

The objective is to assign a position on the quay, a berthing time, and a number of quay cranes to each calling vessel, so as to minimize the overall assignment cost. For each vessel, we consider three different kinds of costs: the cost of waiting before berthing, the cost of delay after the desired departure time, and the cost of deviation from the desired position on the quay. To do this, we use the following given data:

- Set of time periods: $T = \{1, 2, ..., H\}$, where H is the planning horizon.
- Length of the quay: L.
- Number of quay cranes: Q.
- Set of calling vessels: V, with $N = |V|$.

  For each vessel $i \in V$, we know:

  - length: l_i
  - expected arrival time: a_i
  - desired departure time: s_i
  - minimum and maximum number of cranes that can be assigned to the vessel: q_i^{min}, q_i^{max}
  - estimated handling time if it is handled using q cranes: u_i^q
  - desired position at the quay: d_i
  - cost per period of waiting for berthing after the expected arrival time: C_i^w

  - cost per period of delay after the desired departure time: C_i^d
  - cost per length unit away from the desired position at the quay: C_i^p

The waiting time of a vessel is defined as the difference between its assigned berthing time and its expected arrival time. The delay is the difference between the desired departure time and the assigned departure time. Moreover, we consider that each vessel has an ideal position on the quay, which is the position closest to the location of its related containers in the yard. Hence, the deviation from the desired position is the distance between the ideal position and the assigned position on the quay. The handling time of each vessel depends on the number of cranes assigned to it. It can be inversely proportional to the number of cranes, as in Park & Kim (2003), or can take into account the decreasing marginal productivity due to crane interferences, as in Meisel & Bierwirth (2009). Any relationship between the number of cranes and the handling times can be included in the model by specifying the particular handling times u_i^q of each vessel i if q quay cranes are assigned to it.

The assumptions of our approach are the following:

- Time
  - The planning horizon is divided into multiple equal time segments.
  - Vessels are to be moored within the planning horizon.
- Quay
  - Each position on the quay can accommodate one vessel at a time.
- Vessels
  - When a vessel is moored, the berthing position is kept fixed.
  - Once started, the handling of a vessel cannot be interrupted.
  - The handling time of each vessel is considered to be independent of its berthing position. This assumption is reasonable if the quay has enough machinery and workers for container transportation between the yard and the quay at any moment. Hence, the cranes serving each vessel do not need to wait for vehicles. The increased transportation cost produced if the position of the vessel deviates from its desired position is included in the objective function.
  - The handling time of each vessel depends on the number of cranes assigned to it. No specific relation between them is assumed, so it can be either linear or non-linear.

  - The setup and release times of the cranes may be included in the handling time estimates for the vessels provided as parameters. The time spent by cranes in moving from one vessel to another depends on the resulting berth plan and thus cannot be precisely anticipated. However, handling time estimates may include an additional time clearance for those movements.
  - The time for docking and undocking maneuvers is considered to be included in the vessel handling time.
  - Vessels may have different relative importance. Therefore, cost coefficients are specific to each vessel.
  - The inter-ship clearance is included in the vessel length. In general, for vessels longer than 130 m, this clearance corresponds to 10% of its length. For small vessels, the minimum clearance is 10 m.

- Cranes
  - The number of cranes available at the quay is fixed and all the cranes have the same characteristics.
  - All quay cranes can move along the quay, but they cannot cross each other.
  - The time taken by cranes to move along the quay is considered negligible compared to the handling time of vessels and thus it is not taken into account in the schedule.
  - Each quay crane can be assigned to one vessel at most in each time period.
  - The number of quay cranes assigned to a vessel does not change during its stay at the quay.
  - There is a minimum and a maximum number of cranes that can be assigned to a vessel.

4. A mixed integer linear model

We address the continuous BACAP with time-invariant quay crane assignment by proposing the following MILP.

First, we define the following variables:

t_i = berthing time of vessel i

p_i = berthing position of vessel i

h_i = delay incurred in the handling of vessel i

e_i = deviation of vessel i from its desired position on the quay

$$r_{iqt} = \begin{cases} 1, & \text{if the handling of vessel } i \text{ with } q \text{ cranes starts at time } t \\ 0, & \text{otherwise} \end{cases}$$

$$\sigma_{ij} = \begin{cases} 1, & \text{if the departure time of vessel } i \text{ is prior to the berthing time of vessel } j \\ 0, & \text{otherwise} \end{cases}$$

$$\delta_{ij} = \begin{cases} 1, & \text{if vessel } i \text{ is completely below vessel } j \text{ in the space-time diagram, that is,} \\ & i \text{ is positioned completely to the right of } j\text{, looking at them from the quay} \\ 0, & \text{otherwise} \end{cases}$$

These variables and the previously defined parameters are the elements of the following model:

$$Min \sum_{i \in V} (C_i^w (t_i - a_i) + C_i^d h_i + C_i^p e_i) \tag{1}$$

s. t.

$$\sum_{t=a_i}^{H} \sum_{q=q_i^{min}}^{q_i^{max}} r_{iqt} = 1, \qquad \forall i \in V \tag{2}$$

$$t_i = \sum_{t=a_i}^{H} \sum_{q=q_i^{min}}^{q_i^{max}} r_{iqt} \cdot t, \qquad \forall i \in V \tag{3}$$

$$t_j - t_i - \sum_{t=a_i}^{H} \sum_{q=q_i^{min}}^{q_i^{max}} (r_{iqt} \cdot u_i^q) - (\sigma_{ij} - 1)H \geq 0, \qquad \forall i, j \in V, i \neq j \tag{4}$$

$$p_j - (p_i + l_i) - (\delta_{ij} - 1)L \geq 0, \qquad \forall i, j \in V, i \neq j \tag{5}$$

$$\sigma_{ij} + \sigma_{ji} + \delta_{ij} + \delta_{ji} \geq 1, \qquad \forall i, j \in V, i \neq j \tag{6}$$

$$p_i + l_i \leq L + 1, \qquad \forall i \in V \tag{7}$$

$$\sum_{i \in V} \sum_{q=q_i^{min}}^{q_i^{max}} \sum_{\tau=\max(a_i, t-u_i^q+1)}^{t} r_{iq\tau} \cdot q \leq Q, \qquad \forall t \in T \tag{8}$$

$$h_i \geq t_i - s_i + \sum_{t=a_i}^{H} \sum_{q=q_i^{min}}^{q_i^{max}} (r_{iqt} \cdot u_i^q) - 1, \qquad \forall i \in V \qquad (9)$$

$$e_i \geq p_i - d_i, \qquad \forall i \in V \qquad (10)$$

$$e_i \geq d_i - p_i, \qquad \forall i \in V \qquad (11)$$

$$\sigma_{ij}, \delta_{ij}, \in \{0, 1\}, \qquad \forall i, j \in V, i \neq j \qquad (12)$$

$$r_{iqt} \in \{0, 1\}, \qquad \forall i \in V, \forall t \in \{a_i, \ldots, H\}, \forall q \in \{q_i^{min}, \ldots, q_i^{max}\} \qquad (13)$$

$$h_i, e_i \geq 0, \qquad \forall i \in V \qquad (14)$$

$$p_i \geq 1, \qquad \forall i \in V \qquad (15)$$

The objective function (1) is the minimization of the overall planning cost, which consists of the sum of the waiting cost, the delay cost, and the deviation cost incurred by the terminal for each vessel. The handling of each vessel starts only once and with a fixed number of cranes (2). Constraints (3) link the berthing time of a vessel to the variables r_{iqt} to prevent inconsistencies. Constraints (4)–(6) prevent overlaps between vessels in space and time. Constraints (7) forbid berthing positions in which the vessel would exceed the quay length. Moreover, the number of cranes assigned to vessels for each period of time cannot be greater than the total number of cranes available at the quay due to constraints (8). Constraints (9) define the delay of each vessel and constraints (10)–(11) the deviation from the desired position. Finally, constraints (12)–(15) define the domain of the variables. Note that the first position available at the quay is 1, according to constraints (15).

This model is different from previous models in various respects. Unlike the pioneering paper on the variable-in-time BACAP by Park & Kim (2003), the model proposed here does not use a non-binary variable (Y_{jk}) representing the number of cranes assigned to a vessel (k) in a time period (j), nor a binary variable (X_{ijk}) representing whether a coordinate (i, j) of the space-time diagram is covered by the rectangle representing the schedule of a vessel (k). The model we propose also differs from the variable-in-time BACAP model presented by Meisel & Bierwirth (2009). They use a binary variable (r_{it}) which is set to 1 if at least one crane is assigned to a vessel (i) in a time period t and another binary variable (r_{itq}) which is set to 1 if exactly a certain number of cranes (q) are assigned to a vessel (i) in a time period (t). In contrast, our model uses a binary variable r_{iqt} representing whether the handling of a vessel i *starts* with q cranes in time period t, thereby involving a completely different logic for the expression of the constraints and reducing the number of variables for the time-invariant BACAP. Furthermore, Meisel & Bierwirth (2009)'s model assumes that vessels' deviations from their desired positions necessarily impact on their handling times, whereas in our formulation of the problem we assume that deviations force the terminal to incur more costs related to horizontal transport at the yard. Moreover, unlike these previous works, in our proposed model the cost structure is different, as vessels are not allowed to speed up their arrival and there is no tardiness penalty apart from the delay cost. The objective function of our formulation, however, is the same as the one used by Türkoğullari et al. (2014), although their model differs in its proposed variables and thus in its constraints. In particular, in their model a single sort of binary variable (X_{ijt}^{k}) is used, which represents whether a vessel (i) berths at a section on the quay (j) in a time period (t) with a number of cranes (k) assigned to it. This is only possible by discretizing the quay length, and for this reason the number of variables increases considerably with the length and the discretization unit applied. Finally, our proposal also differs from the generalized set partitioning models presented by Iris et al. (2015), since our formulation is not based on such a modelling approach.

5. Enhancing the integer linear model for BACAP: valid inequalities

In this section we develop several families of valid inequalities for the model described in the previous section. These inequalities will reduce the polyhedron of the solutions of the integer model linear relaxation, making it closer to the convex hull of the set of integer solutions, thus improving the lower bounds and hopefully reducing the search in a branch and bound algorithm. Nevertheless, we have to be careful not to overload the model with too many new constraints, because the positive effects can be counteracted by the extra effort needed to solve the linear relaxations. Therefore, in the description of each type of valid inequality, together with the general expression we indicate the way in which we implement it, based on the idea that most of the conflicts between vessels appear in the vicinity of their ideal allocations.

5.1. Strengthening the non-overlapping constraints

If two vessels i and j are concurrent in time according to variables r_{iqt} and r_{jqt}, then they must be separated in space. Although this constraint can be built in general for any pair of times t_i and t_j that ensure the concurrence, we describe it here for the particular case in which they are close to their arrival times, because these are the times at which the vessels will try to be berthed and therefore the constraint to solve the conflict between them will be active.

For each pair of vessels i, j, such that $i \neq j$, $a_i + \min(u_i^q) > a_j$ and $a_j + \min(u_j^q) > a_i$, we can write:

$$\sigma_{ij} + \sigma_{ji} + \sum_{q=q_i^{min}}^{q_i^{max}} \sum_{t=a_i}^{a_j+\min(u_j^q)-1} r_{iqt} + \sum_{q=q_j^{min}}^{q_j^{max}} \sum_{t=a_j}^{a_i+\min(u_i^q)-1} r_{jqt} \leq 2, \tag{16}$$

If both terms involving variables r_{iqt} and r_{jqt} take value 1, the vessels overlap in time and then $\sigma_{ij} = \sigma_{ji} = 0$. According to constraint (6), $\delta_{ij} + \delta_{ji} \geq 1$ and therefore the vessels will be separated in space.

Figure 2 shows an example of concurrence in time. If vessel i starts at a time $t_i \in \{a_i, a_j{+}\min(u_j^q){-}1\}$ and vessel j starts at a time $t_j \in \{a_j, a_i + \min(u_i^q) - 1\}$, they concur in time and therefore $\delta_{ij} + \delta_{ji} \geq 1$, separating them in space.

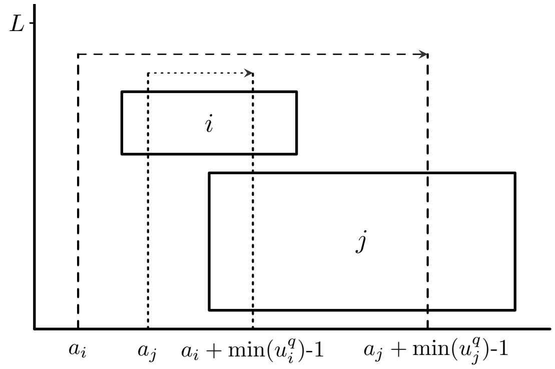


Figure 2: If vessels concur in time, they must be separated in space.

5.2. Minimum separation in time and space for vessels in their desired positions

If a pair of vessels would overlap if they were moored in their desired positions at their arrival times, a minimum separation between them would be needed, in time or in space. Figure 3 shows the case of two vessels i and j at their least-cost assignment. If they are separated in time ($\sigma_{ij} + \sigma_{ji} = 1$), we can compute the minimum separation *minTimeMove* (in the figure: $a_i + \min(u_i^q) - a_j$) and then we have:

$$t_i + t_j \geq a_i + a_j + minTimeMove \cdot (\sigma_{ij} + \sigma_{ji}), \qquad \forall i, j \in V, i \neq j \tag{17}$$

If they are separated in space ($\delta_{ij}{+}\delta_{ji} = 1$), we can compute the minimum separation *minSpaceMove* (in the figure: $d_i + l_i - d_j$) and we have:

$$e_i + e_j \geq minSpaceMove \cdot (\delta_{ij} + \delta_{ji}), \qquad \forall i, j \in V, i \neq j \tag{18}$$

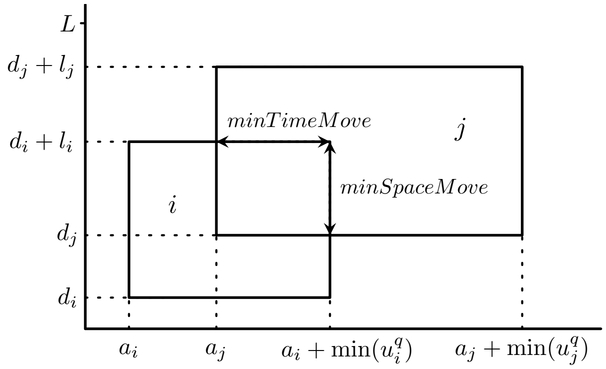


Figure 3: Minimal separation in time and space.

5.3. Cover constraints on ordered sets of variables δ

If we have a set of vessels such that the sum of their lengths exceeds the quay length, then not all of them can be concurrent in space and at least one of them must be separated in time. Figure 4 shows an example of three vessels i, j, k, with $l_i + l_j + l_k > L$. Therefore, we can add the inequality $\delta_{ij} + \delta_{jk} + \delta_{ki} \leq 1$. As this condition is satisfied for any ordering of the vessels, we can add constraints corresponding to other orderings, for instance, $\delta_{ji} + \delta_{ik} + \delta_{kj} \leq 1$.

In general, if we identify a subset of vessels S such that the sum of their lengths exceeds the length of the quay, for each permutation of vessels in S, $i, j, k, .., n, m$ we have a constraint:

$$\delta_{ij} + \delta_{jk} + \cdots + \delta_{nm} + \delta_{mi} \leq |S| - 2, \qquad i, j, k, ..., n, m \in S \tag{19}$$

We only consider minimal subsets S of vessels that do not fit together at the quay and concur in time if they were moored at their arrival times, considering their maximum handling times. We also limit the cardinality of the subsets, using a parameter α.

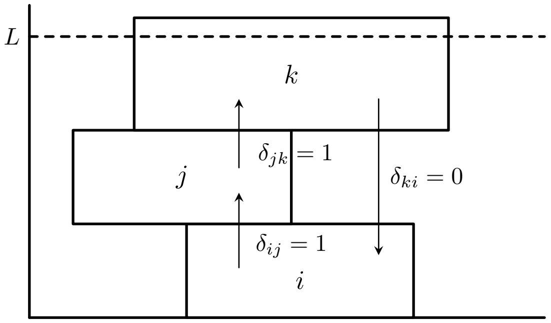


Figure 4: Vessels that do not fit together at the quay.

5.4. Cover constraints on non-ordered sets of variables δ

If we have a set of vessels S such that the sum of their lengths exceeds the quay length, as in the previous case, but instead of taking it as an ordered set we consider simultaneously all its permutations and all the variables separating them in space, as can be seen in Figure 5, a valid inequality for this set is:

$$\sum_{i \in S} \sum_{j \in S, j \neq i} \delta_{ij} \leq \frac{|S|^2 - |S|}{2} - 1 \tag{20}$$

Note that the total number of δ variables in set S is $|S|(|S|-1)$. Moreover, in each pair of variables δ_{ij}, $\delta_{ji}, i, j \in S$, if we assume one vessel in the pair is above the other in the space-time diagram, then one of the variables must be equal to 1, while the other must be 0 by constraint (5). In Figure 5, for example, $\delta_{jk} = 1$ and $\delta_{kj} = 0$. Consequently, the maximum value of the sum of the variables will be $(|S|^2 - |S|)/2$. If the sum of the vessel lengths exceeds L, not all the vessels in S can be stacked up one above the other in the space-time diagram, so the sum of the variables must be lower than or equal to $(|S|^2 - |S|)/2 - 1$.

As in the previous case, we only consider minimal subsets of vessels S that do not fit together at the quay and concur in time if they are moored at their arrival times, considering their maximum handling times.

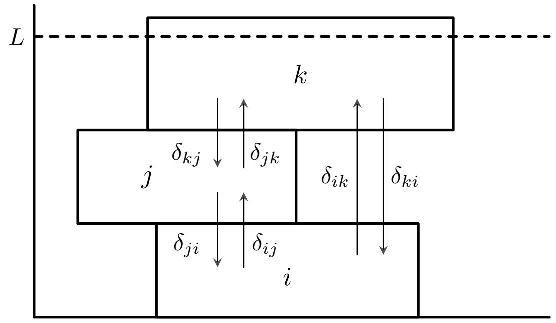


Figure 5: Constraint on a non-ordered set of vessels.

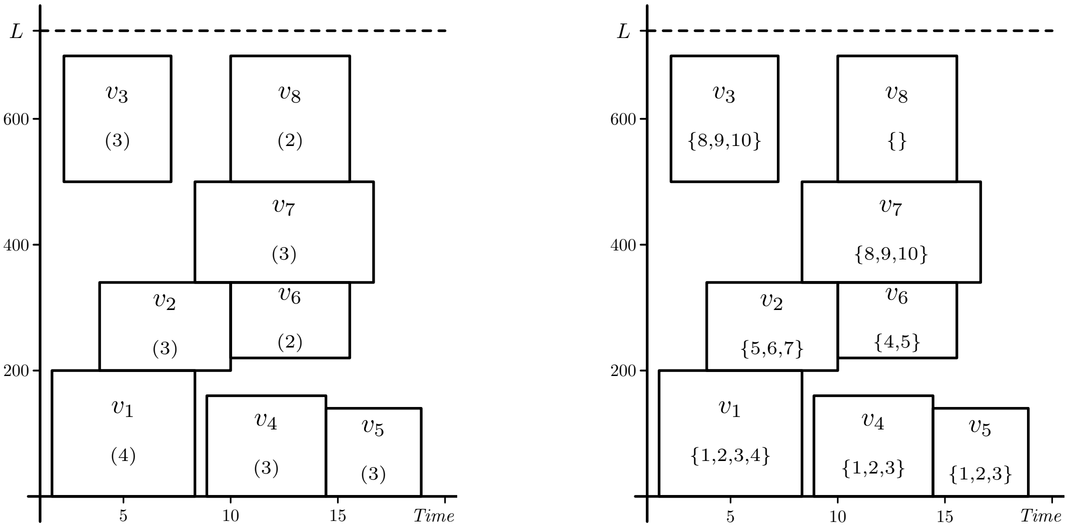
5.5. Cover constraints on variables r_{iqt}

The last family of valid inequalities also refers to groups of vessels that cannot fit together at the quay, but this time we add cover constraints on variables r_{iqt}. Now these subsets of vessels are identified among the vessels that would concur in time if they were assigned to their ideal allocation, considering their corresponding minimum handling times to ensure validity. Let S be a set of vessels satisfying these conditions and $minDep_i^S$ the minimum departure time in the ideal assignment of the vessels in S, not including vessel i. We can add the following inequality:

$$\sum_{i\in S}\sum_{q=q_i^{min}}^{q_i^{max}}\sum_{t=a_i}^{minDep_i^S-1} r_{iqt} \leq |S|-1 \tag{21}$$

6. The problem of assigning specific cranes to vessels

A feasible solution for the BACAP only guarantees that in any period, the number of cranes being used does not exceed the number of available cranes. However, it does not ensure that a feasible assignment of cranes to vessels is possible. Figure 6 illustrates this situation. Each rectangle represents a vessel and the number of the cranes assigned to it appears in parentheses. If there are 10 cranes available, the solution in Figure 6(a) is feasible for the BACAP. Let us assume that the cranes are numbered from 1 to 10, ordered from the beginning of the quay, and we assign cranes to vessels as shown in Figure 6(b). Starting from the beginning of the planning period, cranes 1, 2, 3, and 4 are assigned to vessel 1; cranes 5, 6, and 7 to vessel 2; and cranes 8, 9, and 10 to vessel 3. When vessel 7 starts being processed, vessel 3 has finished and so cranes 8, 9, and 10 can be assigned to vessel 7. Similarly, we can assign cranes 1, 2, and 3 to vessel 4. When vessel 2 has finished, cranes 4 and 5 can be assigned to vessel 6. However, no cranes can be assigned to vessel 8. When it starts being processed, two cranes, 6 and 7, are available, but they cannot be moved to the position of vessel 8 without causing a disruption in the handling of vessel 7. Therefore, in order to ensure a feasible crane assignment, the model described in previous sections has to be extended to include the assignment of specific cranes to each vessel in such a way that each vessel can be processed without interruptions or crane changes. In this section we propose an integer linear model for the BACASP, extending the ideas of the BACAP model, as well as an iterative procedure and a branch-and-cut algorithm, in which the model for the BACAP can be used to obtain the solution for the BACASP.



(a) Feasible solution for the BACAP.

(b) Infeasible solution for the BACASP.

Figure 6: A solution of the BACAP not feasible for the BACASP.

6.1. An integer linear model for the BACASP

Let us consider that the Q cranes on the quay are numbered and ordered from 1 to Q starting from the beginning of the quay. We define *crane groups* as groups of consecutive cranes, with cardinality varying from 1 to Q. There will be Q groups composed of 1 crane, $Q-1$ groups of 2 cranes: $\{1,2\},\{2,3\},\ldots;$ $Q-2$ groups of 3 cranes, and so on, and a final group composed of the Q cranes. We denote the set of these crane groups by QG. Each group $g \in QG$ is defined by 3 parameters:

- n_g= number of cranes in the group
- f_g= number of the first crane in the group
- z_g= number of the last crane in the group

As every vessel i has a minimum and a maximum number of cranes that can be assigned to it, q_i^{min} and q_i^{max}, respectively, not all the crane groups can be assigned to every vessel. We define the set of crane groups that can be assigned to vessel i as $QG_i = \{g \in QG \,|\, q_i^{min} \leq n_g \leq q_i^{max}\}$.

The variables defined in the previous sections also apply here, except that now r_{igt} takes value 1 if vessel i starts at time t being handled by crane group g, and 0 otherwise.

With these changes, a model for the BACASP is:

$$Min \sum_{i \in V}(C_i^w(t_i - a_i) + C_i^d h_i + C_i^p e_i) \tag{22}$$

s. t.

$$\sum_{t=a_i}^{H} \sum_{g \in QG_i} r_{igt} = 1, \qquad \forall i \in V \tag{23}$$

$$t_i = \sum_{t=a_i}^{H} \sum_{g \in QG_i} r_{igt} \cdot t, \qquad \forall i \in V \tag{24}$$

$$t_j - t_i - \sum_{t=a_i}^{H} \sum_{g \in QG_i} (r_{igt} \cdot u_i^{n_g}) - (\sigma_{ij} - 1)H \geq 0, \qquad \forall i, j \in V, i \neq j \tag{25}$$

$$p_j - (p_i + l_i) - (\delta_{ij} - 1)L \geq 0, \qquad \forall i, j \in V, i \neq j \tag{26}$$

$$\sigma_{ij} + \sigma_{ji} + \delta_{ij} + \delta_{ji} \geq 1, \qquad \forall i, j \in V, i \neq j \tag{27}$$

$$p_i + l_i \leq L + 1, \qquad \forall i \in V \tag{28}$$

$$\sum_{i \in V} \sum_{g \in QG_i} \sum_{\tau = \max(a_i, t - u_i^{n_g} + 1)}^{t} r_{ig\tau} \cdot n_g \leq Q, \qquad \forall t \in T \tag{29}$$

$$h_i \geq t_i - s_i + \sum_{t=a_i}^{H} \sum_{g \in QG_i} (r_{igt} \cdot u_i^{n_g}) - 1, \qquad \forall i \in V \tag{30}$$

$$e_i \geq p_i - d_i, \qquad \forall i \in V \tag{31}$$

$$e_i \geq d_i - p_i, \qquad \forall i \in V \tag{32}$$

$$\sigma_{ij}, \delta_{ij}, \in \{0, 1\}, \qquad \forall i, j \in V, i \neq j \tag{33}$$

$$r_{igt} \in \{0, 1\}, \qquad \forall i \in V, \forall g \in QG_i \forall t \in \{a_i, \ldots, H\} \tag{34}$$

$$h_i, e_i \geq 0, \qquad \forall i \in V \tag{35}$$

$$p_i \geq 1, \qquad \forall i \in V \tag{36}$$

However, two more constraints have to be added:

- In each period t a crane group g can be assigned at most to one vessel

$$\sum_{i | g \in QG_i} \sum_{\tau = \max(a_i, t - u_i^{n_g} + 1)}^{t} r_{ig\tau} \leq 1, \qquad \forall t \in T, \forall g \in QG \tag{37}$$

- If vessel i is moored to the right of vessel j, looking at them from the quay, the numbers of the cranes assigned to i must be lower than the numbers of the cranes assigned to vessel j.

$$\sum_{t=a_j}^{H} \sum_{g \in QG_j} f_g r_{jgt} - \sum_{t=a_i}^{H} \sum_{g \in QG_i} z_g r_{igt} \geq 1 - Q(\delta_{ji} + \sigma_{ij} + \sigma_{ji}), \qquad \forall i, j \in V, i \neq j \tag{38}$$

If $\delta_{ji} = 1$ or $\sigma_{ij} = 1$ or $\sigma_{ji} = 1$, the constraint is deactivated. But if $\delta_{ji} = \sigma_{ij} = \sigma_{ji} = 0$, then, by constraint (27), $\delta_{ij} = 1$, and vessel i is to the right of vessel j, that is, i is below j in the space-time diagram. In this case, the constraint ensures that the number of the first crane serving vessel j is greater than the number of the last crane serving vessel i.

6.2. An iterative procedure using the BACAP model

The model described in Section 6.1 is necessarily more complex than the model previously presented for the BACAP, because instead of having an index q for each crane, variables r_{igt} have an index g for each crane group admissible for vessel i. This added complexity resulting from a greater number of variables may limit the applicability of the model for solving problems of large sizes. An alternative procedure for solving the BACASP is to use the BACAP model in an iterative way, adding constraints to the BACAP model if its solution is not feasible for the BACASP until a feasible and then optimal solution is found.

Our iterative procedure takes as its starting point the following definitions and theorems proposed by Türkoğullari et al. (2014), which they use for a specific iterative procedure based on their model. A vessel sequence $v_1, v_2, \dots, v_n$ in a given feasible solution of the BACAP is *complete* if v_1 is the vessel closest to the beginning of the quay, v_n is the vessel closest to the end of the quay, v_i and v_{i+1} are two consecutive vessels with v_i closer to the beginning of the quay, and two consecutive vessels in the sequence concur at least for one time period. A complete sequence is said to be *proper* if the sum of the number of cranes assigned to vessels in this sequence is less than or equal to the number of cranes available at the quay. Otherwise, it is called an improper complete sequence. For example, in Figure 6(b) vessels v_4, v_6, v_7, and v_8 form a proper complete sequence, while vessels v_1, v_2, v_7, and v_8 form an improper sequence. It is easy to show that given an optimal solution of the BACAP, an optimal solution of the BACASP can be obtained from it if and only if every complete sequence of vessels extracted from the solution of the BACAP is proper.

With regard to our iterative procedure, if a complete improper sequence $S = s_1, \dots, s_n$ is found, the following constraint can be added to our BACAP model to prevent vessels in the sequence from being assigned so as to form an improper complete sequence in that specific order at any given time and position:

$$\sum_{i \in S} \sum_{t=a_i}^{H} \sum_{q=q_i^{min}}^{q_i^{max}} q \; r_{iqt} \leq Q + M \left(\sum_{j=s_1}^{s_{n-1}} (\sigma_{j,j+1} + \sigma_{j+1,j} - \delta_{j,j+1}) + n - 1 \right) \tag{39}$$

where: $M = \sum_{i \in S} q_i^{max} - Q$. If any pair of consecutive vessels in the sequence, v_j and v_{j+1}, are separated in space, v_{j+1} being above vessel v_j ($\delta_{j,j+1} = 1$), and are concurrent in time ($\sigma_{j,j+1} = \sigma_{j+1,j} = 0$), then the rightmost term in the expression takes value 0 and the assignment of a number of cranes to each vessel, represented by variables r_{iqt}, must satisfy the condition that their sum does not exceed Q. In any other case, the constraint is satisfied for any crane assignment.

Using this constraint, we have developed the iterative algorithm described in Algorithm 1. If just one constraint is added for every improper sequence found, considering the vessels in the order in which they appear in this sequence, the solution of the next BACAP model very frequently includes the same set of vessels, but the order in which they appear in the sequence is changed. In order to prevent this situation, at least partially, instead of including just one constraint for each improper sequence, we also include other permutations of vessels. Nevertheless, as the length of the sequence can be great in large instances, we only do this if the cardinality of the sequence does not exceed a parameter $MaxSizeSeqForPerm$.

Unlike the iterative procedure presented by Türkoğullari et al. (2014), the use of the relative position variables σ and δ in our proposed constraint (39) allows us to prevent any specific improper complete sequence from appearing in the integer solution regardless of the time and position of the vessels. In their iterative procedure, for each improper complete sequence a constraint is added to prevent the vessels from being placed in a position of time and space close to their specific locations in the solution in which the sequence was found. Only a limited range of positions are forbidden in this way, so the procedure might require additional iterations to prevent close positions. Moreover, no other permutations of the same vessels are considered.

Algorithm 1 Iterative algorithm for the BACAP–BACASP

```
Require: A formulation $P$ for the BACAP.
Ensure: A solution for the BACAP without improper complete sequences
Solve $P$
while there is at least one improper complete sequence in an optimal solution of $P$ do
    for all improper complete vessel sequence $S$ found do
        Insert in $P$ a cut for $S$ using constraint (39)
        if $|S| \leq MaxSizeSeqForPerm$ then
            Insert in $P$ such a cut for each other permutation of the vessels in $S$
        end if
    end for
    Solve $P$
end while
```

6.3. A branch-and-cut procedure using the BACAP model

The iterative algorithm presented in the previous section involves solving the whole model with the new cuts each time a new improper complete sequence is found in its optimal solution. From a theoretical perspective, this method is not efficient and may be improved by identifying the cuts and adding them to the model as soon as a new integer solution is obtained. Thus, if the integer solution found contains new improper complete sequences, the corresponding cuts are added and the process resumes, without having to start again from scratch.

Specifically, the algorithm can be described as a branch and bound applied to the BACAP model in which at each node with a feasible integer solution a procedure is used to find all improper complete sequences and introduce their corresponding cuts (39) as lazy constraints. Therefore, these new constraints only will be checked each time a new candidate integer solution is found during the process. The candidate integer solution will be considered a feasible BACASP solution only if it has no new improper complete sequences and all the lazy constraints checked are satisfied. As in the iterative procedure previously explained, the parameter $MaxSizeSeqForPerm$ is used to limit the number of cuts introduced.

7. Computational experiments

The models and exact algorithms proposed were tested and compared with other recent proposals to assess their efficiency and limits on instances of realistic size. In this section we describe the experiments conducted and discuss their results.

7.1. Test instances and implementation issues

We have generated several sets of instances according to the criteria used in previous papers, extending them to consider the characteristics of the time-invariant BACAP and BACASP and the various types of cost assumed in this study. The first set was generated applying the criteria of Park & Kim (2003). From now on, it will be referred to as *GenPK.* For all the instances in this set we consider a quay with length $L = 1200$ meters and a time horizon $H = 300$ hours, discretized in units of 10 meters and 1 hour respectively. There are $Q = 11$ quay cranes available and $q_i^{min} = 2$, $q_i^{max} = 5$, $\forall i \in V$. The set consists of 50 randomly generated instances, 10 instances for each number of vessels considered: $N \in \{20, 25, 30, 35, 40\}$. The data relating to each vessel are determined by uniform distributions as follows: $U[1, 170]$ for the the arrival time, $U[10, 48]$ for the number of crane-hours required, $U[15, 35]$ for the length, and $U[1, 120]$ for the desired position on the quay. The desired departure time, which is not specified in the paper, is determined by applying the criterion of Meisel & Bierwirth (2009): $s_i = a_i + 1.5 \cdot \min(u_i^q)$. The vessel handling time for each number of cranes results from dividing the number of crane-hours specified for the vessel by the number of cranes, rounded up to the next integer. Finally, the costs are the same for all vessels: $C_i^w = 1000$, $C_i^d = 2000$, $C_i^p = 200$, $\forall i \in V$.

We have also generated a set of instances based on the criteria applied by Meisel & Bierwirth (2009). From this point on, it will be referred to as *GenMB-10m.* It consists of 50 instances, 10 for each number of vessels considered: $N \in \{20, 30, 40, 50, 60\}$. The time unit is 1 hour, the planning horizon is 210 hours, and the quay is 1000 meters long. Moreover, there are 10 cranes on the quay and three different kinds of vessels: Feeder, Medium and Jumbo. Within each instance, 60%, 30% and 10% of the vessels correspond to these classes, respectively. The arrival times of vessels are uniformly distributed between 0 and 168 (one week). The reason why we chose a planning horizon of 210 hours instead of 168 is to prevent the generation of infeasible instances. The lengths, workloads (in crane-hours), and min-max number of cranes of the vessels are computed according to Table 1 taken from the cited paper, while the desired position of each vessel i is generated from $U[1, L + 1 - l_i]$. For each vessel, each handling time results from dividing its workload in crane-hours by the number of cranes assigned to it, rounding up to the next integer. The desired departure time of each vessel i is $a_i + 1.5 \cdot \min(u_i^q)$. With respect to the costs, we consider that they are the same for all the vessels: $C_i^w = 1000$, $C_i^d = 2000$, and $C_i^p = 200$.

We also generated a set, called *GenMB-50m*, with the same instances as *GenMB-10m*, changing the discretization of vessel and quay lengths to 50 meters. In particular, the lengths were discretized and rounded to the next integer, and the desired positions were discretized and rounded to the nearest integer. The cost coefficients are also the same, except in the case of the deviation cost coefficient, which was changed to maintain the same cost per meter as in *GenMB-10m*: $C_i^p = 1000$. The set *GenMB-50m* is only considered in comparisons with previous works.

Table 1: Specifications of the test instances generated by Meisel and Bierwirth

Class	Length	Crane-hours	q_i^{min}	q_i^{max}
Feeder	$U[8,21]$	$U[5,15]$	1	2
Medium	$U[21,30]$	$U[15,50]$	2	4
Jumbo	$U[30,40]$	$U[50,65]$	4	6

All the methods developed in the previous sections as well as those proposed by Türkoğullari et al. (2014) were implemented in C++ using GCC–G++ 6.2 and CPLEX 12.6, limiting the size of the search tree to 30 GiB. The compilation was performed specifying the special parameters *–O3 –march=corei7* in order to generate an executable file that makes the most of the processor. The experiments were run on an Intel Core i7 2600 at 3.4 GHz with 31.4 GiB of RAM, running the operating system Ubuntu 14.04 GNU/Linux 3.13. The branch-and-cut algorithm was implemented on CPLEX defining a *lazy constraint callback* so that the cuts introduced are considered as lazy constraints. As this option disables the automatic parallel mode in CPLEX, the number of threads was manually set to 8 in order to make the most of the processor.

7.2. Evaluation of the MILP for the BACAP and the proposed valid inequalities

In order to evaluate the quality of the model proposed for the BACAP and the influence of the proposed valid inequalities, several experiments were conducted, first considering the proposed MILP alone, and then adding each of the valid inequalities described in Section 5. Each configuration was run on each instance with a time limit of 1 hour. In order to control the number of inequalities (19) added to the model, the parameter α limiting the cardinality of the subsets of vessels involved in a constraint was set to 4. This value was obtained in preliminary experiments, which showed that greater values led to the introduction of so many constraints that computation time was dramatically increased. With lower values very few cover constraints were added and thus their impact was in general very slight.

The results obtained for *GenPK* and *GenMB-10m* are shown in Tables 2 and 3, respectively. For each subset of 10 instances of a given size, the tables show the number of instances solved to optimality, the average computation time in seconds, and the average and maximum gap in percentage. For each instance, the gap was provided by CPLEX as $100 \cdot (ub - lb)/ub$, where lb is the value of the best lower bound obtained within the time limit, and ub is the value of the objective function corresponding to the best integer solution achieved.

Table 2: Solving the BACAP on instances *GenPK*

Model	Vessels	Optimum	Time	Avg. gap	Max. gap
MILP	20	10	1.2	0	0
	25	10	2.3	0	0
	30	10	6.2	0	0
	35	10	39.7	0	0
	40	9	625.2	0.3	3.2
MILP + constraints (16)	20	10	1.2	0	0
	25	10	1.9	0	0
	30	10	5.0	0	0
	35	10	49.5	0	0
	40	9	549.3	0.5	5.0
MILP + constraints (17), (18)	20	10	1.1	0	0
	25	10	2.0	0	0
	30	10	4.8	0	0
	35	10	48.0	0	0
	40	9	585.0	0.5	4.9
MILP + constraints (19)	20	10	1.4	0	0
	25	10	2.1	0	0
	30	10	5.0	0	0
	35	10	72.5	0	0
	40	9	494.5	0.5	4.6
MILP + constraints (20)	20	10	1.5	0	0
	25	10	2.4	0	0
	30	10	5.6	0	0
	35	10	51.6	0	0
	40	9	561.7	0.8	8.0
MILP + constraints (21)	20	10	1.4	0	0
	25	10	2.4	0	0
	30	10	5.2	0	0
	35	10	39.6	0	0
	40	9	532.1	0.5	5.2
MILP + all constraints	20	10	1.2	0	0
	25	10	1.7	0	0
	30	10	4.0	0	0
	35	10	53.8	0	0
	40	9	595.2	0.6	6.4

Table 3: Solving the BACAP on instances *GenMB-10m*

Model	Vessels	Optimum	Time	Avg. gap	Max. gap
MILP	20	10	0.6	0	0
	30	10	5.6	0	0
	40	10	14.6	0	0
	50	5	2440.3	2.6	7.7
	60	0	3607.3	20.4	42.5
MILP + constraints (16)	20	10	0.6	0	0
	30	10	5.3	0	0
	40	10	11.3	0	0
	50	5	2455.6	3.0	8.7
	60	0	3602.0	20.4	45.3
MILP + constraints (17), (18)	20	10	0.7	0	0
	30	10	4.8	0	0
	40	10	15.3	0	0
	50	6	2197.4	2.2	6.4
	60	0	3604.2	16.6	27.5
MILP + constraints (19)	20	10	0.7	0	0
	30	10	4.3	0	0
	40	10	12.5	0	0
	50	5	2351.6	2.4	9.3
	60	0	3602.2	18.8	40.1
MILP + constraints (20)	20	10	0.7	0	0
	30	10	5.6	0	0
	40	10	16.1	0	0
	50	5	2661.3	3.3	10.8
	60	0	3602.3	19.8	39.9
MILP + constraints (21)	20	10	0.6	0	0
	30	10	5.4	0	0
	40	10	19.0	0	0
	50	5	2710.2	3.3	10.2
	60	0	3601.8	18.4	36.2
MILP + all constraints	20	10	0.6	0	0
	30	10	4.5	0	0
	40	10	9.5	0	0
	50	7	2084.4	1.5	6.5
	60	0	3602.3	16.2	30.0

The results in Table 2 show that all but one of the instances in the *GenPK* set were solved to optimality. Only one instance of 40 vessels was not optimally solved, with a gap in the initial configuration of 3.2%. Table 3 shows similar results for the *GenMB-10m* set. All the 40-vessel instances were optimally solved, but only 5 of the instances of 50 vessels and none of the instances of 60 vessels could be optimally solved. For instances of 50 vessels, the gaps for instances not optimally solved were quite low, with a maximum of 7.7% in the initial configuration. For instances of 60 vessels, the gaps were larger, with an average of 20.4% and a maximum of 42.5%. The information in these tables indicates that 50 vessels is the maximum instance size for which the model can be used with a guarantee of obtaining an optimal or quasi-optimal solution.

With respect to the influence of the valid inequalities, Table 2 also shows that the added inequalities, separately or all together, do not improve the good performance of the initial model. However, in Table 3, we can observe that although each inequality by itself does not improve the results, by adding all the valid inequalities two more instances were solved to optimality, the average running times were reduced, except for the case of 60 vessels, and the average and maximum gaps were also decreased. For this reason, we decided to use the model including all the valid inequalities in the next tests.

7.3. Comparison with Türkoğullari et al. (2014)

We compared our model with the BACAP model presented by Türkoğullari et al. (2014), which considers time-invariant crane assignment and solves exactly the same problem. Both models were run over the same sets of instances: *GenPK*, *GenMB-10m*, and *GenMB-50m*, on the same computer. For the set of instances *GenPK*, the model proposed by Türkoğullari et al. (2014) could not even achieve integer solutions for any of the instances within the time limit of 3600 seconds. The reason is that the memory available in the computer (31.4 GiB) was not enough to construct the model. The results for sets *GenMB-10m* and *GenMB-50m* are shown in Table 4. As in previous tables, times are given in seconds and gaps in percentages. The number of instances in each group in which at least an integer solution was found is shown in the rows labelled "Solved". The empty fields indicate that no integer solutions were found within the time limit and thus it was not possible to calculate the statistics.

Table 4: Results for the BACAP using the Türkoğullari et al. model and our MILP.

Vessels		*GenMB-10m*		*GenMB-50m*	
		Model by Türkoğullari et al.	Our MILP with valid inequalities	Model by Türkoğullari et al.	Our MILP with valid inequalities
20	Solved	10	10	10	10
	Optimum	10	10	10	10
	Avg. time	309.6	0.6	14.3	0.4
	Avg. gap	0	0	0	0
	Max. gap	0	0	0	0
30	Solved	10	10	10	10
	Optimum	10	10	10	10
	Avg. time	748.6	4.5	28.2	7.2
	Avg. gap	0	0	0	0
	Max. gap	0	0	0	0
40	Solved	5	10	10	10
	Optimum	5	10	10	10
	Avg. time	1567.0	9.5	47.6	17.0
	Avg. gap	0	0	0	0
	Max. gap	0	0	0	0
50	Solved	1	10	10	10
	Optimum	0	7	9	4
	Avg. time	2301.0	2084.4	963.4	2740.3
	Avg. gap	62.4	1.5	0.2	3.8
	Max. gap	62.4	6.5	2.2	15.1
60	Solved	0	10	10	10
	Optimum	0	0	4	0
	Avg. time		3602.3	2436.9	3601.9
	Avg. gap		16.2	10.7	17.9
	Max. gap		30.0	24.9	28.6

The results show that our approach clearly outperforms the results of the model presented by Türkoğullari et al. (2014) over the set of instances *GenMB-10m* (Table 4). We are able to solve to optimality instances of up to 40 vessels in very short times, and almost all the instances of 50 vessels in half an hour, attaining low gaps when optimality could not be proven within the time limit.

With respect to the set of instances *GenMB-50m*, the model proposed by Türkoğullari et al. (2014) obtains more optimal solutions and lower gaps for large instances with 50 and 60 vessels. The performance of our model over this set of instances is similar to the one attained over the set *GenMB-10m.* The different performance of Türkoğullari et al. (2014)'s model shown on the sets *GenMB-10m* and *GenMB-50m* could be explained by its dependence on the discretization factor applied to the lengths of the quay and the vessels. They consider the position at the berth as an index of their binary variables, and therefore the number of variables increases dramatically when the discretization is finer. In particular, their model contains $O(BNQH)$ binary variables, considering B as the number of quay sections resulting from the discretization, whereas our model contains $O(N^2 + NQH)$ variables: $O(NQH)$ variables r; $O(N)$ variables t, p, h and e; and $O(N^2)$ variables σ and δ. The considerably greater number of variables in their model also explains the memory limitations it experienced in set $GenPK$.

All things considered, we can conclude that our model is a good approach to the BACAP, whatever the discretization factor applied to the lengths, as it does not increase the number of variables. It can be applied to any continuous berth allocation problem, regardless of the discretization used for quays and vessels. As set *GenMB-50m* is a discretization of *GenMB-10m*, the results obtained by the BACAP models on each instance can be compared. The costs of the solutions with a discretization of 10 meters are lower than those with a discretization of 50 meters. Overall, the costs decrease 11% on average, since the better utilization of the quay results in lower costs related to deviations from the desired position of the vessels and better utilization of the cranes, which consequently reduces delays. Unless there are specific reasons at a given terminal for using a discretization of 50 meters, a finer discretization, for instance of 10 meters, will produce better solutions.

7.4. Evaluation of the exact methods for the BACASP

In this section we discuss the results of the MILP proposed for the BACASP as well as those of the iterative procedure based on the BACAP described in Section 6. We also compare them with the iterative procedure proposed by Türkoğullari et al. (2014) on the sets of instances for which their model could achieve feasible solutions. In Algorithm 1 we set $MaxSizeSeqForPerm = 4$, so the permutations of an improper complete sequence are only included for sequences with up to 4 vessels. This value was determined through preliminary experiments, in which lower values of $MaxSizeSeqForPerm$ forced the procedure to iterate over many specific permutations of 4 vessels, while a value of 5 caused the introduction of too many constraints each time an improper complete sequence of 5 vessels was found. $MaxSizeSeqForPerm = 4$ thus appeared as the value that leads to the best computation time.

Tables 5 and 6 contain the results on sets *GenPK*, *GenMB-10m*, and *GenMB-50m*. The tables have the same structure as those of the previous section. The integer model (22)–(38) can fail to obtain an integer solution within the time limit due to the size of the model. The iterative procedures can fail because until they get to an optimal solution of the BACASP, the solutions obtained are not feasible and therefore the problem is not really solved. When using the model, if not all the instances of a subgroup have been solved, the computation times and gaps are calculated considering only those for which an integer solution has been found. In the case of the iterative procedures, there are no gaps, because either the optimal solution is found or there is no feasible solution. Table 5 does not show results for the iterative algorithm proposed by Türkoğullari et al. (2014) because the model on which it is based was not able to obtain even a feasible solution within the time limit for any instance.

Table 5: Comparing approaches to the BACASP on the set *GenPK*

		MILP	B&C	Iterative
Vessels				procedure
20	Solved	10	10	10
	Optimum	10	10	10
	Avg. time	153.3	20.3	1.8
	Avg. gap	0	0	
	Max. gap	0	0	
25	Solved	10	10	10
	Optimum	10	10	10
	Avg. time	329.3	48.0	3.1
	Avg. gap	0	0	
	Max. gap	0	0	
30	Solved	10	10	10
	Optimum	10	10	10
	Avg. time	955.3	304.4	15.4
	Avg. gap	0	0	
	Max. gap	0	0	
35	Solved	10	10	9
	Optimum	6	6	9
	Avg. time	2090.5	1547.2	640.6
	Avg. gap	32.0	7.9	
	Max. gap	98.8	29.3	
40	Solved	4	9	7
	Optimum	1	1	7
	Avg. time	3527.3	3359.2	622.7
	Avg. gap	46.2	16.0	
	Max. gap	86.8	25.4	

Table 6: Comparing approaches to the BACASP on sets *GenMB-10m* and *GenMB-50m*

Vessels		*GenMB-10m*				*GenMB-50m*			
		MILP	B&C	Iterative	Iterative by Türkoğullari	MILP	B&C	Iterative	Iterative by Türkoğullari
20	Solved	10	10	10	9	10	10	10	10
	Optimum	10	10	10	9	10	10	10	10
	Avg. time	28.3	1.3	0.7	687.8	25.8	1.4	0.7	17.5
	Avg. gap	0	0			0	0		
	Max. gap	0	0			0	0		
30	Solved	10	10	10	6	10	10	10	10
	Optimum	9	10	10	6	10	10	10	10
	Avg. time	629.9	125.9	13.5	2077.3	551.2	53.4	18.1	238.9
	Avg. gap	3.6	0			0	0		
	Max. gap	36.1	0			0	0		
40	Solved	10	10	10	0	10	10	10	10
	Optimum	8	9	10	0	8	8	10	10
	Avg. time	1439.8	1000.0	85.8	3632.4	1353.7	1208.0	40.8	191.2
	Avg. gap	4.0	1.4			5.9	2.0		
	Max. gap	22.3	14.0			42.4	12.2		
50	Solved	6	10	2	0	8	10	2	4
	Optimum	0	0	2	0	0	0	2	4
	Avg. time	3611.7	3601.8	2059.0		3602.5	3601.8	1744.0	1132.3
	Avg. gap	69.0	28.5			64.7	28.3		
	Max. gap	91.3	44.8			82.6	40.6		
60	Solved	3	7	0	0	3	8	0	0
	Optimum	0	0	0	0	0	0	0	0
	Avg. time	3616.3	3602.1			3604.0	3602.3		
	Avg. gap	91.9	53.0			87.4	48.6		
	Max. gap	94.8	79.8			91.2	69.5		

Tables 5 and 6 show that the proposed model for the BACASP is able to optimally solve most instances with up to 35 vessels of set *GenPK* and up to 40 vessels of sets *GenMB-10m* and *GenMB-50m*. As the model is more complex than that of the BACAP, the size of the instances optimally solved decreases. Compared with the model, the branch-and-cut algorithm reduces the computation times and gaps and also solves more instances in all sets. Unexpectedly, the proposed iterative procedure is even faster than the branch-and-cut algorithm on the three test sets. This may be explained by the special methods used by the solver when no user cuts are introduced. The main disadvantage of the iterative procedure is that for instances in which the BACAP model is difficult to solve, very few, if any, iterations can be done within the time limit and almost always the procedure fails to obtain a feasible solution. The branch-and-cut, in contrast, is able to obtain feasible solutions for all instances with 50 vessels and most instances with 60 vessels.

Comparing with the iterative procedure proposed by Türkoğullari et al. (2014) on sets *GenMB-10m* and *GenMB-50m*, for which their BACAP model was able to obtain solutions, it can be observed that their procedure works slightly better on instances with 50 vessels in set *GenMB-50m* and clearly worse in general on set *GenMB-10m*, basically repeating the behaviour of the BACAP models. Again, our exact methods have been shown to be very stable, regardless of the discretization factor used.

8. Conclusions

The berth allocation and quay crane assignment problem (BACAP) has become an important topic of research during the last decade. Researchers generally tackle this problem by considering either time-invariant or variable-in-time crane assignment. Although terminal operators aim to maximize the use of cranes, addressing the problem by considering a variable-in-time crane assignment usually introduces important shortcomings in the optimization process, such as the need for too many variables, the need to minimize the movements of the cranes between vessels and the considerable impact that the time spent by the cranes to perform such movements —which is usually ignored in academic studies—, may have on the overall plan. For this reason, the time-invariant version appears as an alternative in which a little sacrifice in the optimal use of the cranes may lead to solutions that are easier to implement in real terminals without further non-optimal changes.

The continuous time-invariant BACAP has received less attention in the academic literature than the variable-in-time version. The models and exact methods in the state-of-the-art proposed for this variant use too many variables and do not take advantage of the continuous nature of the quay. Consequently, new approaches might outperform them in both the computation time required and the size of the instances solved to optimality.

In this study we have proposed new mixed integer linear models and exact methods which overcome the shortcomings in the state-of-the-art and do achieve better results in realistic problem instances. Unlike other previously proposed formulations, the models presented here do not consider a discretized quay length and use instead a continuous variable for the berthing position of the vessels. This makes our methods truly continuous, and consequently their performance does not depend on the discretization factor used for the lengths of the quay and the vessels.

Our initial formulation proposed for the BACAP has also been enhanced by adding several families of valid inequalities, which are able to reduce computation times and gaps. As the computational study shows, the model has a stable behaviour, obtaining optimal or near-optimal results on different classes of instances with up to 50 vessels. In fact, it can solve optimally instances for which the state-of-the-art methods are not able to achieve feasible solutions under the same experimental conditions.

We have also introduced a new mixed integer model for the BACASP, which assigns specific cranes to the service of each vessel and thus produces more realistic solutions that can be used in practical situations. Although this model is more complex than the BACAP model, it is useful to solve instances with up to 40 vessels. Nevertheless, in order to solve larger instances of the BACASP, an iterative procedure and a branch-and-cut algorithm have also been designed, using the BACAP model and a new type of constraint.

Our approach could be extended to the variable-in-time crane assignment version of the BACAP in the future. Another potential line of research would be the integration of the BACAP with the Quay Crane Scheduling Problem, in line with the increasing trend of integrating all problems arising in each area of a container terminal. The BACAP could also be studied considering terminals consisting of a number of quays with different characteristics and numbers of cranes.

Acknowledgements

This study was partially supported by the Spanish Ministry of Economy and Competitiveness, DPI2014-53665-P, cofinanced by FEDER funds, and by Generalitat Valenciana, PROMETEO/2013/049.

References

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. (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.

Blazewicz, J., Cheng, T. C. E., Machowiak, M., & Oguz, C. (2011). Berth and quay crane allocation: a moldable task scheduling model. *Journal of the Operational Research Society*, *62*(7), 1189–1197.

Carlo, H. J., Vis, I. F. A., & Roodbergen, K. J. (2015). Seaside operations in container terminals: literature overview, trends, and research directions. *Flex Serv Manuf J*, *27*(1), 224–262.

Chang, D., Jiang, Z., Yan, W., & He, J. (2010). Integrating berth allocation and quay crane assignments. *Transportation Research Part E: Logistics and Transportation Review*, *46*(6), 975–990.

Elwany, M. H., Ali, I., & Abouelseoud, Y. (2013). A heuristics-based solution to the continuous berth allocation and crane assignment problem. *Alexandria Engineering Journal*, *52*(4), 671–677.

Han, X., Gong, X., & Jo, J. (2015). A new continuous berth allocation and quay crane assignment model in container terminal. *Computers and Industrial Engineering*, *89*, 15–22.

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.

Hsu, H. P. (2016). A HPSO for solving dynamic and discrete berth allocation problem and dynamic quay crane assignment problem simultaneously. *Swarm and Evolutionary Computation*, *27*, 156–168.

Iris, Ç., Pacino, D., Ropke, S., & Larsen, A. (2015). Integrated Berth Allocation and Quay Crane Assignment Problem: Set partitioning models and computational results. *Transportation Research Part E: Logistics and Transportation Review*, *81*, 75–97.

Karam, A., & Eltawil, A. B. (2016). Functional integration approach for the berth allocation, quay crane assignment and specific quay crane assignment problems. *Computers and Industrial Engineering*, *102*, 458–466.

Kim, K. H., & Moon, K. C. (2003). Berth scheduling by simulated annealing. *Transportation Research Part B: Methodological*, *37*(6), 541–560.

Lalla-Ruiz, E., González-Velarde, J. L., Melián-Batista, B., & Moreno-Vega, J. M. (2014). Biased random key genetic algorithm for the Tactical Berth Allocation Problem. *Applied Soft Computing Journal*, *22*, 60–76.

Le, M., Wu, C., & Zhang, H. (2012). An integrated optimization method to solve the berth-QC allocation problem. In *2012 8th International Conference on Natural Computation*, (pp. 753–757). IEEE.

Li, W., Wu, Y., & Goh, M. (2015). Planning and Scheduling for Maritime Container Yards. In *Planning and Scheduling for Maritime Container Yards: Supporting and Facilitating the Global Supply Network*, chap. 2, (pp. 1–110). Cham: Springer International Publishing.

Liang, X., Li, W., Zhao, W., & Li, B. (2012). Multistage collaborative scheduling of berth and quay crane based on heuristic strategies and particle swarm optimization. In *Proceedings of the 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, (pp. 913–918). IEEE.

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.

Meisel, F., & Bierwirth, C. (2013). A Framework for Integrated Berth Allocation and Crane Operations Planning in Seaport Container Terminals. *Transportation Science*, *47*(2), 131–147.

Park, Y., & Kim, K. (2003). A scheduling method for berth and quay cranes. *OR Spectrum*, (25), 1–23.

Raa, B., Dullaert, W., & Schaeren, R. V. (2011). An enriched model for the integrated berth allocation and quay crane assignment problem. *Expert Systems with Applications*, *38*(11), 14136–14147.

Rashidi, H., & Tsang, E. P. K. (2013). Novel constraints satisfaction models for optimization problems in container terminals. *Applied Mathematical Modelling*, *37*(6), 3601–3634.

Rodriguez-Molins, M., Salido, M. A., & Barber, F. (2014). Robust Scheduling for Berth Allocation and Quay Crane Assignment Problem. *Mathematical Problems in Engineering*, *2014*.

Stahlbock, R., & Voß, S. (2008). Operations research at container terminals: A literature update. *OR Spectrum*, *30*(1), 1–52.

Türkoğullari, Y. B., Taşkin, Z. C., Aras, N., & Altinel, I. K. (2014). Optimal berth allocation and time-invariant quay crane assignment in container terminals. *European Journal of Operational Research*, *235*(1), 88–101.

Türkoğullari, Y. B., Taşkin, Z. C., Aras, N., & Altinel, I. K. (2016). Optimal berth allocation, time-variant quay crane assignment and scheduling with crane setups in container terminals. *European Journal of Operational Research*, *254*(3), 985–1001.

UNCTAD (2017). *Review of Maritime Transport*. United Nations Conference on Trade and Development.

Ursavas, E. (2014). A decision support system for quayside operations in a container terminal. *Decision Support Systems*, *59*(1), 312–324.

Xuelian, C., & Zhiying, Y. (2012). An Algorithm for Continuous Berth Allocation with Quay Crane Dynamic Allocation. In *2012 Fifth International Conference on Intelligent Computation Technology and Automation*, (pp. 541–544). IEEE.

Yang, C., Wang, S., & Zheng, J. (2013). The Allocation of Berth and Quay Crane by Using a Particle Swarm Optimization Technique. *Lecture Notes in Electrical Engineering*, *254*, 779–787.

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

Zhang, C., Zheng, L., Zhang, Z., Shi, L., & Armstrong, A. J. (2010). The allocation of berths and quay cranes by using a sub-gradient optimization technique. *Computers & Industrial Engineering*, *58*(1), 40–50.