



Innovative Applications of O.R.

A Set-Partitioning-based model for the Berth Allocation Problem under Time-Dependent Limitations



Eduardo Lalla-Ruiz*, Christopher Expósito-Izquierdo, Belén Melián-Batista,
J. Marcos Moreno-Vega

Department of Computer and Systems Engineering, University of La Laguna, 38271 La Laguna, Santa Cruz de Tenerife, Spain

ARTICLE INFO

Article history:

Received 9 September 2014

Accepted 13 October 2015

Available online 19 October 2015

Keywords:

Berth Allocation Problem

Set Partitioning

Optimization model

Maritime container terminal

ABSTRACT

This paper addresses the Berth Allocation Problem under Time-Dependent Limitations. Its goals are to allocate and schedule the available berthing positions for the container vessels arriving toward a maritime container terminal under water depth and tidal constraints. As we discuss, the only optimization model found in the literature does not guarantee the feasibility of the solutions reported in all the cases and is limited to a two-period planning horizon, i.e., one low tide and one high tide period. In this work, we propose an alternative mathematical formulation based upon the Generalized Set Partitioning Problem, which considers a multi-period planning horizon and includes constraints related to berth and vessel time windows. The performance of our optimization model is compared with that of the mathematical model reported in the related literature. In this regard, the computational experiments indicate that our model outperforms the previous one from the literature in several terms: (i) it guarantees the feasibility and optimality of the solutions reported in all the cases, (ii) reduces the computational times about 88 percent on average in the problem instances from the literature, and (iii) presents reasonable computational times in new large problem instances.

© 2015 Elsevier B.V. and Association of European Operational Research Societies (EURO) within the International Federation of Operational Research Societies (IFORS). All rights reserved.

1. Introduction

Nowadays, maritime container terminals have to face increasingly rough requirements from shipping companies which claim real-time services. This pressure is even higher for terminal managers due to the fierce competitiveness among terminals for being selected as exchange points (hubs) or origin-destination points in shipping routes. Thus, maritime container terminals have to improve their management capabilities with the objective of enhancing their productivity. In this regard, an inefficient use of resources is translated into a bottleneck or a delay of the yard-side and land-side operations. This fact gives rise to a low overall productivity of the maritime container terminal. Notteboom (2006) indicates that 93.6 percent of the delays in the vessel schedules are attributable to terminal operations. In this context, the berth allocation operation can be highlighted as one of the most important ones, since it has a significant impact on terminal productivity (Beškovnik, 2008). Moreover, one of the main indicators used by the shipping companies to measure the performance of a certain maritime container terminal is the berth productivity, in

terms of dwell time on berths and waiting time for a free berth (Yeo, 2010). Therefore, it becomes crucial for the container terminals to efficiently use the berths with the goal of improving their productivity and making them more attractive.

The berthing process consists of several phases carried out sequentially. Once a container vessel arrives to a port, a berth is assigned to it according to its particular features (e.g., dimensions, stowage plan, contractual agreements, etc.). Once the vessels are berthed, their containers are unloaded and transferred to the yard, where they are stored temporally. Finally, the containers are loaded into trucks, trains, or other vessels to continue throughout their route. Exhaustive descriptions of the processes within each phase can also be consulted in the works by Steenken, Voß, and Stahlbock (2004), Stahlbock and Voß (2008), and Rashidi and Tsang (2013).

Complex planning and management problems arise in each one of the aforementioned phases. In this regard, the inherent interrelation among the operations involves to treat the problems arisen at each phase and integrate them within functional schemes (i.e., by preprocessing or by feedback). Some examples of schemes can be consulted in the works Meisel and Bierwirth (2006), Meisel (2009), Meisel and Bierwirth (2013), and Lalla-Ruiz, Expósito-Izquierdo, Melián-Batista, and Moreno-Vega (2013). Thus, the effective scheduling of the isolated operations arising at each phase such as berth allocating

* Corresponding author.

E-mail addresses: elalla@ull.es (E. Lalla-Ruiz), cexposit@ull.es (C. Expósito-Izquierdo), mbmelian@ull.es (B. Melián-Batista), jmmoreno@ull.es (J.M. Moreno-Vega).

incoming vessels may lead to an improvement of the overall performance of the terminal. This can be seen in the fact that a poor utilization of berths affects the yard-side (e.g., container arrival/departure schedule, container transshipment operations) and land-side operations (e.g., train or truck container pick-up schedule). Hence, this would result in a low overall productivity of the container terminal, and therefore dissatisfaction of the shipping companies.

The previous discussion leads to the definition of a logistic problem known as Berth Allocation Problem. Its objective is to assign berthing positions along the quay to incoming container vessels. As indicated by Imai, Nishimura, and Papadimitriou (2001a), the berth allocation problem may be frequently solved since berth space is very limited at most container terminals as well as the high interrelation with other related problems (Meisel, 2009), and thousands of containers must be handled daily. Thus an effective berth allocation is crucial (i) for the efficient management of the container flows, (ii) minimize ships waiting time, and (iii) maximize port's turnaround (Rashidi & Tsang, 2013).

When determining the berth schedule, the maritime container terminal has to consider several factors, such as the vessel and berth time windows, number of loaded/unloaded containers, water depth, and tide constraints. In this regard, those maritime container terminals strongly affected by the tide, the water depth constraints have to consider the specific berth depth, vessel drafts, and tidal conditions over time. Therefore, in order to allocate incoming vessels to berthing positions, terminal managers have to take into account not only berth depths and vessel drafts, but also their arrival and berthing time. In this regard, the available depth at low tide may not be adequate for berthing some vessels since it can give rise to accidents or delays. In other situations, the vessels may require high tide for going through certain sections of their route when they are fully loaded. Moreover, empty vessels may not be able to pass under bridges at high tide. In this context, container terminal managers and shipping companies are aware that a small delay in the service for awaiting proper tidal conditions may be amplified to a greater delay. This fact may result in outstanding economical losses and competitiveness.

The aforementioned issues give rise to the definition of the Berth Allocation Problem under Time-Dependent Limitations (BAPTL), which was introduced by Xu, Li, and Leung (2012). In this problem, the water depth of the berths (subject to the tidal conditions), the draft of the vessels, and the priorities of the vessels are considered.

The main contributions of this paper are the following:

- The optimization model proposed by Xu et al. (2012) for the BAPTL is limited to only a two-period planning horizon, i.e., one low tide period and one high tide period. However, as discussed in Section 3.2, when a longer planning horizon is required for providing a solution, this model extends the second period, either low-tide or high-tide to infinity. This may produce infeasible solutions in practice due to the fact that some vessels can be assigned to infeasible positions. On the other hand, in some cases, those infeasible solutions may have a better objective value than feasible ones within the two-period time horizon. Thus, in this work, we address this by proposing additional constraints to ensure the feasibility of the solutions.
- Developing a mathematical model based upon the Generalized Set-Partitioning Problem (GSPP), and termed GM-BAPTL for the BAPTL that considers a multi-period planning horizon. As discussed in this paper, this model overcomes those scenarios in which the model proposed by Xu et al. (2012) reports infeasible solutions. The computational experiments indicate that GM-BAPTL is able to solve all the problem instances from the literature to optimality in less than 0.5 second, which constitutes a reduction of about 88 percent in the computational time.
- Including container vessel and berth time windows in the definition of the BAPTL. This variant of the problem is referred to as

Berth Allocation Problem under Time-Dependent Limitations and Time-Windows (BAPTL-TW), whereas the variant of GM-BAPTL for addressing the BAPTL-TW is termed GM-BAPTL-TW. In order to evaluate the performance of those set partitioning aimed at solving BAPTL and BAPTL-TW, a benchmark suite is proposed.

The remainder of this paper is organized as follows. The literature review is presented in Section 2. The BAPTL is introduced in Section 3. The Generalized Set-Partitioning-based mathematical models for the BAPTL and BAPTL-TW are provided in Section 4. Afterwards, the computational experiments carried out in this work are summarized in Section 5. Finally, the main conclusions extracted from the work and several lines for further research are drawn in Section 6.

2. Literature review

A considerable number of papers concerning the Berth Allocation Problem (BAP) have been published in the literature over the last years. It has been studied in the literature at operational, tactical, and strategic levels. At operational level, the time horizon goes from one day to one week and the BAP at this level is aimed at optimizing the delays and waiting times of container vessels as well as maximize port's turnaround. The tactical level covers tasks ranging from one week up to several months. Some of the BAP objectives at this level aims to optimize the transshipment flows among vessels, cycling visiting of the vessels, fulfillment of contracts among shipping companies and terminal managers, route design, etc. The planning at strategic level has the longest horizon, which goes from one up to several years. In this level, the decisions tackled are related to the market and trade selection, establish specific and dedicated berths, strategic cooperation agreements between terminal and shipping companies, etc. For a more detailed description, the reader is referred to the works by Christiansen, Fagerholt, Nygreen, and Ronen (2007), Vis and de Koster (2003), and Bierwirth and Meisel (2010).

Due to the large variety of maritime terminal layouts, research has produced a multitude of variants for the BAP. According to the classification proposed by Bierwirth and Meisel (2010), the BAP can be distinguished between the static version (SBAP) and the dynamic version (DBAP). In the static version, vessels are in port before starting the planning horizon. Imai, Nagaiwa, and Tat (1997) study this approach. The SBAP can be reduced to a classical assignment problem, which is known to be polynomially-solvable (see Pinedo, 2008). The DBAP is presented as an extension of the SBAP by Imai, Nishimura, and Papadimitriou (2001b), in which vessels can arrive at any moment of the planning horizon. Since the difficulty when finding an optimal solution accurately, they develop a heuristic based on a sub-gradient method with a Lagrangian relaxation. Monaco and Sammarra (2007) present a stronger formulation for the DBAP and develop a Lagrangian relaxation with a non-standard multiplier adjustment method for solving it. Nishimura, Imai, and Papadimitriou (2001) extend the DBAP by considering different water depths and Imai, Nishimura, and Papadimitriou (2003) also include vessels priorities. Cordeau, Laporte, Legato, and Moccia (2005) introduce two formulations for the DBAP, where the major difference lies on the consideration of the quay. While in one formulation a discrete quay is considered, in the other one a hybrid approach is studied. In order to solve them, two Tabu Search algorithms are proposed. Buharkal, Zuglian, Ropke, Larsen, and Lusby (2011) review the three main mathematical formulations for the DBAP. They highlight that the most appropriate one in terms of quality of the solutions and time requirement is based on a Generalized Set Partitioning Problem proposed by Christensen and Holst (2008). Lalla-Ruiz, Melián-Batista, and Moreno-Vega (2012) address the discrete case of the DBAP by proposing an effective and efficient Tabu Search with Path-Relinking. The work by de Oliveira, Mauri, and Lorena (2012) presents a Clustering Search (CS-SA) with Simulated Annealing for solving the DBAP.

This algorithm provides optimal solutions in all the largest instances proposed by Cordeau et al. (2005). In this regard, Ting, Wu, and Chou (2014) propose a Particle Swarm Optimization algorithm for addressing the DBAP, which is able to obtain optimal solutions by shorter computational times than CS-SA. Moreover, Hansen, Oguz, and Mladenovic (2008) take into account the costs for waiting and handling as well as earliness or tardiness of completion with the purpose of including priorities. Also, they consider a handling cost associated with each berth that varies depending on which berth is used. A Variable Neighborhood Search is developed for addressing it.

Concerning spatial constraints, the BAP can be mainly classified as discrete, continuous, or hybrid. For further details the reader is referred to the work by Bierwirth and Meisel (2010). The discrete case presents a quay divided into sections called berths. Some related works that address this variant are those presented by Imai et al. (2001b), Nishimura et al. (2001), Cordeau et al. (2005), and Lalla-Ruiz, González-Velarde, Melián-Batista, and Moreno-Vega (2014). In the continuous case, there is no division of the quay, so that any incoming vessel can be assigned to the quay taking into account its spatial measures. Some works related to this consideration are due to Lim (1998), Park and Kim (2002), and Wang and Lim (2007). In the hybrid version of the BAP, the quay is divided into berths, but a vessel can occupy more than a single berth or share it with other container vessels. Other spatial constraints are described by Cordeau et al. (2005), Imai, Chen, Nishimura, and Papadimitriou (2008) and Cheong, Tan, Liu, and Lin (2010).

Although some of the above considerations regarding physical factors consider water depth (Cheong & Tan, 2008; Cheong et al., 2010; Guldogan, Bulut, & Tasgetiren, 2012; Nishimura et al., 2001), only a few works in the literature take into account tidal constraints. This is the case of the works by Barros, Costa, Oliveira, and Lorena (2011) and Xu et al. (2012).

Barros et al. (2011) develop a berth allocation model with tidal time windows in bulk ports, where the vessels are allocated mainly considering a set of tidal time windows. They propose a set of instances for this problem, in which most of them include a single berth, whereas only one problem instance considers three berths. For addressing the problem, the authors propose a Simulated Annealing-based algorithm, which solves the instances in a reasonable computational time and provides high-quality solutions. The authors highlight the benefits of this approach to be used in a decision support system in the terminal of São Luís (Brasil).

Xu et al. (2012) consider the Berth Allocation Problem under Time-Dependent Limitations (BAPTL) due to tidal conditions. The container vessels have to be allocated to berths considering different tide periods, which cause changes in the water depth of the berths. They formulate the BAPTL limited to a two-period planning horizon as a Mixed Integer Linear Programming model. Additionally, they propose a set of instances based on real data and an algorithmic procedure for solving them. However, despite the fact that only two periods are considered by these authors, some instances require more than two periods. In those cases, the mathematical model is not able to limit the planning horizon to two periods and extend the second period to infinity. As discussed in Section 3.2, this implies that the model provides infeasible solutions in the majority of the instances under analysis.

3. Berth Allocation under Time-Dependent Limitations

The Berth Allocation Problem under Time-Dependent Limitations (BAPTL) was introduced by Xu et al. (2012). It seeks to determine the berthing position and berthing time for those container vessels arriving to a maritime container terminal considering external limiting factors, such as water depth and tidal constraints in such a way that the service time of each vessel is minimized.

In the BAPTL, the time horizon is divided into regular intervals by a parameter T , which represents the time point where the tide level has reached a certain threshold. The resulting intervals represent the low and high tide periods. Namely, a low-water period (LW), defined in $[0, T]$ and a high-water period (HW), defined in $[T, \infty)$. According to Xu et al. (2012) the second time period is extended to the infinity, that is, it does not have a superior time limit. This assumption leads to infeasible solutions as discussed in Section 3.2. Despite the fact that these intervals can be swapped in order to have other scenarios, i.e., LW-HW or HW-LW, the computational results presented by Xu et al. (2012) show that swapping intervals from low-tide to high tide or vice versa for obtaining new scenarios does not increase or change the complexity of the problem. Furthermore, in the BAPTL there is a set of incoming container vessels $N = \{1, 2, \dots, n\}$ and a set of available berths $M = \{1, 2, \dots, m\}$. For each vessel $j \in N$, there is a given service time p_j , arrival time a_j , weight priority v_j , high-water berth index H_j and low-water berth index L_j , where $L_j, H_j \in \{1, 2, \dots, m\}$. Regarding this notation, a vessel $j \in N$ in high-water period can be assigned to berths $H_j, H_j + 1, \dots, m$. Similarly, during a low-water period, a vessel j can be assigned to berths $L_j, L_j + 1, \dots, m$. According to this, in high and low-water periods, vessels cannot be assigned to berths $1, 2, \dots, H_j - 1$ / $L_j - 1$, respectively. It should be noted that the tide level and vessel draft are taken into account in the L_j and H_j parameters.

3.1. Mixed Integer Linear Programming formulation for the two-period BAPTL

With the aim of making this paper self-contained and for including additional constraints to the already proposed model, in the following we report the mathematical formulation proposed by Xu et al. (2012) for the BAPTL. We will refer to this model as M-BAPTL. The following notations are used.

N	Set of n container vessels
M	Set of m berths
T	Time point at which the tide level has reached a certain threshold
L_j	Low-water berth index, $L_j \in \{1, 2, \dots, m\}$, $j \in N$
H_j	High-water berth index, $H_j \in \{1, 2, \dots, m\}$, $j \in N$
p_j	Processing time of container vessel $j \in N$
a_j	Arrival time of container vessel $j \in N$
v_j	Weight priority of container vessel $j \in N$

The decision variables are shown below:

x_{ij}	$\in \{0, 1\}$, equal to 1 if container vessel j is assigned to berth i , and 0 otherwise.
$I_{ijj'}$	$\in \{0, 1\}$, equal to 1 if container vessels j and j' ($j \neq j'$) are both assigned to berth i and vessel j is processed before vessel j' , and 0 otherwise.
s_j	Starting time of the processing of container vessel j

The M-BAPTL is stated as follows:

$$\min \sum_{j \in N} v_j (s_j + p_j - a_j) \quad (1)$$

subject to

$$\sum_{i \in M} x_{ij} = 1, \quad \forall j \in N \quad (2)$$

$$s_j \geq a_j, \quad \forall j \in N \quad (3)$$

$$s_{j'} \geq s_j + p_j - M(1 - I_{ijj'}), \quad \forall j, j' \in N, \text{ s.t. } j \neq j', \forall i \in M \quad (4)$$

$$I_{ijj'} + I_{ij'j} \leq \frac{1}{2}(x_{ij} + x_{ij'}), \quad \forall j, j' \in N, \text{ s.t. } j < j', \forall i \in M \quad (5)$$

$$I_{ijj'} + I_{ij'j} \geq x_{ij} + x_{ij'} - 1, \quad \forall j, j' \in N, \text{ s.t. } j < j', \forall i \in M \quad (6)$$

$$x_{ij} = 0, \quad \forall j \in N, (i = 1, 2, \dots, H_j - 1) \quad (7)$$

$$s_j \geq T x_{ij}, \quad \forall j \in N, (i = 1, 2, \dots, L_j - 1) \quad (8)$$

$$x_{ij} \in \{0, 1\}, \quad \forall j \in N, \forall j \in M \quad (9)$$

$$l_{ijj'} \in \{0, 1\}, \quad \forall j, j' \in N, \text{ s.t. } j \neq j', \forall j \in M \quad (10)$$

The objective function (1) minimizes the weighted sum of turnaround times of the incoming container vessels. Constraints (2) ensure that each container vessel is assigned to a single berth. Constraints (3) guarantee that the processing time of each container vessel starts after its already arrived time at the terminal. Constraints (4) state that if vessels j and j' are both assigned to berth i and vessel j is processed before vessel j' (i.e., $l_{ijj'} = 1$), then, the starting time of vessel j' cannot be earlier than $s_j + p_j$. Constraints (5) and (6) ensure that one of $l_{ijj'}$ and $l_{ij'j}$ is equal to 1 if vessels j and j' are both assigned to berth i . They also ensure that $l_{ijj'} = l_{ij'j} = 0$ if one of vessels j and j' is not assigned to berth i . Constraints (7) do not allow vessel j to be assigned to berths $1, 2, \dots, H_j - 1$. Constraints (8) do not allow vessel j to be assigned to berths $1, 2, \dots, L_j - 1$ during period $[0, T]$. Finally, constraints (9) and (10) are the integrability constraints for the different kinds of variables.

3.2. Improving the M-BAPTL Formulation

According to Xu et al. (2012), the M-BAPTL is limited to only two single periods, i.e., one low-water and high-water period. However, when the departure time of a container vessel exceeds the two-period planning horizon, the M-BAPTL reports infeasible solutions in some cases, since it extends the second period (either low or high tide period) to infinity. For example, considering the LW-HW scenario, if a vessel is appropriately berthed during the second period (high-water) and its departure exceeds the end of the second period. Then, according to M-BAPTL, the solution would still be feasible, even though the draft of the vessel is greater than the berth water depth due to the change of tide (which returned to low-water). An illustrative example of this can be consulted in Section 5.1.1, where the optimal solution provided by M-BAPTL for a problem instance is depicted. Moreover, in some cases, although an appropriate optimal solution can be obtained during the two-period time horizon, M-BAPTL can provide an infeasible optimal solution if it presents a better objective value. That is the case in which a vessel stay goes beyond the two-period time horizon and the berth where it is allocated turns to be infeasible, and the objective value of that solution presents a better objective function value than a feasible solution within the two-period horizon. Finally, it should be noted that in other cases, the solution may remain feasible if the vessel draft is lower than the berth water depth at low tide. In the following, we provide some ways to overcome this issue.

1. *Two-period limiting constraints*: This set of constraints is aimed at restricting the planning horizon to only two periods in order to enable M-BAPTL to solve to optimality or prove that no solution exists within that horizon. Moreover, these constraints are useful for discarding some cases where, as previously commented, besides an appropriate optimal solution can be obtained during the two-period time horizon, the M-BAPTL provides an infeasible optimal solution since it reports a better objective value than the feasible solution within the two periods. In this regard, the problem instances presented by Xu et al. (2012) are not appropriate since the majority of them require a longer time horizon. This can be easily deduced from the instance data of one of their instances reported in Section 5.1.1, where the sum of the available time of the berths equals to 72 for the two-period time horizon is lower than the sum of service times of the vessels without even taking into account their arrival times which is equal to 77.

Constraints (11) state that each vessel $j \in N$ has to be processed before the end of the two-period planning horizon, T :

$$s_j + p_j \leq 2 \cdot T \quad \forall j \in N \quad (11)$$

By adding these constraints to the M-BAPTL formulation, it will be able to ensure feasible solutions or prove that no solution exists for those instances where the required planning horizon is longer than two periods. In the remainder of this paper, the mathematical formulation including these constraints is referred to as M*-BAPTL.

2. *Constraints for providing suboptimal solutions*: The constraints proposed in the previous point only enable M-BAPTL to provide feasible solutions or prove that no solution exists for the two-period planning horizon. However, for practical matters, in those cases where no solution can be provided by M-BAPTL due to the planning horizon (i.e., a large number of vessels to be allocated beyond two periods), a feasible schedule despite of the number of periods could be required in practice. Hence, additional constraints aimed at allowing M-BAPTL to provide feasible suboptimal solutions regardless of the number of periods are required. With this goal in mind, we propose new constraints that allow M-BAPTL to provide suboptimal solutions in those scenarios where M*-BAPTL will not provide any solution. To do that, the constraints from the formulation of M-BAPTL (7) and (8) should be removed, and the following constraints (12) should be included:

$$x_{ij} = 0, \quad \forall j \in N, (i = 1, 2, \dots, \max(L_j, H_j) - 1) \quad (12)$$

Constraints (12) state that each vessel $j \in N$ can be berthed at one of those berths that allows a safety service regardless of the situation of the tide.

4. BAPTL Generalized Set-Partitioning Approach

In this work, we propose an alternative mathematical formulation for the BAPTL termed GM-BAPTL and based upon the Generalized Set-Partitioning Problem (GSPP). Through this reformulation the previous weaknesses of M-BAPTL are addressed, allowing to obtain a feasible solution in all cases by appropriately considering the change of tides. In the related literature, several optimization problems such as Aircrew Scheduling Problem, Network Design Problem, Vehicle Routing Problem, Berth Allocation Problem, etc., have been already formulated as a Set Partitioning Problem.

GM-BAPTL is described as follows. There is a set of columns denoted as Ω that composes the GSPP matrix. Each column represents a feasible berthing assignment of a single container vessel in accordance with the constraints established by the problem (see Section 3.1). This matrix is composed of two sub-matrices, denoted as A and B , both containing $|\Omega|$ columns. Matrix $A = (A_{i\omega})$ contains a row for each vessel, and $A_{i\omega} = 1$, if and only if column ω represents a feasible assignment of vessel $i \in N$. Therefore, the time-dependent constraints and the vessel drafts are then considered. Each column of A contains exactly one non-zero element. Matrix $B = (B_{p\omega})$ contains a row per (berth, time) position. The rows of B are indexed by the elements belonging to set P by means of time steps of the problem time horizon. Since in BAPTL the planning horizon is not directly specified in the parameters of the problem, it can be calculated by means of a mathematical method or by considering the worst possible case. That is,

$$|P| = \left(\left(\sum_{i \in N} p_i \right) + \max(a_i)_{i \in N} \cdot \lambda \right) \cdot m, \quad (13)$$

where λ is the number of tide indexes, i.e., $\lambda = 2$ since we have two indexes, one for low-tide (L) and one for high-tide (H). The entry $B_{p\omega}$ is equal to 1, if and only if, a position $p \in P$ is contained in the assignment that column ω represents and the related container vessel

is able to berth in the specific time condition. As can be checked $|P|$ depends on the number of vessels, number of tide indexes as well as arrival and processing times. The cost c_ω of any column $\omega \in \Omega$ is the service time of the respective position assignment and it is multiplied by the priority factor w_j . A binary decision variable x_ω is equal to 1 if column ω is used in the solution, and 0 otherwise. Therefore, GM-BAPTL is stated as follows:

$$\min \sum_{\omega \in \Omega} c_\omega x_\omega \quad (14)$$

subject to

$$\sum_{\omega \in \Omega} A_{i\omega} x_\omega = 1, \quad \forall i \in N \quad (15)$$

$$\sum_{\omega \in \Omega} B_{p\omega} x_\omega \leq 1, \quad \forall p \in P \quad (16)$$

$$x_\omega \in \{0, 1\}, \quad \forall \omega \in \Omega \quad (17)$$

The objective function (14) minimizes the service time of the container vessels. The set of constraints (15) ensures that all the vessels are served. Finally, the constraints (16) guarantee that at a time interval, in a berth, only one vessel can be served. Constraints (17) are standard binary restrictions on the decision variables.

Table 1 (a) illustrates an example of GM-BAPTL matrix for a problem instance composed of $n = 3$ container vessels, $m = 2$ berths, and a prefixed time horizon of 4 time steps. Times 1 and 2 correspond to the low tide, whereas times 3 and 4 to the high tide. In order to ease the understanding of this example, it has been included a column that delimits the sub-matrices A and B explained above. Table 1(b) summarizes the arrival time, a_j , processing time, p_j , and draft of each container vessel. Regarding the time point T when the tide changes, in this example it is set to 2. Table 1(c) indicates the possible berths for each vessel depending on the state of the tide. Additionally, Table 1(d) indicates the water depth of the berths during low-tide (wd_L) and high-tide (wd_H). As can be seen, $|\Omega| = 13$; that is, 13 feasible single assignments are included when filling the matrix.

Since container vessel 1 can be assigned to any berth ($L_1 = 1$ and $H_1 = 1$) regardless of the state of the tide, all the possible schedules are included for the planning horizon (columns 1–6). The tidal indexes of container vessel 2 (L_2 and H_2) indicate that at low tide this vessel can only be allocated to berth 2, while at high tide vessel 2 can berth at berths 1 or 2 (columns 7–10). Concerning vessel 3, it is unable to be berthed at any time in berth 1 ($L_3 = H_3 = 2$). Hence, all its possible assignments are located at berth 2 (columns 11–13).

As discussed in Section 3.2, the mathematical model proposed by Xu et al. (2012) is limited to a two-period planning horizon. Nevertheless, GM-BAPTL allows to consider a planning horizon equal to or larger than two periods. This issue has to be taken into account only when creating the sub-matrices A and B.

In this regard, we propose the following statements to be considered when filling the GSPP matrix. Through these statements, selecting feasible berths according to the tidal indexes L_j and H_j when creating the matrices A and B is ensured.

- If $T > \text{mod}(s_j, 2T) \geq 0$ and $p_j \leq (T - \text{mod}(s_j, 2T))$ for a vessel $j \in N$, then the index corresponding to $[0, T]$ is considered.
- If $T > \text{mod}(s_j, 2T) \geq 0$ and $p_j > (T - \text{mod}(s_j, 2T))$ for a vessel $j \in N$, then the index considered is the most restrictive one, that will be the one corresponding to low-water period.
- If $2T > \text{mod}(s_j, 2T) \geq T$ and $p_j \leq (2T - \text{mod}(s_j, 2T))$ for a vessel $j \in N$, then the index corresponding to $[T, 2T]$ is considered.
- If $2T > \text{mod}(s_j, 2T) \geq T$ and $p_j > (2T - \text{mod}(s_j, 2T))$ for a vessel $j \in N$, then the index considered is the most restrictive one, that will be the one corresponding to low-water period.

As can be checked, with these statements when filling the GSPP matrix allows to solve the BAP-TL also for more than two-periods.

Algorithm 1: Procedure for determining the number of columns

```

1 Initialize the total number of columns,  $n_{Col} \rightarrow 0$ 
2 for  $(i=1,2,\dots,m)$  do
3   for  $(j=1,2,\dots,n)$  do
4     for  $(t=a_j, a_j+1, \dots, |P| - p_j)$  do
5       if  $(\text{mod}(t, 2T) \geq T)$  then
6         possibleBerth =  $H_j$ 
7       else
8         possibleBerth =  $L_j$ 
9       if  $(i \geq \text{possibleBerth})$  then
10        if  $(\text{mod}(t+p_j-1, 2T) \geq T)$  then
11          possibleBerth =  $H_j$ 
12        else
13          possibleBerth =  $L_j$ 
14        if  $(i \geq \text{possibleBerth})$  then
15          Generate column for vessel  $j$  at berth  $i$  starting
            in time step  $t$ 
16          Increase the total number of columns,  $n_{Col}$ , one
            unit
17 return  $n_{Col}$ 
  
```

It should be noted that in the case of M*-BAPTL, it is restricted to two-period if the user wants to solve to optimality using the model proposed by Xu et al. (2012) or solve to a sub-optimal solution (not restricted to two periods) using the model (Xu et al., 2012) with the additional constraints proposed in point (2) of Section 3.2.

Furthermore, these statements together with the calculation of P are used when determining the GSPP matrix. Its dimensions depend on $|P| + n$ and the number of columns, $|\Omega|$. In this regard, P is given by Eq. (13) and it depends on the number of vessels, the number of berths and processing times. However, the number of columns is determined by all the possible assignments of each vessel. Therefore, in order to obtain the size of the problem correctly, the number of columns has to be determined, thus, Algorithm 1 is proposed. Through it, the number of columns $|\Omega|$, termed as n_{Col} , is determined. This value is obtained considering all the possible allocations of each vessel $j \in N$ at each berth $i \in M$ starting from their arrival time $t = a_j$ and covering the complete time horizon $|P|$.

4.1. BAPTL with time-windows

In the original formulation proposed by Xu et al. (2012), berth and container vessel time windows constraints are not considered. However, as pointed out by Yang, Zhang, and Lam (2013), Ting et al. (2014), and Moorthy and Teo (2006), in practical applications the container vessels and berths may have time windows. These time windows may be due to contractual agreements between the port and shipping companies. For instance, maximum possible time that a container vessel can stay at the terminal without prejudicing its performance, or other contractual conditions with additional container terminals. Furthermore, regarding the berths, they may have time windows for establishing their availability, which depends on the terminal planning. In this regard, numerous container terminals are closed for cargo handling during nights or weekends (Christiansen et al., 2007). This feature becomes very significant when cargo handling time is longer than one working day or it is scheduled at the end of the week. This could mean that the vessel would have to stay idle much of the time in port. Therefore, the container vessel service time

Table 1

Example of the BAPTL using GM-BAPTL for $n = 3$ container vessels, $m = 2$ berths, and a time horizon of 4 time steps.

(a) GSPP Matrix														
		x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}
	Cost	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_{12}	c_{13}
A	Vessel 1	1	1	1	1	1	1							
	Vessel 2							1	1	1	1			
	Vessel 3											1	1	1
B	Berth 1 / time 1	1												
	Berth 1 / time 2	1	1											
	Berth 1 / time 3		1	1				1						
	Berth 1 / time 4			1				1						
	Berth 2 / time 1				1				1			1		
	Berth 2 / time 2				1	1			1	1		1	1	
	Berth 2 / time 3						1			1	1			1
	Berth 2 / time 4						1				1			

(b) Characteristics of the container vessels

Vessel	a_j	p_j	Draft (m.)
1	1	2	12.0
2	1	2	15.0
3	2	1	16.5

(c) Vessels tidal indexes

Vessel	L_j	H_j
1	1	1
2	2	1
3	2	2

(d) Berth water depths

Berth	wd_L (m.)	wd_H (m.)
1	14.0	16.0
2	17.0	18.0

will depend not only on their arrival and handling times, but also on the schedule of the berths. Consequently, due to the importance and influence of the berths and vessels time windows in the management of the terminal, this paper proposes additional constraints for GM-BAPTL. This new problem is referred to as Berth Allocation Problem under Time-Dependent Limitation with Time Windows (BAPTL-TW). Moreover, since the BAPTL-TW lies within the operational level, the time window constraints are considered within the complete time horizon.

The BAPTL-TW includes the following notations as problem data for including the time windows:

- d_j Maximum departure time of vessel j
- b_{a_i} Start of berth i availability
- b_{d_i} End of the berth i availability

The time windows corresponding to berth $i \in M$ and vessel $j \in N$ are $[b_{a_i}, b_{d_i}]$ and $[a_j, d_j]$, respectively. It should be mentioned that the mathematical formulation for the BAPTL-TW is the same as GM-BAPTL (see Section 4) since the time windows only affect when creating the matrices A and B . However, in this case the number of rows, P , of matrix B can be set to $|P| = \sum_{i \in M} (b_{d_i} - b_{a_i})$.

As stated above, the time windows in BAPTL-TW are only taken into account when creating the sub-matrices A and B . Hence, when generating the Ω columns the following constraints are taken into account:

$$s_j + p_j \leq d_j \quad \forall j \in N \quad (18)$$

$$s_j + p_j \leq b_{d_i} \quad \forall j \in N, \forall i \in M \quad (19)$$

$$(s_j - b_{a_i}) \cdot x_{ij} \geq 0 \quad \forall j \in N, \forall i \in M \quad (20)$$

Constraints (18) establish the maximum possible departure time for the vessels, whereas the berths time windows are defined by constraints (19) and (20). Note that constraints (20) are not linear, nevertheless, these constraints are only considered when filling the A and B matrices. Finally, as can be checked, the time windows considered are hard in the sense that only feasible columns satisfying these constraints are generated. Therefore, in accordance with constraints (15) all the vessels have to be served within their time windows.

The GSPP matrix size to properly provide feasible solutions will depend on $|P + n|$ and the number of columns, $|\Omega|$. In this regard, for determining it, Algorithm 1 can be used by changing line 4 to $\max(a_j, b_{a_i})$ to indicate the starting point of all the allocations for each vessel $j \in N$.

Table 2

Numerical example for the BAPTL-TW.

(a) GSPP Matrix

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
	Cost	c_1	c_2	c_3	c_4	c_5	c_6	c_7
A	Vessel 1	1	1	1				
	Vessel 2				1	1		
	Vessel 3						1	1
B	Berth 1 / time 1	1						
	Berth 1 / time 2	1	1					
	Berth 1 / time 3		1					
	Berth 1 / time 4							
	Berth 2 / time 1			1	1		1	
	Berth 2 / time 2			1	1	1		1
	Berth 2 / time 3					1		
	Berth 2 / time 4							

(b) Vessels information

Vessel	p_j	Time Windows		Draft (m.)
		a_j	d_j	
1	2	1	3	12
2	2	2	4	15
3	1	2	3	16.5

(c) Vessels tidal indexes

Vessel	L_j	H_j
1	1	1
2	2	1
3	2	2

(d) Berths related tidal information

Berth	wd_L (m.)	wd_H (m.)	Time Windows	
			b_{a_i}	b_{d_i}
1	14	16	1	3
2	17	19	2	4

Table 2 illustrates an example of the BAPTL-TW for a problem instance composed of $n = 3$ container vessels, $m = 2$ berths, and a time horizon of 4 time steps. Table 2(b) reports the processing time p_j , arrival time a_j , maximum departure d_j , and draft of the container vessels. Table 2(c) shows the possible berths where each container vessel can berth depending on the state of the tide. Table 2(d) indicates the water depth of the berths during low-tide (wd_L) and high-tide (wd_H), and the time window of each berth (b_{a_i}, b_{d_i}). Regarding the time point in which the tide changes, we set $T = 2$. In this example, the total number of columns that represent all the single assignments of the vessels is $|\Omega| = 7$, which is shorter than the total number of columns generated for the same example without time windows depicted in Section 4. This is due to the constraints added caused by the time windows. Concerning the allocation of the container vessels, vessel 1 can be assigned regardless the state of the tide ($L_1 = H_1 = 1$). Hence, by considering its time window and the availability of the berths, the columns related to the feasible assignments of vessel 1 are defined in

Table 3

Description of the problem instances for assessing BAPTL. Note that sets that range from 3×09 to 8×24 correspond to the instances from the literature by Xu et al. (2012). The instance sets that range from 6×30 to 8×50 correspond to the instances proposed in this work.

Set	No. instances	Effect	m	n	T
3×09	10	Small	3	9	12
3×09	10	Big	3	9	12
4×12	10	Small	4	12	12
4×12	10	Big	4	12	12
5×15	10	Small	5	15	12
5×15	10	Big	5	15	12
6×18	10	Small	6	18	12
6×18	10	Big	6	18	12
7×21	10	Small	7	21	12
7×21	10	Big	7	21	12
8×24	10	Small	8	24	12
8×24	10	Big	8	24	12
6×30	5	Small	6	30	12
6×30	5	Big	6	30	12
7×40	5	Small	7	40	12
7×40	5	Big	7	40	12
8×50	5	Small	8	50	12
8×50	5	Big	8	50	12

columns 1–3. Container vessels 2 and 3 cannot be berthed in berth 1 during low tide ($L_2 = 2$, $L_3 = 2$). Although vessel 2 can be assigned to berth 1 during the high-water period ($H_2 = 1$), it cannot be allocated in that berth due to the time window of the berth ($b_{-a_1} < s_2 + p_2$). Finally, container vessel 3 is only able to berth in berth 2 due to the water depth.

5. Computational experiments

This section is devoted to present the computational experiments carried out with the goal of assessing the performance of the optimization model based on Set Partitioning proposed in this work for the BAPTL. Additionally, we also assess its performance when addressing the extension of the BAPTL with vessel and berth time windows introduced in Section 4.1. All the reported computational experiments presented hereunder have been conducted on a computer equipped with an Intel 3.16 GHz and 4 GB of RAM. Moreover, the optimization models have been implemented by using CPLEX 12.3¹ with an execution time limit of 1 hour for each problem instance.

In order to assess the performance of GM-BAPTL, the set of instances proposed by Xu et al. (2012) and a new set of instances generated in the same way as described in that work are used. Moreover, to consider the time windows constraints related to vessels and berths, a new problem data set including the vessel and berth time window parameters (d_j , b_{-a_i} , b_{-d_i}) as discussed in Section 4.1 have been added. These instances have been generated considering the ones from the work proposed by Cordeau et al. (2005).

Table 3 summarizes the characteristics of the problem instances tackled in this work. Column *Set* is used to differentiate the set of instances by means of size, column *Effect* shows the tidal influence regarding the impact on the container vessel assignment, and columns m and n indicate the number of berths and container vessels, respectively. Finally, the value of the time point where the tide level has reached a certain threshold is reported under the heading T . The distinction of the instances is as follows. The instance sets that range from 3×09 to 8×24 , correspond to the instances from the literature. The instance sets that range from 6×30 to 8×50 correspond to the instances proposed in this work.²

Table 4

Data of instance 0 from instance set 3×09 Big (Xu et al., 2012).

Vessel	a_j	p_j	v_j	L_j	H_j
1	12	3	8	3	1
2	9	12	8	3	1
3	6	10	2	3	1
4	0	12	8	1	1
5	9	8	6	3	1
6	2	10	4	2	1
7	12	4	10	1	1
8	12	11	9	2	1
9	10	7	1	3	2

5.1. Computational results for the literature instances for the two-period BAPTL

As discussed in Section 3.2, the majority of the instances proposed by Xu et al. (2012) are not appropriate for evaluating the performance of their mathematical model, M-BAPTL, and their proposed algorithm since those approaches extend the second period to infinity and the solutions may be infeasible. Therefore in this subsection, we firstly show an example of a solution provided by M-BAPTL to point out this issue. Moreover, with the aim of evaluating and compare their model and the one proposed in this work, GM-BAPTL, we have carried out a comparison for those problem instances where at least one feasible solution can be obtained within a planning horizon of two periods. Table 5 shows a list with the instances from those provided by Xu et al. (2012) that allow a feasible solution within the two-period planning horizon. Note that for some sets there is no feasible solution within the two-period horizon.

5.1.1. Example of an infeasible solution provided by M-BAPTL

In order to show the shortcoming of M-BAPTL, an optimal schedule reported by it for one of the instances used by Xu et al. (2012) is shown in Fig. 1. This instance (instance 3×09 —Big—instance 0) consists of $n = 9$ container vessels and $m = 3$ berths. Table 4 illustrates, for each vessel $j \in N$, its arrival time a_j , processing time p_j , weight priority v_j , and possible berths during the low-water and high-water periods L_j and H_j , respectively. In this solution, container vessel 9 is assigned to berth 1. However, from time step 24, as the tide level changes to low-water, the assignment of container vessel 9 to that berth is infeasible. The reason is found in that at this period the container vessel can only berth at berth 2 (L_9). This happens because constraints (8) are only limited to $[0, T]$. The same situation occurs for container vessel 8, since it is berthed at berth 0 during the high-water period. Unfortunately, this assignment turns infeasible from time step 24 due to the change of tide to low-water. In this period, this container vessel can only berth in berth 1. Clearly, the solution is infeasible in practice because both vessels are berthed during low-water period at infeasible berths.

Moreover, in Fig. 2 we report the optimal solution provided by GM-BAPTL for the same problem instance. It should be noted that M-BAPTL same model with additional constraints, M*-BAPTL, is able to prove that no feasible solution exists within the two-period planning horizon.

5.1.2. Comparison with M-BAPTL

As mentioned previously, a comparison for those instances where M-BAPTL provides a feasible solution within the two-period time horizon. Table 5 shows a list of the instances from those provided by Xu et al. (2012) that allow a feasible solution within the two-period planning horizon. Note that for some sets there is no feasible solution within the two-period horizon.

¹ <http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/>.

² <https://sites.google.com/site/gcports/tidal-berth-allocation-problem>.

Table 5

Comparison among GM-BAPTL and M*-BAPTL for the instances where a solution can be reached within the two-period horizon. M*-BAPTL is the M-BAPTL model with additional constraints to restrict the time horizon to two periods (see Section 3.2).

Set	Effect	Instance #	M*-BAPTL			GM-BAPTL			
			Obj.	Gap (percent)	t (seconds)	Obj.	Gap (percent)	t (seconds)	t _{imp} (percent)
3 × 09	Small	0	606	0.00	0.23	606	0.00	0.01	95.65
		1	511	0.00	1.61	511	0.00	0.04	97.52
3 × 09	Big	3	564	0.00	0.10	564	0.00	0.03	70.00
4 × 12	Small	1	562	0.00	0.82	562	0.00	0.04	95.12
		7	772	0.00	0.11	772	0.00	0.06	45.45
5 × 15	Small	3	899	0.00	5.64	899	0.00	0.09	98.40
		8	673	0.00	3.28	673	0.00	0.25	92.38
6 × 18	Small	0	543	0.00	4.36	543	0.00	0.14	96.79
		3	602	0.00	26.44	602	0.00	0.42	98.41
		6	872	0.00	0.92	872	0.00	0.25	72.83
7 × 21	Small	0	1534	0.13 ^a	3607.76	1534	0.00	0.15	100.00
		4	1213	0.00	3.81	1213	0.00	0.36	90.55
8 × 24	Big	8	1109	0.14 ^a	3609.39	1109	0.00	0.20	99.99
		Average	804.62	0.02	558.81	804.62	0.00	0.16	88.70

^a For those instances CPLEX is not able to prove the optimality within the maximum computational time of 1 hour.

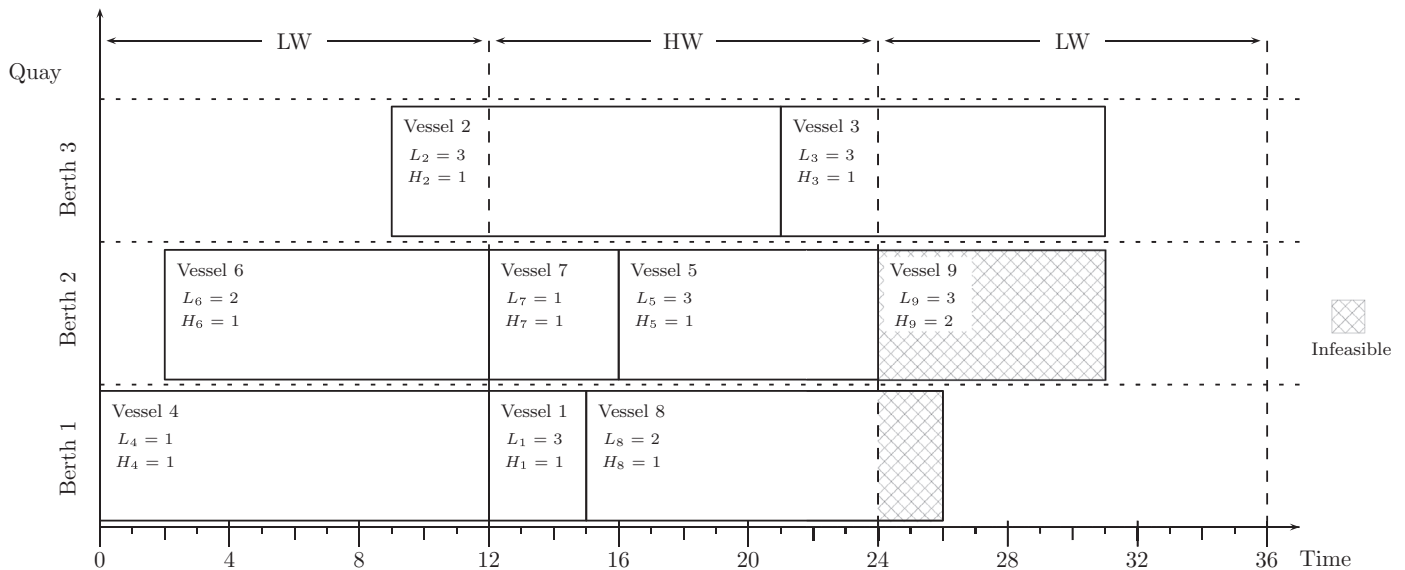


Fig. 1. Optimal solution reported by M-BAPTL (Xu et al., 2012) for the instance 0 (instance set 3 × 09 Big) provided by Xu et al. (2012).

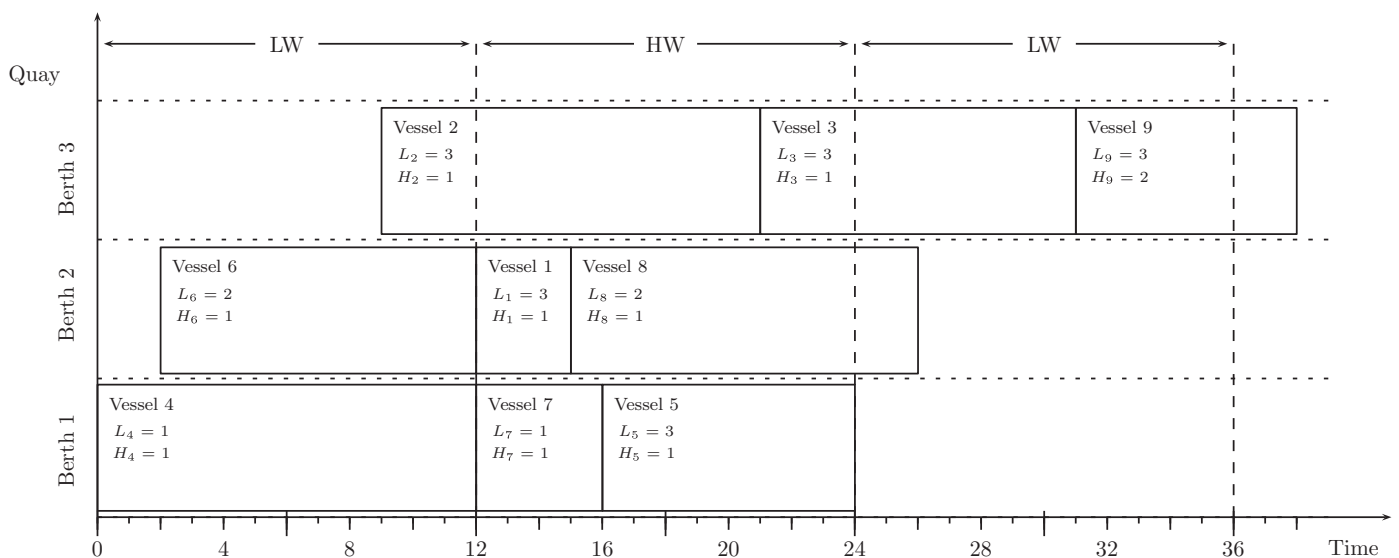


Fig. 2. Optimal solution reported by GM-BAPTL for the instance 0 (instance set 3 × 09–Big) provided by Xu et al. (2012).

Table 6

Comparison among GM-BAPTL and M*-BAPTL for the instances where a solution can be solved to optimality by M*-BAPTL. The number of constraints and variables are those reported in the reduced MIP given by CPLEX. The number of nodes is the required nodes processed to reach the optimal solution

Set	Effect	Instance #	M*-BAPTL			GM-BAPTL		
			Variables	Constraints	Nodes	Variables	Constraints	Nodes
3 × 09	Small	0	139	237	152	122	42	0
	Small	1	194	345	1275	37	20	4
3 × 09	Big	3	132	240	232	113	31	0
	Small	1	177	312	271	38	115	0
4 × 12	Small	7	43	75	3	32	26	0
	Small	3	532	993	2332	22	15	0
5 × 15	Small	8	442	798	1353	374	77	0
	Small	0	987	1850	977	729	114	0
6 × 18	Small	3	923	1733	17.866	547	84	0
	Small	6	733	1389	384	129	72	0
7 × 21	Small	4	1063	1991	1172	1028	133	0
	Average		487.73	905.73	2365.18	288.27	66.27	0.36

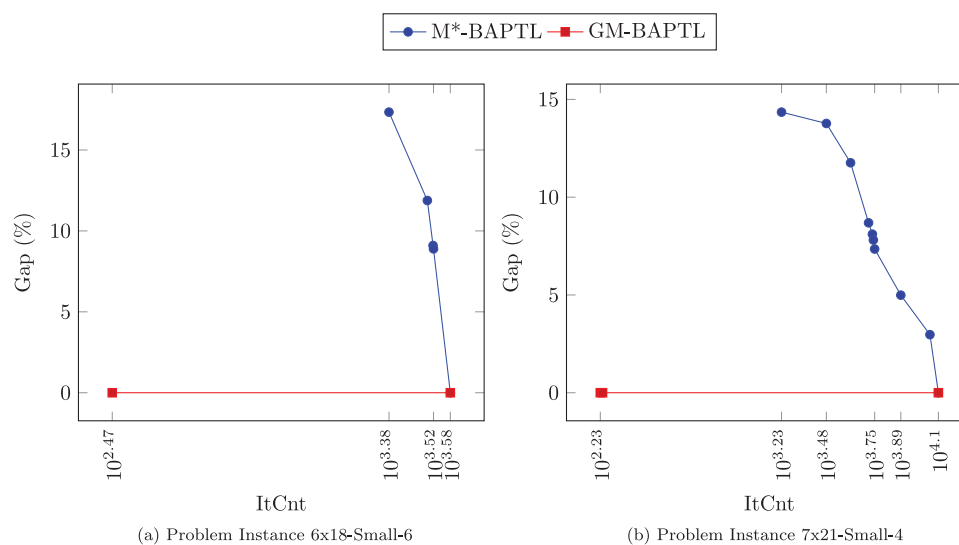


Fig. 3. Comparison of the relative error (Gap (percent)) evolution along the cumulative iterations (ItCnt) reported by CPLEX for the two largest-sized instances solved to optimality by M*-BAPTL and GM-BAPTL.

Table 5 reports the computational results provided by M-BAPTL with the additional constraints proposed in Section 3.2 (M*-BAPTL) and GM-BAPTL constrained to two periods proposed in Section 4. The column *Obj.* shows the objective function values, the columns *Gap* (percent) show the relative error, and the columns *t* (seconds) show the required computational time measured in seconds. Moreover, the column t_{imp} under GM-BAPTL heading shows the time reduction with respect to the time required by M*-BAPTL. The computational results show that GM-BAPTL clearly requires less computational time than M*-BAPTL. On average, GM-BAPTL reduces over a 88 percent of the computational time required for providing the same quality results than M*-BAPTL. In this regard, the temporal performance exhibited by GM-BAPTL is not strongly affected by the dimensions of the instances. In the worst case, it requires 0.42 seconds. The model reported in the literature achieves the optimal solution in all the problem instances. However, although it reaches the optimal solution for two problem instances (7×21 Small—instance 0; 8×24 Big—instance 8), it cannot prove the optimality of them within the maximum computational time of 1 hour.

In Table 6 a comparison in terms of number of variables, constraints and nodes reported by CPLEX for M*-BAPTL and GM-BAPTL is shown. The columns *Variables* and *Constraints* report those provided in the reduced MIP given by CPLEX. The column *Node* reports the number of nodes processed to reach the optimal solution. As can

be checked, GM-BAPTL requires far less nodes than M*-BAPTL, in this regard, in the majority of the cases GM-BAPTL does not require to process any node to provide the optimal solution. The problem size in terms of variables and constraints is smaller in the GM-BATL reformulation than in M*-BAPTL in all cases. Moreover, in Fig. 3 we report the evolution of the relative error (*Gap* (percent)) along the cumulative iterations reported by CPLEX for the largest-sized instances (6×18 Small—instance 6 and 7×21 Small—instance 4) where both models solve to optimality. The rationale of using the CPLEX cumulative iterations provided by CPLEX instead of number of nodes is based on the fact that GM-BAPTL does not require to process nodes for providing the optimal solution in the majority of the cases. As can be checked in the plot, GM-BATL requires less iterations and directly starts from a gap of 0 percent while the stepwise behavior exhibited by M*-BAPTL starts from a gap of about 17 percent and 14 percent, respectively. Also, it should be highlighted that it requires more iterations. Hence, at the light of these results, we can state that GM-BAPTL outperforms M*-BAPTL for tackling the problem instances proposed by Xu et al. (2012), and can be considered as solid alternative when either tackling new scenarios or using embedding the model in more complex solution approaches.

Table 7 presents a comparison between M-BAPTL (Xu et al., 2012) and GM-BAPTL. The literature problem instances used are those for which Xu et al. (2012) indicate that M-BAPTL provides the optimal

Table 7

Comparison between the results obtained to optimality by using M-BAPTL as reported by Xu et al. (2012) and the multiple-period GM-BAPTL model.

Set	Effect	M-BAPTL		GM-BAPTL		ρ (percent)
		Avg. Obj.	Avg. t (seconds)	Avg. Obj.	Avg. t (seconds)	
3 × 09	Small	459.50 ^a	1.36	467.00	0.29	1.63
3 × 09	Big	562.90 ^a	1.91	573.70	0.53	1.92
4 × 12	Small	705.80 ^a	5.48	717.40	0.57	1.64
4 × 12	Big	702.60 ^a	6.72	712.50	0.65	1.41
5 × 15	Small	840.20 ^a	26.92	863.30	1.64	2.75
5 × 15	Big	904.80 ^a	63.39	947.30	1.48	4.70

^a Infeasible solutions are included in the calculation of the average.

Table 8

Computational results obtained by means of GM-BAPTL for the instances proposed by Xu et al. (2012).

Set	Effect	Avg. Obj.	Avg. t (seconds)
3 × 09	Small	467.00	0.29
3 × 09	Big	573.70	0.53
4 × 12	Small	717.40	0.57
4 × 12	Big	712.50	0.65
5 × 15	Small	863.30	1.64
5 × 15	Big	947.30	1.48
6 × 18	Small	970.10	2.91
6 × 18	Big	1057.10	2.95
7 × 21	Small	1279.80	6.15
7 × 21	Big	1195.60	6.06
8 × 24	Small	1420.00	11.55
8 × 24	Big	1394.00	10.35

Table 9

Computational results by means of GM-BAPTL for the instances proposed in this work.

Set	m	n	Effect	Instance #	Obj.	t (seconds)
6 × 30	30	6	Small	1	710	13.14
				2	2131	14.14
				3	1070	18.21
				4	812	15.15
				5	1331	17.64
				Average	1210.80	15.66
6 × 30	30	6	Big	1	968	18.62
				2	1560	23.73
				3	1588	25.34
				4	1280	16.62
				5	991	11.53
				Average	1277.40	19.17
7 × 40	40	7	Small	1	2283	59.08
				2	4641	65.76
				3	1549	59.70
				4	2609	70.76
				5	2960	57.78
				Average	2808.40	62.62
7 × 40	40	7	Big	1	2171	60.58
				2	2567	55.68
				3	1520	58.50
				4	2155	35.12
				5	2366	67.20
				Average	2155.80	55.42
8 × 50	50	8	Small	1	2325	136.04
				2	2155	144.48
				3	1639	150.31
				4	2329	133.01
				5	2936	133.36
				Average	2276.80	139.44
8 × 50	50	8	Big	1	4167	82.18
				2	2279	90.76
				3	3001	141.39
				4	2437	121.80
				5	3302	138.01
				Average	3037.20	114.83

solutions. The rationale behind this is to make us a rough idea of the benefit of considering the regular changes of tides. The benefit of using GM-BAPTL instead of M-BAPTL is given by

$$\rho = \frac{\text{Avg. Obj}_{\text{GM-BAPTL}} - \text{Avg. Obj}_{\text{M-BAPTL}}}{\text{Avg. Obj}_{\text{M-BAPTL}}} \times 100 \text{ percent}, \quad (21)$$

where $\text{Avg. Obj}_{\text{M-BAPTL}}$ is the average objective function value provided by M-BAPTL and $\text{Avg. Obj}_{\text{GM-BAPTL}}$ is the average objective function value provided by GM-BAPTL. Eq. (21) is used for measuring the benefit of GM-BAPTL in comparison with M-BAPTL.

The results reported in Table 7 indicate that GM-BAPTL requires less computational time and provides the optimal solution in all cases. In addition to the fact that many optimal solutions provided by M-BAPTL are infeasible, it requires larger computational times. Concerning the benefit of using GM-BAPTL instead of M-BAPTL, the parameter ρ may indicate a maximum benefit of 4.70 percent for the largest problem instance set. This points out, on the one hand, the advantage of properly considering the changes of tides for the problem instances used. On the other hand, it highlights that GM-BAPTL is a suitable formulation for being used when addressing the berth planning under Time-Dependent Limitations.

5.2. Computational results for the literature instances for the multi-period BAPTL

As discussed above, the instances proposed by Xu et al. (2012) require a time horizon longer than two periods. Therefore, this subsection is devoted to assess the performance of GM-BAPTL when applied to those instances as well as to assess its performance over a new set of instances that considers more realistic sizes and longer planning horizons.

Tables 8 and 9 show the computational results for the instances provided by Xu et al. (2012) as well as for the new set of problem instances proposed in this work. The average objective function value, Avg. Obj., and average computational time, Avg. t (seconds), are reported. As can be seen, GM-BAPTL requires 133.10 seconds on average

to solve the largest problem instances. Moreover, the results reveal that when the number of container vessels increases, GM-BAPTL requires more computational time for solving them.

5.3. Computational results for the BAPTL-TW

This subsection is aimed at evaluating the results of the Berth Allocation Problem under Time-Dependent Limitations with Time Windows constraints (BAPTL-TW) proposed in this work. With this purpose, vessels and berth time windows are added to the problem data provided by Xu et al. (2012) and the instances of the benchmark suite used in Section 5.2.

Tables 10 and 11 reflect the computational results for the instances reported in the literature and in this work with time windows, respectively. Columns Avg. Obj., Avg. Gap (percent) and

Table 10

Computational results for problem instances proposed by Xu et al. (2012) with time windows added.

Set	Effect	Avg. Obj.	Avg. <i>t</i> (seconds)
3 × 09-tw	Small	579.20	0.09
3 × 09-tw	Big	650.70	0.05
4 × 12-tw	Small	858.10	0.15
4 × 12-tw	Big	824.60	0.12
5 × 15-tw	Small	1046.30	0.29
5 × 15-tw	Big	1119.20	0.26
6 × 18-tw	Small	1176.30	0.60
6 × 18-tw	Big	1266.30	0.57
7 × 21-tw	Small	1549.10	1.18
7 × 21-tw	Big	1432.60	1.20
8 × 24-tw	Small	1697.60	2.13
8 × 24-tw	Big	1622.00	1.80

Avg. *t* (seconds) show the average of the objective values, relative error and required time measured in seconds, respectively.

The results reported in Tables 10 and 11 show that the optimization model for the BAPTL-TW exhibits a suitable performance. In the largest problem instance, the maximum computational time required is 7.02 seconds on average. The computational results also indicate that the temporal behavior is not strongly influenced by the time windows and multiple period assumption. In this regard, it has to be noticed that the computational time required is notably lower than the problem instances without time windows. This is due to the reduction given by the time windows, which make the instances easier to solve.

6. Conclusions and further research

In this paper, we propose a mathematical formulation for the Berth Allocation Problem under Time-Dependent Limitations (BAPTL) based upon the Generalized Set Partitioning Problem, referred to as GM-BAPTL. This mathematical formulation considers a multi-period planning horizon. This allows to tackle problem instances and scenarios where the planning horizon may be equal to or larger than two periods. Moreover, we have jointly tackled this problem with some real-world issues at maritime container terminals, such as container vessel and berth time windows. This constitutes the Berth Allocation Problem under Time-Dependent Limitation with Time-windows constraints (BAPTL-TW). Finally, we propose a set of additional constraints for enforcing the two-period mathematical formulation reported in the literature (M-BAPTL) proposed by Xu et al. (2012). This set of constraints ensures the feasibility of the solutions reported by M-BAPTL within the two-period planning horizon. In other case, that mathematical formulation may extend the second period to infinity, which can cause non-practical schedules.

From the computational experiments that have been carried out, it is possible to indicate that GM-BAPTL improves the runtime when compared with M-BAPTL enforced with the additional constraints. In this regard, one of the advantages exhibited by GM-BAPTL is its computational efficiency, which is not impaired when considering the multi-period planning horizon that takes into account the regular change of tides beyond a planning horizon larger than two periods. Regarding the BAPTL-TW, the results indicate that this mathematical formulation presents a satisfactory behavior for all instance sizes within small computational times. It is remarkable that the computational effort required for solving large-size instances is, on average, less than 8 seconds. This characteristic makes the BAPTL-TW appropriate for being applied to heavily congested container terminal scenarios, in which the terminal managers have to face several time windows restrictions.

The aforementioned facts justify the use of both optimization models as support tools for the current practice in maritime con-

Table 11

Computational results for problem instances proposed in this work with time windows added.

Set	<i>m</i>	<i>n</i>	Effect	Instance #	Obj.	<i>t</i> (seconds)
6 × 30-tw	30	6	Small	1	710	1.97
				2	2302	1.42
				3	1070	1.73
				4	852	1.42
				5	1331	1.64
				Average	1253.00	1.64
6 × 30-tw	30	6	Big	1	968	1.91
				2	1580	1.81
				3	1670	2.56
				4	1302	1.17
				5	991	1.34
				Average	1302.20	1.76
7 × 40-tw	40	7	Small	1	2288	3.92
				2	4687	5.17
				3	1561	4.05
				4	2647	4.31
				5	2995	4.06
				Average	2835.60	4.30
7 × 40-tw	40	7	Big	1	2171	4.43
				2	2630	4.75
				3	1526	3.64
				4	2411	3.08
				5	2384	4.83
				Average	2224.40	4.15
8 × 50-tw	50	8	Small	1	2366	6.45
				2	2176	5.48
				3	1639	6.53
				4	2329	6.01
				5	2956	5.50
				Average	2293.20	5.99
8 × 50-tw	50	8	Big	1	4369	6.31
				2	2279	5.82
				3	3003	9.32
				4	2507	7.96
				5	3397	5.73
				Average	3111.00	7.02

tainer terminals since they provide the optimal scheduling in a matter of seconds and as indicated in Imai et al. (2001b) the Berth Allocation Problem is solved frequently as berth space may be limited and thousands of containers within vessels must be handled daily. Furthermore, the time advantage exhibited by GM-BAPTL and BAPTL-TW also suggest that they are suitable for being included in integrated designs, where BAPTL may be embedded with other related seaside problems such as the Quay Crane Allocation Problem, Quay Crane Scheduling Problem, etc. Moreover, in the spirit of dynamic and stochastic frameworks, the proposed models can be used as a way to obtain a baseline schedule, or for assessing 'a posteriori' the decisions taken in order to evaluate the suitability of the solution approaches used.

On the basis of the findings presented in this paper, the next stage of our research will be focused on the integration of BAPTL and BAPTL-TW with other related seaside problems arising in maritime container terminals, such as the Quay Crane Allocation Problem and the Quay Crane Scheduling Problem. Furthermore, another interesting topic for future works will be the analysis of different strategies and ways to overcome scenarios where the vessel time windows cannot be satisfied.

Acknowledgments

This work has been partially funded by the European Regional Development Fund and the Spanish Ministry of Science and Technology (project TIN2012-32608). Eduardo Lalla-Ruiz and Christopher Expósito-Izquierdo would like to thank the Canary Government for the financial support they receive through their doctoral grants.

References

- Barros, V. H., Costa, T. S., Oliveira, A. C. M., & Lorena, L. A. N. (2011). Model and heuristic for berth allocation in tidal bulk ports with stock level constraints. *Computers & Industrial Engineering*, 60(4), 606–613. doi:10.1016/j.cie.2010.12.018.
- Beškovnik, B. (2008). Measuring and increasing the productivity model on maritime container terminals. *Pomorstvo*, 22(2), 171–183.
- 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.
- Buhrkal, K., Zuglian, S., Ropke, S., Larsen, J., & Lusby, R. (2011). Models for the discrete berth allocation problem: a computational comparison. *Transportation Research Part E*, 47(4), 461–473.
- Cheong, C., & Tan, K. (2008). A multi-objective multi-colony ant algorithm for solving the berth allocation problem. In Y. Liu, A. Sun, H. Loh, W. Lu, & E.-P. Lim (Eds.), *Advances of computational intelligence in industrial systems: Vol. 116. Studies in computational intelligence* (pp. 333–350). Berlin/Heidelberg: Springer. doi:10.1007/978-3-540-78297-1_16.
- Cheong, C., Tan, K., Liu, D., & Lin, C. (2010). Multi-objective and prioritized berth allocation in container ports. *Annals of Operations Research*, 180, 63–103.
- Christensen, C. G., & Holst, C. T. (2008). *Berth allocation in container terminals*. Technical University of Denmark (Master's thesis).
- Christiansen, M., Fagerholt, K., Nygreen, B., & Ronen, D. (2007). Chapter 4 maritime transportation. In C. Barnhart, & G. Laporte (Eds.), *Transportation: Vol. 14. Handbooks in operations research and management science* (pp. 189–284). Elsevier. doi:10.1016/S0927-0507(06)14004-9.
- Cordeau, J. F., Laporte, G., Legato, P., & Moccia, L. (2005). Models and tabu search heuristics for the berth-allocation problem. *Transportation Science*, 39, 526–538.
- de Oliveira, R. M., Mauri, G. R., & Lorena, L. A. N. (2012). Clustering search for the berth allocation problem. *Expert Systems with Applications*, 39(5), 5499–5505. doi:10.1016/j.eswa.2011.11.072.
- Guldogan, E., Bulut, O., & Tasgetiren, M. F. (2012). A dynamic berth allocation problem with priority considerations under stochastic nature. In *Advanced intelligent computing theories and applications with aspects of artificial intelligence: Vol. 6839. Lecture notes in computer science* (pp. 74–82). Berlin/Heidelberg: Springer. doi:10.1007/978-3-642-25944-9_10.
- Hansen, P., Oguz, C., & Mladenovic, N. (2008). Variable neighborhood search for minimum cost berth allocation. *European Journal of Operational Research*, 191(3), 636–649. doi:10.1016/j.ejor.2006.12.057.
- Imai, A., Chen, H. C., 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. doi:10.1016/j.tre.2007.03.003.
- Imai, A., Nagaiwa, K., & Tat, C. W. (1997). Efficient planning of berth allocation for container terminals in Asia. *Journal of Advanced Transportation*, 31(1), 75–94.
- Imai, A., Nishimura, E., & Papadimitriou, S. (2001a). The dynamic berth allocation problem for a container port. *Transportation Research Part B: Methodological*, 35(4), 401–417.
- Imai, A., Nishimura, E., & Papadimitriou, S. (2001b). The dynamic berth allocation problem for a container port. *Transportation Research Part B: Methodological*, 35(4), 401–417. doi:10.1016/S0191-2615(99)00057-0.
- Imai, A., Nishimura, E., & Papadimitriou, S. (2003). Berth allocation with service priority. *Transportation Research Part B: Methodological*, 37(5), 437–457. doi:10.1016/S0191-2615(02)00023-1.
- Lalla-Ruiz, E., Expósito-Izquierdo, C., Melián-Batista, B., & Moreno-Vega, J. M. (2013). A metaheuristic approach for the seaside operations in maritime container terminals. In *Advances in computational intelligence* (pp. 28–35). Springer.
- 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*, 22, 60–76.
- Lalla-Ruiz, E., Melián-Batista, B., & Moreno-Vega, J. M. (2012). Artificial intelligence hybrid heuristic based on tabu search for the dynamic berth allocation problem. *Engineering Applications of Artificial Intelligence*, 25(6), 1132–1141.
- Lim, A. (1998). The berth planning problem. *Operations Research Letters*, 22(23), 105–110. doi:10.1016/S0167-6377(98)00010-8.
- Meisel, F. (2009). *Seaside operations planning in container terminals*. Springer. <http://link.springer.com/book/10.1007/978-3-7908-2191-8>.
- Meisel, F., & Bierwirth, C. (2006). Integration of berth allocation and crane assignment to improve the resource utilization at a seaport container terminal. In *Operations research proceedings 2005* (pp. 105–110). Springer.
- 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.
- Monaco, M., & Sammarra, M. (2007). Berth allocation problem: a strong formulation solved by a Lagrangian approach. *Transportation Science*, 41(2), 265–280.
- Moorthy, R., & Teo, C. (2006). Berth management in container terminal: The template design problem. *OR Spectrum*, 28(4), 495–518. doi:10.1007/s00291-006-0036-5.
- 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. doi:10.1016/S0377-2217(00)00128-4.
- Notteboom, T. (2006). The time factor in liner shipping services. <http://dx.doi.org/10.1057/palgrave.mel.9100148>
- Park, K. T., & Kim, K. H. (2002). Berth scheduling for container terminals by using a sub-gradient optimization technique. *Journal of the Operational Research Society*, 53(9), 1054–1062.
- Pinedo, M. L. (2008). *Scheduling: Theory, algorithms, and systems* (3rd). Springer Publishing Company, Incorporated.
- Rashidi, H., & Tsang, E. (2013). Novel constraints satisfaction models for optimization problems in container terminals. *Applied Mathematical Modelling*, 37(6), 3601–3634. doi:10.1016/j.apm.2012.07.042.
- Stahlbock, R., & Voß, S. (2008). Operations research at container terminals: a literature update. *OR Spectrum*, 30, 1–52.
- Steenken, D., Voß, S., & Stahlbock, R. (2004). Container terminal operation and operations research—A classification and literature review. *OR Spectrum*, 26, 3–49.
- Ting, C., Wu, K., & Chou, H. (2014). Particle swarm optimization algorithm for the berth allocation problem. *Expert Systems with Applications*, 41(4, Part 1), 1543–1550 ISSN No. 0957-4174. doi:10.1016/j.eswa.2013.08.051.
- Vis, I. F. A., & de Koster, R. (2003). Transshipment of containers at a container terminal: An overview. *European Journal of Operational Research*, 147(1), 1–16.
- Wang, F., & Lim, A. (2007). A stochastic beam search for the berth allocation problem. *Decision Support Systems*, 42(4), 2186–2196.
- Xu, D., Li, C. L., & Leung, J. Y. T. (2012). Berth allocation with time-dependent physical limitations on vessels. *European Journal of Operational Research*, 216(1), 47–56.
- Yang, D., Zhang, A., & Lam, J. S. L. (2013). Impacts of port productivity and service level on liner shipping operating cost and schedule reliability. In *International forum on shipping, ports and airports (IFSPA) 2013: Trade, supply chain activities and transport: Contemporary logistics and maritime issues*.
- Yeo, H. (2010). Competitiveness of Asian container terminals. *The Asian Journal of Shipping and Logistics*, 26(2), 225–246.