An exact robust approach for the integrated berth allocation and quay crane scheduling problem under uncertain arrival times*

Filipe Rodrigues[a], Agostinho Agra[b]

[a] *ISEG-School of Economics and Management, University of Lisbon*
REM - Research in Economics and Mathematics, CEMAPRE
[b] *Department of Mathematics and CIDMA, University of Aveiro*

Abstract

We consider an integrated berth allocation and quay crane assignment and scheduling problem where the arrival times of the vessels may be affected by uncertainty. The problem is modelled as a two-stage robust mixed integer program where the berth allocation decisions are taken before the exact arrival times are known, and the crane assignment and scheduling operations are adjusted to the arrival times. To solve the robust two-stage model, we follow a decomposition algorithm that decomposes the problem into a master problem and a separation problem. A new scenario reduction procedure for solving the separation problem is proposed as well as a warm start technique for reducing the number of iterations performed by the decomposition algorithm. To scale the proposed decomposition algorithm for large size instances, it is combined with a rolling horizon heuristic.

The efficiency and effectiveness of the proposed algorithms are demonstrated through extensive computational experiments carried out on randomly generated instances with both homogeneous and heterogeneous cranes as well as on instances from the literature.

---

keywords: OR in maritime industry, berth allocation and quay crane scheduling, robust optimization, uncertain arrival times, decomposition algorithm.

1. Introduction

Port management plays a major role in maritime transportation. One of the main strategies to efficiently manage ports is the minimization of the time required to operate the cargo volume of the vessels [29]. However, maritime transportation is frequently affected by several uncertainty factors such as weather conditions and mechanical failures, which may interfere with the port operations planning. Therefore, it is crucial to account for such uncertainties in decision making [21] to improve the efficiency of terminal operations.

One property that characterizes a good quality solution of a practical problem is its ability for dealing with adverse and unpredictable events. This is known as robustness. Different robustness concepts can be found in the literature. In this paper, we study the integrated berth allocation and quay crane assignment and scheduling problem, known by the acronym BACASP [28], where the vessel arrival times are considered uncertain.

*Published in European Journal of Operational Research, 295, 499-516, 2021
https://www.sciencedirect.com/science/article/pii/S037722172100206X

Stochastic programming and robust optimization are two alternative and distinct approaches for dealing with uncertainty, and both approaches have been applied over the years for solving different berth allocation problems (see Section 2). While stochastic programming optimizes over expectations of functions involving random variables, robust optimization optimizes over the worst-case scenario. Optimizing over the worst-case scenario allows to obtain solutions that remain feasible for every possible realization of the uncertainty parameters in the uncertainty set. Obtaining a robust solution for the BACASP means to obtain a berth scheduling that does not need to be readjusted when disruptions on the arrival times of the vessels covered by the uncertainty set occur. This is not the case of stochastic programming, where readjustment strategies are often unavoidable when disruptions occur. *Adjusting the original schedule frequently is something that the port planner does not want to see* [18] since *frequent rescheduling of the berth plan usually results in poor performance of the overall terminal efficiency. Because the berth plan is the very first level of terminal planning and is used as a key input to yard storage, personnel and equipment deployment planning, any berth scheduling adjustment will induce a series of changes to all the other subsequent operations* [32]. Therefore, this paper focus on robust optimization approaches. Our goal is to derive a solution that minimizes the total completion time (i.e., the difference between the departure time of the vessels and their arrival time) assuming the worst scenario amongst all possible vessel arrival delays within the uncertainty set occurs. This follows the general concept that a robust solution is interpreted as a solution that remains feasible for every possible realization of the uncertain parameters in the uncertainty set, see [5, 6] for details. The level of conservatism is controlled through suitable parameters of the uncertainty set. Different concepts of robust solutions have also been used by other authors, such as solutions that exhibit a *good* performance, for example, lower baseline costs, lower probabilities of constraints violation and lower rescheduling costs, when a set of possible scenarios of uncertainty is considered, [9, 11, 16, 24, 31, 32, 35].

The robust BACASP with uncertain arrival times we consider has the following assumptions. The berth is divided into berth sections of equal length and the vessels are allowed to moor at any place along the wharf (known as continuous berth allocation). The time horizon is divided into time periods of equal length. The load and unload operations are performed with a set of cranes that transfer the cargo between the vessels and the storage area at the yard. A crane can be assigned to a vessel and moved to another vessel in the following period while the first vessel is still operating (known as time-variant cranes assignment). Nevertheless, non-crossing constraints and safety spaces between cranes must be obeyed. For a given time horizon, a set of vessels with different lengths and with known cargo quantities to operate is considered. The estimated arrival time of each vessel is known, however, since delays during the planning horizon can occur, the arrival times are assumed to belong to an uncertainty set built upon these estimated (nominal) arrival times. The deterministic variant of this problem (with known arrival times), that motivated this work, was studied in [4]. It is worth mentioning that the robust problem studied in our paper and all the assumptions made arise from a practical application.

In port terminals, some operations, such as the cranes assignment and scheduling, may be easily readjusted after the arrival time of the vessels are observed, while other decisions such as the berth allocation of vessels are hard to be reverted or readjusted. Thus, the allocation of a vessel has a great impact in the allocation of the following incoming vessels. Since the average time of the vessels at ports is in general large [29], a delay on an arrival of a vessel may have repercussions on the forthcoming vessels and in several other port operations (such as quay cranes assignment and scheduling) for the following days. Hence, the robust BACASP under uncertain arrival times studied in this paper integrates two types of decisions: the allocation of arriving vessels to berth positions (known as berth allocation), and the assignment of cranes to vessels and the schedule of their operations (known as quay crane assignment and scheduling). These decisions correspond to two subproblems which, in the literature, have also been considered separately and are referred to as BAP and QCASP, respectively. We assume that the berth allocation decisions are taken before the exact arrival times are known (here-and-now decisions), while the crane assignment and scheduling decisions are taken when the exact arrival times are known (wait-and-see decisions). Therefore, we model the robust BACASP as a two-stage mixed integer program where the berth positions are the first-stage decisions and the cranes assignment and schedule are the second-stage decisions.

To solve the introduced two-stage model, we propose an exact decomposition algorithm [7, 33] where a restricted model, known as master problem, is solved for a subset of scenarios, and a separation problem, known as the adversarial problem, either identifies a scenario that is violated by the solution of the master problem or shows that no such a scenario exists. In case such a scenario is found, then it is added to the master problem and the process is repeated. The general idea of the decomposition algorithm is presented in Figure 1.

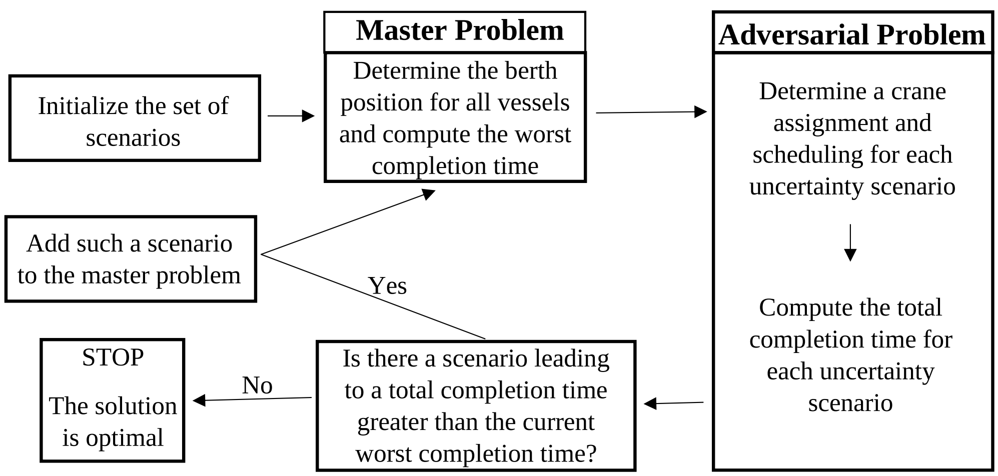

Figure 1: Decomposition algorithm.

The performance of this general procedure greatly depends on how several issues are addressed for each particular problem. In the robust BACASP considered in this paper, the three main issues to be addressed are: (i) Solve the master problem efficiently. The master problem is a hard BACASP with several scenarios that is difficult to solve when the number of scenarios is large; (ii) Solve the adversarial problem efficiently. This implies to solve a max-min problem where the evaluation of each scenario of uncertainty requires to solve a crane assignment and scheduling problem. In other words, it implies to solve a hard mixed integer programming model (with a large number of binary variables) for each scenario of uncertainty. For this reason, the resulting adversarial problem is amongst the hardest adversarial problems in robust optimization (see [13] for details), and therefore, solving it efficiently is a very challenging task; (iii) Minimize the number of global iterations required to achieve the optimal solution. Since at each iteration of the decomposition algorithm both subproblems considered in (i) and (ii) need to be solved, an efficient decomposition procedure should minimize the number of times each subproblem is solved.

For solving the master problem, we propose an exact approach based on a mixed-integer program and a rolling horizon heuristic to solve large size instances. It is worth recalling that in order to ensure that the decomposition algorithm is an exact procedure it is only necessary to solve the master problem to optimality in the last iteration. To solve the adversarial problem to optimality, each scenario must be evaluated. We propose a new scenario evaluation heuristic that quickly determines several heuristic crane assignments and schedules. This heuristic is the basis of a new procedure, called scenario reduction procedure, that allows to identify a large number of irrelevant scenarios of uncertainty. Such scenarios can be discarded in the current iteration of the decomposition algorithm without solving the associated hard mixed integer programming problems. For reducing the number of global iterations performed by the decomposition algorithm, we propose a warm start technique where a carefully chosen scenario of uncertainty is used to initialize the decomposition algorithm through a new heuristic called slack reduction heuristic.

The main contributions of our study are summarized as follows.

i) To the best of our knowledge, this is the first study of the BACASP under uncertain vessel arrival times aiming to minimize the total completion time of the worst-case scenario. This concept of robustness was only followed in [15, 19, 34, 26], however, the first three works only address the BAP, and the last one addresses the BACASP under uncertain quay crane setup costs. From the robust optimization perspective, the BACASP has the unusual property that the worst-case scenario does not necessarily correspond to the maximum allowed delay of the vessels, which makes the problem even more difficult to solve.

ii) A two-stage model is introduced where the berth allocation decisions are taken before the exact arrival times are known (first-stage decisions), and the cranes assignment and schedule decisions are taken when the arrival times are known (second-stage decisions).

iii) An exact decomposition algorithm for the robust BACASP under uncertain vessel arrival times is proposed. Several enhancement strategies for solving the decomposition algorithm efficiently are proposed, namely: 1) a new crane assignment and scheduling heuristic used within a new scenario reduction procedure to discard non-relevant scenarios each time the adversarial problem is solved; 2) a new slack reduction heuristic for finding a relevant scenario to use in a warm start technique for reducing the number of global iterations performed. Such enhancements allow us to work with a number of scenarios much larger than the number of scenarios used in previous works found in the literature.

iv) To solve large size and harder instances, a rolling horizon algorithm is proposed in which the time horizon is divided according to the arrival times of the vessels.

v) Extensive computational experiments on sets of both randomly generated instances (with homogeneous and heterogeneous cranes) and instances from the literature are performed. The obtained results show that 1) the scenario reduction procedure allows to discard around 93% of irrelevant scenarios when the adversarial problems are solved; 2) the warm start technique drastically reduces the number of iterations of the decomposition algorithm; and 3) the proposed solution methods are scalable for large size instances.

The paper is organized as follows. In Section 2 we review the most relevant literature related to our work. A deterministic mixed integer model is presented in Section 3 and the corresponding robust model is given in Section 4. In Section 5 we illustrate a robust BACASP solution and the gains resulting from applying robust optimization techniques. The proposed decomposition algorithm and several enhancements are introduced in Section 6. A rolling horizon heuristic to handle with large instances is proposed in Section 7. The computational tests are reported in Section 8. Finally, in Section 9 we present the main conclusions.

2. Literature review and related work

In this section, we review the literature related to our work. Since berth allocation and quay crane assignment and scheduling problems have received great attention in the last years (see [8] for a survey), we focus on the literature on both BAP and BACASP under uncertainty, with an emphasis on the works on robust approaches.

One of the first works addressing uncertainties in quayside operations is [20]. The authors studied the BAP with uncertain vessel arrival and handling times and proposed a framework to create robust berth allocations. The problem is solved using a simulated annealing algorithm. Stochastic based approaches for handling uncertainties of vessel arrival times and handling times were considered in [12] and [37]. In both works, a genetic algorithm is proposed.

A common approach to obtain robust solutions for both BAP and BACASP is the use of time buffers. Such time buffers are essentially used to absorb delays propagation throughout the whole berth plan, avoiding infeasibilities and/or large rescheduling costs. Buffering strategies were used for the BAP in [9, 16, 32, 35] and for the BACASP in [24].

Du et al. [9] proposed a robust reactive feedback procedure based on the buffer idea to solve the BAP under stochastic vessel arrival times. It starts by using a simulated annealing procedure for solving the deterministic BAP with fixed buffers. The obtained solution is evaluated on a set of arrival delay scenarios and reassignment rules are applied (when needed). Then, heuristic strategies are employed to adjust the time buffer of each vessel. After updating the buffer times, a simulated annealing is used again to solve the new model. Hendriks et al. [15] presented a robust proactive optimization model for the cyclic BAP under uncertain vessel arrival times. The model minimizes the required crane capacity and determines a time window for each vessel arrival time ensuring that any vessel arriving in the determined time window is released with no delays. In [36] a two-stage decision model and a meta-heuristic approach are proposed for a BAP under uncertain arrival and handling time of vessels. Both a proactive strategy to obtain an initial schedule that incorporates a degree of anticipation of uncertainty and a reactive recovery strategy to adjust the initial schedule to handle realistic scenarios are considered. Xu et al. [32] proposed a proactive robust berth scheduling algorithm that integrates a simulated annealing and a branch-and-bound algorithm for the BAP with uncertain vessel arrival and handling times. The problem minimizes the total departure delay time of vessels and maximizes the length of the buffer time (which is constant for all vessels). A bi-objective optimization model for the BAP aiming to minimize costs and maximize robustness of schedules is also presented in Zhen et al. [35]. The proposed proactive model is based on time buffers that are used to absorb delays caused by both vessel arrival and handling uncertain times. Rodriguez-Molins et al. [24] proposed a multi-objective model for the BACASP under uncertain handling times, aiming to minimize the total service time and to maximize the robustness of the solution through buffer times (depending on the size of the vessels). To solve larger instances, the authors proposed a proactive genetic algorithm.

A proactive genetic algorithm was also proposed by Golias et al. [11] for the BAP with both vessel arrival and handling uncertain times as well as by Shang et al. [26] for the BACASP assuming the quay crane setup time of shifting along the quay is uncertain. Golias et al. [11] formulated the BAP as a bi-level bi-objective optimization problem aiming to minimize the average and the range of the total service times (that is, the difference between the total service time of the worst scenario and the total service time of the best scenario). Zhen [34] studied the BAP under uncertain handling times. A stochastic programming and a robust formulation were proposed to minimize the total cost of the baseline schedule plus the average and the maximum cost for handling potential conflicts, respectively. To solve large scale instances, a three-phase heuristic based on the insertion strategy is proposed.

Shang et al. [26] proposed two robust optimization models to formulate the BACASP. The goal is to ensure that the obtained schedule remains feasible when delays occur. Along with a genetic algorithm, an insertion heuristic is also proposed for solving larger instances. Ursavas and Zhu [30] proposed a framework based on a stochastic dynamic programming approach to the BAP, and provided a characterization of optimal policies under stochastic arrival and handling times. Liu et al. [17] introduced an adaptive differential evolution algorithm to the BAP with uncertain arrival and handling times. The BAP under uncertainty is also considered in [18]. Using the concept of conflict, the authors proposed two kinds of service levels and constructed two decision models. A two-stage heuristic algorithm is presented. A mixed-integer program is proposed in [25] for a more general problem where arrival times and handling times are uncertain. They proposed heuristic schemes based on the MIP model. In [27], a BAP where uncertainty resulting from the effect of tides on the berthing schedule and from the arrival time of vessels is considered. A sample average approximation method based on a mathematical model is presented for solving the problem.

Xiang et al. [31] studied the BACASP assuming that four types of disruptions can occur: deviation of the vessel arrival and handling times, calling of unscheduled vessels, and breakdown of quay cranes. To accomplish these unpredictable events, the authors proposed a reactive heuristic strategy that takes the baseline schedule as the reference schedule and minimizes the recovery cost. In [14] the effect of the arrival uncertainty on the container stacking is studied, and different heuristics are proposed accordingly to the stacking principles and with levels of vessel arrival information. Iris and Lam [16] proposed a mathematical model for the BACQAP under vessel arrival and handling times uncertain with both proactive and reactive properties. The proposed model aims to minimize the cost associated with the baseline schedule and the cost associated with resource actions while maximizing the robustness of the solution through buffer times. Harder instances are solved by a two-stage heuristic framework based on an adaptive large neighborhood search heuristic.

The closest approach to the one developed in our paper was recently presented by Liu et al. [19] to solve a BAP under uncertainty. As the case presented here, the uncertainty set uses no probabilistic information. A two-stage robust optimization approach is proposed where the baseline schedule is made before the uncertain data is revealed and the recovery operation is made after the disruptions are known. Three two-stage robust models are presented.

3. The formulation for the deterministic BACASP

In this section we present a formulation for the BACASP, based on the time-space diagram, known as Relative Position Formulation. This formulation is the same used in [4] and it is the basis of the new formulation of the robust problem introduced in Section 4. Consider a set of N heterogeneous vessels and a time horizon of M time periods. Assume that the wharf is divided into $J+1$ berth sections and the total number of cranes operating in the wharf is C. Let us define the following sets: $V = \{1,\ldots,N\}$ the set of vessels, $T = \{1,\ldots,M\}$ the set of time periods, $B = \{0,\ldots,J\}$ the set of berth positions and $G = \{1,\ldots,C\}$ an ordered set of cranes where for all $g, g' \in G$ such that $g < g'$ means that crane g has an operating range (in term of berth positions) lower than crane g'.

The length of each vessel $k \in V$ (measured in number of berth sections) is given by H_k and the cargo volume to be operated and the nominal arrival time are represented by Q_k and A_k, respectively. The maximum number of cranes that can simultaneously work on a vessel $k \in V$ is NC_k. The safety time between the departure of a vessel and the berthing of a new one (measured in number of time units) is denoted by F. Each crane $g \in G$ can operate in the wharf between berth sections S_g and E_g and has a processing rate of P_g.

For each $k, l \in V,\ g \in G$ and $j \in T$ let us define the following variables:

$x_{k\ell}$: binary variable taking value one if ship ℓ berths after ship k had departed;

$y_{k\ell}$: binary variable taking value one if ship ℓ berths below the berth position of ship k;

z_{gkj} : binary variable taking value one if crane g is assigned to ship k in time period j;

b_k : berthing position of vessel k;

t_k : berthing time of vessel k;

c_k : departing time of vessel k.

The deterministic BACASP can be formulated as follows:

$$
\begin{aligned}
\min \quad & \sum_{k \in V} (c_k - A_k) & & (1)\\
s.t. \quad & x_{\ell k} + x_{k\ell} + y_{\ell k} + y_{k\ell} \geq 1, & k, \ell \in V, k < \ell, \quad & (2)\\
& x_{\ell k} + x_{k\ell} \leq 1, & k, \ell \in V, k < \ell, \quad & (3)\\
& y_{\ell k} + y_{k\ell} \leq 1, & k, \ell \in V, k < \ell, \quad & (4)\\
& b_k \leq J - H_k, & k \in V, \quad & (5)\\
& b_k \geq b_\ell + H_\ell + (y_{k\ell} - 1)J, & k, \ell \in V, k \neq \ell, \quad & (6)\\
& t_\ell \geq c_k + F + (x_{k\ell} - 1)M, & k, \ell \in V, k \neq \ell, \quad & (7)\\
& t_k \geq A_k, & k \in V, \quad & (8)\\
& \sum_{k \in V} z_{gkj} \leq 1, & j \in T, g \in G, \quad & (9)\\
& t_k \leq j z_{gkj} + (1 - z_{gkj})M, & j \in T, k \in V, g \in G, \quad & (10)\\
& c_k \geq (j+1) z_{gkj}, & j \in T, k \in V, g \in G, \quad & (11)\\
& \sum_{j \in T} \sum_{g \in G} P_g z_{gkj} \geq Q_k, & k \in V, \quad & (12)\\
& b_k + H_k \leq E_g z_{gkj} + (1 - z_{gkj})J, & j \in T, k \in V, g \in G, \quad & (13)\\
& b_k \geq S_g z_{gkj}, & j \in T, k \in V, g \in G, \quad & (14)\\
& z_{gkj} + z_{g'\ell j} \leq 2 - y_{k\ell}, & j \in T, k, \ell \in V, g, g' \in G, g' < g, \quad & (15)\\
& \sum_{g \in G} z_{gkj} \leq NC_k, & k \in V, j \in T, \quad & (16)\\
& b_k, t_k, c_k \in \mathbb{Z}_0^+, & k \in V, \quad & (17)\\
& x_{k\ell}, y_{k\ell} \in \{0, 1\}, & k, \ell \in V, k \neq \ell \quad & (18)\\
& z_{gkj} \in \{0, 1\}, & k \in V, g \in G, j \in T. \quad & (19)
\end{aligned}
$$

The objective function (1) minimizes the sum of the completion time of all vessels. The completion time of a vessel is the time elapsed between the arrival time and the departure time. Constraints (2)-(4) are the usual constraints enforcing each pair of vessels not to overlap either in time or in space. Constraints (6) and (7) relate the space and the time variables of two vessels, respectively. These constraints also enforce the definition of variables x_{kl} and y_{kl}. Constraints (5) and (8) define the range of variables b_k and t_k, respectively.

Constraints (9) ensure that each crane can operate in at most one vessel in each time period. Constraints (10) guarantee that a crane can be assigned to a vessel only after its arrival, while constraints (11) ensure that if a crane was operating vessel k in time period j, then this vessel can only depart after the end of time period j ($c_k \geq j+1$). Constraints (12) guarantee that the cranes assigned to vessel k are enough to operate all the cargo of that vessel. Constraints (13) and (14) ensure that a crane can be assigned to a vessel if and only if the vessel is berthed within the range of the crane. Constraints (15) prevent cranes from passing each other at any time period. Constraints (16) impose a maximum number NC_k of cranes that can simultaneously work on vessel $k \in V$ and allow to consider a safety distance between cranes. Constraints (17) define the integer variables, while constraints (18) and (19) define the binary variables.

3.1. A discretized reformulation for the BACASP

Model (1)-(19) is based on the big-M constraints (6) and (7), and it is well-known that such constraints lead to a very weak model in the sense that the associated linear relaxation retrieves a poor bound. This problem was identified by Agra and Oliveira [4] who proposed a reformulation of the model that avoids the use of big-M constraints by introducing new variables resulting from the discretization of both time and space variables t_k, c_k and b_k. They showed that with this new set of variables the obtained model is stronger than the original one. Therefore, in this paper, we also use such variables. For each $k \in V$, $j \in T$ and $n \in B$ the new variables are described as follows:

π_{kn} : binary variable taking value one if the first (lower) berth position of vessel k is n;

σ_{kn} : binary variable taking value one if berth section n is assigned to vessel k;

α_{kj} : binary variable taking value one if vessel k starts operating in time period j;

β_{kj} : binary variable taking value one if vessel k is operated in time period j;

γ_{kj} : binary variable taking value one if the last time period vessel k is operated is j.

The set of additional constraints is the same used in [4] and it is as follows:

$$b_k = \sum_{n \in B} n\pi_{kn}, \qquad k \in V, \tag{20}$$

$$\sum_{n \in B} \sigma_{kn} = H_k, \qquad k \in V, \tag{21}$$

$$\sum_{n \in B} \pi_{kn} = 1, \qquad k \in V, \tag{22}$$

$$\pi_{kn} \geq \sigma_{kn} - \sigma_{k,n-1}, \qquad k \in V, n \in B, n > 1, \tag{23}$$

$$\pi_{k1} \geq \sigma_{k1}, \qquad k \in V, \tag{24}$$

$$\pi_{kn} \leq \sigma_{kn}, \qquad k \in V, n \in B, \tag{25}$$

$$\pi_{kn} \leq 1 - \sigma_{k,n-1} \qquad k \in V, n \in B, n > 1, \tag{26}$$

$$y_{k\ell} + \sum_{m=max\{n-H_\ell+1,0\}}^{J} \pi_{km} + \pi_{\ell n} \leq 2, \qquad k,\ell \in V, k \neq \ell, n \in B, \tag{27}$$

$$\pi_{kn}, \sigma_{kn} \in \{0,1\}, \qquad k \in V, n \in B, \tag{28}$$

$$z_{gkj} \leq \sum_{n=S_g}^{E_g-H_k} \pi_{kn}, \qquad k \in V, g \in G, j \in T, \tag{29}$$

$$t_k = \sum_{j\in T} j\alpha_{kj}, \qquad k \in V, \tag{30}$$

$$c_k \geq (j+1)\beta_{kj}, \qquad k \in V, j \in T, \tag{31}$$

$$z_{gkj} \leq \beta_{kj}, \qquad k \in V, j \in T, g \in G, \tag{32}$$

$$\sum_{j\in T} \alpha_{kj} = 1, \qquad k \in V, \tag{33}$$

$$\alpha_{kj} \geq \beta_{kj} - \beta_{k,j-1}, \qquad k \in V, j \in T, j > 1, \tag{34}$$

$$\alpha_{k1} \geq \beta_{k1}, \qquad k \in V, \tag{35}$$

$$\alpha_{kj} \leq \beta_{kj}, \qquad k \in V, j \in T, \tag{36}$$

$$\alpha_{kj} \leq 1 - \beta_{k,j-1}, \qquad k \in V, j \in T, j > 1, \tag{37}$$

$$x_{kl} + \beta_{ki} + \beta_{lj} \leq 2, \qquad k,l \in V, k \neq l, j, i \in T, i \geq j-1, \tag{38}$$

$$\gamma_{kj} \geq \beta_{kj} - \beta_{k,j+1}, \qquad k \in V, j \in T, j < M, \tag{39}$$

$$\gamma_{k1} \geq \beta_{kM}, \qquad k \in V, \tag{40}$$

$$\gamma_{kj} \leq \beta_{kj}, \qquad k \in V, j \in T, \tag{41}$$

$$\gamma_{kj} \leq 1 - \beta_{k,j+1}, \qquad k \in V, j \in T, j < M, \tag{42}$$

$$\sum_{j\in T} \gamma_{kj} = 1, \qquad k \in V, \tag{43}$$

$$\alpha_{kj}, \beta_{kj}, \gamma_{kj} \in \{0,1\}, \qquad k \in V, j \in T. \tag{44}$$

For ease of presentation, we omit the explanation of the constraints. Such an explanation can be found in [4]. The deterministic model for the BACASP is then represented by the objective function (1) and constraints (2)-(44).

4. The robust BACASP formulation

The deterministic model for the BACASP described in the previous section assumes that the arrival times of all vessels are exactly known in advance. However, this assumption is not true in most practical problems since the arrival times may be uncertain due to several factors as weather conditions and mechanical failures. In this section, we present a mathematical model for the robust BACASP under uncertain arrival times. We assume that the probability distribution of the arrival times is not known and the arrival times vary in an uncertainty set. In this paper, we consider an uncertainty set, henceforth referred to as multiple constrained budgeted (MCB) uncertainty set, inspired by the budgeted uncertainty set proposed by Bertsimas and Sim [6]. There are several reasons for the choice of this kind of uncertainty sets: i) it allows to control the degree of robustness of the obtained solutions; ii) it allows to obtain tractable models; iii) it is very easy to be followed by practitioners, and iv) it is being widely used by several authors to model the uncertain nature of parameters like arrival times and traveling times, see [3, 19, 22, 26].

The MCB uncertainty set considered is defined as:

$$\Omega = \left\{ A^\omega \ : \ A_k^\omega = A_k + [\hat{A}_k \delta_k], \ 0 \le \delta_k \le 1, \ k \in N, \sum_{k=1+j[N/ng]}^{\min\{(j+1)[N/ng],N\}} [\delta_k] \le \Gamma_j, \ j \in \{0, 1, \dots, ng-1\} \right\}$$

where A_k is the nominal/deterministic arrival time of vessel $k \in V$, $\hat{A}_k$ is the maximum allowed delay of vessel $k \in V$, and $[\cdot]$ denotes the round operator. Unlike the original budget uncertainty set, the MCB uncertainty set we propose has multiple budget constraints and a round operator to ensure that only integer arrival time values are allowed. Note that we are considering a discrete BACASP, which justifies the use of this round operator. The vessels are grouped into ng groups according to the nominal arrival times, and a budget constraint is associated with each group of vessels. For example, the first budget constraint imposes that at most Γ_0 vessels of the first group (the ones with lower nominal arrival time) can arrive to the port with delay. An uncertainty set with multiple budget constraints was already used by Rodrigues et. al. [23] for the robust inventory problem.

To maximize the utility of the wharf and improve customer satisfaction, terminal planners need to schedule the berthing for each arriving vessel carefully in advance [32]. Therefore, the mathematical model for the robust BACASP we propose is a two-stage model in which the berth positions of the vessels are the first-stage decisions (decisions that have to be taken before the uncertainty is revealed), and the cranes assignment and schedule are the second-stage decisions (decisions taken after the uncertain arrival times are revealed). Since the crane assignment variables z_{gkj} are second-stage decision variables, the berth time, the departure time of the vessels and the discrete time variables $\alpha_{kj}, \ \beta_{kj}, \ \gamma_{kj}$ are forced to be second-stage variables as well. Then, all these variables are redefined as follows:

z_{gkj}^ω : binary variable taking value one if crane g is assigned to ship k in period j when scenario $\omega \in \Omega$ occurs;

t_k^ω : berthing time of vessel k when scenario $\omega \in \Omega$ occurs;

c_k^ω : departing time of vessel k when scenario $\omega \in \Omega$ occurs;

α_{kj}^ω : binary variable taking value one if vessel k starts operating in time period j when scenario $\omega \in \Omega$ occurs;

β_{kj}^ω : binary variable taking value one if vessel k is operated in time period j when scenario $\omega \in \Omega$ occurs;

γ_{kj}^ω : binary variable taking value one if the last time period vessel k is operated is j when scenario $\omega \in \Omega$ occurs.

Note that all the first-stage decision variables ($x_{k\ell}, \ y_{k\ell}, \ b_k, \ \pi_{kn}$ and σ_{kn}) remain unchanged. The formulation of the robust BACASP is as follows:

$$\min \ \theta \tag{45}$$

$$
\begin{aligned}
s.t.\quad & \theta \geq \sum_{k\in V}(c_k^\omega - A_k^\omega), && \omega\in\Omega, && (46)\\
& (2)-(6),\ (18),\ (20)-(28) && && \\
& t_\ell^\omega \geq c_k^\omega + F + (x_{k\ell}-1)M, && k,\ell\in V, k\neq \ell, \omega\in\Omega, && (47)\\
& t_k^\omega \geq A_k^\omega, && k\in V,\omega\in\Omega, && (48)\\
& \sum_{k\in V} z_{gkj}^\omega \leq 1, && j\in T, g\in G, \omega\in\Omega, && (49)\\
& t_k^\omega \leq j z_{gkj}^\omega + (1-z_{gkj}^\omega)M, && j\in T, k\in V, g\in G, \omega\in\Omega, && (50)\\
& c_k^\omega \geq (j+1) z_{gkj}^\omega, && j\in T, k\in V, g\in G, \omega\in\Omega, && (51)\\
& \sum_{j\in T}\sum_{g\in G} P_g z_{gkj}^\omega \geq Q_k, && k\in V,\omega\in\Omega, && (52)\\
& b_k + H_k \leq E_g z_{gkj}^\omega + (1-z_{gkj}^\omega)J, && j\in T, k\in V, g\in G, \omega\in\Omega, && (53)\\
& b_k \geq S_g z_{gkj}^\omega, && j\in T, k\in V, g\in G, \omega\in\Omega, && (54)\\
& z_{gkj}^\omega + z_{g'\ell j}^\omega \leq 2 - y_{k\ell}, && j\in T, k,\ell\in V, g,g'\in G, g'<g, \omega\in\Omega, && (55)\\
& \sum_{g\in G} z_{gkj}^\omega \leq NC_k, && k\in V, j\in T, \omega\in\Omega, && (56)\\
& b_k, t_k^\omega, c_k^\omega \in \mathbb{Z}_0^+, && k\in V,\omega\in\Omega, && (57)\\
& z_{gkj}^\omega \leq \sum_{n=S_g}^{E_g-H_k} \pi_{kn}, && k\in V, g\in G, j\in T, \omega\in\Omega, && (58)\\
& t_k^\omega = \sum_{j\in T} j\alpha_{kj}^\omega, && k\in V,\omega\in\Omega, && (59)\\
& c_k^\omega \geq (j+1)\beta_{kj}^\omega, && k\in V, j\in T, \omega\in\Omega, && (60)\\
& z_{gkj}^\omega \leq \beta_{kj}^\omega, && k\in V, j\in T, g\in G, \omega\in\Omega, && (61)\\
& \sum_{j\in T}\alpha_{kj}^\omega = 1, && k\in V,\omega\in\Omega, && (62)\\
& \alpha_{kj}^\omega \geq \beta_{kj}^\omega - \beta_{k,j-1}^\omega, && k\in V, j\in T, j>1, \omega\in\Omega, && (63)\\
& \alpha_{k1}^\omega \geq \beta_{k1}^\omega, && k\in V,\omega\in\Omega, && (64)\\
& \alpha_{kj}^\omega \leq \beta_{kj}^\omega, && k\in V, j\in T, \omega\in\Omega, && (65)\\
& \alpha_{kj}^\omega \leq 1-\beta_{k,j-1}^\omega, && k\in V, j\in T, j>1, \omega\in\Omega, && (66)\\
& x_{kl} + \beta_{ki}^\omega + \beta_{lj}^\omega \leq 2, && k,l\in V, k\neq l, j,i\in T, i\geq j-1, \omega\in\Omega, && (67)\\
& \gamma_{kj}^\omega \geq \beta_{kj}^\omega - \beta_{k,j+1}^\omega, && k\in V, j\in T, j<M, \omega\in\Omega, && (68)\\
& \gamma_{k1}^\omega \geq \beta_{kM}^\omega, && k\in V,\omega\in\Omega, && (69)\\
& \gamma_{kj}^\omega \leq \beta_{kj}^\omega, && k\in V, j\in T, \omega\in\Omega, && (70)\\
& \gamma_{kj}^\omega \leq 1-\beta_{k,j+1}^\omega, && k\in V, j\in T, j<M, \omega\in\Omega, && (71)\\
& \sum_{j\in T}\gamma_{kj}^\omega = 1, && k\in V,\omega\in\Omega, && (72)
\end{aligned}
$$

$$z^{\omega}_{gkj}, \alpha^{\omega}_{kj}, \beta^{\omega}_{kj}, \gamma^{\omega}_{kj} \in \{0, 1\}, \qquad g \in G, k \in V, j \in T, \omega \in \Omega. \tag{73}$$

The objective function (45) and the constraints (46) ensure that the value of the new auxiliary variable θ corresponds to the highest total completion time of all vessels among all the scenarios in Ω. The description of the remaining constraints is similar to the one presented in the previous section and therefore it is omitted.

Remark 4.1. *The proposed robust model explicitly takes into account all possible realizations of the arrival times of the vessels in the uncertainty set. However, such a model also allows to obtain a solution with some degree of protection against uncertain handling times due to the use of parameter F. This parameter is known in the literature as buffer time and it has been widely used in both BAP and BACASP to absorb delays propagation, see [9, 16, 24, 32].*

5. The importance of a robust solution

In this section, we present an example to illustrate a robust solution for the BACASP and to show the importance of having such kind of solutions.

Let us consider a wharf divided into 7 berth sections and a set of 4 heterogeneous vessels arriving at the wharf during a time horizon of 7 time periods. There are 2 homogeneous cranes operating in the wharf. The length H_k, the nominal (expected) arrival time A_k and the cargo volume $\bar{Q}_k$ to be operated in each vessel $k \in V$ (measured in terms of the number of time periods assuming that only one crane is used) are as follows:

- **Ship 1**: $H_1 = 2$, $A_1 = 1$, $\bar{Q}_1 = 2$;
- **Ship 2**: $H_2 = 2$, $A_2 = 2$, $\bar{Q}_2 = 5$;
- **Ship 3**: $H_3 = 3$, $A_3 = 1$, $\bar{Q}_3 = 1$;
- **Ship 4**: $H_4 = 3$, $A_4 = 3$, $\bar{Q}_4 = 3$.

Let us assume that at most one vessel may arrive 2 time periods later to the wharf.

The first time-space diagram in Figure 2 shows a solution for this BACASP. Ship 1 arrives at the port at time period 1, berths in 5 and departs at time period 3 (completion time equal to 2), ship 2 arrives at the port at time period 2, berths in 0 and departs at time period 7 (completion time equal to 5), ship 3 arrives at the port at time period 1, berths in 1 and departs at time period 2 (completion time equal to 1) and ship 4 arrives at the port at time period 3, berths in 3 and departs at time period 5 (completion time equal to 2). Therefore, the cost of this solution when no delays occur is 10 (2+5+1+2). This solution does not take into account possible delays and, therefore, it is not necessarily a robust solution for the problem. The second diagram in Figure 2 shows the behavior of the solution presented in the first diagram (i.e., keeping the berth positions and the relative position of the vessels) when a delay of 2 time periods on ship 3 occurs. We can observe that the total completion time increases from 10 to 11. This occurs because a delay in ship 3 creates conflicts with ship 2 and ship 4. We can also observe that the impact on the total completion time is only one unit because the second-stage variables (quay crane assignment and scheduling) are optimally adjusted to this scenario.

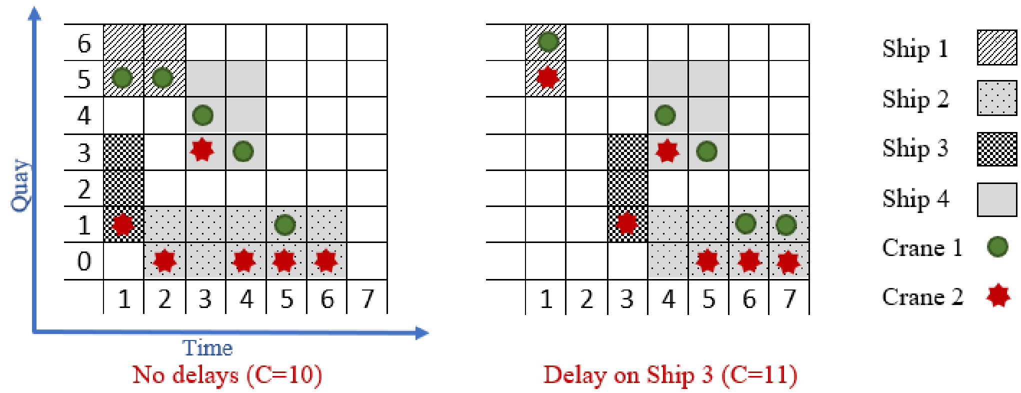

Figure 2: Alternative (not robust) solution for the BACASP.

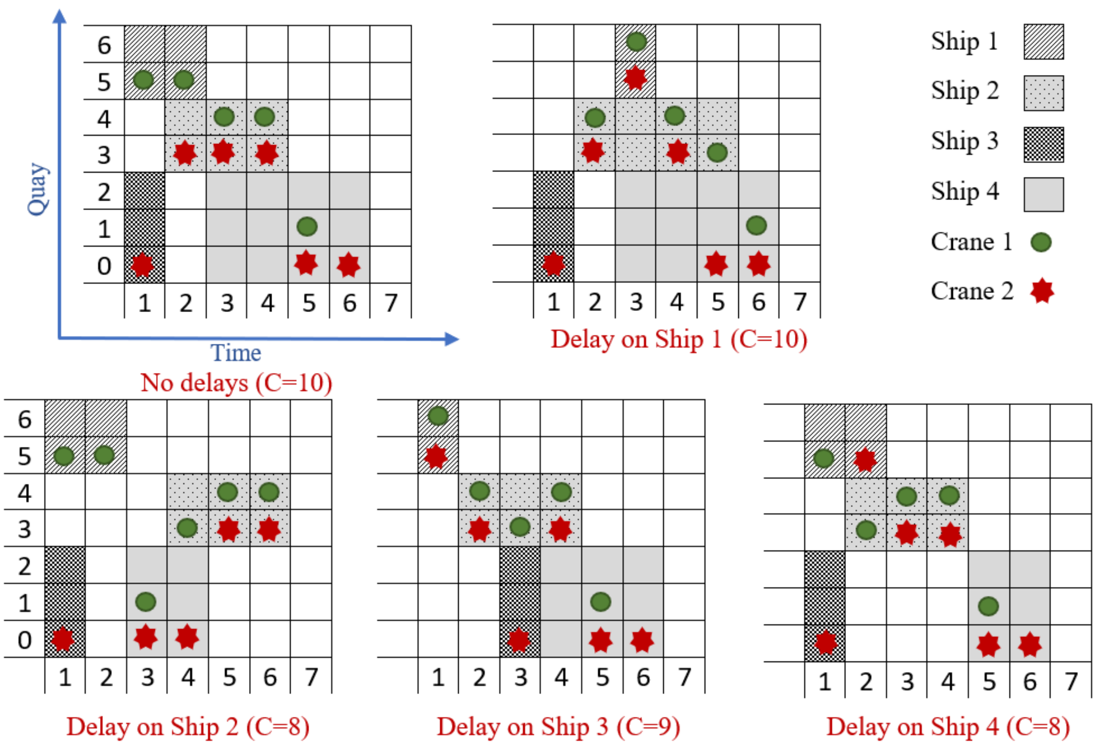

Figure 3: Robust solution for the BACASP.

The first time-space diagram in Figure 3 shows a robust solution for the same BACASP instance. In such a solution, when there are no delays, ship 1 arrives at the port at time period 1 and departs at time period 3 (completion time equal to 2), ship 2 arrives at time period 2 and departs at time period 5 (completion time equal to 3), ship 3 arrives at time period 1 and departs at time period 2 (completion time equal to 1) and ship 4 arrives at time period 3 and departs at time period 7 (completion time equal to 4). Therefore, the cost of the robust solution when no delays occur is 10 (2+3+1+4), the same as the first solution presented in Figure 2. However, this solution takes into account possible delays. If there is a delay of 2 time periods in the arrival time of ship 1 the total completion time of the solution remains the same. If a delay of 2 time periods in the arrival time of ships 2, 3, and 4 occurs, then the total completion time of the solution is reduced to 8, 9, and 8, respectively.

The presented example also illustrates situations where delays may have no negative impact on the total completion time of the solution (as shown in the last three diagrams presented in Figure 3). This contrasts with other robust optimization problems where the worst-case occurs for scenarios that consider the maximum allowed deviations from the nominal values. This is the case of maritime transportation problems with time windows [1] and maritime inventory routing problems [3]. This is a very important remark that has a strong impact on the solution methods presented next.

6. Exact decomposition algorithm

The number of scenarios in the robust model presented in Section 4 is usually large (or even infinite), making the direct use of such a model impractical. However, in general, only a small set of scenarios is needed to obtain an optimal robust solution, since several scenarios are dominated by the others [3]. Finding the most relevant scenarios is a very challenging problem that can be solved by using a decomposition algorithm, see [33].

In this section, we describe an exact decomposition algorithm (DA) for solving the robust BACASP working as follows. We consider a master problem (MP), where the constraints involving second-stage variables are written only for a small subset $\Xi \subset \Omega$ of scenarios. The objective function value of the MP gives the highest sum of the minimum total completion time of all vessels among all scenarios in Ξ, henceforth referred to as *current worst completion time* and denoted by $\bar{\theta}$. Given a feasible first-stage solution to the MP, we check whether there is an omitted scenario $\omega \in \Omega \backslash \Xi$ leading to a total completion time θ^ω greater than $\bar{\theta}$ by solving a separation problem known as adversarial problem (AP). If such a scenario ω is found, then it is added to Ξ and the augmented MP is solved again. Otherwise, when no such a scenario exists, the procedure stops, meaning that the current solution is already an optimal robust solution, see Figure 1. The general idea of a DA is summarized in Algorithm 1.

Algorithm 1 Decomposition algorithm for the robust problem.

1: Initialize $\Xi = \{\omega^0\}$, where ω^0 is the scenario corresponding to the nominal arrival times
2: Solve the master problem for set Ξ and compute the current worst completion time $\bar{\theta}$
3: **while** There is a scenario ω in $\Omega \setminus \Xi$, leading to a total completion time θ^ω greater than $\bar{\theta}$ **do**
4:    Add ω to Ξ and add the corresponding variables and constraints to the MP
5:    Solve the new master problem
6: **end while**

Decomposition algorithms have been widely used in the past for solving complex two-stage robust problems, but its application to each problem is far from being direct. It is important to remark that DA has never been used before for solving the robust BACASP. In the decomposition algorithm we propose, the MP is defined by the objective function (45), by the constraints (2)-(6), (18), (20)-(28) and by the constraints (46)-(73) defined for a subset $\Xi \subset \Omega$ of scenarios. Given a first-stage solution $S_1 = \{\bar{x}, \bar{y}, \bar{b}, \bar{\pi}, \bar{\sigma}\}$ obtained by the MP, the AP is a complex max-min problem defined as follows:

$$
\begin{aligned}
\max_{\omega \in \Omega \backslash \Xi} \quad \min_{t^\omega, c^\omega, z^\omega, \alpha^\omega, \beta^\omega} & \sum_{k \in V} (c_k^\omega - A_k^\omega) \\
s.t. \quad & (47) - (73), \\
& x_{kl} = \bar{x}_{kl} \qquad k, l \in V, \\
& y_{kl} = \bar{y}_{kl} \qquad k, l \in V, \\
& b_k = \bar{b}_k \qquad k \in V.
\end{aligned}
$$

Solving a max-min problem is hard since such a problem is not convertible into a single optimization problem, see [23]. The adversarial problem can be solved as follows. Start by solving the inner minimization problem for each scenario $\omega \in \Omega \backslash \Xi$ independently to obtain a second-stage solution $S_2^\omega = \{t^\omega, c^\omega, z^\omega, \alpha^\omega, \beta^\omega\}$ and the corresponding total completion time θ^ω. Then, the scenario $\bar{\omega}$ leading to the highest completion time is chosen. If such a completion time is greater than the current completion time, then scenario $\bar{\omega}$ is added to the MP. Otherwise, the current solution determined by the MP is optimal since there is not a scenario leading to a completion time greater than the current one.

The efficiency of a DA depends on three factors: the hardness of the master problem, the hardness of the adversarial problem, and the total number of global iterations required to obtain an optimal robust solution. In our case, the MP is a large MIP model that becomes more difficult to solve as the number of scenarios in Ξ increases. Solving the AP implies to solve a large number of MIP subproblems (one for each scenario), and although the first-stage solution is fixed when solving those subproblems, each subproblem is still difficult to solve due to the large number of binary variables presented. Note that each MIP subproblem corresponds to an assignment and scheduling problem which is known to be a very hard problem. Therefore, it is of vital importance to reduce i) the number of global iterations of the DA; ii) the number of MIP subproblems to solve occurring in the AP; and iii) the complexity of the master problem. To achieve these three goals we propose a new warm start technique, a scenario reduction procedure, and a rolling horizon heuristic, respectively. The use of both the warm start technique and the scenario reduction procedure keeps the decomposition algorithm an exact procedure.

6.1. Warm start

In the first step of Algorithm 1, the set Ξ is initialized with a single scenario corresponding to the nominal arrival times. A warm start strategy consists of initializing subset Ξ with a scenario having some arrival time delays such that the obtained solution is *more robust* than the one corresponding to the nominal scenario. By doing so, we expect to reduce the total number of iterations of the DA and therefore to reduce the total computational time.

It is worth mentioning that the scenario with delays used to initialize the DA must be carefully chosen, otherwise, the warm start can produce an undesirable effect by increasing the total computational time. Note that some delays on the arrival times may reduce congestion at the port leading to lower cost solutions than the one corresponding to the nominal scenario (see the example in Section 5). Such kind of scenarios should not be used in a warm start technique.

Based on the intuition that higher cost solutions are associated with scenarios leading to an increase of the port congestion, we propose a heuristic approach to find the scenario to use in the warm start. The main idea of the proposed heuristic is to find a scenario with some delays leading to a reduction of the slacks between the arrival times of the vessels. Therefore, we expect that in such a scenario more vessels arrive at the port almost at the same time or at least the arrivals of the vessels are closer. When several vessels arrive at the port at the same time, limited resources such as quay cranes have to be shared by those vessels, meaning that not all the vessels will be served with the minimum possible service time. Therefore, the total completion time of those vessels increases as well as the aggregated total completion time associated with such a scenario. Hence, it is likely that the worst scenarios are the ones with smaller slacks between the arrivals of the vessels. The proposed heuristic, called slack reduction heuristic, is summarized in Algorithm 2.

Algorithm 2 Slack reduction heuristic

Initialize $MinSlack$ as the total number of periods M and sort all the vessels by increasing order of nominal arrival times.
for each scenario $\omega \in \Omega$ **do**
    $\text{Slack}^{\omega} := \sum_{k=2}^{N-1} \min\{|A_k^{\omega} - A_{k-1}^{\omega}|, |A_{k+1}^{\omega} - A_k^{\omega}|\}$
    if $\text{Slack}^{\omega} < MinSlack$ **then**
        $MinSlack \leftarrow \text{Slack}^{\omega}$
        $\bar{\omega} \leftarrow \omega$
    end if
end for
Return scenario $\bar{\omega}$

To better understand the slack reduction heuristic presented in Algorithm 2, let us analyze the following example. Consider a set of 6 vessels and the corresponding vector of nominal arrival times $n = (1,\ 2,\ 7,\ 8,\ 10,\ 12)$. Also consider two arrival time scenarios $w^1 = (1,\ 4,\ 7,\ 9,\ 10,\ 12)$ and $w^2 = (2,\ 2,\ 8,\ 8,\ 10,\ 12)$, both with 2 delays of one and two time periods. Then, we have

$$\text{Slack}^{n} = \min\{|2-1|, |7-2|\} + \min\{|7-2|, |8-7|\} + \min\{|8-7|, |10-8|\} \min\{|10-8|, |12-10|\}$$

$$\text{Slack}^{n} = 1 + 1 + 1 + 2 = 5$$

$$\text{Slack}^{\omega^1} = 3 + 2 + 1 + 1 = 7$$

$$\text{Slack}^{\omega^2} = 0 + 0 + 0 + 2 = 2$$

This means that scenario ω^2 is more preferable for the warm start than both scenarios n and ω^1 since it increases the port congestion. Note that in this scenario there are more vessels mooring at the same time and consequently not all vessels will be operated with maximum efficiency (at least) in the period corresponding to the arrival time (since the number of cranes is limited).

6.2. Scenario reduction procedure

At each iteration of Algorithm 1 (Step 3), the AP has to be solved to identify the scenario(s) to add to the MP, that is, the scenario(s) leading to a total completion time greater than the current worst completion time. As remarked before, each scenario can be evaluated by solving an hard MIP subproblem and the number of scenarios to evaluate can be large. Since the complexity of each MIP subproblem increases as the number of vessels increases, following this procedure to evaluate a set of scenarios (even small) can be very time consuming. To reduce the time associated with the AP, we propose a new procedure, called scenario reduction procedure (SRP), that allows to discard irrelevant scenarios and therefore to reduce the number of MIP subproblems to solve. Given a first-stage solution $S_1 = \{x, y, b, \pi, \sigma\}$ found by the MP and a scenario $\omega \in \Omega \backslash \Xi$ to be evaluated, we heuristically determine a feasible second-stage solution $S_2^\omega = \{t^\omega, c^\omega, z^\omega\}$ for that scenario, i.e., a feasible crane assignment. By computing the total completion time in scenario ω, i.e., $\theta^\omega = \sum_{k \in V}(c_k^\omega - A_k^\omega)$, we get an upper bound for the value of the optimal crane assignment (obtained when the corresponding MIP subproblem is solved). Therefore, all scenarios ω such that $\theta^\omega \leq \bar{\theta}$ can be discarded in the current iteration since they will not lead to a completion time greater than the current worst completion time. When the feasible crane assignment determined does not allow to discard a given scenario $\omega \in \Omega \backslash \Xi$, a different crane assignment can be generated and the process can be repeated until either the scenario ω is discarded or a predefined number of iterations is reached. If none of the obtained crane assignments allows to discard the scenario, then the MIP subproblem needs to be solved for the corresponding scenario. When solving the MIP subproblem through branch-and-cut, every feasible crane assignment found is evaluated and if the completion time is lower than the current worst completion time the scenario is discarded. Otherwise, the MIP subproblem ends when the optimal crane assignment θ^{ω^*} satisfies $\theta^{\omega^*} > \bar{\theta}$, and scenario ω is added to the MP.

The description of the SRP is summarized in Algorithm 3.

Algorithm 3 Scenario reduction procedure (SRP)

1: **Input:** A first-stage solution $S_1 = \{x, y, b, \pi, \sigma\}$ and the current worst completion time $\bar{\theta}$
2: Set *Feasible*=true
3: **for** each scenario $\omega \in \Omega \backslash \Xi$ **do**
4: Successively determine a feasible crane assignment according to S_1 (and compute θ^ω) until either $\theta^\omega \leq \bar{\theta}$ or a predefined number of iterations is reached
5: **if** $\theta^\omega \leq \bar{\theta}$ **then**
6: Discard scenario ω
7: **else**
8: Solve the MIP subproblem to obtain the optimal crane assignment and compute $\theta^{\omega*}$
9: **if** $\theta^{\omega*} \leq \bar{\theta}$ **then**
10: Discard scenario ω
11: **else**
12: Set *Feasible*=false, $\bar{\omega} = \omega$ and go to Step 16
13: **end if**
14: **end if**
15: **end for**
16: **if** not *Feasible* **then**
17: Return scenario $\bar{\omega}$
18: **end if**

Since the size of the MP rapidly grows as the number of scenarios increases, we only insert one scenario per iteration in the MP. This means that each time the SRP is employed, not all the scenarios need to be evaluated since the SRP stops when the first disrupted scenario is found.

Remark 6.1. *An alternative version of the SRP is to evaluate all the scenarios and then return the scenario leading to the highest completion time. This idea was followed in [2, 3] and, in general, it allows to reduce the number of global iterations performed by the DA comparing with the approach* *that consists of choosing the first scenario found leading to a total completion time greater than the current one. For the reasons mention before, the evaluation of each scenario in the robust BACASP can be very time consuming. Therefore, in this paper, we adopt the first strategy since it avoids evaluating all the scenarios at each iteration and consequently to save a lot of time.*

At Step 4 of Algorithm 3, several crane assignments must be quickly determined. Such crane assignments can be obtained by using a new heuristic (presented in the next section), called scenario evaluation heuristic.

6.2.1. Scenario evaluation heuristic

Given a feasible first-stage solution $S_1 = \{x, y, b, \pi, \sigma\}$, the scenario evaluation heuristic (SEH) aims to find feasible crane assignments for a given scenario. A general description of this heuristic is given in Algorithm 4.

Algorithm 4 Scenario evaluation heuristic (SEH)

1: **Input:** A feasible first-stage solution $S_1 = \{x, y, b, \pi, \sigma\}$ and a scenario $\omega \in \Omega \backslash \Xi$
2: $Cost \leftarrow 0$
3: **for** $j \in T$ **do**
4:     Define $\bar{V}$ as the set of vessels available to be operated in period j
5:     $Cost \leftarrow Cost + \#\bar{V}$
6:     **if** $\#\bar{V} = 1$, i.e., $\bar{V} = \{v_1\}$ **then**
7:         Assign the maximum possible crane capacity to vessel v_1
8:     **end if**
9:     **if** $\#\bar{V} \geq 2$ **then**
10:         Extract a subset $\tilde{V} = \{v_1, v_2\}$ with two vessels from $\bar{V}$
11:         Define a priority vessel (let us assume v_1 as priority vessel)
12:         Assign the maximum possible crane capacity to vessel v_1
13:         Assign the maximum possible crane capacity to vessel v_2 taking into account that cranes assigned to v_1 cannot be assigned to v_2
14:         If beneficial, readjust cranes by moving some cranes assigned to v_1 to v_2
15:     **end if**
16: **end for**
17: Return $Cost$

The SEH is a greedy algorithm that assigns cranes to vessels sequentially over each time period. At each time period $j \in T$, the SEH identifies a set of available vessels to be operated (Step 4), i.e., the vessels that arrived until time j and are not fully unloaded at this time period. If there is only one available vessel (Step 6), then it is operated with the maximum possible crane capacity. We assume that the SEH only assigns cranes to at most two vessels in each time period. Therefore, if more than two vessels are available (Step 9) only two of them are selected to be served (Step 10). Two alternative criteria are used to select the vessels:

S_{SC} - Select the vessels with smaller cargo onboard.
S_{AE} - Select the vessels that arrived earlier.

Having only two vessels in set $\tilde{V}$ we identify a priority vessel to be served (Step 11). Three alternative criteria are used to select the priority vessel:

P_R - Randomly select the priority vessel.

P_{SC} - Select the vessel with smaller cargo as priority.

P_{SS} - Select the vessel with smaller slack as priority. Given a vessel $k \in \tilde{V}$ define set $\tilde{V}_k$ as the set of vessels with an arrival time greater than or equal to A_k^ω sharing berth sections with vessel k. The slack of vessel k is then defined as $s_k = \min_{\ell \in \tilde{V}_k}\{A_\ell^\omega - c_k^{min}\}$, where c_k^{min} is the minimum completion time of vessel k (computed by assuming that vessel k is always served with maximum crane capacity).

After selecting the priority vessel, the maximum possible number of cranes is assigned to that vessel and the maximum number of remaining cranes (not assigned to the priority vessel) is assigned to the non-priority vessel. However, in some cases, some cranes are moved from the priority vessel to the non-priority one (Step 14). This is done in the following way. We start by computing the number (nc_{v_1}) of cranes assigned to the priority vessel in the last period in which it is going to be operated. If such number is lower than the maximum number of cranes that can work simultaneously on vessel v_1 (i.e, $nc_{v_1} < NC_{v_1}$) then we can move at most $NC_{v_1} - nc_{v_1}$ cranes from vessel v_1 to vessel v_2. This procedure does not affect the completion time of the priority vessel but may lead to a lower completion time of the non-priority vessel.

Combining the different ways of selecting vessels in Steps 10 and 11 of Algorithm 4, we obtain six variants of the SEH: $SEH_{S_{SC}+P_{SC}}$, $SEH_{S_{SC}+P_{SS}}$, $SEH_{S_{SC}+P_R}$, $SEH_{S_{AE}+P_{SC}}$, $SEH_{S_{AE}+P_{SS}}$ and $SEH_{S_{AE}+P_R}$. It should be noted that both $SEH_{S_{SC}+P_R}$ and $SEH_{S_{AE}+P_R}$ variants can be applied several times to produce a large set of feasible assignments due to the random operator, which is not the case of the remaining four variants.

Remark 6.2. *The proposed decomposition algorithm is based on the model defined in Section 4 which aims to minimize the worst total completion time. Such a model can easily be modified to optimize different criteria such as the waiting time or the total departure time delay (when maximum departure times are considered). If a different criterion is chosen, the proposed decomposition algorithm and all the heuristics described before can be straightforwardly redefined according to it.*

7. Rolling horizon heuristic

The MP becomes harder to solve as the number of both vessels and scenarios increases, which can make it difficult to find feasible solutions for some instances quickly. This statement was already verified in [4] for the deterministic BACASP. Therefore, to handle large size instances we use a rolling horizon heuristic (RHH). Rolling horizon heuristics have been successfully applied before for solving BACASP variants (see [4, 31]) but they were never applied inside of a decomposition algorithm to solve the MP.

The RHH used in this paper is similar to the one presented in [4]. The main idea of the RHH is to split the planning horizon into smaller sub-horizons, and then solve a tractable subproblem for each sub-horizon at each iteration. The RHH starts by considering only a subproblem with some of the first vessels arriving at the port in the nominal scenario. Then, at each iteration $i > 1$, the subproblem to solve is defined by the inclusion of a new set of vessels (not included yet in subproblem $i-1$). The number nv of vessels included at each iteration is fixed. The time horizon T_i of the subproblem to be solved in iteration i is defined in terms of the later possible arrival time of the last vessel k arriving at the port and considered in the subproblem as

$$T_i := \max_{\omega \in \Omega} A_k^{\omega} + nt$$

where nt is a given tolerance.

At each iteration $i > 1$, the berth locations of the vessels considered in the previous iterations are frozen (as well as the remaining first-stage decision variables). However, to ensure a soft transition between subproblems, the second-stage decision variables are kept free during the last ut time periods of the time horizon T_{i-1} of the previous iteration. The second-stage decision variables in periods from 1 to $T_{i-1} - ut$ are also frozen. Hence, at iteration i, we solve a restricted problem where the berth decisions are considered only for the new nv vessels and part of the scheduled decisions are fixed.

The presented RHH has three parameters:

nv : number of new vessels to consider at each iteration;

nt : number of periods to consider in each subproblem after the arrival time of the last vessel;

ut : number of last periods in time horizon T_{i-1} in which the second-stage variables are free when solving subproblem at iteration i.

The RHH is described in Algorithm 5.

Algorithm 5 Rolling horizon heuristic (RHH)

1: **Input:** nv, nt and ut
2: Define $T_1 := \max_{\omega \in \Omega} A_{nv}^{\omega} + nt$ as the limited time horizon of the first subproblem
3: Define $U := \lceil N/nv \rceil$ as the number of iterations to cover the full planning horizon
4: Solve the first subproblem defined with time horizon T_1 for nv vessels
5: **for all** $i \in \{2, \dots, U\}$ **do**
6: Fix x, y, b, π, σ variables for the vessels considered in iteration $i - 1$
7: Fix t, z, α, β for all vessels considered in iteration $i - 1$ for all time periods $\{1, \dots, T_{i-1} - ut\}$ and for all scenarios in Ξ
8: Define the new time horizon as $T_i := \max_{\omega \in \Omega} A_{i \times nv}^{\omega} + nt$
9: Solve the restricted mixed integer problem for $i \times nv$ vessels and T_i time periods
10: **end for**

7.1. RHH analysis

The proposed RHH is used inside the decomposition algorithm for solving the arising master problems, which are MIP models defined by constraints (45) - (73). The RHH iteratively solves several subproblems independently whose dimension depends on the parameters nv, nt, and ut. Since each master problem is partitioned into several subproblems that are solved independently, the optimality of the obtained solution can be lost, specially when the number of vessels nv considered in each subproblem is small. Considering a large number of vessels nv in each subproblem leads to better quality solutions, but the computational time required for solving each subproblem can significantly increase. The total number of time periods considered in each subproblem highly depends on the range of the expected arrival times of the vessels considered in that subproblem. The larger the range of the expected arrival times, the larger the number of time periods considered, and therefore, the larger the dimension of the subproblem. The parameter ut is used to make soft transitions between subproblems, therefore, the larger the value ut, the softer the transitions between subproblems, and therefore, the higher the probability of obtaining optimal solutions. However, large values of the parameter ut increase the hardness of each subproblem, and therefore, the computational time required for solving them.

Hence, the DA combined with the RHH becomes a heuristic approach since the cost of the obtained solution can be larger than the one obtained by the exact DA (without RHH).

8. Computational results

In this section, we report the computational experiments carried out to test both the exact decomposition algorithm (DA) and the rolling horizon heuristic (RHH). In Section 8.1 we describe the instances used and discuss some implementation details of the algorithms. In Sections 8.2 and 8.3 we report the computational results for the randomly generated instances considering both homogeneous and heterogeneous quay cranes, respectively. In subsection 8.4 we present the results for a set of instances from the literature while in Section 8.6 we solve large size instances with up to 60 vessels and 168 time periods.

All the tests were run on a computer with a CPU Intel(R) Core i7, with 16GB RAM, and using the Xpress Optimizer Version 27.01.02 with the default options.

8.1. Instances used and implementation details

To test both the exact DA and the RHH, we use randomly generated instances as well as instances of the literature. The instances of the literature correspond to the ones used in [4]. The randomly generated instances are obtained as follows.

The number M of time periods is fixed to 60. This value is taken from the larger practical instances given in [4] that motivated this work. It was observed by other authors [25] that, in general, the arrival time of a vessel can only be estimated with accuracy one or two days before its actual arrival. For larger time horizons the solution approaches presented in this paper can still be used by adjusting the time period. In the literature, we can find periods up to 3 hours (see [10]). Note that with an intermediate case of 2 hours, the time horizon corresponds to five working days which may cover most practical cases where a reasonable estimate of the arrival times is known. The safety time F between two vessels is one time period. The total number N of vessels arriving at the port varies in $\{6, 7, \ldots, 15\}$. For each number of vessels, ten instances are randomly generated. We assume that the time between arrivals follows an exponential distribution with rate N/M, thus the arrival times are sequentially generated according to that probability distribution. After generating all arrival times, it is necessary to ensure that all vessels can be operated within the time horizon. To do so, the arrival times are mapped into the interval $[1, M-10]$ according to the formula:

$$A_k^{new} := \left[1 + (A_k^{old} - 1)\frac{(M-10)-1}{A_N^{old} - 1}\right]$$

where A_k^{old} and A_k^{new} represent the generated and the mapped arrival time for vessel k, respectively, and $[\cdot]$ is the round operator. The size of the vessels (measured in terms of the number of berth sections) varies in $\{5, 6, 7\}$ while the cargo (in ton.) is an integer number varying in $[1,000; 8,000]$. The number J of berth sections is 34 and the number of available cranes is 7. The processing rate of each crane is constant and equal to 263.644 ton/hr. We assume that at most $NC_k = 4$ cranes can simultaneously operate in vessel $k \in V$. It is worth mentioning that all the parameter values defined here are based on the real-world instances used in [4]. Since we consider 10 different numbers of vessels and for each of those numbers 10 randomly generated instances are obtained, the test set is composed of 100 instances.

The MCB uncertainty set is built as follows. The vessels are divided into $ng = 3$ groups and at most one vessel of each group can arrive with a delay (i.e., $\Gamma_j = 1$ for all $j \in \{0, 1, 2\}$). The maximum allowed delay $\hat{A}_k$ is the same for all vessels $k \in V$ and is equal to two time periods. Therefore, since we are considering a discrete time horizon, it means that each vessel can arrive either on time or with a delay of one or two time periods. The randomly generated instances used are online available at http://sweet.ua.pt/aagra/. Each of these instances is identified with the name *R_N_i* where N ranges from 6 to 15 and corresponds to the number of vessels used in the instance and i ranges from 1 to 10 and denotes the instance number.

Now we describe some implementation details associated with both the scenario reduction procedure (SRP) and the rolling horizon heuristic (RHH).

In Step 4 of Algorithm 3, a variant of the SEH is successively used to find a feasible crane assignment leading to a completion time lower than or equal to the current worst completion time. For each scenario $\omega \in \Omega \backslash \Xi$, the choice of the SEH variant is defined as follows. The first variants employed are the static SEH variants (the ones with no random steps) in the following order:

$$\text{(Static Variants)} \quad SEH_{S_{SC}+P_{SC}} \rightarrow SEH_{S_{SC}+P_{SS}} \rightarrow SEH_{S_{AE}+P_{SC}} \rightarrow SEH_{S_{AE}+P_{SS}}$$

It is important to recall that when one of the SEH variants finds a crane assignment that allows to discard the scenario, the next variants are not employed. When none of these four variants find a desirable crane assignment, the random SEH variants are applied in the following order:

$$\text{(Random Variants)} \quad SEH_{S_{SC}+P_R} \ (\times 100) \rightarrow SEH_{S_{AE}+P_R} \ (\times 100).$$

Each of these two random variants is executed 100 times, allowing to obtain a large number of different crane assignments and therefore increasing the probability of discard scenarios.

The RHH has three input parameters: the number nv of new vessels to consider in each iteration, the number nt of periods to consider in each subproblem after the arrival time of the last vessel, and the number ut of last periods of time horizon T_{i-1} in which the second-stage variables are free when solving the subproblem corresponding to iteration i. In our experiments, we use nv=3 since, according to the results presented in [4], this is the value leading to a better trade-off between solution cost and computational time for the deterministic BACASP. Note that, in general, the quality of the solutions increases and the computational time required to solve each subproblem decreases as the number nv increases. Taking into account that the maximum cargo of a vessel is 8,000 ton., the maximum number of cranes simultaneously operating in a vessel is 4, and the processing rate of each crane is 263.644 ton/hr, we can estimate the maximum completion time of a vessel (assuming that it is operated with maximum resources) as $\left\lceil \frac{8000}{4 \times 263.644} \right\rceil = 8$. Hence, we must have $nt \geq 8$. Since it may not be possible to allocate the maximum resources to each vessel, using $nt = 8$ can lead to infeasible solutions. To avoid such infeasibilities we use $nt = 10$ and set $ut = nt/2 = 5$.

8.2. Results for BACASP with homogeneous cranes

In this section, we present the results for the randomly generated instances of the robust BACASP with homogeneous cranes described in the previous section. To understand the complexity of our instances and the importance of solving the robust BACASP with a decomposition algorithm enhanced with the proposed SRP, we start by presenting in Table 1 the total number $\#\Omega$ of possible scenarios considered in our instances in terms of the number of vessels. In this table we also report the computational time required to evaluate all the scenarios in one iteration of the DA when i) each exact MIP subproblem is solved through Xpress (Column $Xpress$); ii) only the 4 static SEH variants are used in SRP (Column $Static\ Variants$); and iii) only the two random SEH variants are used (100 times each) in SRP (Column $Random\ Variants$).

Table 1: Total number of scenarios and computational times (in seconds) required to evaluate all the scenarios

N	$\#\Omega$	$Xpress$	$Static\ Variants$	$Random\ Variants$
7	175	117	<1	25
9	343	305	<1	67
11	567	721	3	130
13	841	1356	6	254
15	1331	2627	8	465

Table 1 shows that the total number of scenarios handled is much larger than the number of scenarios considered in previous works (see, for example, [19, 31]). Moreover, the time required by the static variants is very short, meaning that when these variants are able to obtain crane assignments that allow to discard a large number of scenarios the adversarial problem can be solved quickly. The computational time associated with the random variants (executed 100 times each) is much lower than the one associated with solving all subproblems with Xpress. It is worth recalling that the reported computational times refer to the evaluation of all scenarios, but not all the scenarios need to be evaluated at each iteration of the DA.

Now we study the effect of using the scenario reduction procedure (SRP) as well as the warm start (WS) technique in the exact decomposition algorithm (DA). The obtained results are displayed in Table 2. The first column identifies the number of vessels, while the second reports (for each number of vessels) the average cost of the obtained solutions over the set of the ten randomly generated instances. The third and the fourth columns report the average total computational time (in seconds) spent by the DA when solving the master problems and the adversarial problems, respectively, while the fifth column reports the average total time (in seconds) spent on solving each of the ten instances. The sixth column displays the average number of iterations performed by the DA (corresponding also to the average number of scenarios added). The remaining two groups of four columns display similar information for the DA combined with the SRP (DA_{SRP}) and for the DA combined with both SRP and WS (DA_{SRP+WS}).

Table 2: Results for the DA and its improvement strategies for the homogeneous instances

N	Cost	DA T_{MP}	DA T_{AP}	DA T_{Total}	DA $\#it$	DA_{SRP} T_{MP}	DA_{SRP} T_{AP}	DA_{SRP} T_{Total}	DA_{SRP} $\#it$	DA_{SRP+WS} T_{MP}	DA_{SRP+WS} T_{AP}	DA_{SRP+WS} T_{Total}	DA_{SRP+WS} $\#it$
6	33	17	83	100	2.6	13	8	21	2.6	8	6	14	1.6
7	38	19	148	167	2.9	21	11	32	3.0	16	10	26	2.5
8	43	126	444	570	3.5	139	57	196	3.5	116	35	151	3.3
9	41	152	651	803	4.1	84	95	179	3.6	124	73	197	3.0
10	42	176	886	1062	3.9	126	103	229	3.7	102	62	164	2.9
11	46	712	1711	2423	4.9	728	195	923	4.8	220	115	335	3.6

Table 2 reports results for instances with a number of vessels lower than or equal to 11 since the computational time tends to increase as the number of vessels increases and there are already two instances with 11 vessels (*R_11_6* and *R_11_9*) that could not be solved to optimality within 6 hours by the DA without improvements. Therefore, the last line of the table refers to a set of only 8 instances (the ones solved to optimality by the DA within 6 hours).

Table 2 shows that, in general, the largest portion of time required by the DA is spent on solving the adversarial problems. The time required to solve the adversarial problems can be reduced in 88% on average when the DA is combined with the SRP. Furthermore, the total time of the DA can be reduced in around 82% when it is combined with both the SRP and the WS. Columns $\#it$ show that the number of iterations performed (scenarios added) decreases when the warm start strategy is used. This may explain the fact that, in general, the lowest running times correspond to the DA_{SRP+WS} strategy.

The results presented in Table 2 clearly demonstrate the benefits of using both the SRP and WS techniques. Therefore, we only combine the rolling horizon heuristic (RHH) with the DA_{SRP+WS} approach. For ease of notation, the resulting approach is denoted by RHH. The obtained results are displayed in Table 3. The first column identifies the number of vessels. Columns T_{MP}, T_{AP}, T_{Total}, and $\#it$ have the same meaning as in Table 2. To obtain the results reported in the last column, we start by computing the gap of the solution obtained by the RHH with respect to the optimal solution obtained by the DA_{SRP+WS} approach for each of the 10 instances, according with the formula

$$Gap_i(\%) = \frac{C_i^{RHH} - C_i^*}{C_i^*} \times 100, \qquad i = 1, \ldots, 10$$

where C_i^{RHH} denotes the cost of the solution obtained by RHH when solving instance i and C_i^* denotes the cost of the optimal solution of instance i. The last column of Table 3 displays the average of those gaps.

Table 3: Computational results to access the performance of the RHH when solving homogeneous instances

	DA_{SRP+WS}				RHH				
N	T_{MP}	T_{AP}	T_{Total}	$\#it$	T_{MP}	T_{AP}	T_{Total}	$\#it$	$Gap(\%)$
6	8	6	14	1.6	5	5	10	1.7	0.0
7	16	10	26	2.5	11	7	18	2.5	1.0
8	116	35	151	3.3	32	26	57	3.3	0.5
9	124	73	197	3.0	39	70	109	3.3	0.2
10	102	62	164	2.9	20	55	76	2.6	0.7
11	1123	163	1286	4.0	59	164	224	4.0	0.0

The results reported in Table 3 show that RHH is faster than DA_{SRP+WS} specially because it drastically reduces the computational time required to solve the master problems. Moreover, we can see from the last column that the average gaps associated with the solutions obtained by the RHH are lower than 1%, indicating a good performance of this approach in terms of the cost of the obtained solutions. The last row of Table 3 reports the results for only 9 of the 10 instances with 11 vessels since the instance *R_11_6* could not be solved to optimality within 6 hours by the DA_{SRP+WS} approach. That instance was solved with RHH in 1196 seconds.

Table 4 displays the results obtained with RHH when solving the large size instances with a number of vessels ranging from 12 to 15. The first column identifies the number of vessels. The second, third and, fourth columns report the average cost of the obtained solutions, the average total time required by RHH to solve each of the ten instances, and the average number of iterations performed. Column S_{Total} shows the average number of scenarios evaluated per instance when the RHH is employed. Columns S_S and S_R show the average number of scenarios eliminated by the static SEH variants and by the random SEH variants, respectively. The numbers in parenthesis correspond to the percentage of scenarios eliminated in each case.

Table 4: Computational results obtained by RHH when solving large size homogeneous instances

N	Cost	T_{Total}	$\#it$	S_{Total}	S_S	S_R
12	54	120	3.2	1172	1018 (89%)	137 (9%)
13	55	294	3.6	1199	1060 (90%)	73 (4%)
14	68	1829	4.8	1937	1360 (73%)	354 (15%)
15	60	1726	7.1	2678	2126 (80%)	351 (12%)

Table 4 shows that the average number of scenarios evaluated per instance by RHH is large. The obtained results also reinforce the importance of the SRP since, on average, 98% of the scenarios are eliminated by both static and random SEH variants. Is it worth reminding that the random SEH variants are just employed when the static SEH variants are not able to find a crane assignment that allows to discard the scenario. Note also that the total number of scenarios evaluated (Column S_{Total}) is larger than the one presented in Table 1 since some scenarios are evaluated in more than one iteration of the DA.

8.3. Results for BACASP with heterogeneous cranes

In this section, we present the results for the randomly generated instances of the robust BACASP with heterogeneous cranes. The heterogeneous cranes instances were obtained from the homogeneous instances by increasing the processing rate of cranes 3 and 4 from 263.644 to 319.001. The processing rate of the remaining cranes 1, 2, 5, 6, and 7 are kept unchanged and equal to 263.644 units per hour.

Table 5 shows the effect of using both SRP and WS techniques in the exact decomposition algorithm. The description of each column is the same as in Table 2.

Table 5: Results for the DA and its improvement strategies for the heterogeneous instances

		DA				$DA+SRP$				$DA+SRP+WS$			
N	Cost	T_{MP}	T_{AP}	T_{Total}	$\#it$	T_{MP}	T_{AP}	T_{Total}	$\#it$	T_{MP}	T_{AP}	T_{Total}	$\#it$
6	31	34	116	150	2.5	52	31	83	2.5	20	27	47	1.5
7	36	203	233	436	3.4	193	44	237	3.1	74	20	94	2.1
8	36	190	399	589	3.9	245	99	344	4.0	108	19	127	3.4

Table 5 reports results for instances with at most 8 vessels since there are already three instances with 8 vessels (*R_8_1*, *R_8_2*, and *R_8_8*) that could not be solved to optimality within 6 hours by the DA without improvements. Therefore, the last line of the table refers to a set of only 7 instances (the ones solved to optimality by the DA within 6 hours).

The conclusions drawn from Table 5 are according to the ones obtained from Table 2, i.e., the time required to solve the adversarial problems can be reduced in 77% on average when the DA is combined with the SRP and the total time of the DA can be reduced in 74% on average when it is combined with both the SRP and WS. The number of iterations performed also decreases when the warm start strategy is used. It is important to note that there are 3 heterogeneous instances with 8 vessels that were not solved to optimality by the DA without improvements within 6 hours, while all the homogeneous cranes instances with up to 10 vessels were solved by the DA. This indicates that the instances with heterogeneous cranes are more difficult to solve than the ones in which homogeneous cranes are used.

Table 6 displays results similar to the ones presented in Table 3 to evaluate the performance of the RHH.

Table 6: Computational results to access the performance of the RHH when solving the heterogeneous instances

	DA_{SRP+WS}				RHH				
N	T_{MP}	T_{AP}	T_{Total}	$\#it$	T_{MP}	T_{AP}	T_{Total}	$\#it$	$Gap(\%)$
6	20	27	47	1.5	8	22	30	1.5	0.0
7	81	22	103	2.2	28	22	50	2.1	0.6
8	108	19	127	3.4	19	5	24	2.9	0.4

The results reported in Table 6 show that RHH is faster than DA_{SRP+WS} and the average gaps associated with the solutions obtained by RHH are lower than 0.6%, which reveals a good performance of the RHH. The last row of Table 6 reports results for only 7 of the 10 instances with 8 vessels since the instances *R_8_1*, *R_8_2*, and *R_8_8* could not be solved to optimality within 6 hours by the DA_{SRP+WS}.

Table 7 displays the results obtained with RHH when solving the instances with a number of vessels ranging from 9 to 15. The meaning of each column is the same as in Table 4.

Table 7: Computational results obtained by RHH when solving large size heterogeneous instances

N	Cost	T_{Total}	$\#it$	S_{Total}	S_S	S_R
9	37	288	2.9	471	255 (59%)	128 (25%)
10	39	177	3.6	746	547 (76%)	146 (17%)
11	45	583	4.4	1083	763 (74%)	239 (21%)
12	50	705	5.0	1509	1029 (63%)	358 (27%)
13	51	725	3.8	1325	836 (81%)	297 (13%)
14	62	3152	4.0	1525	1021 (69%)	223 (13%)
15	55	4071	6.1	3177	2113 (61%)	533 (19%)

Instances *R_14_9*, *R_14_10*, and *R_15_1* could not be solved by RHH in less than 6 hours, therefore these three instances are not included in Table 7. To obtain robust solutions for such instances, we imposed a time limit of 2 minutes when solving each rolling horizon subproblem. The results for these three instances are reported separately in Table 8.

Table 8: Computational results obtained by RHH for large size instances

Instance	Cost	T_{Total}	$\#it$	S_{Total}	S_S	S_R
R_14_9	61	1959	5	1538	486 (32%)	914 (59%)
R_14_10	64	660	4	1228	1191 (97%)	32 (3%)
R_15_1	62	2899	3	1579	970 (61%)	423 (27%)

Table 7 shows that the average number of scenarios evaluated per instance by RHH is large. Around 88% of the scenarios are eliminated by applying both static and random SEH variants. Table 8 shows that most difficult instances can be heuristically solved in a reasonable time by imposing a time limit to each rolling horizon subproblem. This observation allows to use the RHH for solving instances with more than 15 vessels and 60 time periods.

8.4. Results for BACASP instances of the literature

In this section, we test the proposed RHH and the DA enhanced with both the SRP and WS on the instances used in [4]. It is worth reminding that some of those instances are classified as difficult instances even under deterministic assumptions. The set of instances is composed of 9 instances. All instances identified with index a (I_{7a}, I_{8a}, I_{10a}, I_{12a}, and I_{15a}) are real-world instances, while the instances identified with index b (I_{8b}, I_{10b}, I_{12b}, and I_{15b}) are artificial instances derived from the instances with index a by enforcing the first 4 vessels to arrive at the same time.

The obtained results are displayed in Table 9. The first column identifies the instances. The number of vessels and time periods are given in the second and third columns. Each time the master problem in the DA is solved to optimality, the resulting worst completion time is a lower bound for the optimal value of the robust problem. In column LB we report the (best) obtained lower bounds. We set a time limit of 6 hours (21600 seconds) to solve the DA. It is worth mentioning that when the DA ends before the time limit, the reported lower bound coincides with the value of the optimal solution (all these cases are identified with an asterisk). Columns T_{Total} and $\#it$ report the total computational time required for solving either the DA or the RHH and the total number of iterations performed. Column $Cost$ reports the cost of the robust solution obtained by RHH and column $Gap(\%)$ shows the gap between the cost of the solution obtained by the RHH with respect to the best lower bound obtained by the DA. Since the best lower bounds can be much lower than the optimal solution, the reported gaps can also be interpreted as maximum gaps of the RHH solution in relation to the optimal robust solution value. Instance I_{15b} could not be solved within 6 hours neither by the DA_{SRP+WS} nor by the RHH. Therefore, to obtain a robust solution for this instance using RHH, we imposed a time limit of 2 minutes when solving each rolling horizon subproblem.

Table 9: Computational results for instances of the literature

			DA_{SRP+WS}			RHH			
Inst.	N	M	LB	T_{Total}	$\#it$	$Cost$	T_{Total}	$\#it$	$Gap(\%)$
I_{7a}	7	50	27*	35	2	27	29	2	0.0
I_{8a}	8	55	41*	700	4	41	31	3	0.0
I_{8b}	8	55	48*	6609	2	48	59	1	0.0
I_{10a}	10	60	56*	14235	6	56	1116	7	0.0
I_{10b}	10	60	63	>21600	>2	64	916	3	1.6
I_{12a}	12	65	63*	14629	3	63	677	1	0.0
I_{12b}	12	65	73	>21600	>1	75	7807	4	2.7
I_{15a}	15	65	72	>21600	>3	76	1196	4	5.3
I_{15b}	15	65	72	>21600	>2	104	3669	4	30.8

Table 9 shows that only 5 of the 9 instances were solved to optimality by the DA within the imposed time limit. The total time required by the RHH is much lower than the one required by the DA and RHH was able to obtain optimal solutions for all the 5 instances solved to optimality by the DA. For the 3 instances I_{10b}, I_{12b} and I_{15a} not solved to optimality by the DA, the gap associated with the RHH is still low (lower than 5.3%). The gap associated with instance I_{15b} is high for two reasons: i) the DA_{SRP+WS} was only able to perform 2 iterations during the time limit of 6 hours, which indicates that the reported lower bound can be a poor bound, and ii) the first rolling horizon subproblem corresponds to the first 3 vessels that arrive at the same time, which makes such subproblem difficult to solve. It is important to note that this is an artificial instance (still difficult to solve in the deterministic case) that does not reflect a real-world situation.

8.5. Performance of the robust solutions

In this section, we evaluate the performance of the proposed robust algorithm on the real-world instances I_{7a}, I_{8a}, I_{10a}, I_{12a} and I_{15a} based on the following two indicators:

Deterministic solution. The solution of the deterministic model (where the uncertain arrival times are replaced by their expected values) is frequently used in literature to evaluate the impact of disregarding the uncertainty in the solution procedure, as well as the benefits of using robust solutions. To compute this indicator, we start by solving the deterministic problem. Then, given a set of NS scenarios, we fix the berthing schedule of the vessels and compute the total completion time of that schedule for each scenario independently. This procedure generates a sample of NS values of completion times. In Table 10, we report the average and the maximum of the obtained values (in columns Det).

Perfect information. This is an indicator frequently used in stochastic optimization that we adapt to evaluate the obtained robust solutions. Computing this indicator implies to solve several deterministic problems (one for each scenario) independently and obtain the total completion time for each scenario. This procedure generates a sample of NS values of completion times. In Table 10, we report the average and the maximum of the obtained values (in columns PI).

After obtaining the robust solution generated for the decomposition algorithm for each instance, we fix the berthing schedule of the vessels and compute the total completion time of the schedule for each scenario independently. This procedure generates a sample of NS values of completion times. In Table 10, we report the average and the maximum of the obtained values (in columns *Robust*).

Table 10: Performance of the robust solution analysis

		Average			Worst-scenario		
Inst.	NS	PI	*Robust*	*Det*	PI	*Robust*	*Det*
I_{7a}	175	26.1	26.1	26.5	27	27	27
I_{8a}	245	39.7	39.8	39.7	41	41	41
I_{10a}	441	53.1	53.2	54.3	56	56	60
I_{12a}	729	60.6	60.7	61.8	63	63	69
I_{15a}	1331	73.5	73.7	74.4	75	76	78

Table 10 provides very interesting insights about the obtained robust solutions as well as about the strategy of optimizing the worst-case scenario. First, from the comparison between columns *Det* and *Robust*, it becomes clear that the robust solutions outperform the deterministic solutions not only in terms of the worst-case scenario, but also in terms of the average completion time. Second, the comparison between columns PI and *Robust* shows that our approach (that directly optimizes over the worst-case scenario) also keeps the expectation of the completion times (usually optimized by stochastic programming approaches) under control. This observation clearly justifies the importance of optimizing over the worst-case scenario. Note that the PI indicator corresponds to the not realistic case where the values of the arrival times are exactly known and therefore, the results associated with PI are the best ones that can be achieved. The average results associated with our robust solutions are really close to the ones associated with PI and the worst-case value is almost always the same (except for instance I_{15a}). This shows the good performance of the robust approach.

8.6. Results for large size instances

In this section, we use the proposed decomposition algorithm with both the warm start technique and the SRP (DA_{WS+SRP} approach) combined with the rolling horizon heuristic for solving large size instances. Following the procedure described in section 8.1, we randomly generate a new set of instances (with homogeneous cranes) composed of three subsets of instances with 40, 50 and 60 vessels, respectively. Each subset has 5 different instances and all of them have 168 time periods (corresponding to the number of weekly hours).

For the instances considered in the previous sections (with up to 15 vessels), the proposed uncertainty set divides the set of vessels into $ng = 3$ groups (according to their expected arrival times) and at most one vessel of each group can arrive with a delay of either one or two time periods. However, grouping the vessels into 3 groups is not suitable for instances with a large number of vessels. For example, for instances with 60 vessels, each group has 20 vessels, meaning that at most one of those 20 vessels can arrive later. Therefore, for large size instances, the parameter ng used in the uncertainty set has to be increased. The uncertainty set used for the instances considered in this section has the following parameters: i) the number ng of groups is set to $\left[\frac{3\times N}{10}\right]$, where $[\cdot]$ denotes the round operator; ii) the maximum allowed delay of a vessel $k \in V$, $\hat{A}_k$, is two time periods (as before); and iii) the maximum number of vessels that can be delayed in each group, $\Gamma_j,\ j = 0, \ldots, ng-1$, is one (as before). This uncertainty set configuration implies that each group of vessels is composed of either three or four vessels and at most one of those vessels is allowed to arrive later with a maximum delay of two time periods.

For instances with a number of vessels greater than or equal to 40, the total number of scenarios defined by the uncertainty set is extremely large (up to 3×10^{10}) making impossible to solve such instances considering all vessels in the same model. Hence, we solve such instances by considering several subproblems with a smaller number of vessels. The illustration of the procedure is shown in Figure 4.

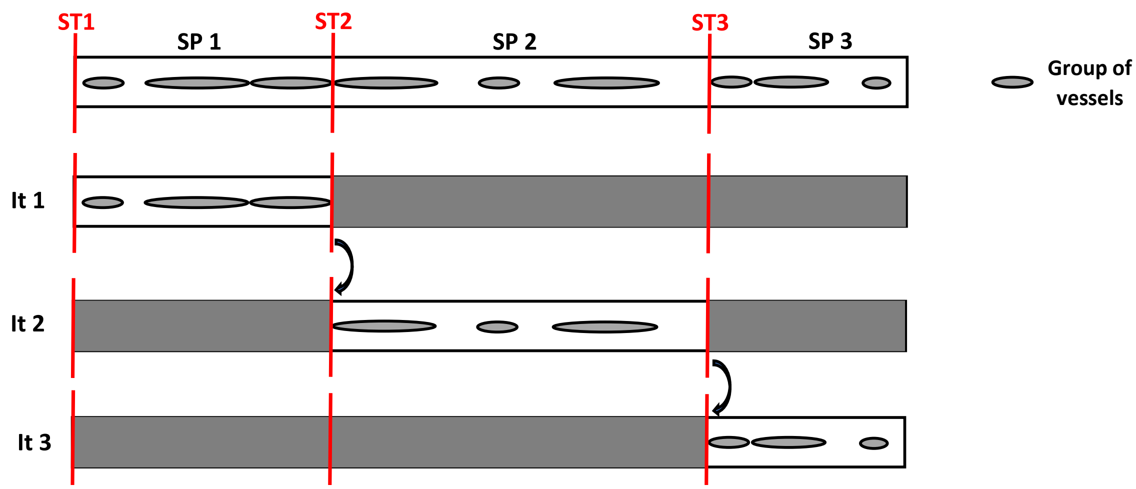

Figure 4: Procedure for solving large size instances.

We start by grouping the vessels according to the groups defined by the uncertainty set. Each circular mark represents a group of vessels, and its size is related to the expected arrival times of the vessels in that group. The first and the last extremities of each mark represent the expected arrival time of the first and of the last vessels of the group, respectively. At each iteration, a subproblem is solved considering only 3 groups of vessels. In the first iteration, the subproblem to solve considers only the first three groups of vessels and the time horizon is defined by the expected arrival time of the last vessel considered plus the maximum allowed delay plus the minimum handling time of that vessel plus a given tolerance (15 time periods in our experiments). After solving the first subproblem, by using the RHH approach, the worst-case scenario is found and the departure time of the vessels is computed.

In the second iteration, the second set of three groups of vessels is considered. The start time for the corresponding problem (ST_2) is the expected arrival time of the first vessels considered. However, there may exist vessels associated with the previous iteration having a departure time in the solution (obtained in the first iteration) greater than ST_2. Such vessels must be also considered in the second iteration as follows:

- If in the solution of the first iteration a vessel started to be handled before time ST_2, then its cargo at the beginning of the second iteration is the difference between the initial cargo quantity and the volume of cargo handled until period ST_2-1. The berth position of the vessel is fixed to the berth position determined in the solution of the first iteration and the start time of the handling operations in that vessel is fixed to ST_2.

- If in the solution of the first iteration a vessel started to be operated after time ST_2, then its cargo at the beginning of the second iteration is the initial cargo. The berth position of the vessel is not fixed and the start time of the handling operations in that vessel is fixed to the one determined at the first iteration.

The same process is repeated for the next iterations until the end of the time horizon and the resulting subproblems are always solved by using the RHH approach. This procedure reflects several practical situations at ports since not all the arrival times of the vessels are known at the beginning of the planning horizon. Such arrival times are usually successively known over the time.

The obtained results for the large size instances are displayed in Table 11. The number of vessels considered is shown in the first column. The average cost of the obtained solution is given in the second column. Columns T_{MP} and T_{AP} report the average computation time spent when solving all the master problems and all the adversarial problems, respectively. Column T_{Total} shows the average total computational time, while the last column reports the average number of iterations.

Table 11: Computational results for large size instances with 168 time periods

N	Cost	T_{MP}	T_{AP}	T_{Total}	$\#it$
40	181	916	1689	2605	17
50	227	1735	1582	3317	21
60	238	1821	2726	4581	27

The results presented in Table 11 shows that the proposed approach is scalable for large size instances since the average computational times are lower than two hours. Furthermore, these results also reinforce the difficulty of solving each adversarial problem since a large portion of the total time is devoted to that task.

9. Conclusion

In this paper, we study the robust BACASP assuming that the vessel arrival times are uncertain and belong to a multiple constrained budget uncertainty set, inspired by the original budget uncertainty set proposed by Bertsimas and Sim [6]. A two-stage mathematical model is proposed, where the berth decisions are first-stage decisions and the crane assignments are second-stage decisions. Such a model aims to minimize the worst sum of the completion times of all vessels taking into account all possible realizations of the arrival times in the uncertainty set.

Since a large number of scenarios is considered, an exact decomposition algorithm that splits the original problem into a master problem and a separation problem (adversarial problem) is proposed to obtain an optimal robust solution. Taking into account that solving the adversarial problem requires to solve a hard MIP subproblem for each scenario of uncertainty, we propose a new scenario reduction procedure that discards irrelevant scenarios, and therefore reduces the number of MIP subproblems to solve as well as the time associated with the adversarial problem. A warm start technique, where a carefully chosen scenario is used to initialize the decomposition algorithm, is also proposed to reduce the number of global iterations performed by the decomposition algorithm. Large size instances are solved heuristically by the decomposition algorithm combined with a rolling horizon heuristic.

Computational experiments are conducted on both randomly generated instances and instances from the literature. The results for the randomly generated instances are obtained considering both homogeneous and heterogeneous cranes. The obtained results reveal that the instances become more difficult to solve when heterogeneous cranes are considered. However, in both homogeneous and heterogeneous cases, the conclusions are similar. First, the proposed scenario reduction procedure drastically reduces the computational time required for solving the adversarial problem, since around 93% of the scenarios are discarded. Second, the warm start technique reduces the number of iterations performed by the decomposition algorithm and therefore reduces the global computational time. These two improvements together reduce the total running time of the decomposition algorithm by around 78%. Third, the rolling horizon heuristic allows to solve large size instances in a short time and the gap associated with the obtained solutions, with respect to the optimal solution value, is very low. Finally, the proposed methods are scalable for large size instances since we are solving instances with up to 60 vessels and 168 time periods in less than two hours.

Acknowledgements

The research was partially supported by the Center for Research and Development in Mathematics and Applications (CIDMA) through the Portuguese Foundation for Science and Technology (FCT), references UIDB/04106/2020 and UIDP/04106/2020. The research of the first author was also partially supported by the Project CEMAPRE/REM - UIDB/05069/2020 - financed by FCT/MCTES through national funds.

References

[1] A. Agra, M. Christiansen, R. Figueiredo, L. M. Hvattum, M. Poss, and C. Requejo. The robust vehicle routing problem with time windows. *Computers & Operations Research*, 40(3):856–866, 2013.

[2] A. Agra, M. Christiansen, L. M. Hvattum, and F. Rodrigues. A MIP based local search heuristic for a stochastic maritime inventory routing problem. In A. Paias, M. Ruthmair, and S. Voß, editors, *Lecture Notes in Computer Science, Computational Logistics*, volume 9855, (Springer International Publishing), pages 18–34, 2016.

[3] A. Agra, M. Christiansen, L. M. Hvattum, and F. Rodrigues. Robust optimization for a maritime inventory routing problem. *Transportation Science*, 52(3):509–525, 2018.

[4] A. Agra and M. Oliveira. Mip approaches for the integrated berth allocation and quay crane assignment and scheduling problem. *European Journal of Operational Research*, 264(1):138–148, 2018.

[5] A. Ben-Tal, L. E. Ghaoui, and A. Nemirovski. *Robust optimization.* Princeton Series in Applied Mathematics. Princeton University Press, 2009.

[6] D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52:35–53, 2004.

[7] D. Bienstock and N. Özbay. Computing robust basestock levels. *Discrete Optimization*, 5(2):389–414, 2008.

[8] C. Bierwirth and F. Meisel. A follow-up survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, 224(3):675–689, 2015.

[9] Y. Du, Y. Xu, and Q. Chen. A feedback procedure for robust berth allocation with stochastic vessel delays. In *2010 8th World Congress on Intelligent Control and Automation*, pages 2210–2215, 2010.

[10] G. Giallombardo, L. Moccia, M. Salani, and I. Vacca. Modeling and solving the tactical berth allocation problem. *Transportation Research Part B: Methodological*, 44(2):232–245, 2010.

[11] M. Golias, I. Portal, D. Konur, E. Kaisar, and G. Kolomvos. Robust berth scheduling at marine container terminals via hierarchical optimization. *Computers & Operations Research*, 41:412–422, 2014.

[12] X. Han, Z. Lu, and L. Xi. A proactive approach for simultaneous berth and quay crane scheduling problem with stochastic arrival and handling time. *European Journal of Operational Research*, 207(3):1327–1340, 2010.

[13] G. A. Hanasusanto, D. Kuhn, and W. Wiesemann. K-adaptability in two-stage robust binary programming. *Operations Research*, 63(4):877–891, 2015.

[14] Y. He, A. Wang, and H. Su. The impact of incomplete vessel arrival information on container stacking. *International Journal of Production Research*, 57:1–15, 2019.

[15] M. Hendriks, M. Laumanns, E. Lefeber, and J. T. Udding. Robust cyclic berth planning of container vessels. *OR Spectrum*, 32(3):501–517, 2010.

[16] Ç. Iris and J. S. L. Lam. Recoverable robustness in weekly berth and quay crane planning. *Transportation Research Part B: Methodological*, 122:365–389, 2019.

[17] C. Liu, X. Xiang, C. Zhang, and L. Zheng. A decision model for berth allocation under uncertainty considering service level using an adaptive differential evolution algorithm. *Asia-Pacific Journal of Operational Research*, 33(6):615–627, 2016.

[18] C. Liu, X. Xiang, and L. Zheng. Two decision models for berth allocation problem under uncertainty considering service level. *Flexible Services and Manufacturing Journal*, 29:312–344, 2017.

[19] C. Liu, X. Xiang, and L. Zheng. A two-stage robust optimization approach for the berth allocation problem under uncertainty. *Flexible Services and Manufacturing Journal*, 32(2):425–452, 2020.

[20] R. Moorthy and C. Teo. Berth management in container terminal: the template design problem. *OR Spectrum*, 28:495–518, 2006.

[21] K. G. Murty, J. Liu, Y. Wan, and R. Linn. A decision support system for operations in a container terminal. *Decision Support Systems*, 39(3):309–332, 2005.

[22] F. Rodrigues, A. Agra, M. Christiansen, L. M. Hvattum, and C. Requejo. Comparing techniques for modelling uncertainty in a maritime inventory routing problem. *European Journal of Operational Research*, 277(3):831–845, 2019.

[23] F. Rodrigues, A. Agra, C. Requejo, and E. Delage. Lagrangian duality for robust problems with decomposable functions: the case of a robust inventory problem. *INFORMS Journal on Computing*, 2020.

[24] M. Rodriguez-Molins, L. Ingolotti, F. Barber, M. A. Salido, M. R. Sierra, and J. Puente. A genetic algorithm for robust berth allocation and quay crane assignment. *Progress in Artificial Intelligence*, 2:177–192, 2014.

[25] X. Schepler, S. Balev, S. Michel, and E. Sanlaville. Global planning in a multi-terminal and multi-modal maritime container port. *Transportation Research Part E: Logistics and Transportation Review*, 100:38 – 62, 2017.

[26] X. T. Shang, J. X. Cao, and J. Ren. A robust optimization approach to the integrated berth allocation and quay crane assignment problem. *Transportation Research Part E: Logistics and Transportation Review*, 94:44 – 65, 2016.

[27] A. Sheikholeslami and R. Ilati. A sample average approximation approach to the berth allocation problem with uncertain tides. *Engineering Optimization*, 50(10):1772–1788, 2018.

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

[29] UNCTAD. Review of maritime transport, 2019. Technical report, United Nations, New York and Geneva, 2019.

[30] E. Ursavas and S. X. Zhu. Optimal policies for the berth allocation problem under stochastic nature. *European Journal of Operational Research*, 255(2):380 – 387, 2016.

[31] X. Xiang, C. Liu, and L. Miao. Reactive strategy for discrete berth allocation and quay crane assignment problems under uncertainty. *Computers & Industrial Engineering*, 126:196–216, 2018.

[32] Y. Xu, Q. Chen, and X. Quan. Robust berth scheduling with uncertain vessel delay and handling time. *Annals of Operations Research*, 192:123–140, 2012.

[33] B. Zeng and L. Zhao. Solving two-stage robust optimization problems using a column-and-constraint generation method. *Operations Research Letters*, 41(5):457–461, 2013.

[34] L. Zhen. Tactical berth allocation under uncertainty. *European Journal of Operational Research*, 247(3):928–944, 2015.

[35] L. Zhen and D.-F. Chang. A bi-objective model for robust berth allocation scheduling. *Computers & Industrial Engineering*, 63(1):262–273, 2012.

[36] L. Zhen, L. H. Lee, and E. P. Chew. A decision model for berth allocation under uncertainty. *European Journal of Operational Research*, 212(1):54–68, 2011.

[37] P. Zhou and H. Kang. Study on berth and quay-crane allocation under stochastic environments in container terminal. *Systems Engineering - Theory & Practice*, 28(1):161–169, 2008.