Engineering Applications of Artificial Intelligence 103 (2021) 104276

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

A simulation–optimization framework for enhancing robustness in bulk berth scheduling

Alan Dávila de León [a], Eduardo Lalla-Ruiz [b,*], Belén Melián-Batista [a], J. Marcos Moreno-Vega [a]

[a] Department of Computer Engineering and Systems, University of La Laguna, Spain
[b] Department of Industrial Engineering and Business Information Systems, University of Twente, Netherlands

ARTICLE INFO

Keywords:
Maritime logistics
Berth Allocation Problem
Metaheuristics
Simulation–optimization
Buffer management
Simheuristics

ABSTRACT

The service time of the vessels is one of the main indicators of ports' competitiveness. This, together with the increasing volume of bulk transportation, make the efficient management of scarce resources such as berths a crucial option for enhancing the productivity of the overall terminal. In real scenarios, the information available to port operators may vary once the planning has been elaborated. Unforeseen events, errors, or modifications in the available information can lead to inefficient terminal management and the initial scheduling might become unfeasible. This implies that the use of deterministic approaches may not be enough to maximize productivity. Therefore, in this work, proactive simulation–optimization approaches that utilize the information collected during the simulation for guiding the optimization search to provide robust solutions are proposed. Moreover, a multi-objective approach based on the Non-Dominated Sorting Genetic Algorithm II (NSGA-II) for jointly tackling the problem objective as well as the deviations because of stochastic changes is developed. Finally, we also investigate the contribution of time management strategies such as buffers to absorb stochastic modifications and hence increase solutions' robustness. The computational results indicate, on the one hand, the benefit of integrating both types of objectives (i.e., deterministic and stochastic) to guide the simulation–optimization process, and on the other hand, the benefit of using the multi-objective approaches like NSGA-II. Finally, the incorporation of buffers leads to better performance in terms of reducing penalty costs due to disruptions, shortening the planning risks related to only considering deterministic planning.

1. Introduction

The international transport of goods plays an important role in the development and consolidation of important economic sectors such as trade, industry or tourism. Their competitiveness and efficiency are, therefore, essential for the economic and social progress of regions. A port can be represented as a system, which we will refer to as a port system, formed by subsystems or interrelated elements. The nature and granularity of the representation are closely linked to the level of detail required and to the operational functioning of the port. In a port, three clearly defined zones are identified which are related in a complex way to each other: seaside, yard, and landside.

The Berth Allocation Problem (BAP) considers a quay divided into sections or berths and a group of vessels that must be served within a well-defined planning horizon. In its simplest version, this problem aims to assign berthing position and berthing time to each vessel, trying to optimize a given objective function. Due to its great impact on the rest of the problems involved in the maritime supply chain, the BAP has been widely studied in the literature. In this paper, we focus on a variant of the BAP, the Bulk Berth Allocation Problem (Bulk-BAP, Umang et al., 2013) that takes place in bulk ports. That problem variant considers that each vessel arrives in a given instant of time, so it cannot berth before that time. However, in order to increase the reliability of the port system, in this paper, the arrival times are considered stochastic parameters (Guldogan et al., 2011; Umang et al., 2017) and handling times are subject to measured or continuous random distributions (Ganji et al., 2010; Na and Zhihong, 2009).

In real environments, port terminal operators must deal with scenarios where unforeseen circumstances and uncertain factors might occur (Notteboom, 2006; Vernimmen et al., 2007). In the seaside subsystem, vessels generally operate on closed-circuit routes following a pre-established schedule. In practice, when a vessel travels on its shipping route, several uncertainty factors (port congestion, adverse weather conditions, etc.) may happen occasioning planning delays, even when some type of uncertainties have been already considered when planning that route. Furthermore, handling times are also affected by uncertainty due to diverse factors involved such as the

* Corresponding author.
E-mail addresses: adavilal@ull.es (A.D. de León), e.a.lalla@utwente.nl (E. Lalla-Ruiz), mbmelian@ull.es (B. Melián-Batista), jmmoreno@ull.es (J.M. Moreno-Vega).

https://doi.org/10.1016/j.engappai.2021.104276
Received 20 May 2020; Received in revised form 21 April 2021; Accepted 23 April 2021
Available online 18 May 2021
0952-1976/

quantity of cargo, mechanical problems, technical inspections, among others. These uncertainties cause a progressive deterioration in the current planning and may lead to unfeasible scheduled operations. The consequences of not considering uncertainties range from a decrease of terminal's productivity in the short term to losing planned calls in the long run. Furthermore, some researchers have shown that the events produced in some subsystems that define the port system end up affecting the whole system (Lalla-Ruiz, 2017). In this sense, any unforeseen event can have an impact on the overall performance of the port system and may also affect other systems with which the port interacts (for example, the nearby road network). Finally, several authors have highlighted the importance of considering the penalties and consequences associated with the deviations produced in the baseline scheduling (Park and Kim, 2005; Imai et al., 2007; Chang et al., 2010). The economic consequences imposed on the terminal by the ocean carriers because of service delays (Theofanis et al., 2009), e.g., end of cargo processing and departure of the docked ship, leads to the necessity of minimizing the deviation between planned and realized operations. Thus, a risk analysis is required in order to implement actions to absorb or reduce the effect of harmful results caused by disruptive events.

In the related literature, there are two major proposals to minimize the impact of disruptive events on port system performance: (*i*) proactive and (*ii*) reactive. In purely proactive approaches, a baseline schedule is defined according to the information available before the berthing and processing of the vessels, and it is not possible to adapt it based on the new information available. Therefore, the scheduling gained from this type of approach should lead to reducing the deviation between the baseline schedule and the current schedule. To minimize the impact of disruptive events, the uncertainty of the problem being addressed must be analysed, thereby leading to fault-tolerant baseline scheduling. For this reason, historical data is needed to define protection and prevention plans capable of managing existing risks and improving the robustness of the system. This data can be used to extract probability distributions and measure the level of robustness of the solutions. On the other hand, reactive approaches aim at bouncing back disruptions once they happen. They involve decisions aimed at minimizing the impact of disruptive events happening in real-time. This means that decisions adapted to current information can be taken. These actions might modify the baseline schedule to maximize the efficiency of the system with the new information available. Unlike proactive approaches, a purely reactive approach generates solutions without taking into account robustness metrics. Thus, schedules created from a purely reactive approach are less suitable for absorbing the impact of disruptive events than proactive ones; so it is necessary to put in place recovery strategies with the consequent additional cost associated (Theofanis et al., 2009; Chang et al., 2010).

Different proactive approaches have been proposed in different berth allocation problems as further discussed in the literature review section. The majority of them address either stochastic vessel arrival times or stochastic handling times. As solution strategies, few authors (Du et al., 2010; Xu et al., 2012a; Iris and Lam, 2019) focus on incorporating buffer times between vessels to minimize the negative impacts of the disruptive events, while others authors (Zhen and Chang, 2012; Rodriguez-Molins et al., 2014) try to maximize the intrinsic vessel-specific buffer times (see Section 2 for further details). With regards to the Bulk-BAP, to the best authors' knowledge, no proactive approach has been proposed for this problem. The unique approach coping with uncertainties in Bulk terminals is the one by Umang et al. (2017) that presents a reactive approach for the Bulk-BAP under uncertainty in handling times and arrival times.

Considering the above discussion, in this work, a proactive approach for the Bulk-BAP considering disturbances on planned operations due to vessels' information change (i.e., arrival of vessels and handling times) is developed. This way the impact of uncertainty can be minimized, providing a robust starting point for reactive approaches like the one presented in Umang et al. (2017). With this aim, a schedule robustness measure is introduced to evaluate the risk of solutions by simulation. The integration of simulation in the search process allows the use of robustness measures to evaluate solutions under stochastic circumstances as well as feedback to the search process. Therefore, it is necessary to use multi-objective algorithmic approaches that provide high quality in terms of the objective function value solutions with low risk. With this objective, simulation-based Non-dominated Sorting Genetic Algorithm II (NSGA-II, Deb et al., 2002) is proposed to generate a robust berth schedule proactively for the Bulk Berth Allocation Problem. In addition, dynamic vessel-specific buffer times are used to improve robustness.

The contribution of this paper is threefold:

- Proactive simulation–optimization approaches are considered to generate robust baseline schedules for the Bulk-BAP by simulating disruptive events related to vessels' arrival and handling times. Among them, a novel simheuristic framework that considers the information collected during the simulation phase within the metaheuristic search process is proposed and assessed.
- A multi-objective approach is proposed to address the Bulk-BAP by jointly considering its objective as well as the solution robustness with regards to uncertain conditions. In doing so, the first NSGA-II simheuristic approach for solving an optimization problem is developed.
- Vessel-specific buffer times are proposed to absorb the impact of disruptive events and improve the robustness of baseline schedules. A data-driven heuristic approach is proposed to reliably and efficiently determine them by taking into account uncertainty in the arrival and handling times. In addition, different dynamic and static strategies are proposed to manage these during the search process.

The organization of the paper is as follows. Section 2 presents a comprehensive literature review. Section 3 describes the Bulk Berth Allocation Problem. Section 4 describes the main features of simheuristics and the algorithms used to tackle this problem. The results obtained by the proposed solution approaches are discussed in Section 5. Finally, Section 6 presents the conclusions together with future research lines.

2. Literature review

In the related literature, the Berth Allocation Problem (BAP) consists of assigning positions and berthing times to arriving vessels at the port with the aim of optimizing a given objective function. It has been widely studied in the literature (Bierwirth and Meisel, 2015) presenting different variants according to spatial and time constraints. Within the spatial restrictions, the quay can be considered *(i)* discrete: the quay is divided into equal sections called berths (Hansen et al., 2008; Cordeau et al., 2005; Monaco and Sammarra, 2007; Lalla-Ruiz et al., 2012); *(ii)* continuous: the quay is treated as a continuous section allowing vessels to berth at any point within it (Lee et al., 2010; Imai et al., 2005); and *(iii)* hybrid: the quay is divided into sections enabling a vessel to occupy more than one section (Cordeau et al., 2005; Umang et al., 2013). Moreover, concerning the temporal restrictions, the BAP can be treated as *(iv)* static: all the vessels are at the terminal when the planning is going to be conducted (Imai et al., 1997); *(v)* dynamic: the vessels arrive along the planning horizon (Imai et al., 2001; Cordeau et al., 2005); and *(vi)* time-dependent: the availability of the berths changes along the time horizon (Xu et al., 2012b; Lalla-Ruiz et al., 2016). Furthermore, in the BAP the arrival time has also been considered as a decision variable to maximize the reliability of the schedule minimizing vessel delayed departures (Golias et al., 2009).

Among the previously cited papers, Umang et al. (2013) address the Berth Allocation Problem in bulk ports (Bulk-BAP) which considers a hybrid quay layout that divides the quay into sections and where each section can only be occupied by a vessel at each instant of time, but a vessel can occupy more than one section. Regarding temporal constraints, the Bulk-BAP is included in the dynamic variant category. The BAP in bulk ports is also studied in Ribeiro et al. (2016). The authors propose a MILP model considering maintenance, demurrage, and dispatch values for handling the vessels and an ALNS which yields good solutions on a set of instances based on real data. The research work (Ernst et al., 2017) addresses the continuous BAP in order to minimize delays in a bulk terminal taking into account tidal constraints. Two MILP models are proposed, one based on the sequence variables, the other based on time-indexed variables together with a two-phase method in order to enhance the performance of this model. Finally, in Pratap et al. (2017) a framework for integrating berth allocation and vessel un-loader allocation is provided. Two different approaches are proposed, one solves the problems by solving them sequentially and the other solves both problems simultaneously. Chemical Reaction Optimization is proposed to solve the second phase of a sequential approach and a Genetic Algorithm is used to solve the first phase of the sequential approach and the integrated approach.

The previous approaches address the problem from a deterministic point of view. Nevertheless, handling and arrival times might vary and, hence, can be considered stochastic to minimize the total waiting time of calling vessels (Zhou and Kang, 2008) or to maximize berth efficiency and maintain the integrity of the schedule (Golias, 2011). In this sense, any unforeseen event or change significantly alters the value of one or more parameters characterizing the port system, resulting in a change of scenario. Moorthy and Teo (2007) underline the need for a model that contemplates these stochastic circumstances to improve the performance of the terminal. Various approaches have been proposed in the literature to address uncertainty while vessels are being handled, thereby maximizing the effectiveness of the terminal (Schepler et al., 2019). Below is a literature review of proactive approaches to address the BAP in stochastic environments.

One of the first studies carried out in this area is proposed by Zhou and Kang (2008). The authors consider arrival and handling times as stochastic variables to improve the quality of service in the tactical BAP. The uncertainty in both variables is modelled through a normal distribution, where the median represents the expected value and the deviation is calculated according to historical data. Their objective is to minimize the average waiting time, for which they use a Genetic Algorithm. Zhou et al. propose a stochastic mathematical programming model instead of simulation to evaluate and optimize the allocation of resources.

Lu et al. (2010) also address the tactical BAP but minimizing the expected value plus the standard deviation of the total service time and the weighted waiting time of all vessels. Unlike other papers, it prioritizes vessels based on their relative importance. It uses normal distributions to model uncertainty for both arrival and handling times. In order to obtain robust solutions under conditions of uncertainty, they use an algorithm based on a Genetic Algorithm, complemented by a local search, the evaluation function of which contains a simulation phase to evaluate the solutions under stochastic conditions. To evaluate the performance and robustness of the proposed plans, the expected value and the standard deviation of the target value are calculated.

Du et al. (2010) aim at maximizing the robustness of BAP scheduling under conditions of uncertainty only considering vessel arrival times. An inconsistency cost is used as a measure of robustness, measuring the deviation produced in position and berthing times. At the same time, they seek to minimize the delay in the departure of vessels. Their proposal is to properly manage buffer times that absorb the uncertainty of arrival times, which results in more robust scheduling. Uncertainty follows a normal distribution, where the median corresponds to the expected time of arrival, while the standard deviation varies depending on the level of stochasticity sought. The proposed feedback procedure consists of three phases which are repeated iteratively until a certain stopping criterion is met: (*i*) the deterministic problem is solved by Simulated Annealing, (*ii*) modified scheduling is obtained using a reallocation rule for each of the delay scenarios, (*iii*) and a heuristic method which contains two adjustment rules is applied to adjust the buffer times for each of the vessels. The scheduling is iteratively adjusted according to the feedback details and robust scheduling is obtained. To simulate uncertainty, training and validation scenarios are generated for each instance.

Hendriks et al. (2010) address the cyclic berthing for a continuous and dynamic BAP and, unlike other papers, use time windows to model uncertainty in arrival times, instead of using expected arrival times. They propose a robust mixed-integer linear programme (MILP) model that minimizes the maximum capacity required of cranes. Scheduling is deemed robust if there is a feasible solution for each arrival scenario in which all vessels arrive within their time windows.

Zhen and Chang (2012) study the BAP under uncertainty in both arrival times and handling times. They use a normal distribution for the generation of uncertainty in both variables, where the median matches the expected value. Unlike other papers, the probability distribution does not apply to all vessels, but a selection is made of a subset of vessels that will undergo a deviation from their estimated arrival time. Another selection is also made of a subset of vessels that will experience a deviation from their estimated handling time. They propose a multi-objective optimization model to find a balance between robustness and cost. They also propose an SWO heuristic to address large-scale instances. Once the solutions are obtained, they are evaluated based on stochastic scenarios. The robustness of the solutions is measured by a weighted sum of the buffer times allocated to each vessel. Buffer times are intrinsic to the problem, i.e., no extra time is added between vessels, but rather the goal is to maximize the time between the time of departure of one vessel and the berthing of the next one.

Xu et al. (2012a) address the Robust Berth Allocation Problem (RBAP) under uncertainty in handling and arrival times. Their objective is to minimize the total departure delay time of vessels minus the robustness measure (i.e. the length of buffer time). As Du et al. (2010) and Zhen and Chang (2012), buffer times are used to providing a more robust baseline schedule, however, they are non-vessel-specific. Propose heuristic method combining Simulated Annealing and branch-and-bound algorithm, using simulation (random scenarios) to determine the robustness and service level of baseline schedule in an uncertain environment.

Karafa et al. (2013) consider stochastic handling times for the BAP, modelling uncertainty through normal distribution. They formulate the BAP as a bi-objective optimization problem, where the idea is to minimize the total service time of the vessels and the robustness associated with the solution. This robustness is modelled as the variability of the start and end time of handling. They use a heuristic based on an evolutionary algorithm together with a simulation-based Pareto pruning algorithm to solve the problem. The Genetic Algorithm generates a set of non-dominated solutions, which are evaluated by a Monte Carlo simulation. The one that minimizes the total service time on average is then selected.

Golias et al. (2014) address the discrete berth scheduling problem by modelling uncertainty in arrival and handling times through a triangular distribution. The goal is to simultaneously minimize the average and range of the total service time required to serve all the vessels at the terminals. The second objective formulation is also used as a measure of robustness of the scheduling. Because two objectives are used, the problem is formulated as a bi-objective bi-level optimization problem. They solve the problem by using a Genetic Algorithm whose solutions are evaluated through the use of scenarios.

Rodriguez-Molins et al. (2014) address the tactical BAP with uncertainty in handling times using a Genetic Algorithm. They use a multi-objective optimization approach where the goal is to minimize the total service time while maximizing the robustness of the solutions. For the search process to work correctly, they normalize and allocate a weighting to each of the objective functions, and thus work with a single objective function. To increase the robustness of the solutions, they seek to maximize the use of buffer times. Like in Zhen and Chang

(2012), they do not allocate buffer times between vessels, but the buffer times correspond to the time between the departure of one vessel and the start of handling the next one. To evaluate the robustness of the proposed solutions, they use scenarios with incidents in the handling time, modelled as randomly generated delays based on a certain range.

Zhen (2015) make two proposals based on the historical data available to obtain robust scheduling for the Tactical Berth Allocation Problem with uncertainty in handling times. The first approach is applied when there is no historical data on handling times, so the maximum and minimum values must be estimated to define uncertainty. If there is historical data, probability distributions are used. For simplicity purposes, for both proposals, they use a uniform distribution for handling times, where the lower and upper limits are generated randomly. To model uncertainty, only handling times are taken into account, although their model is prepared to withstand uncertainty in times of arrival. They also propose scalable metaheuristics to solve large-scale problems. The objective function sets out to minimize the total deviation between the allocated berthing time and the estimated one. They use scenarios to simulate uncertainty during the experiment.

Ursavas and Zhu (2016) propose a stochastic dynamic programming algorithm to obtain optimal policies for the BAP with stochastic arrival and handling times. They propose a framework based on stochastic dynamic programming approach to model the berth allocation problem and characterize optimal policies under stochastic arrival and handling times for different types of calling vessels. They categorize the vessels into deep-sea vessels and barge of feeders. The objective is to minimize the total expected cost by determining to which terminal the vessels should be allocated. Following case-based literature (Karafa et al., 2013; Rodriguez-Molins et al., 2014), they use the Poisson process to model uncertainty for arrival times and an exponential distribution for handling times.

Shang et al. (2016) address the BACAP by modelling uncertainty in handling times through a symmetric distribution. They propose a robust optimization model and another robust optimization model controlling the level of conservation through cost restrictions. The objective is to minimize the total weighted handling time and the waiting time of all vessels. In addition, they present a Genetic Algorithm and an insertion heuristic to deal with large-scale problems.

Finally, Yan et al. (2019) address the dynamic BAP with stochastic arrival times (SDFBAP) formulated as an integer multi-commodity network flow problem and propose a general resolution framework. The objective function minimizes penalties for not being able to service all vessels within the planning horizon and minimize the sum of the expected values of unanticipated schedule delay costs. To evaluate the solutions obtained under unforeseen circumstances, they propose the following simulation-based approach. In each iteration, the arrival times are generated randomly following a probability distribution, while according to a set of rules the necessary adjustments are made to the planning and the different effectiveness metrics are calculated, e.g., berthing time of a vessel minus its actual arrival time. Finally, the average of these metrics is calculated to evaluate the solution.

Related to the Bulk-BAP, recently, Umang et al. (2017) model uncertainty on arrival and handling times through probability distributions. With the objective of minimizing the cost of scheduling, they propose a reactive approach for the real-time rescheduling of vessels in case of disruptive events and new information. They develop two recovery algorithms to reschedule before the occurrence of disruptive events. To assess the robustness of the proposed solutions, each solution is subjected to one hundred disruptive scenarios.

Based on the literature review discussed above, the contributions of this paper are the following. First, the Bulk-BAP with uncertainty in handling and arrival times is addressed proactively. With this aim, simulation-based optimization approaches using metaheuristics (i.e., simheuristics, Juan et al., 2015) are proposed. Second, to jointly consider the deterministic and stochastic objectives, a multi-objective approach based on NSGA-II is investigated. Finally, the proposed algorithms are modified to dynamically manage specific buffer times per vessel to increase the robustness of the proposed solutions. Based on our knowledge, no paper in the current literature presents a multi-objective algorithm that integrates the dynamic management of buffer times and a simulation phase to guide the search and obtain a Pareto front based on the expected average objective function value and robustness of the solutions.

3. Bulk berth allocation problem

The Bulk Berth Allocation Problem (Bulk-BAP) is an NP-hard problem proposed by Umang et al. (2013) that seeks to determine the berthing position and berthing time of bulk vessels arriving at the port over a well-defined time horizon. It addresses the berthing operations at bulk terminals and mainly tackles the service operations involving two areas, i.e., the quay and the yard. In the Bulk-BAP, the consideration of the quay is an extension of the hybrid model (Umang et al., 2013; Robenek et al., 2014), in which the quay is divided into sections and a vessel can occupy more than one section, but a section can only be occupied by a vessel in each time instant. Each section is associated with a cargo type in function of the equipment facilities, i.e., pipeline, conveyor, and crane cargo. In the Bulk-BAP, we are given a set of incoming bulk vessels, N, a quay of length λ divided into a set of sections, M, and a set of cargo types, W. Each section, $k \in M$, has several features such as the type of cargo that can handle w_k, its length l_k, the starting coordinate of the section b_k, and the number of available cranes n_k^w. Similarly, each vessel $i \in N$ also has several features such as the estimated time of arrival ETA_i, the cargo type that carries W_i, the quantity of cargo Q_i, its length L_i, and draft D_i. The deterministic objective function of the Bulk-BAP (1) aims to minimize the total service time of all the vessels arriving at the port.

$$f(s) = \sum_{i \in N} (m_i - ETA_i + c_i) \tag{1}$$

In the expression (1), m_i represents the time instant in which the vessel starts to be served and c_i its corresponding service time. This objective function corresponds to the deterministic case. In the following section, it is further extended to consider stochastic conditions, namely, $h(s)$ (see Section 4.1.1).

3.1. Bulk berth allocation problem under stochastic conditions

From the literature review (see Section 2), it can be observed that the probability distributions that model uncertainty vary depending on the research work. For instance, Yu et al. (2018) indicate that only 20% of vessels arrive at their ETA, while the rest arrive before or after it. Based on the data extracted from 2066 container ship arrivals, they conclude that the deviation between the actual time of arrival (ATA) and the estimated time of arrival (ETA) follows a normal deviation with a median equal to 0.5 and a standard deviation equal to 4. However, they indicate that the probability distributions must be determined from the historical data of the port. Moreover, they remark that the efficient and reasonable use of buffer times is beneficial for terminal scheduling. Therefore, in our solution approach, the probability functions as described in Umang et al. (2017) are used. These probability distributions are based on observations made in the bulk port of Saqr, Ras Al Khaimah, UAE. They use a reactive approach to address uncertainty in a real-time environment. In contrast to that work, in our work a proactive approach is developed with the aim of generating robust baseline schedules that consider uncertainty conditions, e.g., variation of vessels' arrival and handling times).

Based on the data provided in Umang et al. (2017), the uncertainty associated with the estimated time of arrival ETA_i of a vessel $i \in N$ is modelled by a uniform probability distribution $[ETA_i - \delta, ETA_i + \delta]$. The δ parameter is used to control the level of uncertainty of the problem. The uncertainty associated with handling times is significantly less than the uncertainty associated with arrival times. For this reason, the handling time follows a truncated exponential distribution in the

$[H_{ik}, \gamma H_{ik}]$ interval. The lower bound H_{ik} and the upper bound γH_{ik} represent the minimum and maximum handling time of the vessel due to uncertainty, respectively. The expected handling time of vessel $i \in N$ berthed at position k is given by:

$$h'_{ik} = \frac{-\ln(\epsilon - \rho_h(\epsilon - \epsilon^{\gamma}))}{\tau} \quad (2)$$

$$\epsilon = e^{-\tau H_{ik}} \quad (3)$$

$$Prob(h_{ik} \leq h'_{ik}) = \rho_h \quad (4)$$

Both τ and p_h are constant, equal to 0.5 and 0.95, respectively. The γ parameter is used to control the level of uncertainty of the problem. For more details, see Umang et al. (2017) research work.

4. Solution approaches

Considering an environment with multiple sources of uncertainty, it is necessary to implement mechanisms to minimize the impact of possible events that might modify the initial or baseline schedule. To this end, proactive approaches are proposed to maximize port productivity by maximizing the robustness of the solutions provided. In this sense, simheuristic based algorithms are proposed as a general resolution framework to address real problems under uncertain conditions.

Taking the discussion above into account, when considering robustness as an additional optimization objective, a multi-objective approach can be adopted. In a multi-objective optimization problem, there is usually no global optimum which optimizes all objective functions, but rather there is a set of non-dominated solutions that make up a Pareto front. Therefore, in order to obtain a Pareto front that considers the objective of the problem at hand and the stochastic impact of the solutions, a multi-objective optimization algorithm based on the well-known Non-dominated Sorting Genetic Algorithm II (NSGA-II, Deb et al., 2002) is proposed in Section 4.3. In addition, Genetic Algorithm approaches to be used as a reference to analyse the contribution of NSGA-II are proposed in Section 4.2. It is worth pointing out that both approaches consider the information collected during the simulation for guiding the search process while in traditional simheuristics approaches this information is not considered in an integrative way.

4.1. Simheuristic

Simheuristics (Juan et al., 2015) provide a framework that enables addressing real problems with uncertain components by jointly using metaheuristics and simulation. This permits obtaining computational efficient solutions while considering their impact on stochastic scenarios. The simulation process considered in the framework allows to model and recreate complex stochastic systems. This way, by analysing the information provided by the simulation, it is possible to estimate the feasibility of the best solutions in stochastic scenarios. Based on that, we are able to select those solutions that maximize robustness. That is, the selected solutions are not minimizing the objective function value of the deterministic problem at hand, but the one that also meets certain criteria considering the uncertainty, e.g., maximizes the robustness of the port system.

4.1.1. Simulation

In the simheuristics template, a simulation process is incorporated for obtaining a metric that indicates the robustness of a solution generated by a metaheuristic. The simulation consists of an interactive process that is responsible for recreating the appearance of unexpected events and thus evaluating the behaviour of the solution. After the simulation, to quantify the risk related to the stochastic changes, a penalty cost that considers the total delay time in vessels' planned departure is determined. Hence, the minimization of this penalty cost is considered as the robustness objective of the problem. Considering the above, the lower the penalty cost (i.e., the risk), the higher the robustness.

The simheuristic process is as follows. First, a solution s is generated by a metaheuristic and provided as the input parameter for the simulation process. Then, for each simulation iteration $j < iterMax$, each stochastic component (e.g., handling and arrival times) is modified using a probability distribution that defines its uncertainty (see Section 3.1). That generates new scenario conditions for the solution s that leads to the disrupted solution s'_j. The modification of the stochastic components does not affect the deterministic problem because the latter is only used for obtaining the deterministic berth planning solution, s. For each solution s'_j, total delay time in vessel departure is determined using Eq. (5) and stored in the set Z as shown in (6). From this set, Eq. (7) is defined as the robustness-related optimization objective for the Bulk-BAP, that minimizes the average penalty cost associated with the total delay time in vessel departure, where Γ represents a cost of 800$ per unit of time (Chang et al., 2010). The median of the set of assessments is chosen since it includes 50% of the assessments under conditions of uncertainty while eliminating the influence of peaks. Notice that in the case of the simheuristics scheme proposed in this work, the simulation process is applied over each generated solution during the metaheuristic search process.

$$g(s, s') = \sum_{i \in N} (m_i^{s'} + c_i^{s'}) - (m_i^s + c_i^s) \quad (5)$$

$$Z = \{g(s, s'_1), g(s, s'_2), \ldots, g(s, s'_{iterMax})\} \quad (6)$$

$$h(s) = \Gamma \cdot median(Z) \quad (7)$$

4.1.2. General scheme of simheuristics

In Algorithm 1, the generic procedure defined by simheuristics is shown. The simheuristic receives as input parameter a stochastic problem SP. However, since metaheuristics have to be enabled to deal with this type of problems, it is necessary to transform it into a deterministic problem DP. In this work, the transformation process lies in replacing each stochastic parameter with its expected value (line 1). Then, a phase where alternating metaheuristics and simulation begins (line 4–9). First, the metaheuristic generates the solution s_i for the DP (line 5). After that, a fast simulation is carried out to determine the penalty cost of that solution s_i (line 6). This process between the metaheuristic and simulation process is repeated until a certain stopping criterion is reached (line 4). If the number of iterations is low, it is said to be a fast simulation, while if it is high it is considered an intensive simulation. A subset of elite solutions is selected (line 10) from the penalty cost ($h(s)$) provided by the fast simulation and the objective function value ($f(s)$). Lastly, an intensive simulation process is applied to them (line 11–12). The goal of this last intensive simulation is to obtain a more accurate measure of the penalty cost. Then, a subset of optimal solutions is obtained together with a risk analysis (line 14), which allows selecting the most robust baseline schedules.

4.2. Novel simheuristic framework

Our proposed simheuristic framework incorporates a feedback process between the simulation and metaheuristic phases, so the search process is now guided with the information obtained during the simulation. The used metaheuristic within the framework is based on a Genetic Algorithm (GA, Davis, 1991) in which the fitness function incorporates a fast simulation phase. As indicated earlier, this work also extends that GA framework to multiple objectives by proposing a multi-objective NSGA-II simheuristic approach.

Algorithm 1 Simheuristic process

Require: Bulk-BAP instance with stochastic components, SP
DP = transformIntoDeterminist(SP)
solutions = ∅
i = 0
while !stop criterion **do**
  s_i = resolveDeterministicProblem(DP)
  $h(s_i)$ = fastSimulation(s_i)
  solutions = solutions ∪ (s_i, $h(s_i)$)
  i = i + 1
end while
eliteSolutions = elitistSelection(solutions)
for s ∈ eliteSolutions **do**
  intensiveSimulation(s)
end for
reliabilityAnalysis(eliteSolutions)

Unlike Algorithm 1, the fast simulation phase is integrated during the search process of the GA in order to evaluate the robustness of the solutions, see Algorithm 2. This way, the fast simulation is applied to all the solutions in the population for evaluating their performance. The benefit of this is that the search process is not limited to either the deterministic or stochastic case, on the contrary, it can be guided by the deterministic and stochastic functions (i.e., $f(s)$ and $h(s)$).

Algorithm 2 GA integrated with simulation

Require: Bulk-BAP instance with stochastic components, SP
DP = transformIntoDeterminist(SP)
solutions = ∅
initialize population P
for $s \in P$ **do**
  evaluate(s)
  $h(s)$ = fastSimulation(s)
  solutions = solutions ∪ (s, $h(s)$)
end for
while !stop criterion **do**
  Q = selectionAndReproduction(P)
  Q = mutate(Q)
  for $s \in Q$ **do**
    evaluate(s)
    $h(s)$ = fastSimulation(s)
    solutions = solutions ∪ (s, $h(s)$)
  end for
  P = nextGeneration(P, Q)
end while
eliteSolutions = elitistSelection(solutions)
for s ∈ eliteSolutions **do**
  intensiveSimulation(s)
end for
reliabilityAnalysis(eliteSolutions)

4.2.1. Initialization

In order to generate the initial population, a Greedy Randomized Algorithm (GRA, Resende and Ribeiro, 2010) is developed. This algorithm uses a Restricted List of Candidates (RLC) that contains a subset of the best k candidates to add to the solution constructed so far. The candidates are the pairs vessel-section, whose heuristic evaluation corresponds to the sum of their waiting and handling times, that is, for a pair $i \in N$ and $k \in M$, $m_i - ETA_i + c_i^k$. Each vessel can be served in different sections so that at each iteration all possible mappings between each pair vessel-section are generated. It should be noted that only feasible allocations containing not pre-assigned vessels are considered. The population is initialized using the GRA algorithm with $|RLC| = 4$.

4.2.2. Reproduction

A roulette wheel selection is used to select the solutions for their reproduction according to the fitness value.

4.2.3. Crossover

The crossover operator used in this work is composed of two crossover operators explained below. In each crossover operation, one of the crossover operators is randomly selected.

Crossover by section

To generate a new offspring solution from s_1 and s_2, a section $k \in M$ is randomly selected. The vessels of each parent solution are divided into two sets according to that section: $N_s = A_s^k \cup B_s^k$, where

1. A_s^k refers to s-solution vessels that are berthed in sections after $k \in M$.
2. B_s^k refers to s solution vessels that are berthed in sections before $k \in M$.

Each set consists of triplets: (i, m_i, s_i) where $i \in N$, m_i is the time instant in which the vessel starts to be served, and s_i is the starting section of vessel i. The triplets of each set are sorted into ascending order according to the identifier of the start section s_i. If two vessels share the same start section, they are sorted into ascending order based on the time instant m_i.

The sets are alternated in the following order: B_1, B_2, A_1, A_2. In other words, the vessels of set B_1 are inserted first, then B_2, and so on. For each set, the triplets alternate according to the established order and are inserted into the offspring solution. Before each insertion, a check is made to determine whether the insertion makes the solution unfeasible. If so, the berthing times of already inserted vessels would be modified to ensure feasibility. In this sense, subsequent berthing times are assigned to ensure feasibility, but vessels are not allowed to change section.

Crossover by handling start time

In the crossover by handling start time, the same procedure is followed as with the crossover by section, but to generate a new offspring solution from s_1 and s_2, a handling start time $m_i, i \in N$ is chosen randomly. As in the previous crossover method, vessels of each parent solution are split into two sets: $N_s = A_s^{m_i} \cup B_s^{m_i}$, where

1. $A_s^{m_i}$ refers to s-solution vessels that are berthed in sections after the instant m_i.
2. $B_s^{m_i}$ refers to s solution vessels that are berthed in sections before the instant m_i.

4.2.4. Mutation

The mutation is randomly applied to solutions based on a given probability. The mutation procedure consists of applying an iteration of the pair of destruction and repair methods of the Large Neighbourhood Search metaheuristic (LNS, Shaw, 1998). The destroy method consists of removing certain elements (berth allocations) of the solution to promote diversification. In this work, the degree of destruction is fixed at the beginning of the search procedure, 0.15. Moreover, the repair method, GRA, consists of re-introducing in the solution the elements previously removed.

4.2.5. Solution evaluation

During the evaluation phase, it is necessary to obtain the fitness value of each solution from the population in order to guide the search process. The metrics used are the following:

- Total service time ($f(s)$).
- Robustness metric $h(s)$ obtained through the simulation process as shown in Section 4.1.1.
- A normalized metric considering $f(s)$ and $h(s)$ of all individuals of the population. This objective is used mainly to compare the solutions of the single-objective algorithms and the NSGA-II. In doing so, the distance to the point (0, 0) is used as the fitness value.

Therefore, these functions lead to different versions of the above described GA, i.e., GA-Sim($f(s)$), GA-Sim($h(s)$), and NGA. These versions are later evaluated in Section 5.3.5.

4.3. Non-Dominated Sorting Genetic Algorithm II

In contrast to the Genetic simheuristic algorithm described above, where the problem objective and the robustness-related objective are individually optimized or both objectives are normalized and added into one objective, in this section, we propose multi-objective simheuristic approach. This way, the Non-Dominated Sorting Genetic Algorithm II (NSGA-II, Deb et al., 2002) is adapted. This algorithm incorporates the concept of dominance, elitism within the population, and crowding distance, so that an approximate Pareto front is obtained. In this sense, this provides a set of multiple efficient solutions allowing decision-makers the possibility of choosing the solution based on their current needs.

The initialization of the population as well as the mutation and crossover operators are the same as those described in the previous section. The general procedure of this algorithm is described below:

1. The current population P and offspring Q are combined to obtain the combined population $R = P \cup Q$.
2. Non-dominated sorting is performed and Pareto fronts are identified $F = (F_1, F_2, \dots)$.
3. $i = 0$ and $P = \emptyset$.
4. While $|P| + |F_i| \leq N$ do $P = P \cup F_i$ and $i = i + 1$. If the current front exceeds the population size, $|P| + |F_i| > N$, the crowding distance is calculated. Based on this distance, the most dispersed solutions are selected and included in the population until the maximum is reached, $N - |P|$.
5. The new offspring population Q is generated from P through the selection, crossover and mutation operators. Return to Step 2 whenever a certain stop criterion is not met.

It is worth noting that the simheuristic version of this algorithm only requires the proper incorporation of the objective of the problem at hand as well as the robustness-related objective obtained by means of the simulation process. That is, every time a new individual is generated or altered by means of an operator then it is evaluated through $f(s)$ and a simulation process over this individual is conducted to obtain $h(s)$. Finally, in Step 2, an efficient non-dominated sort using a binary search strategy (ENS_BS, Zhang et al., 2014) is used to obtain Pareto fronts efficiently.

4.3.1. Selection

The selection of the individuals that pass to the next generation during the reproduction is done using the crowded tournament selection operator.

4.3.2. Solution evaluation

The evaluation of the solutions within NSGA-II considers the optimization problem objective ($f(s)$) and the robustness-related objective ($h(s)$) obtained after the simulation process. This way, with the first objective, the idea is to minimize the total service time of the proposed solutions, while the second one minimizes the penalty cost.

4.4. Buffers times

In this work, the use of buffers to increase the margin of action when stochastic changes happen is investigated. A buffer is defined as a frame of periods during which a berthing section is reserved for a vessel. This additional slack time in the planning permits to absorb the impact of disruptive events such as delays in handling or arrival times. It should be noted that the drawback of incorporating buffers involves a decrease in resource availability. In this sense, an excessive amount of buffers or an inaccurate definition of them can lead to an overall reduction in the terminal capacity. For this reason, in this section, a data-driven heuristic approach for calculating them based on scenario information and historical data is proposed.

For each vessel, a heuristic approach to calculate the maximum buffer time is applied. The two sources of uncertainty presented in this work are analysed separately: (*i*) handling time and (*ii*) arrival time. If we consider the uncertainty associated with the handling time, the maximum buffer time needed to mitigate that uncertainty source is $h_i^j \cdot (\gamma - 1)$, $i \in N, j \in M$, where γ is used to control the level of uncertainty associated with the handling time. On the other hand, if the arrival time is selected as a source of uncertainty, it is necessary to consider the start of the vessel's handling time in order to reliably calculate the maximum buffer time. Since each vessel might be delayed $\delta + ETA_i$, $i \in N$, it is necessary to take into account its starting handling time as it can cause delays in the baseline schedule. If the handling time of a vessel begins during the interval $[ETA_i, \delta + ETA_i]$, delays may affect the baseline schedule. However, if the handling time begins during the interval $(\delta + ETA_i, \infty]$, the vessel is not affected by the uncertainty in the arrival time, since it is certain that the handling process of the vessel cannot start as the vessel is not yet berthed at the port.

According to the above discussion, the maximum buffer time for vessel i berthed in section j is calculated using Eq. (8) as follows:

$$Buffer_i^j = HT_i^j + AT_i^j \tag{8}$$

$$HT_i^j = h_i^j \cdot (\gamma - 1) \tag{9}$$

$$AT_i^j = \begin{cases} 0 & m_i \geq ETA_i + \delta \\ ETA_i + \delta - m_i & m_i < ETA_i + \delta \end{cases} \tag{10}$$

Buffer times can be integrated into the solution approaches described in Sections 4.2 and 4.3. The same buffer approach is used for all proposed algorithms.

Two strategies can be used when managing buffer times: (*i*) static and (*ii*) dynamic. In the case of applying a static approach, each vessel is always allocated in the half of the maximum buffer time needed to absorb uncertainty, i.e., $Buffer_i^j/2$. This value is updated if, during the search process, the berthing section or the time instant in which the vessel starts to be served is changed. It is known as a static approach because a vessel i berthed at section j always has the same buffer time allocated to it. On the other hand, if a dynamic approach is used, the buffer times vary during the search process. In this approach, during the population initialization, each vessel is assigned a buffer time in the interval $[(Buffer_i^j/2) \cdot 0.5, (Buffer_i^j/2) \cdot 1.5]$. During the crossover, a buffer time is assigned to each vessel of the offspring solution. In this case, this value is the average of the corresponding vessel's buffer value in the parent solutions. Finally, during the mutation operation, each vessel is assigned a buffer time in the interval defined during the population initialization.

5. Numerical experiments

This section presents the computational results obtained for the different proposed solution approaches. All numerical experiments are carried out on a personal computer with a 3.3-GHz i7-5820k processor and 8 GB of RAM.

The set of 54 instances used in this work is the same of Umang et al. (2013) and it is based on data from SAQR port, Ras Al Khaimah (RAK), UAE. Depending on the number of vessels and berthing sections each instance is labelled as $N \times M$: 10×10, 10×30, 25×10, 25×30, 40×10, and 40×30. This divides the original set into 6 subsets of 9 instances.

In the comparison between the approaches, the following values are selected for the probability distributions used to model the uncertainty in the Bulk-BAP: *(i)* $\delta = 7.5$ and *(ii)* $\gamma = 1.15$. Those values were determined from the historical data presented by Umang et al. (2017).

The experiments conducted in this section are the following:

- In Section 5.1, the solutions provided by 4 different algorithms (i.e., heuristics, constructive, metaheuristic, and evolutionary algorithm) are analysed under conditions of uncertainty. The goal of this study is to analyse how stochastic arrival and handling times affect after initial solutions are generated. This permits obtaining an analysis of the impact of those uncertainties on the methods and instances.
- Since Monte Carlo simulation is jointly considered with metaheuristics, in Section 5.2, a trade-off between the number of iterations and mean average error for the different simulation phases (i.e., fast and intensive simulation) is obtained. Based on this study, the number of iterations of the simulation that are used in the approaches is determined.
- The proposed solution approaches to proactively handle the impact of disruptive events are compared with each other in Section 5.3. This experiment aims at providing insights on the different simulation schemes proposed in this work, i.e., standard, integrated, and multi-objective simheuristics.
- The contribution that buffer time management strategies have on the objective function value and robustness of the solutions are analysed in Section 5.4.
- Finally, the proposed solution approaches are analysed by means of the well-known hyper-volume measure (Zitzler and Thiele, 1999) in Section 5.5.

5.1. Uncertainty analysis

An uncertainty analysis is carried out to determine the average impact of the disruptive events associated with the Bulk-BAP. To do this, four solutions approaches are used to provide a solution to the deterministic case. On those solutions, a very long Monte Carlo simulation to measure the total delay time of the solutions is carried out. This permits studying the performance of the solutions provided by the algorithms for each instance under stochastic conditions. The four solution approaches used during this experiment are the following:

- First-Come First-Served (FCFS): vessels are sorted according to their arrival time and berthed in the best available section.
- Greedy Randomized Algorithm (GRA, see Section 4.2.1): $|RLC| = 4$.
- Large Neighbourhood Search (LNS, see Section 4.2.4): degree of destruction = 0.15; repair method = GRA with $|RLC| = 4$; destroy method designed by removing berth allocations at random; stopping criterion defined by a fixed number of 100 iterations.
- Basic Genetic Algorithm (Basic-GA, Davis, 1991): population size = 100; mutation rate = 1%; crossover operator (see Section 4.2.3); selection operator (see Section 4.2.2); mutation operator (see Section 4.2.4); initialization (see Section 4.2.1); fitness function: $f(s)$; stopping criterion defined by a fixed number of 100 iterations.

Table 1
Uncertainty analysis grouped by $N \times M$.

Instance		Estimated total delay time in vessels departure				
		Min.	Q1	Median	Q3	Max.
10 × 10	Min.	0.00	9.73	17.01	25.16	57.28
	Avg.	0.00	12.59	19.54	27.27	65.07
	Max.	0.00	14.40	22.43	32.02	77.09
10 × 30	Min.	0.00	9.44	16.82	25.13	57.18
	Avg.	0.00	12.60	19.38	26.92	64.76
	Max.	0.00	14.64	22.25	31.12	75.93
25 × 10	Min	0.00	30.63	46.46	61.98	132.40
	Avg.	0.00	41.09	58.82	77.47	164.36
	Max.	0.16	55.31	76.27	95.25	189.10
25 × 30	Min.	0.00	35.14	50.24	65.20	135.35
	Avg.	0.02	44.36	61.75	79.80	162.71
	Max.	0.40	56.79	75.79	94.78	185.16
40 × 10	Min.	0.00	50.41	79.98	109.72	222.08
	Avg.	0.14	77.00	105.05	134.08	263.22
	Max.	1.92	106.78	136.30	164.87	295.02
40 × 30	Min.	0.00	60.96	89.78	118.65	225.22
	Avg.	0.62	87.10	114.90	142.52	260.08
	Max.	4.11	116.17	146.05	173.19	288.17

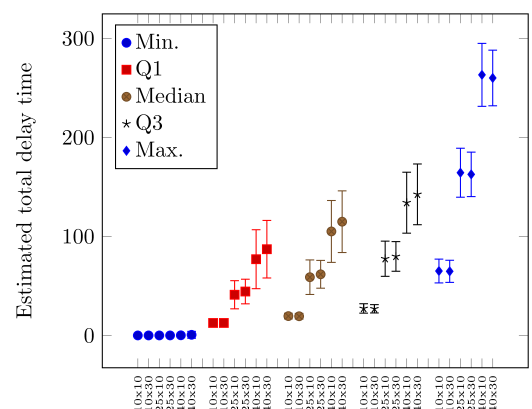

Fig. 1. Uncertainty analysis grouped by $N \times M$.

Firstly, it is necessary to determine whether any solution minimizes the total delay time. In doing so, each algorithm-instance pair is run 1000 times, giving a total of 4000 solutions per instance. To analyse the behaviour of the solutions under conditions of uncertainty, a very intensive Monte Carlo simulation of 100000 iterations is applied to each of these solutions.

After finishing the simulation process for a given solution, the Z set (see Section 4.1.1) is obtained, meaning that the behaviour of the solution in stochastic scenarios can be determined. Once the Z set is obtained for each of the 4000 solutions associated with an instance I, it is possible to determine whether there are solutions that minimize the risk of total delays for said instance.

Using this information, the uncertainty analysis of each instance can be obtained. Furthermore, to simplify it, a new set Z' is defined, made up of the following values of the Z set: the minimum value, the first, second and third quartiles, and the maximum value. If variations in the Z' elements are observed for the same instance, this means that there are solutions with different risk. To do so, the minimum, average, and maximum values are calculated for each of the Z' elements, so it is possible to analyse whether there are variations in them. This information is shown in Table 1, where for simplicity purposes the data is grouped by type of $N \times M$ instance. Fig. 1 shows the results presented in Table 1, where the results are grouped by quartile, e.g., the first grouping corresponds to the variability associated with the Q0 quartile (Min.) for all instances, or the last grouping corresponds to the Q4 quartile (Max.) obtained for each type of instance. Instances are ordered

Table 2
Uncertainty analysis grouped by algorithm.

Algorithm	Expected value	Estimated total delay in vessels departure					Penalty cost
	$f(s)$	Min.	Q1	Median	Q3	Max.	$h(s)$ ($)
FCFS	792.05	0.00	35.36	53.48	73.28	166.32	42784
GRA	780.31	0.15	49.74	67.84	86.24	165.46	54272
LNS	737.78	0.17	48.30	65.40	82.83	160.50	52320
Basic-GA	658.70	0.21	49.63	66.06	82.82	160.80	52848

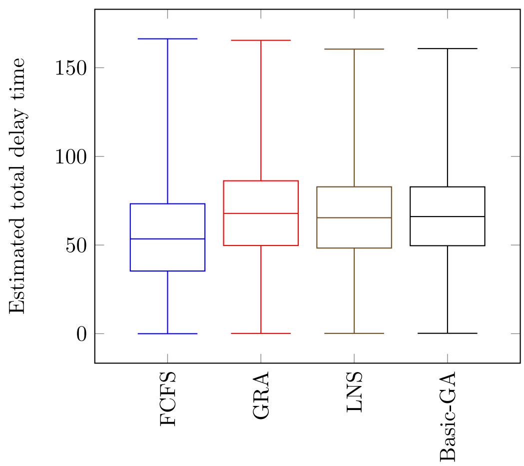

Fig. 2. Estimated total delay time of the solutions provided by FCFS, GRA, LNS and Basic-GA algorithms. Data is grouped by algorithm.

from smallest to largest within each grouping. For example, the median of the 25×30 instance group in Table 1 has a minimum value equal to 46.46, an average value of 58.82, and a maximum value of 76.27. This information is displayed in the third grouping (Median), labelled 25×30 under the x-axis in Fig. 1.

Moreover, it can be seen that the larger the instances, the more risk there is, as the minimum, average, and maximum values of each of the Z' elements move away from 0. Notice that 0 means that there was no variation with regards to the initial planning. The same behaviour is observed if the variation of each of the Z' elements is analysed. The larger the instance, the greater the variation. As can be seen, the minimum values have little to no variability. However, for the rest of the Z' elements, variability is observed, indicating that there are solutions with more risk than others. For example, the difference between the maximum and minimum values of the median for the 10×10 instances is 5.42, while it is 56.27 for the 40×30 instances. Meanwhile, the average median value for the 10×10 instances is 19.54, while it is 114.90 for the 40×30 instances.

We should point out that after having performed such intensive Monte Carlo simulation of 100000 iterations, there is a reduced possibility that the results shown depend on the randomness of the simulation.

However, as shown in Table 2 and Fig. 2 in this experiment none of the algorithms outperforms on average the others in terms of minimizing the total delay for all instances. The expected average objective function value $f(s)$ (from the deterministic case), estimated total delay in vessels departure, and penalty cost $h(s)$ due to the stochastic changes (i.e., $800 per delayed time unit) grouped by algorithm are presented in Table 2. Fig. 2 shows a multiple box-plot comparison of the estimated total delay time reported in Table 2. As can be seen in the table, all of the algorithms provide a similar estimated total delay in vessels departure and penalty costs, with a slight advantage over FCFS heuristics. The reasons why the FCFS heuristic provides lower-risk solutions than the rest of the approaches (i.e., GRA, LNS, and Basic-GA) is because its baseline solution is worse, this is further explained in Section 5.3.5. Secondly, the risk analysis indicates that, although the expected objective values of the solutions are dissimilar, the associated penalty cost is not. For example, although the GRA with $|RLC| = 4$ algorithm obtains an estimated value equal to 780.31 compared with 658.70 obtained by the Basic-GA, the penalty cost is very similar, 54272$ and 52848$, respectively. Likewise, for the rest of the algorithms.

The existence of low-risk solutions and the fact that none of the used algorithms is capable of outperforming the other approaches when stochastic components are considered, suggests that it is necessary to use simulation-based optimization algorithms that take them into account during the search process as discussed in the next sections.

5.2. Parameter selection

The solution approaches incorporate a Monte Carlo simulation phase while evaluating the solutions. Although simulation is necessary in order to determine the penalty cost of the solutions, it is an expensive process in terms of computational effort. If the number of iterations is too high, then the computation burden will be high, while if it is very low, the precision of the penalty cost derived from the stochastic conditions could be greatly affected. It is, therefore, necessary to identify a number of iterations that provide a good trade-off between computation time and accuracy.

In this experiment, a set of 1000 solutions, S, is generated using the constructive algorithm GRA with $|RLC| = 4$ for each of the 54 instances (i.e., 54000 solutions). For each of solution, the Monte Carlo simulation is applied for different number of iterations $I = [10, 50, 100, 500, 1000, 5000, 10000, 50000]$ in order to determine the mean absolute error $MAE_i, i \in I$, given by Eq. (11). The Monte Carlo simulation with 100000 iterations is used as a reference point in the parameter analysis.

$$MAE_i = \frac{1}{|S|} \cdot \sum_{s \in S} |h(s,i) - h(s,100000)|, \quad (11)$$

where $h(s,i)$ represents the penalty cost associated with the disruptive events for solution s under stochastic conditions using a Monte Carlo simulation of i iterations (see Section 4.1.1).

Table 3 shows the MAE_i obtained for each of the iterations, along with the average time, in milliseconds, required to perform the simulation (Time). During the fast simulation, a high percentage of precision is necessary to avoid misguiding the search process. To achieve a total feasible computation time, however, the additional computation cost has to be minimal. It is necessary to take into account that the fast simulation is applied to all solutions generated, so the computation time used is increased geometrically. Because of this, it is decided that the number of iterations necessary to minimize the error during the fast simulation is 500, with a computation time per solution of less than 1 millisecond. On the other hand, in the intensive simulation, the time constraint is not relevant as it is only applied to a small subset of solutions. Thus, a simulation of 100000 iterations over each solution is applied to ensure a high precision.

The following sections compare different approaches based on GA and NSGA-II. The parameters of these algorithms are set considering preliminary experiments as well as based on those considered in de León et al. (2017). In that work, the selection of the best algorithm for solving the deterministic Bulk-BAP for diverse scenarios was dpne by a Machine Learning-based system. To train such a system, multiple

Table 3
Comparison of MAE_i and Time (ms) under different numbers of iterations of Monte Carlo simulation.

Iterations	MAE_i	Time
10	7.13	0.01
50	3.36	0.04
100	2.39	0.09
500	1.07	0.40
1000	0.76	0.78
5000	0.33	3.91
10000	0.23	7.82
50000	0.08	39.03

experiments were conducted on various algorithms including the Basic Genetic Algorithm (Basic-GA) considered in this paper. The set of 720 instances used in de León et al. (2017) was generated from the set of 54 instances used in this paper. Since in previous research we approached the Bulk-BAP in a deterministic and single-objective way using the same instances and operators (i.e., crossover, selector, and mutation operator), the obtained knowledge is extrapolated to this paper with regards to parameter tuning. Based on this, together with the previous experiments performed (e.g. see Section 5.1), all experiments are carried out using the same parameters:

- The initial population is generated by GRA with $|RLC| = 4$.
- The population size is set to 500.
- The time limit for the solver is set to 30 s.
- Mutation rate is set to 1%.
- Each pair instance-algorithm is executed 30 times.

5.2.1. Convergence

In order to compare the proposed single-objective and multi-objective approaches under the same conditions, a time limit is used as stop condition for the solver. In this regard, a convergence study of GA-Sim($f(s)$) and NSGA-II is carried out. The time limit is tuned based on the average results obtained by both approaches. Since each pair instance-algorithm is executed 30 times, the results shown in Figs. 3 and 4 report the average values.

Fig. 3 reports the evolution of the minimum (Min. $f(s)$), average (Avg. $f(s)$), and maximum (Max. $f(s)$) deterministic objective function value of the population (y-axis) obtained by GA-Sim($f(s)$) over time (x-axis). The evolution of the average hyper-volume obtained by NSGA-II is reported in Fig. 4. For each instance, the procedure followed to obtain the hyper-volume is as follows:

- The maximum and minimum value of $f(s)$ and $h(s)$ are obtained by analysing the 4500 non-dominated fronts generated (150 iterations · 30 repetitions = 4500).
- The sets of non-dominated points are normalized (min–max feature scaling).
- Based on the reference point, the hyper-volume of each non-dominated fronts is calculated.
- For each iteration, the average hyper-volume achieved in the 30 repetitions is calculated.

It is clear that GA-Sim($f(s)$) converges around the 20 s of execution, unlike NSGA-II which begins to converge after 30 s of execution. Because the objective is to perform an experiment under the same conditions, it is decided that the time limit is 30 s, since it allows both proposals the time needed to converge. Moreover, the use of 30 s per run also fits with the type of problem (i.e., operational). In addition, as discussed in Section 5.2, the simulation process is an expensive process in terms of computational time, so a sufficiently long time limit might be necessary.

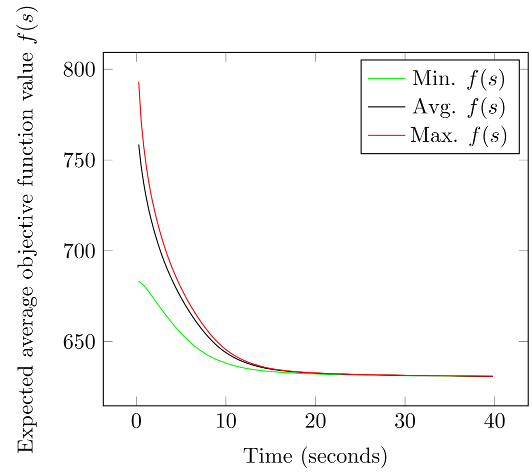

Fig. 3. Convergence of GA-Sim($f(s)$).

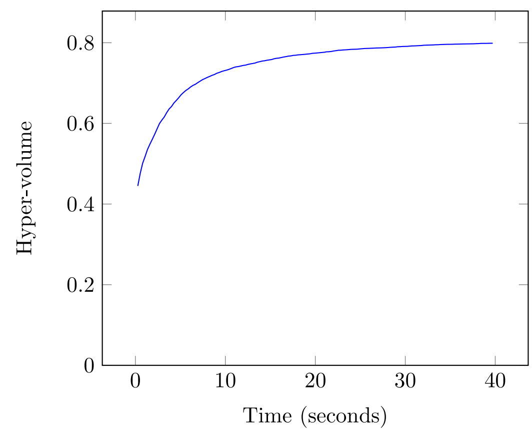

Fig. 4. Convergence of NSGA-II.

5.3. Simheuristic approaches

To analyse the different solution approaches in detail, in this section we first describe the algorithms' development specifications and later compare their performance by means of different experiments.

5.3.1. Single-objective GA with simulation non-integrated into the search process (GA)

The first solution approach considered in this study is the single-objective GA where the simulation is not integrated into the metaheuristic search process as defined in the simheuristic general scheme (Juan et al., 2015). Including this algorithm permits later assessing the contribution that the simulation process, when integrated into the metaheuristic search process, has on the solution robustness.

In this solution approach, for each solution provided by the metaheuristic, a fast simulation phase is executed. Based on a certain criterion, a subset of elite solutions is then selected and an intensive simulation phase is applied to these. It is important to note that the fast simulation phase is not considered during the metaheuristic search process but at the end of it, once a solution is provided (see Algorithm 1).

Two criteria are used for selecting the elite solutions for the intensive simulation. These are: (*i*) $f(s)$ and (*ii*) $h(s)$. Using the first criterion ignores the results obtained from the simulation and only takes the objective function value into account. The second criterion leads to ranking solutions based on the penalty cost associated with the total delay time in vessel departure caused by the stochastic changes.

5.3.2. Single-objective GA with simulation integrated into the search process (GA-Sim)

The second solution approach aims to study if a single-objective approach is sufficient to minimize the penalty cost and objective function value simultaneously when the simulation outcome is considered in the metaheuristic search process. This way, unlike the previous solution approach, a fast simulation phase is integrated during the search process to evaluate the robustness of the solutions, see Algorithm 2.

Similarly as in the preceding section, two criteria are used to guide the search process as well as construct the ranking of elite solutions for the intensive simulation: (*i*) $f(s)$ and (*ii*) $h(s)$.

5.3.3. Normalized GA with simulation integrated into the search process (NGA)

The third solution approach aims to investigate the contribution of jointly optimizing the deterministic and stochastic objectives simultaneously by means of normalizing and merging them. As in the previous case (see Section 5.3.2), the search process considers the outcome from the simulation phase using this normalized objective. This way, balanced solutions for both objectives can be provided.

5.3.4. NSGA-II with simulation integrated into the search process (NSGA-II)

The fourth solution approach aims at studying the benefits of using a multi-objective approach as discussed in Section 4.3 that allows the decision-market to treat both objectives simultaneously and obtain a Pareto front. From this set of non-dominated solutions, the users can select the one that suits best their current needs.

5.3.5. Simheuristic approaches comparison

In this section, the following classification is used to refer to the different approaches:

1. GA: Single-objective GA with simulation non-integrated into the search process. The metaheuristic algorithm uses either the objective function of the Bulk-BAP (i.e., GA($f(s)$)) or the one seeking to minimize the penalty cost once the stochastic changes are considered (i.e., GA($h(s)$)).
2. GA-Sim: Single-objective GA with simulation integrated into the search process. Similarly, as in the previous case, two versions are considered, i.e., GA-Sim($f(s)$) and GA-Sim($h(s)$).
3. NGA: GA that considers both objective functions at the same time by normalizing and summing them up. The function of this algorithm considers the simulation integrated into the search process.
4. NSGA-II: Non-Dominated Sorting Genetic Algorithm II (NSGA-II) with simulation integrated into the search process.

The results obtained from comparing the four solution approaches described in Sections 5.3.1–5.3.4 are summarized in Table 4. The expected average objective function value $f(s)$, estimated total delay in vessels departure, and the penalty costs $h(s)$ are provided for each solution approach. Fig. 5 shows a multiple box-plot comparison of the estimated total delay time reported in Table 4.

Firstly, the results provided by GA are analysed. Slight differences are observed in the objective function value and the penalty cost obtained depending on the criterion selected to generate the ranking. If it is constructed based on the criterion $f(s)$, slightly higher quality solutions are obtained, but with a slightly higher penalty cost than if the criterion $h(s)$ is used, and vice-versa.

The above-reported results indicate that applying the simulation at the end of the search process as defined by simheuristics and done in GA is not sufficient to obtain robust baseline schedules. This raises the research question about using the simulation outcome to guide the search process of the metaheuristic. Hence, to answer that question, the results obtained by considering the simulation in the search process (i.e., GA-Sim) are then analysed. Similarly, as before, two criteria are used to guide the search process as well as construct the ranking of elite solutions. The highest-quality solution in terms of expected average value is given by GA-Sim($f(s)$) with a value of 631.91. However, they are high-risk when compared to the rest of the solution approaches. For example, while GA-Sim($f(s)$) obtains a penalty cost equal to 53833$, GA-Sim($h(s)$) and NSGA-II($h(s)$) obtains 18188$ and 28099$, respectively.

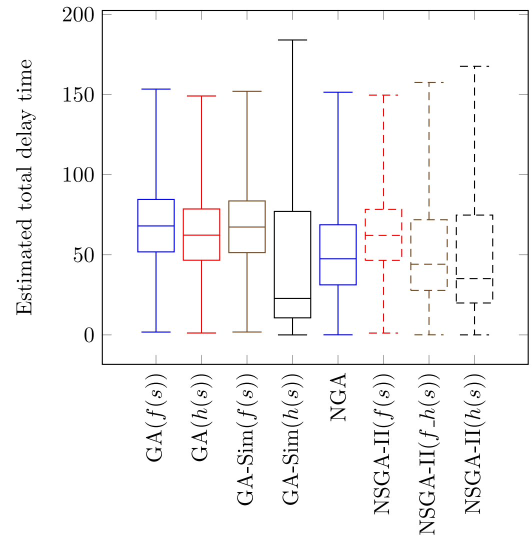

Fig. 5. Estimated total delay time comparison of solutions provided by GA, GA-Sim, NGA and NSGA-II based solution approaches.

On the contrary, if $h(s)$ is used to guide the search process, solutions are obtained with a clear decrease in penalty cost, but at the expense of a significant deterioration in the expected average objective function value, $f(s)$, of the solutions. Specifically, the clear decrease in the quality in terms of $f(s)$ of the solutions is since GA-Sim is capable of detecting that, if a vessel i starts processing later than time $ETA_i + \delta$, its uncertainty only depends on the handling time. This means that GA-Sim, instead of distributing the vessels in different sections to minimize waiting time, accumulates the vessels in the same sections, so that most of them start processing after that instant. This implies a considerable increase in the objective function value and a decrease in the penalty cost of the solutions. However, the expected average objective function value of the solutions make it unfeasible to use this strategy. In this regard, the results presented in Section 5.1 for the FCFS heuristic are also due to this fact.

The results of the previous solution approaches, GA and GA-Sim, focus on optimizing $f(s)$ or $h(s)$ individually. Although these strategies make it possible to obtain high-quality solutions for some of the objectives individually, they are not able to obtain balanced solutions for $f(s)$ and $h(s)$. In this regard, NGA and NSGA-II use different strategies to generate feasible solutions considering both objectives. While NGA normalizes both objectives to work with a single value, NSGA-II focuses on obtaining a Pareto front. Given the fact that NSGA-II provides a Pareto front, three of these solutions are selected to make the comparison with the rest of the solution approaches:

1. The solution with the lowest objective value $f(s)$.
2. The solution with the lowest normalized distance to the origin of coordinate $f\_h(s)$.
3. The solution with the lowest penalty cost $h(s)$.

NGA and NSGA-II manage to increase the solutions' robustness at the expense of increasing the function value $f(s)$ when compared to GA-Sim($f(s)$). Moreover, both solution approaches are capable of providing balanced solutions in terms of $h(s)$ and $f(s)$. Nevertheless,

Table 4
Performance comparison between GA, GA-Sim, NGA and NSGA-II based solution approaches.

Algorithm	Expected value	Estimated total delay in vessels departure					Penalty cost
	$f(s)$	Min.	Q1	Median	Q3	Max.	$h(s)$ ($)
GA($f(s)$)	644.59	1.76	51.81	67.99	84.51	153.35	54393.86
GA($h(s)$)	669.33	1.20	46.58	62.23	78.59	149.09	49783.37
GA-Sim($f(s)$)	631.91	1.79	51.34	67.29	83.58	151.96	53833.80
GA-Sim($h(s)$)	1403.62	0.00	10.65	22.74	77.03	184.00	18188.54
NGA	703.38	0.07	31.23	47.50	68.72	151.39	37999.33
NSGA-II($f(s)$)	635.04	1.14	46.49	62.07	78.30	149.59	49654.05
NSGA-II($f\_h(s)$)	749.99	0.07	27.74	44.07	71.87	157.49	35253.69
NSGA-II($h(s)$)	1002.07	0.02	19.90	35.12	74.74	167.56	28099.44

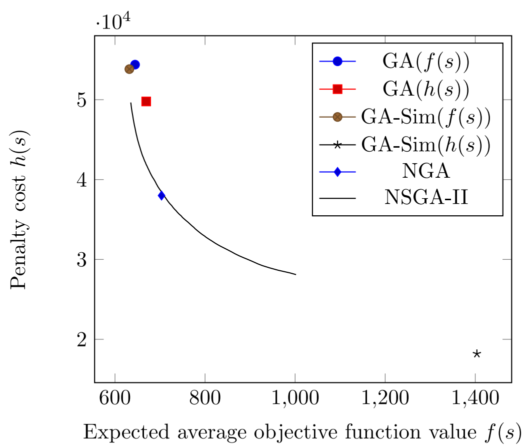

Fig. 6. Performance comparison between GA, GA-Sim, NGA and NSGA-II approaches.

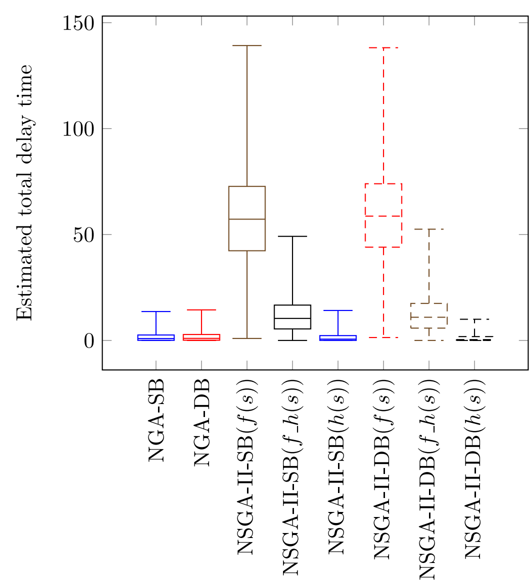

Fig. 7. Estimated total delay time comparison between buffer times based solution approaches.

it is necessary to take into account that the normalized GA does not provide a Pareto front but a solution, while NSGA-II does it. In this sense, if the solutions with the lowest objective value obtained by NSGA-II are compared against GA-Sim($f(s)$), we see that NSGA-II($f(s)$) provides solutions of the very slightly worse quality in terms of $f(s)$ but better penalty cost. However, compared to GA($f(s)$), NSGA-II provides solutions that dominate that one. In the case of seeking to only minimize the penalty cost, NSGA-II provides the best results of all the solution approaches, with the exception of the solutions provided by GA-Sim($h(s)$). Fig. 6 illustrates the values provided in Table 4, where the vertical axis represents the penalty cost ($h(s)$), while the horizontal axis represents the expected average objective function value ($f(s)$). In the case of NSGA-II, the Pareto front is delineated. Although the NGA approach is capable of providing a better solution than NSGA-II, the latter is capable of providing a Pareto front with low-risk and high-objective-value solutions, balanced solutions, and high-risk and low-objective-value solutions in the same execution.

Based on the reported results, we can point out that the NSGA-II is the most convenient solution approach due to the diversity of solutions that make up the Pareto front. The obtained results demonstrate how NSGA-II can compete simultaneously with other solution approaches that are focused on a single objective. That is, in the same execution, it can provide a Pareto front with solutions with the same features as the rest of the solution approaches combined.

5.4. Simheuristic approaches with buffer times

The previous section showed diverse ways to address Bulk-BAP considering changes in the arrival and handling times. Independently of the selected solution approach, it is also possible to increase the robustness of the solutions by adding extra buffer times between vessels to absorb the impact of disruptive events.

Considering the previous results, in this experiment the best approaches so far, i.e., NGA and NSGA-II, are selected for incorporating buffers, where NGA-B corresponds to NGA with buffers, while NSGA-II-B corresponds to NSGA-II with buffers. Moreover, depending on the type of buffer strategy (see Section 4.4), NGA-B and NSGA-II-B can be divided into two groups depending on the used buffer strategy. NGA-SB and NSGA-II-SB represent the case where static buffers are considered, while NGA-DB and NSGA-II-DB the cases where a dynamic buffer strategy is applied.

For each solution approach, the expected average objective function value $f(s)$, estimated total delay time in vessels departure, and penalty costs $h(s)$ are provided in Table 5. Fig. 7 shows a multiple box-plot comparison of the estimated total delay time reported in Table 5. Additionally, Fig. 8 shows a performance comparison between all the solution approaches proposed in this work.

As can be seen, NGA-B, regardless of the strategy used to manage buffer times (i.e., static or dynamic), produces an increase in $f(s)$ together with a significant decrease in the penalty cost compared with the results obtained by NGA without buffers. Considering Fig. 8, if we compare NGA-SB with NGA-DB, the latter achieves an improvement in the objective function value at the cost of a slight deterioration in robustness. In the case of NSGA-II-B, if the Pareto fronts provided by NSGA-II-SB and NSGA-II-DB are compared the same insight as

Table 5
Performance comparison between buffer times based solution approaches.

Algorithm	Expected value	Estimated total delay in vessels departure					Penalty cost
	$f(s)$	Min.	Q1	Median	Q3	Max.	$h(s)$ ($)
NGA-SB	851.90	0.00	0.05	0.90	2.58	13.65	721.54
NGA-DB	802.96	0.00	0.11	0.98	2.81	14.43	781.80
NSGA-II-SB($f(s)$)	642.10	0.94	42.35	57.24	72.73	139.22	45793.12
NSGA-II-SB($f\_h(s)$)	724.26	0.00	5.45	10.40	16.69	49.14	8316.94
NSGA-II-SB($h(s)$)	979.17	0.00	0.02	0.62	2.29	14.15	495.37
NSGA-II-DB($f(s)$)	639.33	1.38	44.03	58.68	73.98	138.20	46946.51
NSGA-II-DB($f\_h(s)$)	703.63	0.00	5.82	10.95	17.48	52.52	8761.98
NSGA-II-DB($h(s)$)	881.13	0.00	0.02	0.39	1.82	10.00	308.08

SB: indicates static buffers; DB: indicates dynamic buffers.

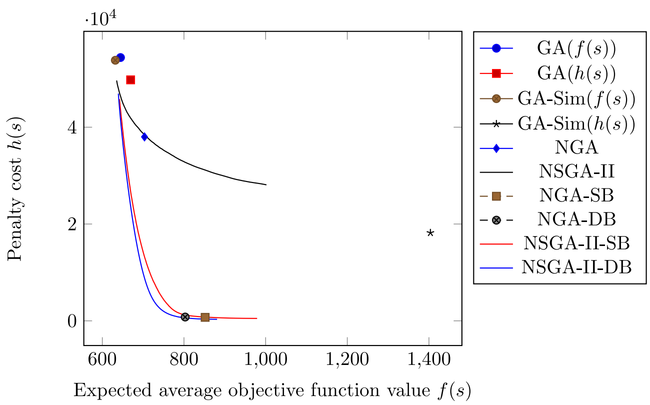

Fig. 8. Performance comparison of all approaches.

before is observed, i.e., NSGA-II-DB exposes dominance over NSGA-II-SB. Moreover, considering NSGA-II without buffers, it is observed that the majority of its Pareto front is contained inside the Pareto fronts provided by the approaches incorporating buffers. Specifically, by employing NSGA-II-DB($h(s)$) the lowest-risk solutions in this paper are obtained with penalty cost close to 308$, while NSGA-II-DB($f(s)$) provides the best buffer-based approach with regards to $f(s)$. In this regard, when compared with a single-objective approach such as NGA-SB or NGA-DB, we can observe from Fig. 8 that the provided solutions are dominated by the solutions in the Pareto front of NSGA-II-B.

Furthermore, regardless of the strategy used to manage buffer times (i.e., static, dynamic, or neither), the proposed solutions fall within a similar quality range in terms of the objective function value $f(s)$, $[631, 1002]$. In contrast, there are clear differences in the penalty cost associated with the solutions. In this sense, the incorporation of buffer times significantly reduces the penalty cost of solutions. That is, solutions are obtained with an associated penalty cost within the range $[308, 46946]$, while if buffer times are not used, the range obtained is $[18188, 54393]$. It is, therefore, necessary to identify whether this reduction in penalty cost leads to a large increase in the objective function value. In this regard, Fig. 8 clearly shows that the cost of increasing the robustness of the solutions is lower when buffer times are used. For example, consider the difference in slope of the line tangent to the Pareto front provided by NSGA-II without buffer time management strategy and NSGA-II-SB at point 650. In addition, by using buffer times it is possible to find solutions of the same expected average value as if no buffer times were used, but also with a lower associated penalty cost. For example, for a solution with an objective function value equal to 720, if we apply the NSGA-II-DB, we obtain a penalty cost equal to 4950$, while if we use NSGA-II without buffer time management strategy we obtain a penalty cost close to 37040$.

Table 6
Hyper-volume.

Algorithm	Min.	Q1	Median	Q3	Max.
GA-Sim($h(s)$)	0.15	0.22	0.25	0.28	0.35
GA-Sim($f(s)$)	0.02	0.05	0.07	0.09	0.12
NGA	0.23	0.26	0.28	0.30	0.34
NSGA-II	0.40	0.44	0.46	0.49	0.53
NGA-SB	0.78	0.80	0.81	0.82	0.84
NGA-DB	0.82	0.84	0.85	0.86	0.87
NSGA-II-SB	0.94	0.95	0.95	0.95	0.96
NSGA-II-DB	0.94	0.95	0.96	0.96	0.97

SB indicates static buffers; DB indicates dynamic buffers.

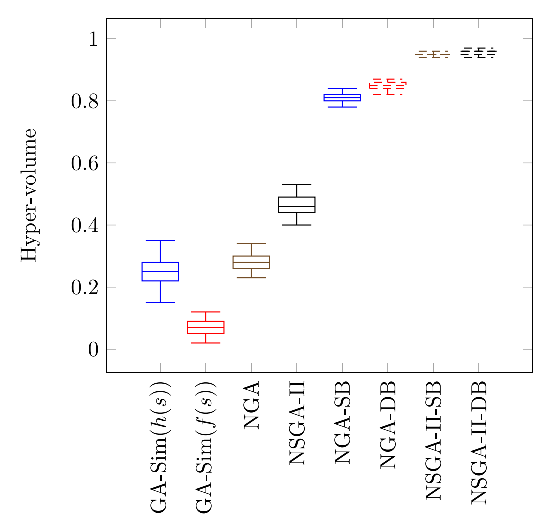

Fig. 9. Hyper-volume.

From the above discussion and analysis, it can be concluded that the dynamic buffer time management strategy overcomes the static one used in NSGA-II-B. In turn, the NSGA-II-SB exposes dominance over NSGA-II when buffers are not used, and therefore advises its application in Bulk-BAP or related contexts.

5.5. Hyper-volume analysis

In order to compare the different approaches, the hyper-volume measure (Zitzler and Thiele, 1999), also known as the size of the space covered (i.e., S-measure), is used. In the two-dimensional case, given a set of non-dominated solutions such as $S = \{(f(s_1), h(s_1), \dots, f(s_n), h(s_n))\}$ and a reference point R, the hyper-volume $HV(S, R)$ is defined as the objective value space covered by S with respect to R.

The reference point, R, does not dominate any solution of the Pareto front, so usually, the nadir point is used. Because the worst possible point is unknown, the nadir point is determined from the sets to be compared. Concretely, for each objective, the worst value is selected from all the sets being compared (While et al., 2011). In this paper, the objective value space is normalized so that the nadir point is $(1, 1)$. The greater the hyper-volume value, the better the approximation set of non-dominated solutions S.

Because the maximum values of the objective functions are unknown, the hyper-volume calculated in this paper cannot be used to measure the absolute performance of the approaches (Knowles et al., 2006). However, it can be used to compare the relative performance among approaches. Table 6 shows the hyper-volume obtained for each instance-algorithm pair and Fig. 9 illustrates those values in a multiple box-plot comparison. For single-objective approaches, only the best solution from the population is used in the Pareto front. For each problem instance, the procedure to obtain the hyper-volume is as follows:

1. The maximum and minimum values of $f(s)$ and $h(s)$ are obtained by analysing the 240 non-dominated fronts generated (8 algorithms $\cdot$ 30 repetitions = 240).
2. The sets of non-dominated points are normalized (min–max feature scaling).
3. Based on the reference point, the hyper-volume of each non-dominated fronts is calculated.
4. Each approach is assigned the average hyper-volume.

By analysing the results of the hyper-volume reported in Table 6, it can be determined that simheuristics approaches with buffers outperform the approaches without them. In addition, it is clear that the dynamic buffer management strategy improves the results obtained by the static strategy. Moreover, Table 6 shows that the interquartile range (IQR) of the approaches with buffers times is lower than the approaches without them. This indicates greater robustness in the results obtained by these approaches. It can be concluded that the NSGA-II-DB outperforms the rest of approaches in terms of hyper-volume. This corroborates the results discussed in Sections 5.3–5.4.

6. Conclusions

In this work, we have tackled the Bulk Berth Allocation Problem (Bulk-BAP) with stochastic vessels' arrival and handling times by means of proactive approaches and buffer time management strategies. The proposed approaches aim at increasing the robustness of the provided solutions when unplanned disrupting events occur in order to minimize the impact on the port system. On the other hand, the incorporation of buffers is aimed to absorb and increase preparedness in case stochastic events happen. Different simulation–optimization schemes were proposed. Namely, (i) an adaptation to the Bulk-BAP of the simheuristic standard scheme as depicted in Juan et al. (2015), (ii) a novel integrated scheme that considers the stochastic related information during the metaheuristic search, and (iii) a multi-objective approach based on NSGA-II for jointly optimizing both the deterministic and stochastic problem.

Extensive computational results were conducted to analyse the impact of stochastic changes on the Bulk-BAP as well as the contribution of the solution approaches on that matter. In this regard, the integrated scheme, where both objectives (i.e., deterministic and stochastic) are used to guide the search, provides better solutions than the standard simheuristic. Moreover, the multi-objective scheme using NSGA-II provides Pareto fronts dominating the majority of the single-objective approaches' solutions. Finally, the benefit of incorporating buffers was shown. Solution approaches using them exhibited better performance with regards to penalty costs due to disruptions. On the other hand, the dynamic buffer time management strategy overcomes the static one, which suggests its application in real-world environments where the Bulk-BAP has to be solved.

As limitations of this work, the developed approaches only tackle the Bulk Berth Allocation Problem, that is, without considering other related operational problems such as quay crane scheduling problem, internal vehicles management, workforce scheduling, etc. On the other hand, the consideration of this problem incorporating other types of transportation units (e.g., containers) can lead to further insights with regards to the utilization of the proposed approaches. Finally, the proposed simheuristics considered the main two stochastic features used in the literature related to the BAP (e.g., arrival and handling times), however other features related to either the vessel or the port might be interesting to be considered (e.g., berth time-windows, early departure, temporarily forbidden berths, etc.).

As future work, the following research lines keep open:

- The study of incorporating reactive strategies on the proposed approaches to handle disruptions in real-time once they happen. On the other hand, we aim at incorporating forecasting components (e.g., Pani et al., 2014; Karasu et al., 2020).
- We also aim at studying the implications and contributions of our proposed approaches in other relevant logistics problems such as vehicle routing or resource allocation related problems.
- With the aim of minimizing computation times or allowing to tackle larger problems, it is intended to study parallel metaheuristics. Specifically, a future line of research is the study of the parallelization of multi-objective population-based metaheuristics under stochastic conditions. The parallelization of these algorithms is not trivial, so different models from the literature should be contemplated. This might involve parallelizing either the simulation of disruptive events or the embedded population-based approaches, e.g., master–slave, distributed, and cellular models. In multi-objective environments, the previous point requires the definition of the necessary mechanisms to build and evolve a single Pareto front in a distributed way.

CRediT authorship contribution statement

Alan Dávila de León: Conceptualization, Methodology, Software, Validation, Formal analysis, Data curation, Writing - original draft, Writing - review & editing, Investigation, Validation, Visualization. **Eduardo Lalla-Ruiz:** Conceptualization, Methodology, Formal analysis, Writing - original draft, Writing - review & editing, Investigation, Validation, Visualization, Supervision. **Belén Melián-Batista:** Conceptualization, Methodology, Writing - review & editing, Supervision. **J. Marcos Moreno-Vega:** Conceptualization, Methodology, Writing - review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work has been partially supported by the Ministerio de Economía y Competitividad, Spain with FEDER funds (project TIN2015-70226-R) and by the Agencia Estatal de Investigación (Spain) (project PID2019-104410RB-I00/AEI/10.13039/501100011033).

Appendix. Acronyms

See Table A.7.

Table A.7
List of acronyms.

ALNS	Adaptive Large Neighbourhood Search
ATA	Actual time of arrival
BACAP	Berth Allocation and Quay Crane Assignment Problem
BAP	Berth Allocation Problem
Bulk-BAP	Bulk Berth Allocation Problem
Basic-GA	Basic Genetic Algorithm
DP	Deterministic problem
ENS_BS	Efficient non-dominated sort using a binary search
ETA	Estimated time of arrival
FCFS	First-Come First-Served
GA	Genetic Algorithm
GA($f(s)$)	Single-objective GA with simulation non-integrated into the search process seeking to minimize $f(s)$
GA($h(s)$)	Single-objective GA with simulation non-integrated into the search process seeking to minimize $h(s)$
GA-Sim	Single-objective GA with simulation integrated into the search process
GA-Sim($f(s)$)	Single-objective GA with simulation integrated into the search process seeking to minimize $f(s)$
GA-Sim($h(s)$)	Single-objective GA with simulation integrated into the search process seeking to minimize $h(s)$
GRA	Greedy Randomized Algorithm
HV	Hyper-volume
LNS	Large Neighbourhood Search
MAE	Mean absolute error
MILP	Mixed-integer linear programme
NGA	GA that considers $f(s)$ and $h(s)$ at the same time by normalizing and summing them up
NGA-B	NGA with buffers
NGA-DB	NGA-B where a dynamic buffer management strategy is applied
NGA-SB	NGA-B where a static buffer management strategy is applied
NSGA-II	Non-Dominated Sorting Genetic Algorithm II
NSGA-II($f\_h(s)$)	NSGA-II where the solution with the lowest normalized distance to the origin of coordinate are selected from the Pareto front
NSGA-II($f(s)$)	NSGA-II where the solution with the lowest objective value $f(s)$ are selected from the Pareto front
NSGA-II($h(s)$)	NSGA-II where the solution with the lowest objective value $h(s)$ are selected from the Pareto front
NSGA-II-B	NSGA-II with buffers
NSGA-II-DB	NSGA-II-B where a dynamic buffer management strategy is applied
NSGA-II-SB	NSGA-II-B where a static buffer management strategy is applied
NSGA-II-DB($f\_h(s)$)	NSGA-II-DB where the solution with the lowest normalized distance to the origin of coordinate are selected from the Pareto front
NSGA-II-DB($f(s)$)	NSGA-II-DB where the solution with the lowest objective value $f(s)$ are selected from the Pareto front
NSGA-II-DB($h(s)$)	NSGA-II-DB where the solution with the lowest objective value $h(s)$ are selected from the Pareto front
NSGA-II-SB($f\_h(s)$)	NSGA-II-SB where the solution with the lowest normalized distance to the origin of coordinate are selected from the Pareto front
NSGA-II-SB($f(s)$)	NSGA-II-SB where the solution with the lowest objective value $f(s)$ are selected from the Pareto front
NSGA-II-SB($h(s)$)	NSGA-II-SB where the solution with the lowest objective value $h(s)$ are selected from the Pareto front
RAK	Ras Al Khaimah
RBAP	Robust Berth Allocation Problem
RLC	Restricted List of Candidates
SDFBAP	Dynamic and Flexible Berth Allocation Model with Stochastic Vessel Arrival Times
SP	Stochastic problem
SWO	Squeaky Wheel Optimization
UAE	United Arab Emirates

References

Bierwirth, C., Meisel, F., 2015. A follow-up survey of berth allocation and quay crane scheduling problems in container terminals. European J. Oper. Res. 244 (3), 675–689.

Chang, D., Jiang, Z., Yan, W., He, J., 2010. Integrating berth allocation and quay crane assignments. Transp. Res. E 46 (6), 975–990.

Cordeau, J.-F., Laporte, G., Legato, P., Moccia, L., 2005. Models and tabu search heuristics for the berth-allocation problem. Transp. Sci. 39 (4), 526–538.

Davis, L., 1991. Handbook of Genetic Algorithms. CumInCAD.

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: Nsga-II. IEEE Trans. Evol. Comput. 6 (2), 182–197.

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

Ernst, A.T., Oğuz, C., Singh, G., Taherkhani, G., 2017. Mathematical models for the berth allocation problem in dry bulk terminals. J. Sched. 20 (5), 459–473.

Ganji, S.S., Babazadeh, A., Arabshahi, N., 2010. Analysis of the continuous berth allocation problem in container ports using a genetic algorithm. J. Mar. Sci. Technol. 15 (4), 408–416.

Golias, M.M., 2011. A bi-objective berth allocation formulation to account for vessel handling time uncertainty. Marit. Econ. Logist. 13 (4), 419–441.

Golias, M., Portal, I., Konur, D., Kaisar, E., Kolomvos, G., 2014. Robust berth scheduling at marine container terminals via hierarchical optimization. Comput. Oper. Res. 41, 412–422.

Golias, M.M., Saharidis, G.K., Boile, M., Theofanis, S., Ierapetritou, M.G., 2009. The berth allocation problem: Optimizing vessel arrival time. Marit. Econ. Logist. 11 (4), 358–377.

Guldogan, E.U., Bulut, O., Tasgetiren, M.F., 2011. A dynamic berth allocation problem with priority considerations under stochastic nature. In: International Conference on Intelligent Computing. Springer, pp. 74–82.

Hansen, P., Oğuz, C., Mladenović, N., 2008. Variable neighborhood search for minimum cost berth allocation. European J. Oper. Res. 191 (3), 636–649.

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

Imai, A., Nagaiwa, K., Tat, C.W., 1997. Efficient planning of berth allocation for container terminals in Asia. J. Adv. Transp. 31 (1), 75–94.

Imai, A., Nishimura, E., Papadimitriou, S., 2001. The dynamic berth allocation problem for a container port. Transp. Res. B 35 (4), 401–417.

Imai, A., Sun, X., Nishimura, E., Papadimitriou, S., 2005. Berth allocation in a container port: Using a continuous location space approach. Transp. Res. B 39 (3), 199–221.

Imai, A., Zhang, J.-T., Nishimura, E., Papadimitriou, S., 2007. The berth allocation problem with service time and delay time objectives. Marit. Econ. Logist. 9 (4), 269–290.

Iris, Ç., Lam, J.S.L., 2019. Recoverable robustness in weekly berth and quay crane planning. Transp. Res. B 122, 365–389.

Juan, A.A., Faulin, J., Grasman, S.E., Rabe, M., Figueira, G., 2015. A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. Oper. Res. Perspect. 2, 62–72.

Karafa, J., Golias, M.M., Ivey, S., Saharidis, G.K., Leonardos, N., 2013. The berth allocation problem with stochastic vessel handling times. Int. J. Adv. Manuf. Technol. 65 (1–4), 473–484.

Karasu, S., Altan, A., Bekiros, S., Ahmad, W., 2020. A new forecasting model with wrapper-based feature selection approach using multi-objective optimization technique for chaotic crude oil time series. Energy 212, 118750.

Knowles, J.D., Thiele, L., Zitzler, E., 2006. A tutorial on the performance assessment of stochastic multiobjective optimizers. TIK-Report 214, ETH Zurich.

Lalla-Ruiz, E., 2017. Intelligent Management of Seaside Logistic Operations At Maritime Container Terminals (Ph.D. thesis). Springer.

Lalla-Ruiz, E., Expósito-Izquierdo, C., Melián-Batista, B., Moreno-Vega, J.M., 2016. A set-partitioning-based model for the berth allocation problem under time-dependent limitations. European J. Oper. Res. 250 (3), 1001–1012.

Lalla-Ruiz, E., MeliáN-Batista, B., Moreno-Vega, J.M., 2012. Artificial intelligence hybrid heuristic based on tabu search for the dynamic berth allocation problem. Eng. Appl. Artif. Intell. 25 (6), 1132–1141.

Lee, D.-H., Chen, J.H., Cao, J.X., 2010. The continuous berth allocation problem: A greedy randomized adaptive search solution. Transp. Res. E 46 (6), 1017–1029.

de León, A.D., Lalla-Ruiz, E., Melián-Batista, B., Moreno-Vega, J.M., 2017. A machine learning-based system for berth scheduling at bulk terminals. Expert Syst. Appl. 87, 170–182.

Lu, Z.-q., Xi, L.-f., et al., 2010. A proactive approach for simultaneous berth and quay crane scheduling problem with stochastic arrival and handling time. European J. Oper. Res. 207 (3), 1327–1340.

Monaco, M.F., Sammarra, M., 2007. The berth allocation problem: A strong formulation solved by a Lagrangean approach. Transp. Sci. 41 (2), 265–280.

Moorthy, R., Teo, C.-P., 2007. Berth management in container terminal: The template design problem. In: Container Terminals and Cargo Systems. Springer, pp. 63–86.

Na, L., Zhihong, J., 2009. Optimization of continuous berth and quay crane allocation problem in seaport container terminal. In: 2009 Second International Conference on Intelligent Computation Technology and Automation, vol. 3. IEEE, pp. 229–233.

Notteboom, T.E., 2006. The time factor in liner shipping services. Marit. Econ. Logist. 8 (1), 19–39.

Pani, C., Fadda, P., Fancello, G., Frigau, L., Mola, F., 2014. A data mining approach to forecast late arrivals in a transhipment container terminal. Transport 29 (2), 175–184.

Park, Y.-M., Kim, K.H., 2005. A scheduling method for berth and quay cranes. In: Container Terminals and Automated Transport Systems. Springer, pp. 159–181.

Pratap, S., Nayak, A., Kumar, A., Cheikhrouhou, N., Tiwari, M.K., 2017. An integrated decision support system for berth and ship unloader allocation in bulk material handling port. Comput. Ind. Eng. 106, 386–399.

Resende, M.G., Ribeiro, C.C., 2010. Greedy randomized adaptive search procedures: Advances, hybridizations, and applications. In: Handbook of Metaheuristics. Springer, pp. 283–319.

Ribeiro, G.M., Mauri, G.R., de Castro Beluco, S., Lorena, L.A.N., Laporte, G., 2016. Berth allocation in an ore terminal with demurrage, despatch and maintenance. Comput. Ind. Eng. 96, 8–15.

Robenek, T., Umang, N., Bierlaire, M., Ropke, S., 2014. A branch-and-price algorithm to solve the integrated berth allocation and yard assignment problem in bulk ports. European J. Oper. Res. 235 (2), 399–411.

Rodriguez-Molins, M., Ingolotti, L., Barber, F., Salido, M.A., Sierra, M.R., Puente, J., 2014. A genetic algorithm for robust berth allocation and quay crane assignment. Prog. Artif. Intell. 2 (4), 177–192.

Schepler, X., Absi, N., Feillet, D., Sanlaville, E., 2019. The stochastic discrete berth allocation problem. EURO J. Transp. Logist. 8 (4), 363–396.

Shang, X.T., Cao, J.X., Ren, J., 2016. A robust optimization approach to the integrated berth allocation and quay crane assignment problem. Transp. Res. E 94, 44–65.

Shaw, P., 1998. Using constraint programming and local search methods to solve vehicle routing problems. In: International Conference on Principles and Practice of Constraint Programming. Springer, pp. 417–431.

Theofanis, S., Boile, M., Golias, M., 2009. Container terminal berth planning: Critical review of research approaches and practical challenges. Transp. Res. Rec. 2100 (1), 22–28.

Umang, N., Bierlaire, M., Erera, A.L., 2017. Real-time management of berth allocation with stochastic arrival and handling times. J. Sched. 20 (1), 67–83.

Umang, N., Bierlaire, M., Vacca, I., 2013. Exact and heuristic methods to solve the berth allocation problem in bulk ports. Transp. Res. E 54, 14–31.

Ursavas, E., Zhu, S.X., 2016. Optimal policies for the berth allocation problem under stochastic nature. European J. Oper. Res. 255 (2), 380–387.

Vernimmen, B., Dullaert, W., Engelen, S., 2007. Schedule unreliability in liner shipping: Origins and consequences for the hinterland supply chain. Marit. Econ. Logist. 9 (3), 193–213.

While, L., Bradstreet, L., Barone, L., 2011. A fast way of calculating exact hypervolumes. IEEE Trans. Evol. Comput. 16 (1), 86–95.

Xu, Y., Chen, Q., Quan, X., 2012a. Robust berth scheduling with uncertain vessel delay and handling time. Ann. Oper. Res. 192 (1), 123–140.

Xu, D., Li, C.-L., Leung, J.Y.-T., 2012b. Berth allocation with time-dependent physical limitations on vessels. European J. Oper. Res. 216 (1), 47–56.

Yan, S., Lu, C.-C., Hsieh, J.-H., Lin, H.-C., 2019. A dynamic and flexible berth allocation model with stochastic vessel arrival times. Netw. Spat. Econ. 19 (3), 903–927.

Yu, J., Tang, G., Song, X., Yu, X., Qi, Y., Li, D., Zhang, Y., 2018. Ship arrival prediction and its value on daily container terminal operation. Ocean Eng. 157, 73–86.

Zhang, X., Tian, Y., Cheng, R., Jin, Y., 2014. An efficient approach to nondominated sorting for evolutionary multiobjective optimization. IEEE Trans. Evol. Comput. 19 (2), 201–213.

Zhen, L., 2015. Tactical berth allocation under uncertainty. European J. Oper. Res. 247 (3), 928–944.

Zhen, L., Chang, D.-F., 2012. A bi-objective model for robust berth allocation scheduling. Comput. Ind. Eng. 63 (1), 262–273.

Zhou, P.-f., Kang, H.-g., 2008. Study on berth and quay-crane allocation under stochastic environments in container terminal. Syst. Eng.-Theory Pract. 28 (1), 161–169.

Zitzler, E., Thiele, L., 1999. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. IEEE Trans. Evol. Comput. 3 (4), 257–271.