Berth planning and real-time disruption recovery
a simulation study for a tidal port

van der Steeg, Jaap Jan; Oudshoorn, Menno; Yorke-Smith, Neil

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Berth planning and real-time disruption recovery: a simulation study for a tidal port

Jaap-Jan van der Steeg[1,2] · Menno Oudshoorn[2] · Neil Yorke-Smith[1]

Abstract

With the increasing volume of container freight transport, future port planning is crucial. Simulation models provide a means to gain insight in the effects of terminal expansions. Detailed simulations incorporate berth allocation: assigning vessels a time and location at the quay wall, where the vessel is loaded and unloaded. This article develops decision models for both offline preliminary berth planning and for online recovery of this plan during simulation. First, we develop an optimisation-based approach that incorporates realistic aspects—cyclic vessel arrivals, tidal windows, and minimisation of vessel draught during low water periods—in order to develop a cyclic baseline berth allocation plan. The approach can proactively incorporate slack for increased robustness. Exploiting a constraint-based solver, we can obtain optimal or satisficing solutions for a year's operation of a large port. The resulting preliminary berth plan is used as a basis for the arrival times. However, disruptions can occur, such as vessel arrival and loading times varying from the planned. Hence, second, we develop a real-time disruption management decision model. This multi-level heuristic approach reacts to disruptions while minimising perturbation of the original berth plan. Computational experiments with a high-resolution simulator show our recovery approach finds good solutions until a tipping point of disturbance. Results also show that when the expected occupation of a terminal is higher, strengthening robustness of the preliminary plan has increased importance. The approach described in the article is implemented for a major European inland tidal port, forming the basis of a simulation-based decision support tool for operational planning and exploring port expansion options.

Keywords Decision support · Simulation · Port expansion · Berth allocation · Disruption management · Constraint programming

✉ Neil Yorke-Smith
   n.yorke-smith@tudelft.nl

Extended author information available on the last page of the article

1 Introduction

With the increasing volume of container freight transport and the increasing size of container vessels, port planning is crucial. Simulation models provide a means to gain insights into operational plans and future port expansions (Benedicto et al. 2018; Clausen and Kaffka 2016). This article is motivated by analysing a planned new terminal for an existing European port, for which a simulation study is required by the port operator.

In order to give insight into capacity and operational dimensions of the new terminal, detailed study of vessel berthing and processing is needed. Two of the processes to be simulated are *vessel arrivals* and *berth allocation*. Realistic factors must be captured in the study. These include simulating a port that has tide-restricted entrance and exit windows, and simulating the way many container shipping companies and terminals work: with cyclically-calling vessels from 'loops' (Hendriks et al. 2013).

First, this article contributes a novel simulation-informed optimisation approach incorporating vessel arrival generation and berth allocation in a simulated environment. Our approach focuses on the generation of cyclic baseline planning on which the vessel arrival times and the actual berth allocation are based (Iris et al. 2018). By creating this baseline planning one can incorporate both the tidal windows of vessels and the cyclic nature of realistic container vessel arrivals. Unlike works in the literature that address the strategic berth planning problem, we face a situation where the arrival times of vessels are not known but are generated in the simulation; only the number of arrivals per vessel class is known. Further, tidal windows and vessel loops must be incorporated.

At a technical level, our approach is to derive a berth planning template as a nonlinear mathematical model. We use the optimisation technique of constraint programming (CP) to automatically derive specific berth allocation plans. Empirical study by means of simulation shows that a CP approach is feasible in practice.

Second, we treat the *real-time dynamics* of variation from the baseline plan through a multi-level heuristic approach which reacts to disruptions while minimising perturbation of the original berth plan. Disruptions include vessel arriving early or late, or their loading and unloading varying from expected. Our approach has the property of responding quickly and also, given sufficient time, repairing the baseline plan while minimising perturbation using a modified CP model (Fukunaga 2009). Empirical study by means of simulation show the recovery approach finds good solutions until a tipping point of disturbance. Results also show that when the expected occupation of a terminal is higher, strengthening robustness of the preliminary plan has increased importance.

Third, we employ our approach for a case study of a large tide-dependent river delta port in Europe.[1] We study both existing terminals and a planned new terminal. As terminal occupancy increases we examine the effect of busyness on the runtime of the CP

---

[1] For reasons of commercial confidentiality, the name of the port cannot be disclosed.

```
┌─────────────────────────────────────┐
│   1. Determine design parameters,    │
│       including simulation period    │
└─────────────────────────────────────┘
                   │
                   ▼
┌─────────────────────────────────────┐
│ 2. Generate vessels for the simulation period │
└─────────────────────────────────────┘
                   │
                   ▼
┌─────────────────────────────────────┐
│ 3. Perform initial berth allocation planning │
│               (Section 4)            │
└─────────────────────────────────────┘
                   │
                   ▼
┌─────────────────────────────────────┐
│   4. Execute berthing plan, updat-   │
│   ing it in (simulated) real time    │
│               (Section 5)            │
└─────────────────────────────────────┘
```
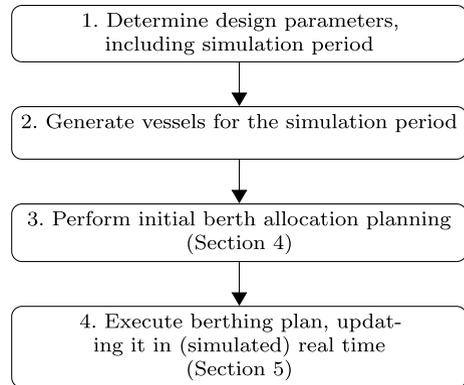
model using a state-of-the-art solver, and show that we can either obtain the optimal berth plan or a good approximation of it, within one hour for a large terminal. Plan repair can be achieved within a few minutes of computation time.

Figure 1 visualises the high level of the proposed simulation-informed optimisation approach. The simulator (Steps 1 and 2) existed previously. It is used for the evaluation of the initial berthing plan (Step 3) and in a feedback loop during simulated planning-and-execution (Step 4). The approach can also be used effectively with live data, in which case the input from the simulation to the optimisation is replaced by real-time data feeds.

The work reported in this article is implemented as part of a bespoke commercial simulation platform in use for decision support at multiple container ports worldwide. We provide managerial insights for the capacity planning of terminals and the robustness of operational plans to uncertainty.

Summarising, the contribution over the state-of-the-art is as follows:

1. An integrated optimisation-based approach to cyclic berth planning accounting for tidal windows.
2. A perturbation-minimising approach to rescheduling to adjust to variation from the planned baseline.
3. A case study of our optimisation approach for a major European inland port for one whole year.

The remainder of the article is structured as follows. The next section surveys related literature. Section 3 identifies the specific problem treated. Sections 4 and 5 describe the simulation-informed optimisation approaches developed for planning and for re-planning. Section 6 presents an analysis on data from the case study port. Section 7 summarises the managerial implications and discusses future directions.

2 Background and related work

2.1 Container terminal simulation

Port simulation is an established technique for analysing, optimising and planning container terminals (Dragović et al. 2017), of value for berth planning, quay crane scheduling, yard scheduling, and other problems in practice.

In this article we adopt simulation-based optimisation in order to analyse a planned terminal expansion. Simulation-based optimisation typically blends an optimisation approach to a decision problem with a simulation model that incorporates uncertainty. The paradigm has diverse use (Zúñiga et al. 2020), including long-term planning (Turan et al. 2020). Castilla-Rodríguez et al. (2020), for instance, approach the management of trans-shipment operations through simulation–optimisation, resulting in robust schedules for quay cranes.

This article addresses vessel arrival generation for a terminal simulation, provides a simulation-informed optimisation approach to berth allocation with realistic factors, and further provides a means to handling uncertainty through re-planning. We say 'simulation-informed optimisation' rather than 'simulation–optimisation' because the simulator is not invoked by the planner, rather the real-time output of the simulator is fed into the planner, and the planner's decisions are then enacted in the simulator. Thus the two components interact in a feedback loop. Hence in this section we briefly survey relevant work in arrival generation, berth allocation, uncertainty, and constraint programming.

2.2 Arrival generation

An important design choice that impacts the simulation process is the generation of vessel arrivals. This design decision influences everything from the busyness in the port's entrance waters to the allocation of berths and other terminal-related planning. Simulation designers largely adopt one of two approaches: either follow a random arrival process or derive the arrival process from historical data Examples of the first approach are Pachakis and Kiremidjian (2003), Thiers and Janssens (1998), Yeo et al. (2007).

In the second approach for arrival generation, when historical data is available and is relevant, simulation models can use it as a basis for vessel arrival generation (Hassan 1993). Huang et al. (2013), for instance, use the actual arrival historical times in the simulation. This is valid if the historical arrival times are representative of the simulation period; in particular, this assumption can be false when simulating future design options for the port. Cimpeanu et al. (2017) assume that a yearly schedule for vessel arrivals are known beforehand. These are based on distributions of inter-arrival times derived from historical data provided by a port terminal company. Van Asperen et al.(2003) examine arrival generation for jetty facilities. The data provided in their case was yearly arrival numbers for the vessels, and not information about any kind of specific distributions.

Tidal patterns have been included in some simulation models. The early work of Groenveld (1983) included tidal conditions post hoc as a check whether a vessel is allowed to sail or not; a vessel that is not allowed to sail due to tidal restrictions needs to wait in the anchorage. Similarly, both Hassan (1993) and Cimpeanu et al. (2017) keep vessels restricted by tidal conditions waiting in a queue. Wadhwa (1992) also takes tidal patterns into account when entering and exiting the port. Here, the notion of loading a vessel according to a certain allowable draught is introduced. Thiers and Janssens (1998) divide vessels into tide-dependent and not tide-dependent. For vessels that are tide-dependent there are certain threshold locations which the vessel can only pass during a certain time-frame. These tidal windows are defined as the time frame in which the vessel can depart from a certain location and pass all the thresholds without issues.

With the exception of Huang et al. (2013), all the works surveyed generate vessels according to inter-arrival times. In cases where historical data is provided, the inter-arrival times are often based on data; on the other hand, if no data is available, different random arrival processes are incorporated. None of the models have included the cyclic nature of arrivals and only some of them included tidal patterns. The latter, however, are modelled as a check whether or not a vessel is allowed to enter the port or not, rather being taken into account in the arrival generation. By contrast, in reality shipping companies do attempt to take tides into account to reduce waiting time.

2.3 Berth allocation

Successful berth planning has major impact on key performance indicators for a terminal (Ding et al. 2016). Berth allocation is the process of allocating terminal resources to vessels: assigning vessels a time and location at the quay wall where the vessel is loaded and unloaded. While berth allocation has received extensive attention in the literature (Wawrzyniak et al. (2020) being a recent example), to the best of our knowledge, no work uses a baseline berth allocation as a base for a simulation model. Table 1 overviews the literature which we summarise next.

The creation of a baseline berth planning itself is found in the literature: for a thorough overview we refer to Bierwirth and Meisel (2015). However, one important difference between the berth allocation papers in the literature and actually using the allocation as a base for a simulation: in the former the expected arrival time and expected service time are usually assumed to be known.

In the most generic case of berth allocation, one tries to find an allocation that plans vessels as close to their expected arrival time as possible. Often also some preferred berthing location of a vessel is taken into account. The objective function optimised in real life is a combination of maximising such preferences, maximising the vessel service levels (i.e., minimise the departure lateness) and minimise operating costs during a planning horizon (Ding et al. 2016).

As noted in the introduction, many container shipping companies and terminals operate cyclically-calling vessels from 'loops' (Hendriks et al. 2013). A loop can be seen as a series of ports that are visited in order. Because of the size of these loops,

Table 1 Selected berth allocation literature overview

Paper	Baseline planning	Dynamic arrival times	Dynamic service times	Water depths	Tidal windows	Cyclic loops	Multi-objective	Quay crane planning	Approach	Other aspects
Ding et al. (2016)	y	–	–	y	y	–	y	y	Heur. + LP	Deployed
Hendriks et al. (2013)	–	–	–	–	–	y	–	y	Heur.	–
Zhen et al. (2011)	–	–	y	–	–	y	–	y	Heur.	–
Imai et al. (2014)	y	–	–	–	–	y	–	–	IP	Strategic
Iris et al. (2018)	y	–	–	–	–	y	–	–	Set packing	
Qin et al. (2016)	–	–	–	y	y	–	–	–	CP	–
Song et al. (2019)	–	–	–	y	–	–	–	–	EA	–
Zhen et al. (2017)	–	–	–	y	y	–	–	y	IP	–
Xiang and Liu (2021)	–	y	y	–	–	–	y	y	Robust opt.	–
Imai et al. (2001)	–	–	–	–	–	–	–	–	Heur.	–
Cordeau et al. (2005)	–	–	–	–	–	–	y	–	Various	–
Nishi et al. (2020)	–	–	y	–	–	–	–	–	DP	–
Zhen (2013)	–	–	y	–	–	–	–	–	Heur.	Stochastic
Schepler et al. (2019)	–	y	–	–	–	–	–	–	Various	Stochastic
Zampelli et al. (2013)	–	–	–	–	–	–	y	y	CP	Side constraints
This work	y	y	y	y	y	y	–	–	CP	Deployed

container shipping companies usually have multiple vessels sailing in the same loop. In reality, a terminal plans the arrivals of these loops using a cyclic baseline berth planning: a schedule for, for example, a week, in which every loop receives a certain time when and location where it is expected to arrive each week. This baseline plan is modified during the week's execution because, due to external factors and uncertainties, vessels often encounter deviation from the baseline plan.

Hendriks et al. (2013) are one of the few authors who incorporate the cyclic nature of realistic berth planning into a model. They divide a cyclic period into discrete time sub-periods. Similarly, Zhen et al. (2011) also divide the planning horizon into discrete time periods.

Unlike these, and most, works which focus on the tactical or operational aspects to berth planning, Imai et al. (2014) consider a strategic dimension to berth allocation. These authors select a subset of the possible vessels for berthing, and then schedule them within a planning horizon. Notably, Imai et al. also accommodate cyclically-calling vessels. Iris et al. (2018) explore integer programming and set packing approaches for the same problem, concluding that the latter has best performance. We adopt the same approach of a cyclic-like baseline plan, but in a simulation-informed optimisation paradigm. Two further important differences from work on the strategic berth template problem are that, first, in a simulation setting the arrival times of vessels are not known, and second, tidal windows are crucial, as we next discuss.

Other authors treat another aspect of berth planning which is important for our contribution: time-varying water constraints. Qin et al. (2016) consider berth allocation with time-varying water depth at a tidal river port. As we discuss further below, these authors characterise situations when a constraint programming approach is superior to an integer programming approach. Zhen et al. (2017) solve berth allocation and quay crane scheduling considering tides and channel flow control constraints. Xiang and Liu (2021) also solve integrated berth allocation and quay crane scheduling, with a robust optimisation approach.

These works all follow the assumption outlined by Hendriks et al. (2012): namely that strategic-level decisions are made (or have been made) about the time of arrival and departure, and these decisions are used to create the (cyclic) berth allocations. However, since these times are not known in a simulated environment and need to be generated from yearly arrival numbers, these models cannot be used directly in our case. Instead, the same mindset of rectangle packing on a cylinder instead of a plane, as explained by Zhen et al. (2011), is adopted in our work, with the modification that we disallow vessels straddling the boundary of the cyclic periods.

2.4 Dynamicism and uncertainty

In the literature, 'dynamic' berth planning more commonly has the meaning that the ships have individual arrival times over the planning horizon. This contrasts to 'static' berth planning where all ships are assumed to already be present at the port at the time of the planning (Imai et al. 2001). Other authors define the 'dynamic' berth allocation problem to recognise that the service time of a vessel when berthed

may differ depending on the berth on the quayside at which it is serviced (Cordeau et al. 2005). Dynamic, then, in this line of the literature, refers to the service time depending on the location; it does not refer to dynamic in the sense of uncertainty in the arrival or service times per se, nor to change in which vessels are arriving.

As with the service-time-fixed 'static' berth allocation problem, the service-time-variable 'dynamic' version has received extensive attention in the literature. As a recent example, Nishi et al. (2020) provide an exact dynamic programming-based approach and analyse how it scales better than previous approaches.

Our approach to uncertainty fits with the simulation-based paradigm, as Sect. 3 describes. We treat the 'static' problem as defined above, while allowing uncertainty in vessel arrival and service times—and indeed re-planning in face of exogenous events.

There are stochastic approaches to berth planning and other terminal operation problems. Zhen (2013), for instance, admits uncertainty over berthing time and berthing positions. Schepler et al. (2019) consider stochastic arrival times of vessels. The authors find that a proactive–reactive approach, based on heuristic search and stochastic dynamic programming, works well when uncertainty is limited; a purely reactive approach dominates when uncertainty is larger.

Such types of modelling approaches have the advantage of reasoning with uncertainty, and the disadvantage compared to simulation–optimisation of more complex mathematical models and limitations on the type of uncertainty or events that can be modelled (Liu et al. 2020).

2.5 Disruption recovery

Recognition of uncertainty is necessary for berth allocation in practice, since berth allocation must be feasible operationally real time. While a berth plan which proactively incorporates robustness is valuable, the capability to react to schedule deviations and exogenous events is necessary. A plan that is fully robust to foreseen disruptions is too inefficient; and we cannot rule out that unforeseen disruptions (Lv et al. 2020).

The first type of reactive approach is *full re-planning*: creating a completely new schedule while optimising the original performance measure. Carette (2016) addresses the berth allocation problem with this idea. Here, the optimal plan is recalculated at given time intervals. At each interval, new information is collected and based on this information an optimisation method is used to create entirely new schedule.

The second type of reactive approach is *disruption management*: adjusting the original berthing plan to accommodate disruptions while keeping as close as possible to the original berthing plan. Its advantage over full re-planning is to minimise actual relocation of equipment and resources which have been prepared according to the original plan. Further, in cases where there are many incoming vessels an updated plan can usually be achieved faster. Zhang et al. (2016), for instance, introduce a disruption management strategy based on a lexicographic optimisation of vessels according to their type.

Yang et al. (2016) simulate disruptions and based on this extract stable states for vessels. Based on these stable states, a rescheduling processes aims to influence vessels' schedules as little as possible. Similarly, Zeng et al. (2011) find an optimal berth allocation through repeated simulation runs and solving disruptions that arise; the authors treat simultaneously allocation and quay crane scheduling. The performance measure is the familiar two-fold measure: on one hand the original optimisation function must be optimised and on the other hand the deviation from the original problem must be minimised. The authors propose the idea of local rescheduling which only considers a limited time and space window to solve the disruption, and find local rescheduling is more efficient than a full rescheduling approach.

Iris and Lam (2019) assume a finite set of scenarios that give the value of different uncertain variables, adopting the robust optimisation paradigm. For the baseline planning a vessel specific buffer is added which acts as robustness. The authors also generate recovery plans for each scenario: for each, a plan is created which subsequently are used to create the baseline plan. Schepler et al. (2019), noted earlier, assume known probability distributions capture uncertainty about arrival times. They present a two-stage stochastic programming model for berth allocation planning, and a stochastic dynamic programming approach for disruption recovery.

Jia et al. (2020) study combined berth allocation of deep-sea vessels and arrival scheduling of other vessels. Focusing on a case study where the number of feeder vessels is much larger than the number of deep-sea vessels, the authors seek to minimise the departure delays of deep-sea vessels and the schedule displacements of feeders. A three-phase simulation–optimisation method handles uncertainty in the service times of the feeders.

The works surveyed are not real-time recovery approaches. Rather they aim at optimising the schedule given a series of simulation runs and corresponding disruptions. By contrast, Umang et al. (2017) present a real-time approach for disruption recovery in a bulk port. The authors assume a baseline schedule and aim to stay as close to that as possible when disruption happen. During a run, the model expects vessels to update their arrival time and handling time. Each hour, a schedule recovery is attempted using the updated values. The authors consider rescheduling in one of two ways: a greedy algorithm and an optimisation algorithm. The greedy algorithm places vessels at berthing locations where the total cost of reassignment is minimal. The optimisation algorithm minimises the objective function through reformulating the optimisation function as a set-partitioning problem where all the feasible assignments are investigated. The authors find that the optimisation algorithm out-performs the others based on objective value while the greedy algorithm stays closer to the original baseline plan.

Compared with Umang et al. (2017), our disruption management approach is more general and more scaleable. First, the simulation model for our case study port is large—one of the largest ports in Europe—and has simulation period of at least one month. Second, Umang et al. only examine test scenarios of 10–25 vessels and a time horizon of 5 days, and only update their planning each hour. We respond to disruptions in near real time: in the order of minutes per vessel. Lastly, Umang et al. assume fixed (hybrid) berthing positions whereas we allow continuous quays for which vessels can be berthed anywhere along the quay wall.

2.6 Constraint programming

Constraint programming (CP) is a flexible artificial intelligence approach to modelling and solving combinatorial optimisation problems (Pesant 2014). CP can be viewed as a generalisation of integer programming (IP).

In contrast to IP, the origins of CP are in artificial intelligence (Pesant 2014; Puget and Shaw 2014). The strengths of CP are expressive modelling flexibility, which yields more compact and understandable declarative models than IP. Further, CP allows us to decouple the modelling from the solving (Freuder 1997; Wallace 2020).

Kizilay et al. (2018), for instance, use CP to address a decision problem in integrated port container terminal operations. Li et al. (2014) use CP to plan vessel rotations (loops) among terminals within a port. In solving a berth allocation problem, Qin et al. (2016) show that CP outperforms integer programming in three instances, namely when using dynamic arrivals over static arrivals, when using fine time granularity, and when the restriction of the objective towards the decision variables is not tight. Closest to our work, Zampelli et al. (2013) use CP to address a combination of berth and crane allocation. Although accommodating many realistic operational constraints and multiple objectives, they do not consider cyclic vessel loops, tidal windows, or dynamicism.

Since the optimisation problem we will formulate has no tight restriction towards the decision variables—only that their values be close to preferred locations—and uses a very fine time granularity of 1 min, we adopt CP and formulate a compact and understandable optimisation model in Sect. 4.

3 Berth planning problem

Having surveyed the literature, in this section we identify the specific problem treated. It has two aspects: planning and re-planning.

First, we address the creation of a cyclic baseline berth allocation of vessel arrivals in a simulation model. The allocation must respect physical and operational constraints. It must allow for tidal windows, and operator/terminal preferences on berthing locations.

This baseline berth planning serves as a basis for the vessel arrival times in the simulation, as follows. Based on provided data about expected vessel arrivals in the port, vessels will be generated for the simulation model. Each of these vessels will be allocated a berth at a certain time during the simulation. Thus each vessel will be assigned the combination of a certain time and a certain position at the quay wall, where the vessel can be loaded and unloaded without interfering with other vessels. Based on the time of this allocation, the actual time the vessel will arrive in the simulation can be decided. That is, due to simulation stochastics, the arrival time of a vessel when the simulation executes can (deliberately, from the point of view of the simulation designer) differ from the planned arrival time.

In this way we treat uncertainty by means of the simulation paradigm rather than through formulating, for instance, a stochastic programming optimisation model. We

also note the ability of simulation to capture behavioural factors, although this is not the focus of the current paper.

Second, we address re-planning of the berth allocation in reaction to dynamic changes to vessel arrival times and orders, loading and unloading times, equipment breakdown, and other exogenous factors. The recovery approach must operate fast enough to be use in real-time operations, and, preferably, minimise perturbation from the baseline plan.

In order to develop a realistic data-based simulation, we take advantage of the following commonly-available data.

1. *A yearly forecast for the simulation period.* That is, for each class of container vessels, the port manager provides an expected yearly number of arrivals. To make these arrivals more realistic, the vessel classes can be sub-divided by the different approach routes to the port. That is, the different 'paths' from shipping lanes into the port entrance; not all vessels can traverse all approach routes, for instance due to water depth in a channel. What vessels belong to what vessel classes can be specific to the application. In the case study of Sect. 6, vessel classes are defined by their size, amount of Twenty-foot Equivalent Unit (TEU) that can be carried, and the handling time distributions. Hence each vessel from the same class has the same size, TEU and handling time distribution.
2. *The division of draught for each class.* That is, how many vessels are expected yearly with what draught. The draught of a vessel is essential knowledge since it determines whether or not a vessel can sail given the tidal situation.
3. *The tidal windows for each draught.* These are required in a similar fashion as in Thiers and Janssens (1998). The tidal window at a certain location represents the first and last moment a vessel with a specific draught is allowed to pass that location, such that it will not encounter any tidal issues over the course of the journey to or from the port. To make a good estimation about the berthing times after a vessel can pass through the tidal window, data about travel times in and around the port are used.
4. *Additional information for each vessel class.* Regarding, for instance, the length of the vessel and the distribution of the handling time for that class.

Note that we do not rely on the existence of (accurate) historical data about vessel arrivals.

For the subsequent parts of a port simulation, in which berth planning is incorporated, more data could be necessary; for example, information about traffic rules, tug boat assistance and external conditions like weather. We discuss further in Sect. 7 at the conclusion of the article.

Fig. 2 Six steps of the berth allocation approach

```
┌─────────────────────────────┐
│  1. Determine simulation    │
│     period and cyclic period│
└─────────────────────────────┘
              │
              ▼
┌─────────────────────────────────────┐
│ 2. Generate vessels for the simulation period │
└─────────────────────────────────────┘
              │
              ▼
┌─────────────────────────────────┐
│ 3. Divide arrivals over cyclic periods │
└─────────────────────────────────┘
              │
              ▼
┌─────────────────────────────────┐
│ 4. Deduce arrivals in projection plan │
└─────────────────────────────────┘
              │
              ▼
┌─────────────────────────────┐
│ 5. Solve the projection plan │
└─────────────────────────────┘
              │
              ▼
┌─────────────────────────────┐
│ 6. Plan each cyclic period  │
└─────────────────────────────┘
```
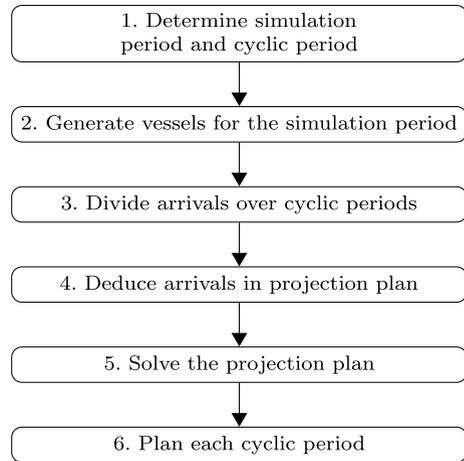
4 Optimisation approach: planning

Our approach is to derive a berth planning template as a non-linear mathematical model. This approach integrates all the required aspects of the problem into a single model. We use the optimisation technique of constraint programming to automatically derive and to dynamically update specific berth allocation plans from the model.

The six steps for the planning are illustrated in Fig. 2. We describe each in turn. In the subsequent Sect. 5 we describe the re-planning.

4.1 Cyclic berth planning

Step 1: Determine simulation period and cyclic period. Any simulation design requires some foundational decisions regarding the simulated period and the cyclic loop period. The decision for the simulation period is important as upon this depend the tidal windows. The cyclic period determines how often the cyclical berth allocation pattern repeats itself. In practice this period is usually 7, 10 or 14 days (Hendriks et al. 2013). The combination of this simulation period and cyclic period determine how often the cyclic pattern is repeated. The cycle period can be chosen to the user's needs. Note that such decisions on the simulation are not part of the planning approach; rather, the planning depends upon them.

Step 2: Generate vessels for the simulation period. The second step concerns the generation of all the vessels that will arrive in the simulated period. This is achieved through a repeated random selection with stratification. A list of vessel arrivals is generated that represents the total amount of arrivals that is expected in the simulation period, according to the provided number of yearly arrivals for each class and draught within that class.

Step 3: Divide arrivals over cyclic periods. We treat the cyclic nature of the arrivals with the assumption that vessels from the same vessel class can belong to the

same loop. Consequently, for each instance where a vessel class has arrivals in every cyclic period we may assume there is one loop for that vessel class. In this step, then, the division of vessels over the cyclic periods is made. The goal is to divide the arrivals from each vessel class as evenly as possible over the cyclic periods.

For example, suppose that the cyclic period is one week and the simulation period covers five weeks, then the vessel arrivals are divided over five periods. Now, if from a certain class 13 arrivals are expected in the simulation period we will assign two arrivals to each week and these will be considered cyclic arrivals. The remaining three arrivals that cannot cover each cyclic period will be assigned to random periods where the same period is not chosen twice. These remaining arrivals are considered non-cyclic arrivals.

Step 4: Deduce arrivals in projection plan. Now that the arrivals of each class per cyclic period is known, the next step is to deduce a projection plan. A projection plan is a berth allocation for one period on which the arrivals for each cyclic period are projected.

For example, if we know that from a certain loop each week two vessels will come to the port, we need two spaces in the berth plan where these vessels can arrive. By doing this, the actual arrivals can be projected on this plan since there is a space where they fit.

The arrivals of the projection plan can be deduced as follows. First, for each cyclic loop that has an arrival each period, one vessel from that class is added to the projection plan. These loops will be planned at the same time and place each period. Second, since the projection plan is the berthing plan on which every period's arrivals will be projected, room for non-cyclic arrivals needs to be incorporated as well. This is done by adding the number of non-cyclic vessels of the period that has the most non-cyclic arrivals. This way, the projection plan will have enough planned spaces such that each period can fit all their planned arrivals.

Step 5: Solve the projection plan. In this step the goal is to reserve a time slot and location for all the vessel class arrivals that are expected in the projection period. As explained earlier, the arrivals in the projection period are not actual arrivals but placeholders on which the actual arrivals for each period will be projected. This means that a decision needs to be made on how much space and how much time needs to be reserved for each arrival. We treat the cases of cyclic and non-cyclic arrivals separately.

First, for the cyclic arrivals this means that each arrival is from the same class which means that the vessel length and estimated handling time are likely rather similar. In any case, for each of the cyclic arrivals the longest vessel length and the longest estimated handling time for that vessel class needs to be reserved. In our case study, vessels from the same class all have the same vessel length and handling time distribution, which means that this length and expected handling time from this class can be used. This is to make sure that when planning the actual arrivals and projecting it on this space that the arrival will stay within the reserved boundaries.

Second, the non-cyclic arrivals are more difficult, since the same spot can be used by different vessels of different classes in each period. It is necessary to make sure that the spaces that are reserved are large enough such that every cyclic period each non-cyclic vessel fits in a space that is large enough to prevent overlap. To do this,

we check the non-cyclic arrivals and find the arrival in each period which takes the longest handling time and has the largest vessel length. This is the time and space that is reserved for the longest non-cyclic arrival in each period. By reserving this space we ensure that the largest non-cyclic arrival each period will have enough space. Next we check each period for the second-to-longest handling time and vessel length and reserve that space, and so forth. In this way it is guaranteed that all the spaces reserved in the projection plan are boundaries for the actual arrivals and no vessel will be outside its planned spot.

For all vessels, cyclic or not, we can anticipate disruptions might occur and proactively incorporate slack time in the plan for increased robustness. The trade-off between a more robust plan versus its reduced efficiency (if no disruptions) is captured by a hyper-parameter ρ. This parameter is the percentage of the each vessel's handling time to add as slack. For example, $\rho = 0.5$ means that a handling time of 10 h is treated as 15 h, i.e., with 5 h (50%) slack. The simulation user can thus express her attitude to risk by varying ρ. We discuss the limitations on how much slack can be added in practice, in Sect. 6.3.

Note that the buffers between vessels provided by adding slack time are a proactive mitigation against uncertainty—delays—in vessel processing times. The observation is that vessels with a large handling time have a larger influence on the planning. This is because a vessel with a relatively short handling time can likely be moved without disturbing many other vessels. However, this observation does not imply that handling time is the most influential source of uncertainty. Section 5 therefore explains how we couple reactive disruption recovery with initial berth planning.

To solve the berth allocation for the projection plan we adopt an optimisation approach. We formulate a compact non-linear model. The model is specified by its parameters, decision variables, objective function and constraints, as follows:

4.1.1 Parameters

$V =$ set of vessels to be served in time horizon
$l_i =$ length of vessel i
$t_i =$ estimated handling time of vessel i
$L =$ quay length
$T =$ cyclic period planning time horizon
$b_{ip} =$ a score for vessel i to be placed at location p

4.1.2 Decision variables

$x_i =$ start berthing time vessel i
$y_i =$ berthing position vessel i
$yObj_i =$ the objective score value for vessel i
$xInterval_i =$ an interval variable covering the interval $\{x_i, x_i + t_i\}$
$yInterval_i =$ an interval variable covering the interval $\{y_i, y_i + l_i\}$

4.1.3 Objective and constraints

$$\max \sum_{i \in V} yObj_i \tag{1}$$

$$\text{NoOverlap2D}(xInterval, yInterval) \tag{2}$$

$$0 \le x_i \le T - t_i \qquad \forall i \in V \tag{3}$$

$$0 \le y_i \le L - l_i \qquad \forall i \in V \tag{4}$$

$$\text{Element}(y_i, b_i, yObj_i) \qquad \forall i \in V \tag{5}$$

$$yObj_i \ge 1 \qquad \forall i \in V \tag{6}$$

The objective function (1) is to maximise the score given to each vessel regarding a certain preferred berthing location for that vessel. As we discuss below, alternative objectives can be incorporated.

Constraint (2) is a CP-specific constraint which constrains each pair of *xIntervals* and *yIntervals*, which form rectangles, to be non-overlapping rectangles (also known as the *diffn* constraint (Beldiceanu and Contejean 1994)). Constraints (3) and (4) ensure that the values x and y can take on such that they remain within the boundaries of the berthing plan. Constraint (5) ensures that the value taken up by $yObj_i$ equals the score for vessel i to be berthed at location y_i.

The final constraint (6) constrains the minimum score that each vessel must have. This last constraint is optional; the minimum score depends on the application. This can be used if certain vessels need to be guaranteed at a certain location. In our case study a value of 1 was chosen to be able to deal with the situation of discontinuous quays (see also Ma et al. (2019)). Namely, in this situation the discontinuous quay can be treated as a continuous quay but on the split a score of 0 is assigned. This way vessels can never be located over the split.

Constraints used in this model follow the naming as implemented in the OR-Tools CP solver (Perron and Furnon 2019) used in our empirical case study in Sect. 6.

Note that there is no constraint enforcing vessel passage in line with feasible tidal windows for that vessel. Indeed, it is not possible to take tidal windows into account with an optimisation approach like this that plans a repeated periodic cycle, as the windows can vary from week to week. Tidal windows are handled in the final step of the planning, described next.

Step 6: Plan each cyclic period. The final step is to project each period's arrivals on the projection berth allocation as created in the previous step. In this procedure the tidal windows will have an important role in deciding which vessel from a loop will arrive in which period. This can be optimised under the assumption that when a shipping line plans arrivals during a low water period, it does not plan a very deep vessel, but rather lets the vessel arrive less heavily laden. This assumption was
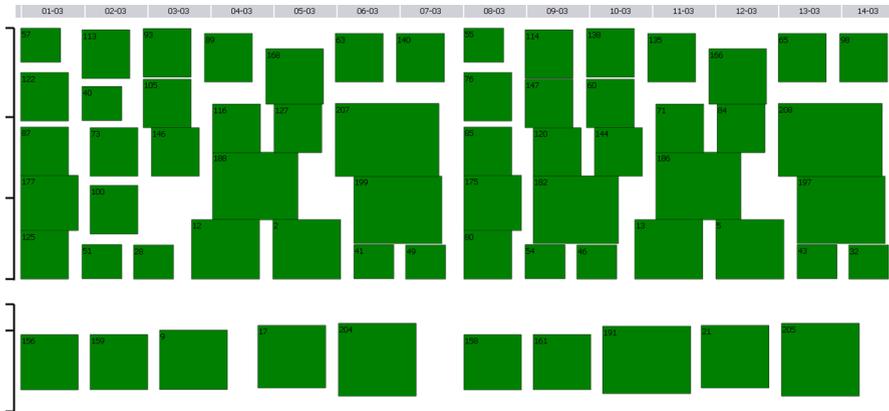
Fig. 3 Example berth allocation: simulation period 2 weeks, cyclic period 1 week. The horizontal axis is the time factor of the berth allocation and the vertical axis shows the quay wall. One can see by the split in the vertical axis that this terminal has a discontinuous quay wall and consists of two sections of quay. The green boxes represent vessels planned in this time frame. Note that in the length of the vessels as visualised (the vertical length of the boxes) also a necessary safety distance is incorporated. Thus, boxes that touch do not touch in real life

provided by domain experts.[2] Hence, for each cyclic loop for which a time and place is reserved in the projection plan, vessels will be planned in such a way that, if the vessel arrives exactly at the time that is planned, the vessel will not have any issues with entering the port due to tidal restrictions. Further, when the expected handling time is over, the vessel should be able to leave as soon as possible.

In practice, this means that when planning the vessels from a loop, first the vessel with the largest draught will be planned. For each period one can use the estimated travel time from the tidal window threshold point to the berth to find the time the vessel would pass that point if it would arrive exactly on time at the planned berth in that period. With this time we can check if the vessel would be able to enter the port at that time according to the tidal patterns. Every period in which the vessel can enter immediately will be selected as a potential period that the vessel will arrive. If the vessel cannot enter immediately in any of the periods, the period with the short- est time to high water will be selected. Now, for each of the selected periods the time after handling is done until the time it can exit the port because of the tides is cal- culated for each period. The period in which this is shortest is selected as the period in which the vessel will arrive. If multiple periods have the same time until exit, we break the tie randomly.

An example output berth allocation from our approach can be seen in Fig. 3. The figure shows a simulation period of two weeks with a cyclic period of one week. Arrivals are randomly generated; the terminal has discontinuous quay walls.

---

[2] The alternative of having the vessels planned to arrive at a higher-water period can be accommodated by our approach. Further, unexpected delays due to travel time and water level can be handled through the disruption recovery (Sect. 5).

Note that in all situations, all the vessels are served. This because over the planning period, due of the extra spaces, we will always find more spaces in the planning to serve the vessels than are needed.

Lastly, note that our planning cyclically-calling vessels on a 2D cylinder—where each rectangle corresponds to a set of similar vessels with a recurring nominal arrival time at each period (Zhen et al. 2011)—means that we could allow vessels to be planned over the 'edge' into the next period. These period-straddling vessels should be taken into account at the start of the planning of the next period. Instead, we require vessels to be scheduled within a period, as can be seen in Fig. 3 by the small break between the end of day 7 and the start of day 8. This significantly simplifies the mathematical optimisation model.

4.2 Alternative objectives

The CP model described above has objective function (1) which maximises the score given to each vessel. The score is derived from the preferred berthing location(s) for that vessel, as given by the terminal operator (who may take into account preferences of the vessel operator). A further advantage of CP is its modelling flexibility (Wallace 2020); alternative objectives can be readily incorporated.

For example, the above model contains no variable for the end of berthing time of vessel $i \in V$. This is because the berthing time is given by parameter t_i. If, instead, t_i is given as the *minimal* rather than expected berthing time, and vessels are allowed to stay quayside longer than t_i, then the model can minimise vessel handling times as follows:

$$\min \sum_{i \in V} z_i - x_i \tag{7}$$

$$\text{NoOverlap2D}(xzInterval, yInterval) \tag{8}$$

$$0 \le x_i \le T \qquad \forall i \in V \tag{9}$$

$$0 \le z_i \le T \qquad \forall i \in V \tag{10}$$

$$z_i - x_i \ge t_i \qquad \forall i \in V \tag{11}$$

$$0 \le y_i \le L - l_i \qquad \forall i \in V \tag{12}$$

where z_i is the end berthing time of vessel i, and $xzInterval_i$ is an interval variable covering the interval $\{x_i, \ldots, z_i\}$.

5 Optimisation approach: re-planning

The preliminary berth plan developed in the last section is used as a basis for the vessel arrival times in the simulation. However, disruptions can occur, such as vessel arrival and loading times varying from the planned. This motivates the importance of treating the dynamic aspects of such disruptions.

5.1 Disruption recovery through localised re-planning

Recall that our approach to disruption is *disruption management*: adjusting the original berthing plan to accommodate disruptions while keeping as close as possible to the original plan. While this is a primarily reactive approach, we include an adjustable percentage of slack in the berthing plan as described earlier. As with planning, our approach is embedded in a simulation context.

It is useful to introduce a simplified description of the simulator's operations is as follows. When the *berth controller* (BC) module receives, in real time, a berthing request for a vessel, it checks whether the proposed berth in the plan is free at the minimum arrival time of the vessel and whether scheduling the vessel thusly will result in any collisions. If both these checks pass, then the vessel can be scheduled at that location and at its minimum arrival time.

However, if either check fails—or indeed if disruptions occur with another vessel—then the BC needs to recover the berth schedule in such a way that every vessel fits in the planning again. This has to happen in real-time and for every vessel and so must be relatively fast: in the order of minutes per vessel. Further, the terminal prefers to stay as close to the original berth plann as possible.

Our approach to disruption recovery leverages and combines two complementary methods. We next describe the quality measure used to assess how good is a planned recovery, then the two methods to obtain recovery plans, and finally how they are used to make decisions.

Quality measure Determining the quality of a repair to a berthing plan has two important factors, according to domain experts with whom we consulted (compare Zeng et al. (2011)): the deviation in berthing time for a vessel, and the deviation in berthing location for a vessel. This gives the penalty for a single vessel i berthed at time x and location y:

$$p_i = c_1 * |x - t| + c_2 * ((1 - \delta) * |y - b| + \delta * L)) \tag{13}$$

where the berthing time x is compared with its previously-planned berthing time t, the berthing location y is compared with b its previously-planned location, δ we explain next, and c_1 and c_2 are cost parameters, weighting the deviation in time and space respectively.

Terminals can have more than one quay wall. In the case of disruption recovery, it could be a good solution to berth a vessel at a different quay wall than at its originally-planned quay wall when disruptions occur. This has been incorporated in the quality formula with the binary variable δ, which takes value 1 if the vessel is moved

to a different quay wall, and 0 otherwise. If the vessel is moved to a different quay wall it is impossible to decide how large the displacement is and therefore the decision has been made to add a large penalty L which is the length of all the quay walls in the terminal combined. This penalty also makes sense in real life because moving resources to a different quay wall is a costly operation.

The quality of a complete realised berthing plan is the average penalty over all the vessels:

$$P = \frac{1}{n} \sum_i^n p_i \tag{14}$$

In consultation with the domain experts, for c_1 the value of 1 is chosen and for c_2 the value of 0.2 is chosen. This means that a 5 m offset is punished similarly to a 1 min delay.

In addition to this quality measure, there may be non-binding requirements about the recovery. These requirements can be seen as preferences of the operator. For example, the case study port has a non-binding requirement that a recovery is not allowed to change the berthing time and location of other vessels that are expected to arrive in four or more days. Any recovery plan which satisfies this *soft constraint* is preferable to any plan which does not.

5.2 Re-planning methods

We leverage one heuristic and one exact re-planning method for disruption management. The heuristic has two parts: the distinct point method and the chain recovery, which are explained next. The exact re-planning method uses CP.

5.2.1 Distinct point method

The heuristic method begins with the *distinct point method*. It finds the empty spaces in a berthing plan. These are used in the chain recovery, described below. Its advantage, compared to brute-force checking each (x, y)-pair, is being a more efficient means to find empty spaces. The resulting *boxes* are of interest because each of these empty spaces represents a potential berthing place and time for the vessel to be planned.

First, the region of interest needs to be determined; this is the sub-space to be searched. For each vessel to be planned, only a small portion of the berthing plan is of interest since this gives the potential berthing locations after its arrival time and within its maximum allowable waiting time.[3] The region of interest therefore starts with the minimum arrival time of the vessel. The end time of the region of interest is the minimum arrival time of the vessel with the addition of the minimum allowable waiting time and the handling time of the vessel. The maximum

---

[3] This maximum is a requirement only relevant to the distinct point method, because otherwise it might delay vessels excessively.

allowable waiting time can be set based on the user's requirements; we use 16 h, as proscribed in Sect. 6.

To determine the empty boxes in a berth plan, the distinct points of interest for both, the x- and the y- axis, are first determined. The distinct points of interest on the x-axis are the start and end times of all the vessels within the region of interest in the planning, the start time of the region of interest, and the end time of the region of interest. The distinct points of interest on the y-axis are the begin and end location of the vessels on the quay within the region of interest, the first point at the quay (at 0 m) and the last point at the quay (the length of the quay). In the case that the terminal is split into multiple quays, the split is also added as a point of interest. In Fig. 4a the selection of distinct points of interest are shown for an example situation.

Now that the distinct points of interest are known, we can use each combination of two *x* points and two *y* points to form boxes. These boxes are known as the potential empty boxes. The creation of empty boxes is shown in Fig. 4b.

The last step is to determine the actual empty boxes. For this, first each of the potential boxes are checked for collisions with the vessels in the region of interest. Second, each box that does not collide with vessels in the region of interest is checked with the other empty boxes. If a box has a 100% intersection with another box it is completely surrounded by the latter. In this case, the bigger box is kept and the smaller box is removed. Third, after this, only the largest possible boxes which completely fill up the empty space are kept. Fourth, only the boxes that start within the maximum allowable waiting time are retained, because these are the boxes that can actually be used as potential berthing locations for the vessel.

In our example situation, Fig. 4c depicts how the final set of boxes are chosen. For clarity, in the figure some selected boxes are omitted. These boxes are the narrow boxes that reach along the whole x-axis and on the y-axis are between 'End of quay' and 'End of 1', between 'Front of 2' and 'Split', between 'Split' and 'End of 3' and between 'Front of 3' and 'Start of Quay'.

5.2.2 Chain recovery

The boxes of interest from the distinct point method are used in *chain recovery*. This completes the heuristic approach to disruption management. Chain recovery approach follows the critical path—the chain—of vessels. Along this chain, vessels are pushed back in time to solve the disruptions (Policella et al. 2004).

This chain starts with the vessel to be planned, or the vessel of interest. When a berthing location is chosen for this vessel, the collisions it makes with other vessels in the berthing plan are calculated. The vessels that it collides with are pushed back (i.e., berthing delayed) until the two vessels do not collide any more. The boxes tell us how far we can move each vessel. However, since these vessels have been pushed backwards, they might have caused new collisions. These collisions are solved repeatedly until no further collisions are caused. Note that the time slacks that exist in the plan—the extra time beyond the handling time of each vessel—act as sponges to absorb the collisions.
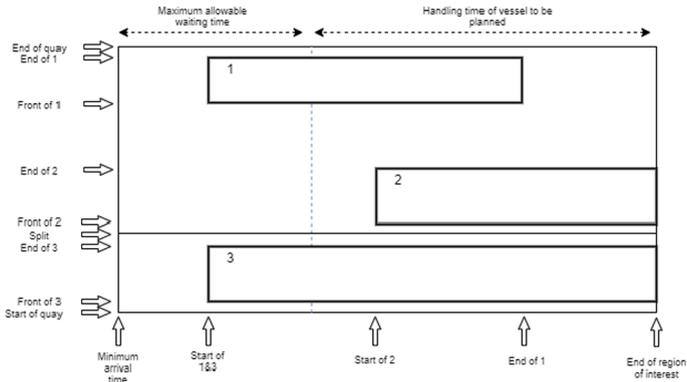
(a) The way distinct points of interest are selected on the x-axis and y-axis in an example situation
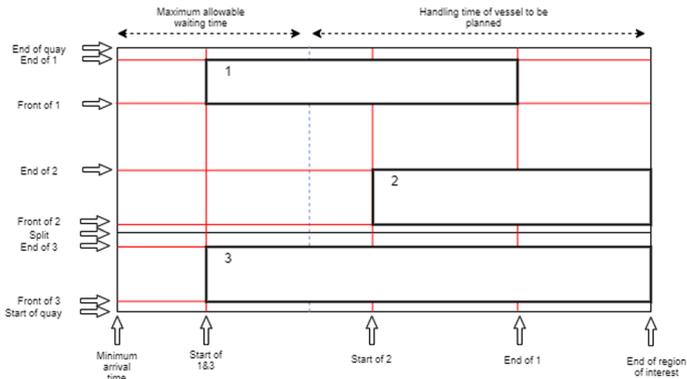
(b) How empty boxes are created in an example situation
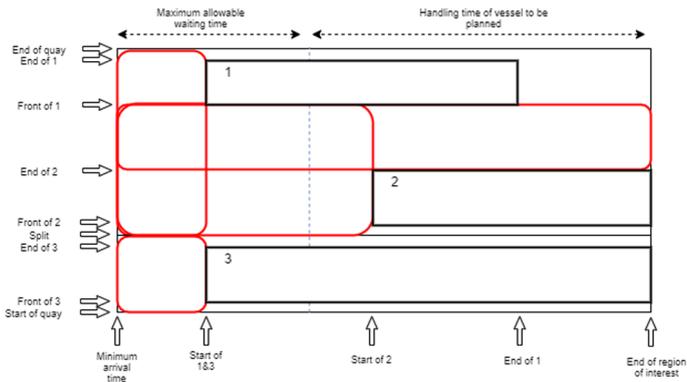
(c) How the largest empty boxes are selected that do not intersect with the vessels

Fig. 4 Example situation of a berth planning and how empty boxes are selected. Note that for reasons of clarity, 4 empty boxes have been left out of (**c**). These boxes are the narrow boxes that reach along the whole x-axis and on the y-axis are between 'End of quay' and 'End of 1', between 'Front of 2' and 'Split', between 'Split' and 'End of 3' and between 'Front of 3' and 'Start of Quay'

Chain recovery is a relatively fast method. It is sub-optimal in the sense that it considers only one type of recovery action: delaying a vessel. It does not consider other actions such as changing a vessel's scheduled location.

5.2.3 Local CP recovery

The second method for disruption recovery is an optimisation-based method, akin to how we develop the initial berthing plan (Sect. 4.1). The motivation is that there might be situations in which heuristics to recover the berth planning—the distinct point method and chain recovery—do not give satisfactory results.

Hence, we use a revised CP model to locally reschedule the berth plan. In principle, this approach can find the optimal recovery plan in terms of minimising perturbation from the original berthing plan (Fukunaga 2009). The CP model is called *local* because it aims to keep the region of interest that will be replanned as local as possible, thereby influencing as little other vessels as possible. This region of interest consists of the entire quay in the y direction, and an expanding time span in the x direction. Expanding means that an increasing time span is used until the model is able to find a feasible solution.

The parameters, decision variables, objective function and the constraints for the local CP model are as follows:

5.2.4 Parameters

V = set of vessels to be served in time horizon
$xOriginal_i$ = original x position for vessel i
$yOriginal_i$ = original y position for vessel i
l_i = length of vessel i
t_i = handling time of vessel i
$Locked_i$ = 1 if the vessel cannot be moved, 0 otherwise
L = quay length
$Split$ = split location of quay
T = planning time horizon

5.2.5 Decision variables

x_i = start berthing time vessel i
y_i = berthing position vessel i
$yObj_i$ = berthing position penalty for deviation from y
s_i = 1 if vessel i is located above the split, 0 otherwise
$xInterval_i$ = an interval variable covering the interval $\{x_i, \ldots, x_i + t_i\}$
$yInterval_i$ = an interval variable covering the interval $\{y_i, \ldots, y_i + l_i\}$

5.2.6 Objective and constraints

$$\min \sum_{i \in V} (x_i + 0.2 * yObj_i) \tag{15}$$

$$\text{NoOverlap2D}(xInterval, yInterval) \tag{16}$$

$$xOriginal_i \leq x_i \leq T - t_i \qquad \forall i \in V \tag{17}$$

$$0 \leq y_i \leq L - l_i \qquad \forall i \in V \tag{18}$$

$$Locked_i = 1 \implies x_i = xOriginal_i \land y_i = yOriginal_i \qquad \forall i \in V \tag{19}$$

$$s_i = 1 \implies y_i \geq Split \qquad \forall i \in V \tag{20}$$

$$s_i = 0 \implies y_i + l_i \leq Split \qquad \forall i \in V \tag{21}$$

$$yObj_i \geq |y_i - yOriginal_i| \qquad \forall i \in V \tag{22}$$

$$(s_i = 1 \land yOriginal_i \leq Split) \lor (s_i = 0 \land yOriginal_i \geq Split) \\ \implies yObj_i = L \qquad \forall i \in V \tag{23}$$

The objective function (15) minimises the difference with the vessels original berthing time and position by minimising their arrival time and penalty for deviation in berthing position. The values of the constants are set as described in Sect. 5.1.

The new constraints (17) and (18) ensure that the boundaries of the values of x and y are respected. The main point of interest here is that x can only be put after *xOriginal* because vessels cannot be forced to come earlier. Constraints (19) ensures that if a vessel is locked it cannot be moved. Constraints (20) and (21) ensure that a vessel cannot be positioned on a split. The final three constraints, (22)–(23) appropriately constraint the value of *yObj*. Constraint (22) ensures that the value taken up by *yObj* is the absolute difference between the original location and the actual location. The last constraint determines whether or not the vessel is put at the other side of the split. If this is the case *yObj* takes on the value for the total length. These constraints will never lead to infeasible results because by definition we know that L is always larger than the absolute difference between the original location and the actual location.

The above CP model does not include soft constraints on the recovery plan. Below we discuss how we handle the one soft constraint relevant to the case study. It is possible to include soft constraints in the model, whereupon a soft CP solving approach is relevant (Schiendorfer et al. 2018).
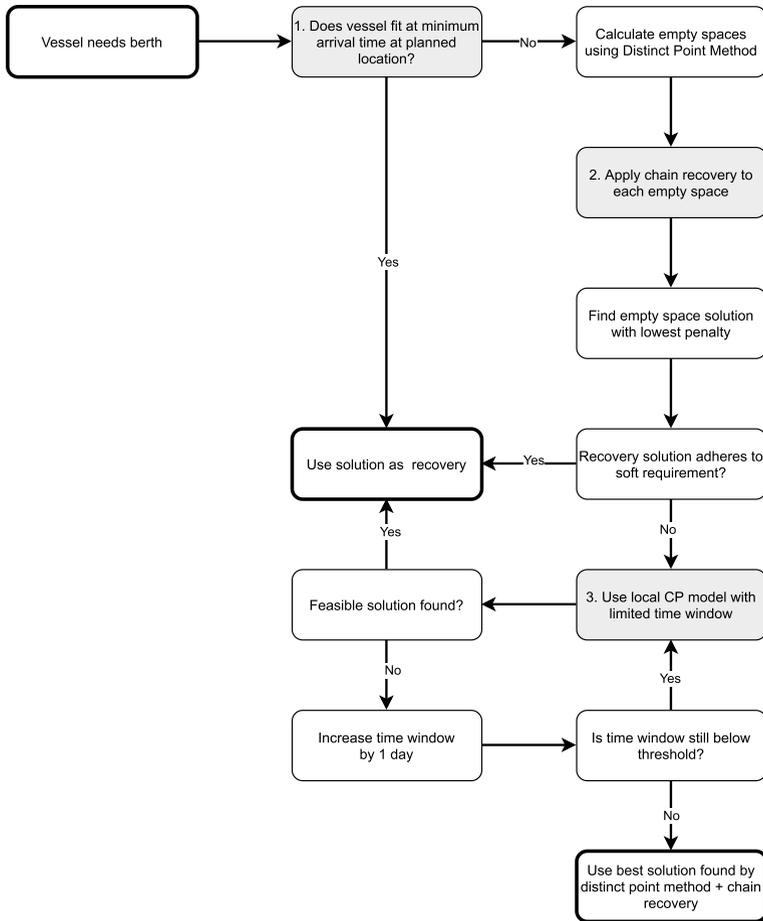
Fig. 5 Flow chart visualising the decision making process for the disruption recovery model. The three main steps are shown with shaded background

5.3 Decision making for re-planning

The two methods just described—chain recovery and local CP—are used together in a multi-level heuristic approach for disruption management, as depicted in Fig. 5.

The decision flow is as follows. Recall that first (Step 1) the Berth Controller checks whether the vessel fits at its minimum arrival time at its planned location. If this fits and no disruptions happen, we know that the vessel v is scheduled at the optimal location and time.

However, if the check fails, then the distinct point method is applied to the region of interest in the berth plan for v. This gives a set of empty boxes of interest. Whether an empty box for v is considered promising is determined by checking the width of the box. First, the length of the vessel must fit in the box. If so, the longer

the time span of the box, the more promising it is. We can next evaluate the solutions corresponding to the empty boxes and choose the best recovery regarding the quality measure Eq. (13).

Second, then in Step 2, the chain recovery method yields the recovery corresponding to an empty box. The vessel to be planned is positioned in the empty box with its berthing time equal to the start time of the box and its berthing position as much as possible to the side of the box. There is one exception: if the original planned berthing position of the vessel v is contained in the empty box, this will be the berthing position of the vessel. If all the disruptions are recovered, the penalties for each vessel is calculated and summed up and compared with the recoveries corresponding to the other empty boxes. Finally, the best recovery is also compared to the situation where v is simply placed at its originally-planned location and chain recovery is applied from there. The best recovery is selected as the recovery for v; BC updates the berthing plan.

Third, however, there is one important exception. It might be that the recovery found in the previous step does not satisfy the soft constraints. If not, Step 3 invokes the local CP model, the exact method (if run to completion). In particular, in the case study port, the only non-binding requirement is that a recovery may not change berthing time and location of vessels four days in the future. It is natural therefore to apply the local CP with iterative deepening on the time window. That is, first, a time window of one day for the repair is tried. If the CP model is not able to find a feasible solution, this time window is extended with one day and the local CP model tried again. This is repeated until the time window also hits the four day limit. In this situation, no good solution can be found within the required time and, since we must provide a solution, the best recovery from the previous step is applied anyway.

6 Case study and discussion

We examine a case study for a major tide-dependent inland river delta port in Europe. The goal is to create the baseline berth allocations for three terminals in this inland port; one terminal is in design and does not exist yet. These baseline berth allocations are used in a simulation model intended to give the port insight into the effect of the addition of this extra terminal under various uncertainty scenarios. The port provided a yearly vessel arrival forecast for the year 2030. Our work is implemented as part of a bespoke commercial simulation platform in use at several ports, and in this case, being used to analyse and plan the port expansion and future operations.

We face a typical situation in which historical data about vessel arrivals is available but not applicable. First, a new terminal will be simulated of which no historical data is available. Second, we are simulating ten years in the future, and vessel arrivals (both the amount and the distribution over the classes) in the forecast provided by the port management are very different from the historical arrivals. Third, the river delta port is highly tide dependent, meaning that vessels with a large draught can have very long waiting times due to tidal restrictions and sometimes even have a tidal window of mere minutes.

The terminals have the following specifications:

- Terminal 1 is relatively short terminal with one quay wall of about 1500 ms.
- Terminal 2 is a large terminal with two quay walls: one of about 2500 ms and a smaller one of about 800 ms.
- Terminal 3 is a new terminal in design. In this case study the terminal consists of two quay walls: one of about 1500 ms and one of about 600 ms.

The case study port imposed a number of specific requirements. These include a time granularity of 1 min. This arises because the planning must be able to account for very deep vessels which sometimes only have a window of a few minutes when entering the port. It is to be emphasised that our approach is generic and different requirements can be accommodated. That is, the CP model of Sect. 4.1 is not specific to the case study. It can be modified with case-specific constraints, and the operational parameters can be easily varied.

We undertake three sets of experiments. These study, respectively, the berth allocation planning, the re-planning through disruption management, and the trade-off between proactive and reactive robustness.

6.1 Experiments: planning

In our first experiment, we examine the planning performance under different levels of problem hardness. The goal is to study the feasibility and potential optimality of our approach to arrival generation and berth allocation. We vary the average occupancy of the terminals and measure the time it takes for the CP model to obtain a (first) solution. The solution time of the CP model is of interest because it is the most time consuming part of the planning approach described in Sect. 4.

We undertook an initial experiment to compare the CP optimisation model with an integer programming (IP) optimisation model. Formulating the problem with the linear constraints of IP led to an artificial increase in model size, and poor performance in initial tests. Results, not reported here, agree with Qin et al. (2016) who demonstrated the superiority of a CP formulation over an IP formulation in the type of problem we solve in the planning.

The hardness of the problem in this study is defined by increasing the occupancy through inflating or deflating the yearly forecasts, thus keeping the ratios in terms of classes and terminals the same. All other variables are held constant, e.g., the terminal specifications, the simulation period and the basis of the yearly forecasts per terminal. The motivation is that the higher the occupancy, the more vessels will be arriving in the simulation, and the more difficult it will be to allocate berths.

The experimental setup consists of the three above-mentioned terminals and their respective yearly forecast for 2030 as provided by the port. Each run is for the simulation period of one month in 2030 with a cyclic period of one week. We stop the solver when it has found and proved an optimal solution, made no solution improvement for two hours, or proved infeasibility. The latter occurs when, for instance, the projection week is too large and the problem is infeasible. For each occupation,

Table 2 Runtime in seconds of the CP model for three terminals

Occupancy		0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
Terminal 1	Time (s)	5.21	5.99	5.68	6.39	7.38	25.54	89.49	45.78	109.87	200.33
	Infeas. %	0	0	0	0	0	0	0	0	0	0
Terminal 2	Time (s)	7.92	10.31	11.49	13.03	15.58	13.78	17.86	70.84	251.49	252.04
	Infeas. %	0	0	0	0	0	0	0	0	0	0
Terminal 3	Time (s)	4.92	6.23	7.24	8.69	8.58	10.12	11.23	12.06	12.1	16.51
	Infeas. %	0	0	0	0	0	0	0	0	0	0

Occupancy		0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
Terminal 1	Time (s)	924.52	1778.04	3461.22	4009.1	4158.97	5772.3	5487.1	8772.91	0
	Infeas. %	0	0	0	0	0	20	20	60	100
Terminal 2	Time (s)	313.48	1699.62	2803.3	3378.02	3774.64	4822.1	5680.25	0	0
	Infeas. %	0	0	0	0	0	20	40	100	100
Terminal 3	Time (s)	136.17	612.28	1146.91	1534.25	2058.14	2659.1	3669.27	0	0
	Infeas. %	0	0	0	0	0	0	20	100	100

For each terminal both the runtime and the percentage of runs that yield timed-out or infeasible results are provided

starting from 5% up to 95% with 5% increments, we report the average runtimes of five runs. The differences between the five runs for each occupancy are found in the random generation of vessel arrivals from the yearly forecast and the division over the cyclic periods.

Experiments were run on a computer with 8GB RAM and an Intel i7 2.80GHz CPU (4 cores). For solving the CP model we used the state-of-the-art OR-Tools CP solver (Perron and Furnon 2019), version 7.5 running with all 4 cores. This solver combines constraint propagation with cuts and linear programming relaxations, in a conflict-directed search framework. Note that the solver is complete: when run without a timeout, it produces the globally-optimal solution or proof of infeasibility.

Results and discussion

The results are presented in Table 2, per terminal.[4] We see that, even though the speed with which the runtimes increase differs per terminal, the general trend is the same. At first, until about 50% occupation, the CP solver is able to find a good solution easily and is also able to prove its optimality. The table reports the time from start until last found solution (if any), or until timeout (if not). This changes in the region from 50% to 60%: here the runtimes start increasing and the solver is not able to prove the optimality without timing out after two hours of no improvement.[5] Note that we report together as 'Infeasible' those instances which timed-out and those instances which were proved infeasible.

From the 65% mark, none of the runs are able to prove optimality any more within the set timeout. As soon as we reach the extremely busy cases (above 80% and higher) we start seeing runtimes of more than 90 min and occasional proven infeasible results. At 90% and higher most results are proven infeasible within two hours. The consequence of these findings is that for normal occupancy of between 50–70% runtimes of above an hour are very unusual.

We have presented our results to the management team of the case study port. Based on multiple demonstrations, the management is well satisfied with the speed of our approach and the quality of the berth plans created. They state that the quality meets their expectations and that the plans produced are realistic.

6.2 Experiments: re-planning

In our second experiment, we examine the preliminary berth planning together with the disruption management. The goal is to study the effect of uncertainty on the ability of the disruption management approach to find good recovery berth schedules. This experiment is run for the same month in 2030 for the same three terminals, again with the yearly forecasts for 2030 provided by the case study port.

---

[4] For the management of the port, it is important to have data for the terminals separately. This is because each terminal has its own specifications (length, distribution of arriving vessel types, a split or not) which speaks to the diversity in which our approach can be applied.

[5] We performed further probing experiments with a 24 h timeout and found that, if the optimal solution is not proven within two hours, the solver makes little further progress up to 24 h. This suggests that near-optimal solutions are found within the time limit, but their optimality is not proven.

The experimental setup is as follows. First, a baseline berth plan is created for the entire simulation period. Next, the simulation is run and the disruption management approach used to respond. We measure the quality of the resulting schedule, and compare it against a simple baseline recovery approach, and against an ablated approach where the CP model is never used. We also report runtimes.

For the experiment we consider three main sources of disruptions. The first source, handling time, is based on a distribution which is based on data provided by the case study port. The second source of uncertainty, travelling time, is included in the simulation model and varies based on busyness on the water with other traffic and corresponding sailing rules. The travelling time is the time for a vessel to traverse from the entrance to the river to the entrance to the port. The third source of uncertainty is the arrival times of vessels: that is, the arrival time to the entrance to the river (not to the berth).

Whereas the first two sources are included in the simulation, the third, the arrival time disruption, can be altered and serve as a measure of uncertainty. To formalise this, if we model the arrival time as a normal distribution with the planned arrival time as mean, uncertainty is defined as the standard deviation in hours in this normal distribution. For example, if the uncertainty is 6 h, the actual vessel arrival will be sampled from a normal distribution with its planned arrival time as mean and 6 h as standard deviation. In this example, following the standard rules of statistics, about 68% of the vessels in this simulation run will arrive within one standard deviation, 6 h before or after the planned arrival time, and about 95% of the vessels will arrive within two standard deviations, 12 h before or after the planned arrival time.

With the increasing uncertainty we study the percentage of the vessel arrivals that are solved by which recovery method:

– **DRCP**: Disruption Recovery with CP. Our full three-step method as described in Sect. 5.2.
– **DRnCP**: Disruption Recovery no CP. We ablate DRCP by removing the CP; thus DRnCP has only the initial check, and then the heuristic distinct point method and chain recovery.
– **Baseline**: A baseline recovery approach that always plans the vessel at its planned location; if disruptions appear it performs chain recovery to make sure that the planning is physically possible again. The baseline preforms no lookahead, and like DRnCP, ignores soft constraints.

To be able to reliably compare these three methods we conducted the experiment with the same random seed for each. The average over 10 runs is taken as an indication for the quality of the resulting recovered berth plan under a certain level of uncertainty. Since simulation run simulates three terminals at once, the presented runtime is the average resulting time over all three of the terminals.

7 Results and discussion

The results are presented in Figs. 6 and 7, and in Tables 6, 7, 8, 9 in the appendix.
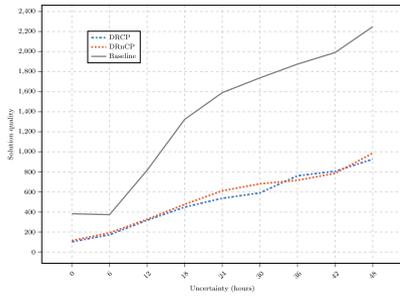
(a) Solution quality for different uncertainties for terminal 1
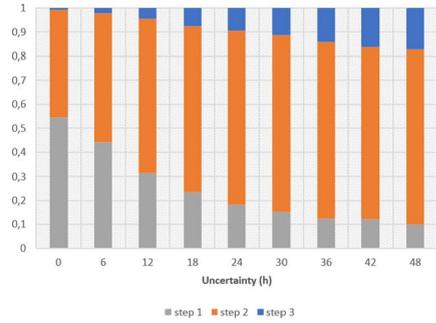
(b) Division between solution steps for terminal 1
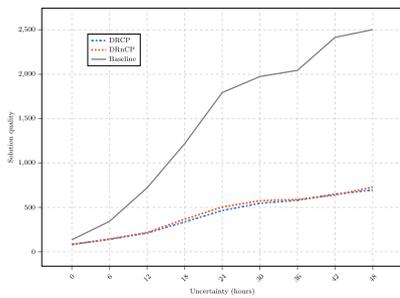
(c) Solution quality for different uncertainties for terminal 2
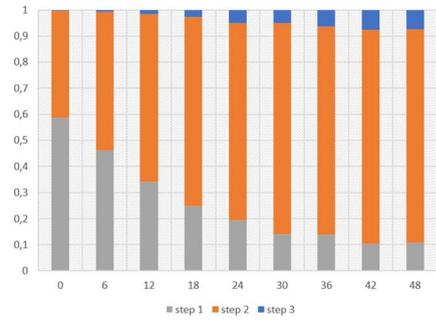
(d) Division between solution steps for terminal 2
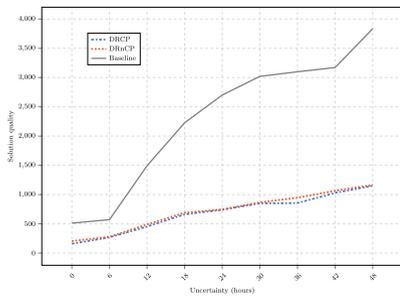
(e) Solution quality for different uncertainties for terminal 3
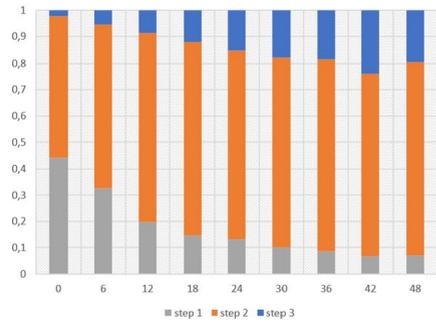
(f) Division between solution steps for terminal 3

Fig. 6 For each of the three terminals: the solution quality for the different recovery approaches for different uncertainties, and the division between different solution steps for different uncertainties

Figure 6 gives for each terminal the solution quality for the three different recovery algorithms (left column) and, for DRCP, the count of solution usage from the three steps (minimal fit, chain recovery, local CP: see Fig. 5). Recall that solution quality is defined as the average penalty over all the vessels (14). In more
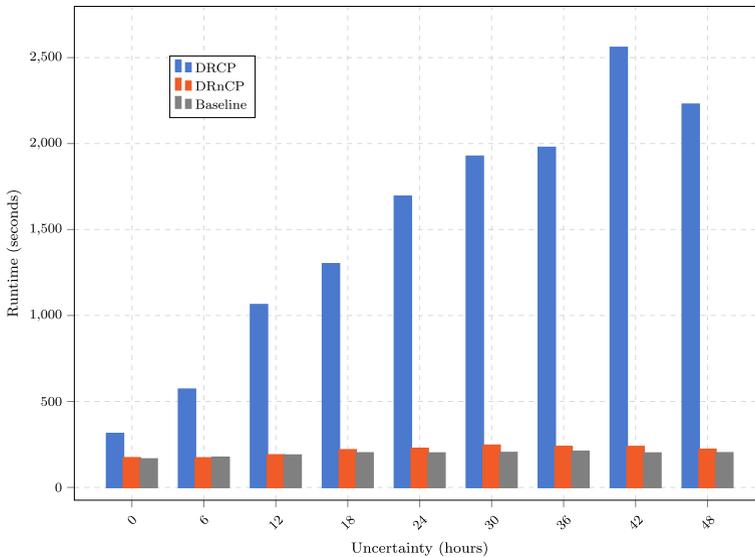
Fig. 7 Runtime (in seconds) for the different recovery approaches as uncertainty (in hours) grows

detail, the right column shows, for each step, the percentage of vessels for which the solution from that step is used.

As expected given its relative simplicity, the baseline has inferior performance.

Figure 7 gives the total runtime of the three methods. DRCP is much slower than the others. This can be expected from DRCP sometimes using the CP solver to obtain the optimal re-schedule, whereas DRnCP and baseline are purely heuristic. Further, looking together at the right column of Figs. 6 and 7, we see how the runtime of DRCP indeed correlates with how often Step 3 (the local CP model) is necessary. Greater uncertainty means that DRCP invokes the local CP model more often, because chain recovery (Step 2) does not provide a feasible solution.

However, interestingly, we see that DRnCP achieves comparable quality of DRCP even at high levels of uncertainty. This is despite the local CP model re-scheduling all vessels, resulting in a solution with vessels likely closer together than the solution obtained by the chain recovery, which pushes vessels into the future. The explanation is that the planning is updated *every time* a new vessel arrives. Hence, although DRnCP can only push back the berthing time of a vessel in the berth plan, it still arrives following its planned arrival time in the baseline berth plan. This means that as soon as the vessel actually arrives, it might be planned at the berthing time and location as indicated before—but it is also likely that DRnCP might find a new time and location for this vessel that is not as far in the future. Thus, even though the 'intermediate' solutions of DRnCP are worse than those of DRCP, there is a good

chance that the problems of the inferior intermediate solution are dissipated when more vessels arrive, more information is available, and the vessels are moved again.

We cannot conclude the higher runtime of DRCP is for nothing, however. The fortuitous situation for DRnCP just described is not guaranteed: it depends on the current situation in the port, the arrival times of other vessels and the handling time of other vessels. In Fig. 6a, c, e we see that the standard deviation of DRnCP is higher than that of DRCP. The greater robustness of DRCP, and its satisfaction of soft constraints (if possible), is the advantage it obtains for its higher runtime.

7.1 Occupancy and robustness

In our third experiment, we examine the influence of the occupancy rate of berth occupation and the value of adding robustness in the preliminary planning. The goal is to study the trade-off between the occupancy rate and robustness.

The experimental set up is as follows. First, a baseline berth plan is created based on a certain expected occupation and while ensuring a certain robustness in the plan. Next, the simulation is run and the quality of the realised berth schedule is measured.

The average occupation of the berths in a terminal is a very important measurement to grasp the busyness in the terminal. A domain expert expressed her interest in the effect of occupancy on how well the disruption recovery would work. This is because, in real life, from a certain occupancy rate, it becomes very difficult for the port manager to keep the terminal running: at a certain tipping point it becomes difficult to keep the waiting times low and stay close to the theoretical planning. One of the main goals in this experiment is to investigate whether such a tipping point exists and, if so, at what occupancy rate it occurs.

Recall that adding robustness to the preliminary berth planning can be done in a proactive manner by adding a time slack after each vessel, controlled by the parameter ρ. By guaranteeing this time slack after each vessel in the plan, the planning is less susceptible to disruptions since the smaller disruptions can be captured within this time buffer. In practice, while the port manager might wish to set ρ to be large, the larger its value, the larger the operational implications for efficiency. In particular, when more vessels are expected the potential length these time slacks can take on decreases. This is because more vessels are required to be fit in the same space, which obviously results in less empty space in the planning. This leads to the goal of the experiment.

For each combination of occupancy rate and robustness level, the average of 10 simulations runs is reported. In line with the soft constraints of the case study port, the DRCP approach is used. A disruption standard deviation of 12 h is assumed: hence 95% of the vessels in these runs will arrive in the time window of one day before and after their planned arrival time.

Results and discussion

Figure 8 and Tables 3, 4, and 5 show the results of the experiment regarding occupancy rates and robustness. '–' denotes that the planning was not able to find a feasible preliminary berth plan (compare Table 2). In this case the recovery algorithm
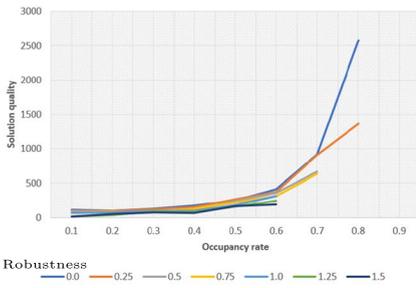
(a) The solution quality for different robustness levels with increasing occupancy rates for terminal 1
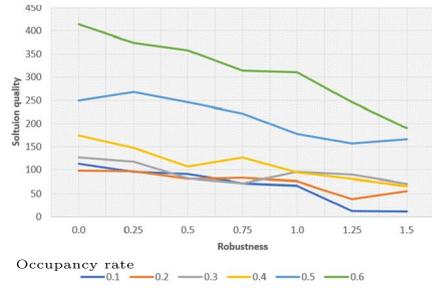
(b) The solution quality for different occupancy rates with increasing robustness levels for terminal 1
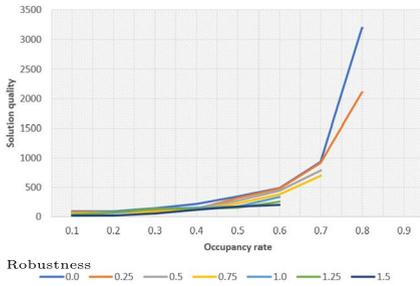
(c) The solution quality for different robustness levels with increasing occupancy rates for terminal 2
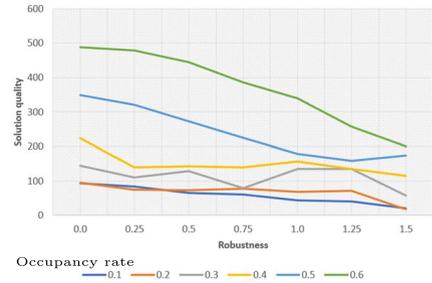
(d) The solution quality for different occupancy rates with increasing robustness levels for terminal 2
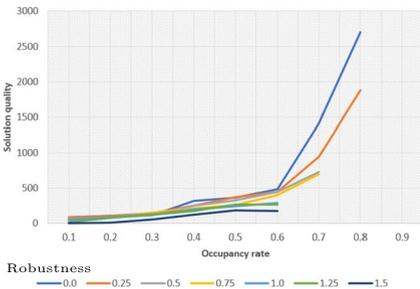
(e) The solution quality for different robustness levels with increasing occupancy rates for terminal 3
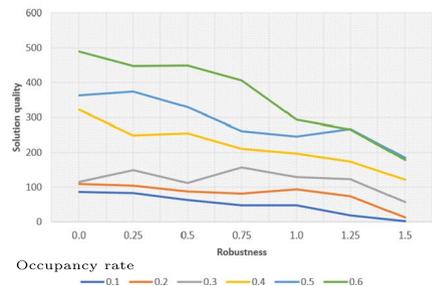
(f) The solution quality for different occupancy rates with increasing robustness levels for terminal 3

Fig. 8 For each of the three terminals: the solution quality for different occupancy rates and different robustness levels. Graphs in the right column show the same data as those in the left column, with occupancy rate and robustness inverted

would still work but since there is no initial plan, the returned quality values are not meaningful.

Table 3 Numerical results for the quality of the realised berth plan given different occupancy rates and robustness levels for the terminal 1

	Occupancy								
Robustness	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.0	113.59	98.61	126.75	174.53	249.71	414.42	915.52	2578.49	–
0.25	95.66	97.2	117.76	148.02	268.57	374.73	906.84	1366.18	–
0.5	91.28	80.6	81.55	107.64	245.87	358.45	662.49	–	–
0.75	71.21	82.8	71.24	126.99	222	314.51	638.31	–	–
1.0	66.78	76.57	95.79	94.53	177.71	310.89	–	–	–
1.25	12.09	38.28	89.51	80.78	157.6	246.18	–	–	–
1.5	10.89	54.71	70.91	66.09	166.76	190.14	–	–	–

Lower score is better. '–' denotes that initial baseline planning found no feasible plan

Table 4 Numerical results for the quality of the realised berth plan given different occupancy rates and robustness levels for the terminal 2

	Occupancy								
Robustness	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.0	92.97	95.81	144.51	223.53	348.91	488.36	933.19	3206.22	–
0.25	84.25	75.53	109.83	139.81	320.08	479.39	915.31	2101.55	–
0.5	65.43	73.8	129.36	143.65	273.52	444.35	793.67	–	–
0.75	60.2	77.81	79.58	139.74	225.21	385.73	701.81	–	–
1.0	43.87	69.27	135.52	156.68	177.77	339.63	–	–	–
1.25	41.22	70.95	135.12	134.39	158.98	257.01	–	–	–
1.5	20.71	18.07	57.88	115.98	174.16	200.99	–	–	–

Lower score is better

'–' denotes that initial baseline planning found no feasible plan

Table 5 Numerical results for the quality of the realised berth plan given different occupancy rates and robustness levels for the terminal 3

	Occupancy								
Robustness	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.0	86.6	108.84	114.67	322.18	362.81	489.49	1419.93	2708.56	–
0.25	82.64	104.22	148.93	249.67	374.47	448.13	945.01	1878.1	–
0.5	62.35	87.77	112.04	254.46	329.85	449.96	728.77	–	–
0.75	47.95	81.84	156.36	210.7	261.25	406.2	697.84	–	–
1.0	47.59	92.99	129.46	196.11	245.88	294.09	–	–	–
1.25	18.28	74.27	122.07	173.57	267.88	265.69	–	–	–
1.5	1.12	12.48	57.23	121.25	185.04	178.48	–	–	–

Lower score is better

'–' denotes that initial baseline planning found no feasible plan
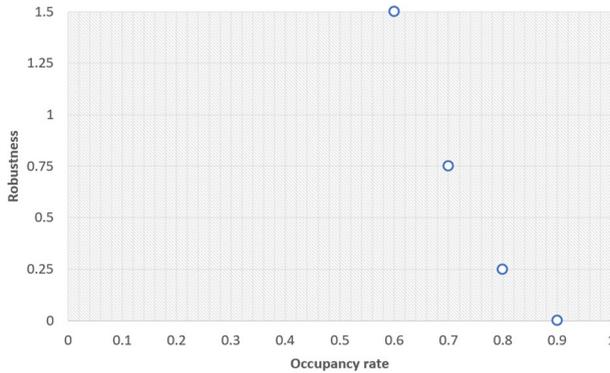
Fig. 9 Trade-off between occupancy rate and robustness

Figures 8a, c, e show the effect of occupancy on the solution quality for the different terminals. We see that, as expected, the quality becomes worse, the higher the occupancy rate. This is expected because the more vessels are anticipated in the same time frame, the more disruptions will happen and therefore higher penalties will be made. What can also be noted is that the tipping point indeed exists. For every terminal, a sudden increase in penalty can be seen at 70% occupation and this trend increases rapidly when 80% occupation is reached. This confirms the prediction that a tipping point exists from where the recovery approach has much more difficulty to find good solutions.

For terminal 2, the CP solver is invoked less frequently than for terminals 1 and 3: less than 10% compared to more than 15% at high uncertainty levels for terminal 1 and more than 20% for terminal 3. Terminal 1 exhibits some small edge in solution quality for DRnCP over DRCP at medium levels of uncertainty, whereas the other terminals have almost no difference.

Figures 8b, d, f show the same data, but with robustness on the horizontal axis. From these graphs can be seen how the solution quality improves when the robustness increases. It can also be seen that for higher occupancy rates, the addition of robustness has more effect. It should be noted that the lines for the occupancy rates of 0.7 and 0.8 where omitted to better show the effect of robustness at the lower levels. However, if these would have been added, one would see that they show the same downward trend, but with steeper fall, the higher the robustness becomes. From this the conclusion can be drawn that the higher the expected occupation, the more important it is to include robustness in the preliminary berth planner.

As noted, it is not possible to add endless time slacks to a berth schedule. Depending on the occupancy more or less robustness can be added before the plan becomes infeasible. The effect of this can be seen best in the tables where the empty spaces represent problems that are infeasible. Figure 9 presents the Pareto frontier where, for each occupancy, the points have been drawn where the first infeasible runs where detected. This information can be used in simulation models to proactively and dynamically add robustness to a berth plan based on the estimated occupancy for that simulation period.

8 Discussion and conclusion

This work is motivated by analysing a planned new terminal for a large river delta port in Europe. The case study is a tidal port, for which a crucial aspect of berth planning is to take into account the tidal windows. A simulation study is required by the port operator.

We develop a novel, generic approach, focussed on the generation of a cyclic baseline planning in simulation on which the vessel arrival times and the actual berth allocation are based. We simultaneously incorporate realistic aspects of the problem: (1)the tidal windows of vessels, (2) the cyclic nature of certain container vessel arrivals, and (3) the minimisation of vessel draught during low water periods. We obtain the baseline plan using a constraint programming model. By adopting constraint programming, we can formulate a compact and understandable declarative model. Further, CP allows us to decouple the modelling from the solving (Freuder 1997).

Our approach is successfully employed for the case study port. With increasing terminal occupancy, we examine the effect of busyness on the runtime of the CP model using a state-of-the-art solver, and show that we can either obtain the optimal berth plan or a good approximation of it, within one hour for a large port. It is to be emphasised that the approach presented is generic, and can be applied to other ports, and indeed to non-tidal ports and also ports with additional constraints such as channel capacities.

The approach is developed with simulation providing input into the optimisation. It can also be used in a live setting, with real-time operational data replacing the input from the simulation.

For a simulation, our approach serves as a basis for the vessel arrival times and the planned berth allocation. When the simulation is run, most vessels will not arrive exactly according to this baseline plan; further, the departure time is an expected time from which the simulation can deviate. The actual arrival time in the simulation will be the planned time with a deviation that is based on a distribution. This distribution can be learned from historical data where one compares real arrivals with their theoretical planned counterparts. Further, the baseline plan is generated based on expected handling times and expected travel times. In the actual simulation, it is also likely that there will be deviation from these expected values. This, plus the deviation from the planned arrival time, requires a disruption recovery module that manages the berth planning in real time.

Hence, leveraging the simulation-informed optimisation paradigm, this article also shows how to re-plan the berth allocation when vessels arrive early or late, their (un)loading takes longer or shorter than expected, or exogenous events occur. We accommodate this uncertainty by formulating a multi-level heuristic approach, including a modified CP model which repairs the plan while minimising perturbation. For the case study port, disruption recovery can be achieved within a few

minutes of computation time. Further, we also allow for some uncertainty in the number of ships calling at the port, by the planning of extra spaces which are only used in some of the weeks and left open in other weeks. Additional sources of uncertainty can accommodated explicitly with the simulation-informed optimisation paradigm.

A distinguishing feature of this article is the combination of berth planning, which has a horizon of months, and its use as an *input* in real-time disruption recovery, to gain insight into future operations. This contribution is enabled by the simulation-informed optimisation paradigm. Having developed an optimised baseline berth allocation plan, we use this baseline plan in simulated arrival generation; and then, in real time, adapt the actual plan in the simulation to minimise disruptions.

We work with a forecast of arrivals in a given year—in the case study presented, the year 2030—and plan for a month of (future) operation. Note that stationarity of arrivals cannot be assumed. If berth plans of longer duration are required, then the baseline planning can be repeated with forecasts for another year, and the plans concatenated: for instance, 2030 and 2031.

The work reported in this article is implemented as part of a bespoke commercial simulation platform in use for decision support at multiple container ports worldwide. We provide managerial insights for the capacity planning of terminals and the robustness of operational plans to uncertainty. Empirical study shows that the recovery approach finds good solutions until a tipping point of disturbance. Results also show that when the expected occupation of a terminal is higher, strengthening robustness of the preliminary plan has increased importance.

Our ongoing work contains three parts. The first is enriching the simulation model itself. For instance, considering that a vessel may be able more or less constrained with respect to the tidal windows according to the cargo to be loaded or unloaded. The second part is enhancing the optimisation models, for instance to remove the restriction of vessels being unable to lie on the boundary of two cyclic periods or the assumption of a single cyclic period for all vessels. The last part is expanding the operational use of the simulation-based decision support tool. For instance, connecting it to simulation models for terminal crane allocation (Jonker et al. 2019), trans-shipment ports (Lv et al. 2020) and to hinterland shipping.

Appendix

See Tables 6, 7, 8, and 9

Table 6 The quality of the realised berth plan and the standard deviation over 10 runs for the different recovery methods for different levels of uncertainty for the terminal 1

		0	6	12	18	24	30	36	42	48
DRCP	mean	102.67	172.76	317.56	448.65	536.83	590.22	760.94	806.09	926.05
	std	14.31	47.48	61.34	98	73.44	186.3	189.87	181.63	176.22
DRnCP	mean	114.89	192.19	327.32	476.5	611.35	681.25	716.03	785.68	986.7
	std	22.88	41.61	54.71	123.88	118.2	312.02	211.41	260.26	240.08
Baseline	mean	382.3	373.25	815.82	1323.85	1590.65	1737.69	1875.11	1990.62	2247.75
	std	229.03	40.73	295.65	707.39	468.23	317.91	341.06	529.03	765.36

Lower scores are better

Table 7 The quality of the realised berth plan and the standard deviation over 10 runs for the different recovery methods for different levels of uncertainty for the terminal 2

		0	6	12	18	24	30	36	42	48
DRCP	mean	83.66	139.55	210.55	336.05	464.46	546.17	579.59	649.38	695.03
	std	8.15	19.17	36.69	61	73.09	32.07	81.16	91.46	99.08
DRnCP	mean	79.8	143.11	216.95	366.51	503.95	575.13	589.25	637.96	727.36
	std	10.56	24.76	45.11	75.17	80.94	89.33	149.71	123.99	177.78
Baseline	mean	135.04	342.53	719.98	1217.71	1794.16	1974.04	2044.14	2415.56	2502.77
	std	30.99	81.86	157.61	229.3	509.89	635.82	719.54	574.72	627.16

Lower scores are better

Table 8 The quality of the realised berth plan and the standard deviation over 10 runs for the different recovery methods and different levels of uncertainty for the terminal 3

		0	6	12	18	24	30	36	42	48
DRCP	mean	159.15	268.03	453.55	661.01	739.74	849.58	854.97	1028.13	1150.06
	std	47.22	41.42	59.34	93.74	120.68	143.25	174.43	170.38	160.67
DRnCP	mean	205.86	278.43	489.4	689.75	747.07	866.22	945.54	1064.8	1165.45
	std	66.51	56.95	67.32	140.61	134.95	183.76	194.86	215.77	241.9
Baseline	mean	511.68	571.22	1492.8	2227.56	2701.32	3020.28	3098.68	3170.95	3834.59
	std	314.25	230.58	541.73	649.47	390.35	283.2	533.01	370.61	618.98

Lower scores are better

Table 9 The runtime in seconds for the three terminals combined for the different recovery methods and different levels of uncertainty

	0	6	12	18	24	30	36	42	48	48
DRCP	313.24	570.97	1062.82	1300.54	1693.32	1925.96	1977.42	2558.88	2228.53	926.04
DRnCP	172.02	171.34	188.92	218.46	226.63	244.82	237.80	237.52	221.01	176.21
Baseline	165.16	174.83	187.71	200.46	199.70	202.78	209.52	199.62	201.02	986.70

References

van Asperen E, Dekker R, Polman M, de Swaan Arons H (2003) Modeling ship arrivals in ports. In: Proceedings of 2003 winter simulation conference (WSC'03), vol 2, pp 1737–1744

Beldiceanu N, Contejean E (1994) Introducing global constraints in CHIP. Math Comput Model 20(20):97–123

Benedicto MAI, Morales RMG, Marino J, de los Santos F (2018) A decision support tool for port planning based on monte carlo simulation. In: Proceedings of the 2018 Winter Simulation Conf. (WSC'18), pp 2885–2896

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

Carette B (2016) Berth allocation in a stochastic environment. Master's thesis, Vrije Universiteit Amsterdam

Castilla-Rodríguez I, Izquierdo CE, Melián-Batista B, Aguilar RM, Moreno-Vega JM (2020) Simulation-optimization for the management of the transshipment operations at maritime container terminals. Expert Systems with Applications 139. https://doi.org/10.1016/j.eswa.2019.112852

Cimpeanu R, Devine MT, O'Brien C (2017) A simulation model for the management and expansion of extended port terminal operations. Trans Res E Logist Transpn Rev 98:105–131

Clausen U, Kaffka J (2016) Development of priority rules for handlings in inland port container terminals with simulation. J Simul 10(2):95–102

Cordeau J, Laporte G, Legato P, Moccia L (2005) Models and tabu search heuristics for the berth-allocation problem. Transp Sci 39(4):526–538. https://doi.org/10.1287/trsc.1050.0120

Ding Y, Jia S, Gu T, Li C (2016) SGICT builds an optimization-based system for daily berth planning. Interfaces 46(4):281–296. https://doi.org/10.1287/inte.2016.0848

Dragović B, Tzannatos E, Park N (2017) Simulation modelling in ports and container terminals: literature overview and analysis by research field, application area and tool. Flexib Serv Manufact J. https://doi.org/10.1007/s10696-016-9239-5

Freuder EC (1997) In pursuit of the holy grail. Constraints Int J 2(1):57–61. https://doi.org/10.1023/A:1009749006768

Fukunaga AS (2009) Search spaces for min-perturbation repair. In: Proceedings of 15th international conf. on principles and practice of constraint programming (CP'09), Springer, Lecture Notes in Computer Science, vol 5732, pp 383–390

Groenveld R (1983) Harborsim, a generally applicable harbour simulation model. Tech. Rep. Internal report 1983-1, TU Delft, Section Hydraulic Engineering

Hassan SA (1993) Port activity simulation: an overview. ACM SIGSIM Simul Digest 23(2):17–36

Hendriks M, Armbruster D, Laumanns M, Lefeber E, Udding J (2012) Strategic allocation of cyclically calling vessels for multi-terminal container operators. Flex Serv Manuf J 24(3):248–273

Hendriks MPM, Lefeber E, Udding JT (2013) Simultaneous berth allocation and yard planning at tactical level. OR Spectrum 35(2):441–456

Huang SY, Hsu WJ, Fang H, Song T (2013) A marine traffic simulation system for hub ports. In: Proceedings of 1st ACM SIGSIM conference on principles of advanced discrete simulation, pp 295–304

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

Imai A, Yamakawa Y, Huang K (2014) The strategic berth template problem. Trans Res E Logist Trans Rev 72:77–100. https://doi.org/10.1016/j.tre.2014.09.013

Iris Ç, Lam JSL (2019) Recoverable robustness in weekly berth and quay crane planning. Trans Res B Methodol 122:365–389

Iris Ç, Lalla-Ruiz E, Lam JSL, Voß S (2018) Mathematical programming formulations for the strategic berth template problem. Comput Ind Eng 124:167–179

Jia S, Li CL, Xu Z (2020) A simulation optimization method for deep-sea vessel berth planning and feeder arrival scheduling at a container port. Trans Res B Methodol 142:174–196. https://doi.org/10.1016/j.trb.2020.10.007

Jonker T, Duinkerken MB, Yorke-Smith N, de Waal A, Negenborn RR (2019) Coordinated optimization of equipment operations in a container terminal. Flexible Serv Manufact J. https://doi.org/10.1007/s10696-019-09366-3

Kizilay D, Eliiyi DT, Hentenryck PV (2018) Constraint and mathematical programming models for integrated port container terminal operations. In: Proceedings of 15th international conf. on integration of constraint programming, artificial intelligence, and operations research (CPAIOR'18), Springer, Lecture Notes in Computer Science, vol 10848, pp 344–360

Li S, Negenborn RR, Lodewijks G (2014) A distributed constraint optimization approach for vessel rotation planning. In: Proceedings of 5th international conference on computational logistics (ICCL'14), Springer, Lecture Notes in Computer Science, vol 8760, pp 61–80

Liu C, Xiang X, Zheng L (2020) A two-stage robust optimization approach for the berth allocation problem under uncertainty. Flex Serv Manuf J 32:425–452

Lv X, Jin JG, Hu H (2020) Berth allocation recovery for container transshipment terminals. Maritime Policy Manage 47(4):558–574. https://doi.org/10.1080/03088839.2020.1725672

Ma HL, Chung SH, Chan HK, Cui L (2019) An integrated model for berth and yard planning in container terminals with multi-continuous berth layout. Ann Oper Res 273(1–2):409–431

Nishi T, Okura T, Lalla-Ruiz E, Voß S (2020) A dynamic programming-based matheuristic for the dynamic berth allocation problem. Ann Oper Res 286(1):391–410

Pachakis D, Kiremidjian AS (2003) Ship traffic modeling methodology for ports. J Waterw Port Coast Ocean Eng 129(5):193–202

Perron L, Furnon V (2019) OR-Tools. https://developers.google.com/optimization/

Pesant G (2014) A constraint programming primer. EURO J Comput Optim 2(3):89–97. https://doi.org/10.1007/s13675-014-0026-3

Policella N, Smith SF, Cesta A, Oddi A (2004) Generating robust schedules through temporal flexibility. In: Proceedings of 14th international conference on automated planning and scheduling (ICAPS'04), pp 209–218, http://www.aaai.org/Library/ICAPS/2004/icaps04-026.php

Puget JF, Shaw P (2014) Background and theory of constraint programming: Background and history. In: 39th LNMB conference on the mathematics of operations research, Lunteren, The Netherlands, www.lnmb.nl/conferences/2014/programlnmbconference/

Qin T, Du Y, Sha M (2016) Evaluating the solution performance of IP and CP for berth allocation with time-varying water depth. Transp Res E Logist Transp Rev 87:167–185. https://doi.org/10.1016/j.tre.2016.01.007

Schepler X, Absi N, Feillet D, Sanlaville E (2019) The stochastic discrete berth allocation problem. Euro J Transp Logist 8(4):363–396. https://doi.org/10.1007/s13676-018-0128-9

Schiendorfer A, Knapp A, Anders G, Reif W (2018) MiniBrass: soft constraints for MiniZinc. Constraints 23(4):403–450. https://doi.org/10.1007/s10601-018-9289-2

Song Y, Zhang J, Liu M, Chu C (2019) The berth allocation optimisation with the consideration of time-varying water depths. Int J Prod Res 57(2):488–516. https://doi.org/10.1080/00207543.2018.1449975

Thiers GF, Janssens GK (1998) A port simulation model as a permanent decision instrument. Simulation 71(2):117–125

Turan HH, ElSawah S, Ryan MJ (2020) A long-term fleet renewal problem under uncertainty: a simulation-based optimization approach. Expert Syst Appl 145:113158. https://doi.org/10.1016/j.eswa.2019.113158

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

Wadhwa LC (1992) Planning operations of bulk loading terminals by simulation. J Waterw Port Coast Ocean Eng 118(3):300–315

Wallace M (2020) Building decision support systems—using MiniZinc. Springer, Cham

Wawrzyniak J, Drozdowski M, Sanlaville E (2020) Selecting algorithms for large berth allocation problems. Eur J Oper Res 283(3):844–862

Xiang X, Liu C (2021) An almost robust optimization model for integrated berth allocation and quay crane assignment problem. Omega p 102455, https://doi.org/10.1016/j.omega.2021.102455

Yang J, Gao H, Liu W, Liu T (2016) A novel strategy for berth and quay crane allocation under disruption in container terminal. Int J Innovat Comput Inform Control 12(5):1535–1551

Yeo GT, Roe M, Soak SM (2007) Evaluation of the marine traffic congestion of north harbor in Busan port. J Waterw Port Coast Ocean Eng 133(2):87–93

Zampelli S, Vergados Y, Schaeren RV, Dullaert W, Raa B (2013) The berth allocation and quay crane assignment problem using a CP approach. In: Schulte C (ed) Proceedings of 19th international conference principles and practice of constraint programming, CP 2013, Uppsala, Sweden, September 16-20, 2013. Springer, Lecture Notes in Computer Science, vol 8124, pp 880–896, https://doi.org/10.1007/978-3-642-40627-0_64

Zeng Q, Hu X, Wang W, Fang Y (2011) Disruption management model and its algorithms for berth allocation problem in container terminals. Int J Innovat Comput Inform Control 7(5):2130–2142

Zhang Q, Zeng Q, Yang H (2016) A lexicographic optimization approach for berth schedule recovery problem in container terminals. Transport 31(1):76–83

Zhen L (2013) Yard template planning in transshipment hubs under uncertain berthing time and position. J Oper Res Soc 64(9):1418–1428. https://doi.org/10.1057/jors.2012.108

Zhen L, Chew EP, Lee LH (2011) An integrated model for berth template and yard template planning in transshipment hubs. Transp Sci 45(4):483–504

Zhen L, Liang Z, Zhuge D, Lee LH, Chew EP (2017) Daily berth planning in a tidal port with channel flow control. Transp Res B Methodol 106:193–217. https://doi.org/10.1016/j.trb.2017.10.008

Zúñiga ER, Moris MU, Syberfeldt A, Fathi M, Rubio-Romero JC (2020) A simulation-based optimization methodology for facility layout design in manufacturing. IEEE Access 8:163818–163828. https://doi.org/10.1109/ACCESS.2020.3021753

Authors and Affiliations

Jaap-Jan van der Steeg[1,2] · Menno Oudshoorn[2] · Neil Yorke-Smith[1]

Jaap-Jan van der Steeg
J.J.vanderSteeg@student.tudelft.nl

Menno Oudshoorn
m.oudshoorn@macomi.nl

[1]   Delft University of Technology, Delft, The Netherlands

[2]   Macomi B.V., Rotterdam, The Netherlands