Received 27 February 2024; revised 23 April 2024; accepted 7 June 2024. Date of publication 12 June 2024; date of current version 19 July 2024.

Digital Object Identifier 10.1109/OJITS.2024.3413197

Container Relocation and Retrieval Tradeoffs Minimizing Schedule Deviations and Relocations

ROBERT KLAR [1,2] (Graduate Student Member, IEEE), ANDERS ANDERSSON [3], ANNA FREDRIKSSON [1], AND VANGELIS ANGELAKIS [1] (Senior Member, IEEE)

[1]Department of Science and Technology, Linköping University, Campus Norrköping, 60 174 Norrköping, Sweden

[2]Department of Traffic Analysis and Logistics, Swedish National Road and Transport Research Institute (VTI), 581 95 Linköping, Sweden

[3]Department of Vehicle Systems and Driving Simulation, Swedish National Road and Transport Research Institute (VTI), 581 95 Linköping, Sweden

CORRESPONDING AUTHOR: R. KLAR (e-mail: robert.klar@liu.se)

This work was supported by the Trafikverket Sweden as part of the Triple F (MODIG-TEK) Project under Grant 2019.2.2.16.

ABSTRACT Ports are striving to improve operational efficiency in the context of constantly growing volumes of trade. In this context, port terminal storage yard operation is key, since complexity and poor coordination lead to containers stacked without consideration of retrieval schedules, resulting in time- and energy-consuming reshuffling operations. This problem, known as the block relocation (and retrieval) problem (BRP), has recently gained considerable attention. Indeed, there are promising solutions to the BRP. However, the literature views the problem in isolation, optimizing one operational parameter for one of the many port stakeholders. This often leads to efficiency losses since port processes involve different stakeholders and port parts. In this work, we explicitly focus on scheduling trucks for pick-up for hinterland distribution. Appointments are often postponed in order to minimize reshuffling operations, leading to losses for the transport forwarders and decreasing the competitiveness of the port. We discuss the trade-off between minimizing container reshuffling operations while maintaining scheduled time windows for container retrieval. We describe the multi-objective optimization problem as a weighted sum of the two objectives. Given the complexity of the problem, we also present a greedy heuristic. Our results indicate that the number of schedule deviations can be reduced without significantly affecting the number of relocations compared to solutions that consider only the latter. Ideally, a weighting of 0.4 and 0.6 should be applied, reflecting schedule deviations and relocations, respectively, to achieve the highest joint optimization potential. This demonstrates that in complex environments, such as ports, with multiple interacting stakeholders and processes, coordination of solutions yields significant benefits.

INDEX TERMS Container relocation problem, ports, optimization, digital twins, schedule deviations.

I. INTRODUCTION

ONE OF the key functions in ports is to provide a temporary container storage buffer to match the difference in scale between sea- and land- transport of goods [1]. With the increasing volume of international trade and the growing capacity of cargo vessels, ports are pressed to store containers efficiently until they are further transported to their final destination [2]. On average more than 80% of the containers are transferred to the storage yard before they are further delivered to their final destination [3]. Therefore, the optimization of container stack logistics within the port storage yard has become one of the most important topics of research in the optimization of container terminal processes.

Despite increasing container throughput, many ports have little potential for expansion of port facilities due to land use conflicts between cities and ports [4]. The lack of ability to increase the land use of port facilities, combined with increasing numbers of containers, requires not only optimization of stacking techniques, but also the ability to effectively and efficiently serve the trucks coming to retrieve containers for further distribution inland. This requires above

The review of this article was arranged by Associate Editor Fangfang Zheng.

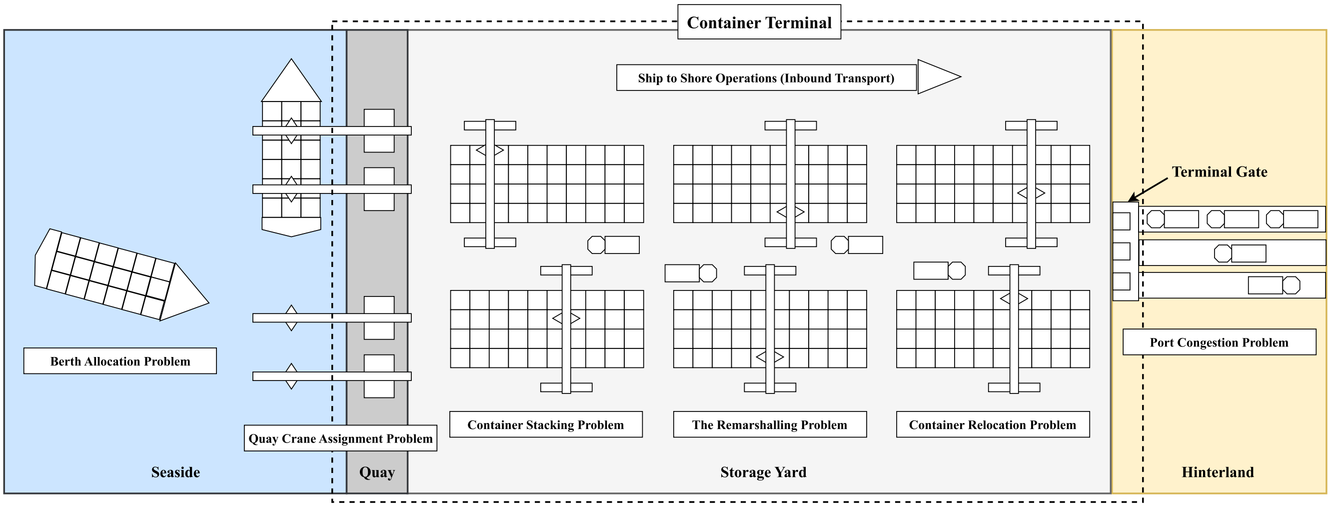


FIGURE 1. The interconnection of port-related problems.

all an uninterrupted flow of terminal gate operations. Gate operations comprise essential tasks for terminal operations, including identifying vehicles, cargo, and drivers, inspecting for damage, providing relevant information to the truck driver (such as where to deliver or retrieve cargo within the terminal), and conducting security checks [5].

In Figure 1 we sketch a cargo port, denoting some of the most common inter-related problems in all areas from seaside to hinterland. These challenges span from berth allocation [6] to efficient gate operation management [7]. As as the port is a complex system of systems, catering numerous processes involving multiple actors, ports must align the interests of a wide range of stakeholders [8]. Within the Container Terminal, to make the best use of limited space available for the Storage Yard, containers are typically stacked side by side and on top of each other. Stacks are aligned to form bays and blocks, as this configuration optimizes the space utilization and allows for crane operations. Figure 1 presents a Storage Yard configuration of six blocks consisting of 10 bays each. However, adopting such storage policy creates a trade-off between space saving and handling efforts for loading and unloading operations [9].

Due to the typically poor coordination across stakeholders, and due to the increasing pace of operations, a common assumption for the storage yard that the exact retrieval order of containers is largely unknown at the time of stacking. Indeed, it is not uncommon that, at retrieval time, one or more containers will be on top of a target container. With, the most commonly used container handling equipment for storage yard being the Gantry crane, which can access only the topmost containers of a stack, blocking containers will have to be relocated before retrieval. Since these relocation moves are very costly (time and energy), finding a sequence of container moves to retrieve containers in a given time frame with as few moves as possible is central in storage yard efficiency. Moreover, since crane throughput is one of the indicators of efficiency of a container terminal, reducing undesired relocations is critical to container operations decisions [10].

Another critical aspect of efficient Container Terminal management is the planning of Terminal Gate operations, which interfacing between the hinterland and the yard grants external actors access to the port. The number of trucks that can enter the port per time window and an estimate of the truck turnaround time are necessary to jointly optimize yard operations and gate queues, which has been identified as one of the key challenges in gate operations planning [11]. Inefficient gate operations lead to congestion on the land side of ports and, in the worst case, long queues of trucks on the roads leading to the port gates. Reducing truck congestion is therefore also important for the cities surrounding ports [12]. A recent comprehensive literature review of port emissions strategies identified reducing port congestion as a major contributor to land transport measures [13]. Ports therefore aim to reduce truck emissions by reducing congestion both outside, at the gate and inside terminals. Furthermore, from a supply chain perspective, disruptions to the flow of trucks should also be avoided, as uncertainty leads to increased costs in the transportation chain, which often represents a major bottleneck and is responsible for overall 60% of the cost of global maritime transportation [14].

The lack of coordination between the various actors involved in container stacking and retrieval therefore reduces the attractiveness of ports by causing delays and increased costs in the forwarding of goods. Since port operations are intertwined, overall port efficiency is not the sum of isolated port operations, but rather how well they are coordinated. Collaborative decision-making among port stakeholders, including terminal operators, ship and truck operators, rail shippers, industry associations, and government agencies, is central to effective and efficient coordination. Joint decision-making can be supported by digital twins, which enabling data exchange and shared views on efficiency losses can become a tool to improve multi-stakeholder collaboration [15]. However, existing optimization models of port activities usually treat one problem at a time and thus

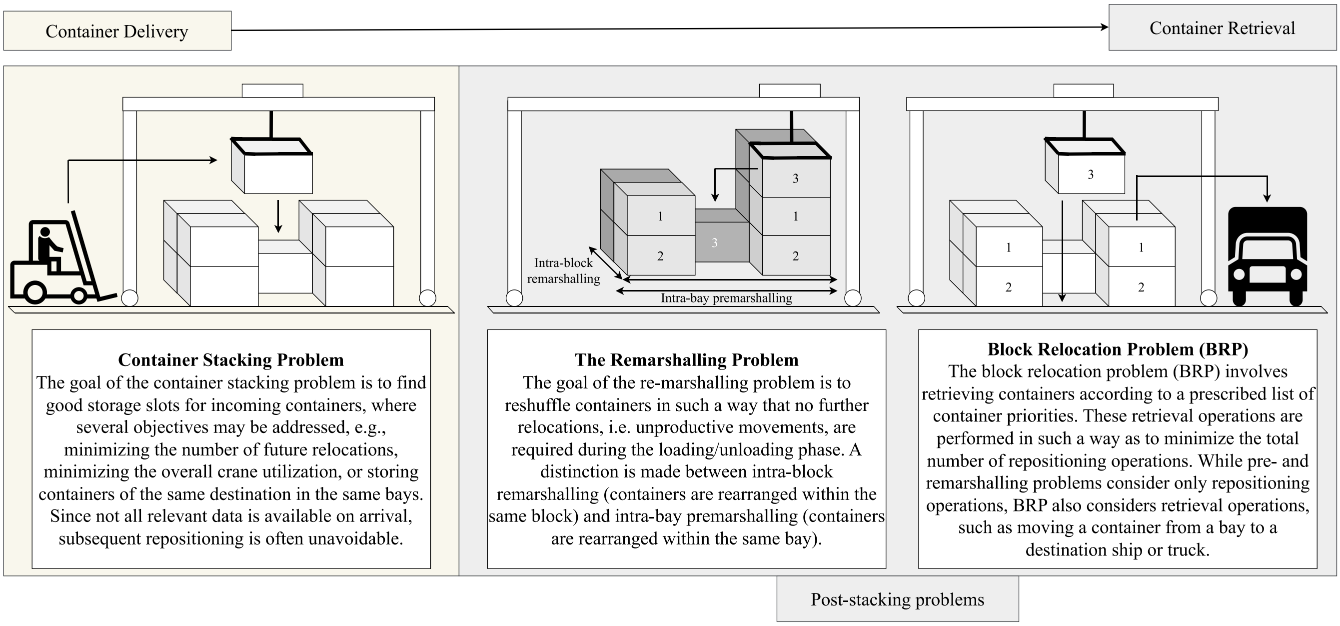


FIGURE 2. Classification of container stacking and container rehandling problems in the storage yard.

do not provide the necessary basis for coordination between stakeholders.

The goal of this paper is to align the container relocation problem with reliable truck scheduling to improve not a single isolated process, but the overall efficiency of the port. More specifically, this paper contributes to the terminal operations optimization literature by

- Providing thorough analysis and discussion of container stacking and retrieval operations at the port, leading to the identification of the importance of aligning the relocation and retrieval of containers with truck scheduling to reduce port congestion.
- Proposing a multi-objective optimization model to obtain the optimum for different trade-offs of minimizing container relocations and schedule deviations, and a heuristic for solving the problem in a timely manner.
- Analysing and discussing of different configurations and use cases.

The paper is organized as follows. In Section II, we discuss the three major storage yard container stacking and retrieval problems, identifying the container relocation problem as most relevant (Section II-A), yielding in a discussion of previously related work (Section II-B). In subsequent Section III we then motivate why focusing on truck appointments is of value and thus complements previous works. Section IV includes the mathematical formulation of the problem (Section IV-A), our simulation framework (Section V) and our proposed algorithm (Section V-B). Section VI presents the results of the optimization model (Section VI-A) and our proposed algorithm (Section VI-C) for several bay configurations. The results are then discussed in Section VII under consideration of practical use cases and policy implications.

II. BACKGROUND

Storage space assignment is about finding the best allocation of containers to storage spaces, with the goal of reducing the cycle time of storage yard operations (i.e., the time required for storage, retrieval, and transfer). The suitability of a storage space allocation depends on the availability and quality of information on arrival and departure times for the handled import, export and transshipment containers [16]. Due to often insufficient information regarding future retrieval times and the duration of the containers in the storage yard as well as disruptions (such as pick up delays, which can happen doe to road congestion or extended (un)loading times) and other unforeseen events, it is often necessary to reallocate the containers.

Drawing on the work of Caserta et al. in [17], Figure 2 presents a classification of container stacking problems from delivery to retrieval, distinguishing between proactive stacking and post-stacking. While the container stacking problem aims to find the best storage slots for incoming containers, both the remarshalling and the block relocation and retrieval problem aim to relocate containers in such a way that no or little reshuffling when retrieving the containers is needed. Both the remarshalling and the block relocation and retrieval problem assume that no containers are entering the bay while solving the problem. Compared to the remarshalling problem, the block relocation and retrieval problem allows to retrieve and relocate containers, leading to reduction of the number of containers in the bay. Although new technologies, such as advances in information and communication technology, new equipment, terminal

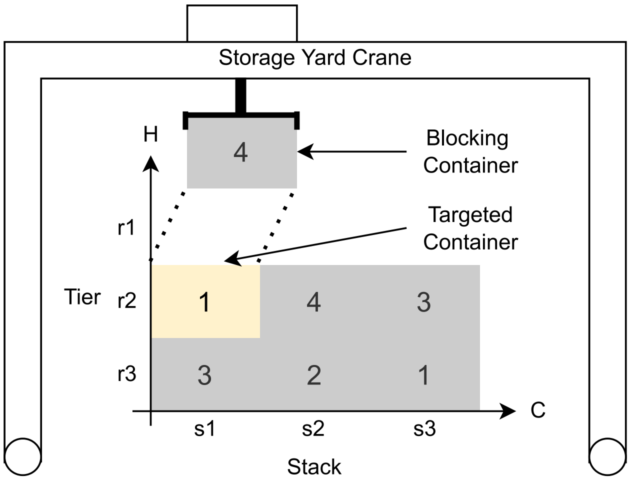


FIGURE 3. **Illustration of a bay in the storage yard consisting of 3 stacks and 3 tiers. The number in each block denotes the retrieval priority.**

redesign, etc., lead to increased terminal efficiency, the efficiency of container terminal operations can also be improved by optimizing the way these operations are performed, especially in the case of stacking or retrieving containers in the storage yard [18]. In the following subsections, the problem of block relocation and retrieval is explored in more depth, as this forms the basis for the work in this paper.

A. THE BLOCK RELOCATION AND RETRIEVAL PROBLEM (BRP)

The block relocation and retrieval problem, often also simply termed block or container relocation problem, is considered as one of the most challenging problems, as it affects the terminal's stacking strategies as well as the time needed for the reshuffling operations. According to a recent survey by Kizilay and Eliiyi in [19] considering 117 yard side publications, the number of container moves is considered as the most significant performance criteria, corresponding to 45 out of 117 yard side publications. The containers that are already stored in the storage area of the yard have already predefined priorities or departure times. It is further assumed that no further containers are obtained while the block relocation and retrieval problem is performed. The goal of the classic block relocation and retrieval problem is to retrieve all containers with respect to their departure times with a minimum number of relocations [20].

The problem is presented in Figure 3. It illustrates an example of a layout with 7 containers stored in 3 stacks $(s1, s2, s3)$ with a height limit of 3 tiers $(r1, r2, r3)$. Consequently, the containers are stored as a 2D array and can be accessed via their Tier and Stack indices. Containers are further indexed by retrieval time or priority and must be retrieved in ascending order. In the first turn, Container $(r1, s1)$ with priority 4 (marked as blocking container) must be relocated to access the (yellow) target container $(r2, s1)$ with priority 1. After container $(r2, s1)$ has been retrieved, container $(r2, s3)$ needs to be relocated to $(r2, s1)$ in order to access container $(r3, s3)$. Afterward, the containers $(r1, s2)$ and $(r2, s2)$ must be relocated to $(r3, s3)$ and $(r2, s3)$ respectively. Afterwards, no further relocations are needed and the remaining containers can be retrieved in the sequence $(r3, s2)$, $(r2, s1)$, $(r3, s1)$, $(r2, s3)$, $(r3, s3)$ according to their priority. Finally, a total of four relocations and 11 container moves are required to empty the bay.

In the restricted version of the block relocation and retrieval problem only the top most block above the target container can be removed [21]. According to the restricted version, only the topmost container within the slots (r1, s1) as shown in Figure 3 or (r2, s3) as these two containers are blocking the target containers with priority 1. In the unrestricted version, all topmost containers, including (s2, r2) could be relocated.

B. PREVIOUS WORK

The block relocation and retrieval problem (BRP) has been studied extensively in the literature from many perspectives since its introduction by [22] as a combinatorial optimization problem with the objective to retrieve a sequence of containers while relocating blocking containers in as few moves as possible.

Depending on the application scenario, the retrieval priority of containers in the container relocation problem is either distinct or grouped [10]. The distinct BRP goes seamlessly with container handling between the yard and the seaside [23], when ships are loaded or unloaded according to the stowage plan, in which a specific slot on the ship is specified for each container [24]. The grouped version of the BRP, on the other hand, is suitable for transporting containers between the yard and the hinterland, as they are delivered or picked up either in groups by rail or by several trucks within the same time window [25]. Such a scenario considering a truck scheduling system is studied by Ku and Arthanari in [26], who propose a stochastic dynamic programming model to calculate the minimum expected number of relocations for a stack of containers that all have departure time windows.

Recent approaches to solving the BRP include (1) heuristic algorithms [27], such as greedy algorithms [25], [28] or genetic algorithms [3], aimed at finding a good solution in a timely manner without necessarily guaranteeing optimality; (2) model-based approaches [29] aimed at creating computational models of the problem to find optimal solutions within the constraints of the model; (3) exact methods [21] that systematically explore the entire solution space to identify the optimal solution, such as branch-and-bound algorithms [30]; and (4) machine learning methods, such as random forests [31] or deep reinforcement learning [32].

Due to the high complexity of the BRP and the fact that the restrictive version is NP-hard [33], the BRP is often considered only in isolation, instead of considering the interconnected operations of the terminal that directly affect the relocation plan determined from the BRP [34]. Previous work involving other interrelated terminal processes include

TABLE 1. Comparison of papers.

Year	Reference	Objectives	Applied Methodology	Conclusions
2019	Zeng et al. [38]	Reducing the need for container rehandling by taking into account partial truck arrival information.	A binary integer programming model considering container retrieval is proposed to minimize the total number of relocations. Further heuristic approaches are suggested.	Jointly optimizing the strategy for relocating containers and the pickup sequence can reduce the total number of rehandles without causing excessive delays.
2021	Hsu et al. [41]†	Coordinating yard crane and yard truck scheduling to achieve better overall performance for a container terminal.	A MILP model is proposed to simultaneously model yard crane and yard truck scheduling. Additionally, a framework for developing hybrid approaches is suggested.	Coordinating yard cranes and yard trucks results in better overall performance for a container terminal. Hybrid approaches are a promising method to achieve this.
2022	Azab and Morita [34]*	Solving the block relocation problem under consideration of truck appointment scheduling.	Two binary integer programming models are proposed and evaluated on data of previous BRP studies.	Container handling operations at the terminal yard can be improved by incorporating truck appointment scheduling into the BRP.
The present paper		Joint minimization of crane relocation movements and schedule deviations. Examination of various use cases and weightings, reflecting the need of different port stakeholders.	An binary integer programming problem is proposed with the objective of minimizing two weighted sums. A greedy heuristics is provided to jointly optimize both problems in a timely manner.	The highest joint optimization potential is achieved by expressing the problem as two weighted sums, with a weighting of 0.6 for minimizing crane relocations and 0.4 for schedule deviations.

* Paper [25] extends the authors' work by proposing a bi-level lexicographical function to achieve the terminals' goal of minimizing container relocations while secondarily minimizing schedule deviations

† Paper [42] expands on the authors' previous work by presenting a MILP model and hybrid approaches to achieve balanced energy consumption and terminal efficiency in an automated container terminal.

the integration of the stowage plan for export containers for ships and yards [23], [35], [36], the linking of the BRP with yard crane scheduling [37], or the integration of container pickup sequence based on truck arrival information with the BRP to reduce container relocations under the consideration of truck appointment scheduling [25], [34], [38], [39], [40].

The resulting block relocation and retrieval problem under consideration of truck appointment scheduling (BRPAS) as proposed by Azab and Morita, is extensively studied in [34]. In their subsequent paper [25], Azab and Morita present a bi-level lexicographical function. The primary objective is to minimize crane relocation movements, while the secondary objective is to minimize the average appointment schedule deviations.

Other related work, which does not directly address BRP but links yard operations with truck retrieval or scheduling, is presented below. In paper [41], Hsu et al. present four different hybrid approaches developed to simultaneously address yard crane scheduling and yard truck scheduling for export containers in the container storage yard. In a follow-up paper [42], Hsu et al. propose a hybrid model to address the need for balancing operational efficiency and energy consumption in automated container terminals. The model aims to solve the integrated scheduling problem of automated quay cranes, automated lift vehicles, and automated stacking cranes in such terminals. Both papers provide a mixed integer linear programming (MILP) formulation for the integrated problems.

Table 1 provides an overview of recent related papers that demonstrate the potential and the need to further explore the joint coordination and optimization of BRP with truck scheduling. Although these papers address the linkage of these problems, their focus is still on the BRP by considering truck retrieval appointments as an input parameter [34], [38] or by approaching the joint optimization solely from the terminal operator's perspective, thus primarily minimizing quay crane relocation movements [25]. This paper therefore builds on these recent works by considering the problem not only from the terminal operator's perspective, but also from the truck haulier's perspective, or from mixed perspectives expressed by different weightings, as introduced in Section II-A. The motivation behind the transition from container relocation minimization to multi-objective optimization with the goal of minimizing both container relocation and schedule deviation using different weightings is described in more detail in Section III.

III. MOTIVATION FOR JOINT OPTIMIZATION OF BRP AND SCHEDULE RELIABILITY

Gate and yard congestion is a major cause of disruptions in container ports, preventing trucks from moving freely and causing bottlenecks that limit port productivity [43]. According to Notteboom et al. in [2], the performance of a terminal's storage yard is significantly impacted by the utilization of the yard cranes, the average yard dwell time, the average truck or railcar turnaround times, and the average gate waiting time for trucks. While the first two are closely related to the container relocation problem, since minimization of unproductive relocations leads to more efficient yard crane utilization and reduction of container dwell time in the yard, the latter two are a direct product of efficient truck scheduling.

More and more ports are thus using truck appointment systems to better allocate trucks. When using truck appointment systems, trucking companies first enter the container information into the booking system and then select the most suitable option from the time periods available in the system for the appointment, provided that the maximum number of trucks that can be accepted for the period is not already completely exhausted [44]. Prior research indicates that the presence and use of a truck scheduling systems can lead to significant improvements in the port's system performance [45].

Although recent studies highlight the efficiency gains of truck appointment systems, many ports apply a same-day appointment policy, which means that trucks can make appointments and arrive on the same day. This makes it more difficult to know truck arrival times in advance and requires the terminal to deploy the equipment more expeditiously [44].

While previous BRP studies significantly increased the efficiency of yard crane utilization, the majority of these studies did not take into account the container pick-up times of the trucks, resulting in a loss of efficiency of the other performance indicators. Although truck appointments were included as an input parameter in the case of BRPAS studies [34], the primary goal of these studies was still to minimize container relocations.

In this paper, we thus combine the objectives of minimizing container relocations and schedule deviations to increase overall port efficiency. We term the resulting multi-objective optimization problem as Block Relocation and Schedule Reliability Problem (BRSRP).

IV. PROBLEM FORMULATION

This sections presents the problem formulation of the BRSRP problem to jointly minimize container relocations and truck schedule deviations to avoid unproductive crane movements, enable schedule reliability, and thus increase overall port efficiency. First, we present the mathematical formulation of the problem using integer programming. Then, we provide a brief overview of the problem's complexity and motivate the need of a heuristic to solve the problem in a timely manner.

A. THE BRSRP PROBLEM

To mathematically formulate the BRSRP problem, we depart from the BRPAS problem formulation of Azab and Morita in [34], using the same set of assumptions: (1) the initial bay configuration is known in advance; (2) all containers in the bay are picked up within the planning horizon, and no containers are received during the retrieval process; (3) each container has a predefined preferred pickup time window, which can be shifted depending on the tradeoff between relocations and schedule deviations; (4) depending on the use case, the weighting between minimizing container relocations and schedule deviations and the resulting tradeoff may change. We include one final assumption (5), which states that containers can be relocated from any slot in the bay, as in the unrestricted version of BRP [21].

The indices, parameters, and (decision) variables used for our problem formulation are defined in Table 2. For each container, a unique index $i \in \{1, \dots, N\}$ is used to identify each container within the bay. The bay layout consists of C stacks and H tiers, representing a 2D matrix by which each container i can be assigned to a slot (s, r), where stack $s \in \{1, \dots, C\}$ is indexed from left to right and tier $r \in \{1, \dots, H\}$ from bottom to top. Each container i is assigned a retrieval time p_i, which assigns each container to be retrieved within a time window t, where $t \in \{1, \dots, T\}$. In each time window t a limited number of trucks can be served based on the parameter L. To serve these L trucks, each aiming to retrieve a single container, a limited number of crane movements are available based on the parameter G, where each stage $k \in \{1, \dots, G\}$ represents a yard crane movement (to either relocate or retrieve containers), where $G \geq L$. If a container cannot be retrieved within the given time horizon, i.e., all crane movements k are deployed, either a remaining container must be retrieved in a later time window $t+1$ or one of the (blocking) containers is retrieved earlier than scheduled, provided it is within the time shift tolerance expressed by δ. The initial bay layout is defined by the parameter I_{isr}, which represents a binary encoding of the original stacking of the N containers spanning the bay. BRSRP Model:

$$\text{minimize} \quad z : \alpha\left(\sum_{i \in \mathrm{N}}\left(\sum_{s \in \mathrm{C}}\sum_{r \in \mathrm{H}}\sum_{k \in \mathrm{G}}\sum_{t \in \mathrm{T}}((|t - p[i]|) \cdot v[i, s, r, k, t])\right)\right) + \beta\left(\sum_{i \in \mathrm{N}}\sum_{s \in \mathrm{C}}\sum_{r \in \mathrm{H}}\sum_{k \in \mathrm{G}}\sum_{t \in \mathrm{T}} y[i, s, r, k, t]\right) \tag{1}$$

Subject to:

$$p[i] - \sum_{s=1}^{C}\sum_{r=1}^{H}\sum_{k=1}^{G}\sum_{t=1}^{T}(t \cdot v[i, s, r, k, t]) \leq \delta, \forall i \in \{1, \dots, N\} \tag{2}$$

$$\sum_{s=1}^{C}\sum_{r=1}^{H}\sum_{k=1}^{G}\sum_{t=1}^{T}(t \cdot v[i, s, r, k, t]) \leq \delta + p[i], \forall i \in \{1, \dots, N\} \tag{3}$$

$$\sum_{i=1}^{N}\sum_{s=1}^{C}\sum_{r=1}^{H}\sum_{k=1}^{G} v_{isrk}^{t} \leq L, \forall t \in \{1, \dots, T\} \tag{4}$$

$$\sum_{i=1}^{N}\sum_{s=1}^{C}\sum_{r=1}^{H} v_{isrk}^{t} + \sum_{i=1}^{N}\sum_{s=1}^{C}\sum_{r=1}^{H} x_{isrk}^{t} \leq 1, \quad \forall k \in \{1, \dots, G\}, t \in \{1, \dots, T\} \tag{5}$$

$$\sum_{i=1}^{N} x_{isrk}^{t} \leq \sum_{i=1}^{N}\left(u_{isrk}^{t} - u_{is(r+1)k}^{t}\right), \forall s \in \{1, \dots, C\}, \quad r \in \{1, \dots, H-1\}, k \in \{1, \dots, G\}, t \in \{1, \dots, T\} \tag{6}$$

$$\sum_{s'=1, s' \neq s}^{N}\sum_{r=1}^{H} y_{is'rk}^{t} \geq \sum_{r=1}^{H} x_{isrk}^{t}, \forall i \in \{1, \dots, N\},$$

TABLE 2. Notation list.

	Notation	Explanation
Parameters	N	Number of containers in the initial configuration of the bay
	C	Number of stacks
	H	Maximum allowed height of bay
	G	Maximum allowed container moves (either retrievals or relocations) per time window
	L	Maximum queue length at bay (appointments per time window)
	T	Number of time windows
	p_i	Scheduled container pickup time window, $i \in 1, ..., N$
	δ	Maximum allowed container pickup time window shift
	I_{isr}	Whether container i occupies slot (s,r) in the initial bay layout, $I_{isr} \in 0, 1$
Indices	i	Index of container, $i \in 1, ..., N$
	s	Index for stack, $s \in 1, ..., C$
	r	Index for tier, $r \in 1, ..., H$
	k	Index of the stage, $k \in 1, ..., G$ where each stage k represents one possible container move
	t	Index for time window, $t \in 1, ..., T$
Variables	u^t_{isrk}	$\begin{cases} 1, & \text{if container i occupies the slot (s, r) at stage k of time window t} \\ 0, & \text{otherwise} \end{cases}$
	x^t_{isrk}	$\begin{cases} 1, & \text{if container i is relocated from slot (s, r) at stage k of time window t} \\ 0, & \text{otherwise} \end{cases}$
	y^t_{isrk}	$\begin{cases} 1, & \text{if container i is relocated to slot (s, r) at stage k of time window t} \\ 0, & \text{otherwise} \end{cases}$
	v^t_{isrk}	$\begin{cases} 1, & \text{if container i is picked up from slot (s, r) at stage k during time widow t} \\ 0, & \text{otherwise} \end{cases}$

u^t_{isrk}, x^t_{isrk}, y^t_{isrk}, v^t_{isrk} are the decision variables.

$$s \in \{1, \ldots, C\}, k \in \{1, \ldots, G\}, t \in \{1, \ldots, T\} \quad (7)$$

$$\sum_{i=1}^{N} v^t_{isrk} + \sum_{i=1}^{N} y^t_{isrk} + \sum_{i=1}^{N} x^t_{isrk} \leq 1, \quad \forall s \in \{1, \ldots, C\}, k \in \{1, \ldots, G\}, t \in \{1, \ldots, T\} \quad (8)$$

$$u^1_{isr1} = I_{isr}, \forall i \in \{1, \ldots, N\}, s \in \{1, \ldots, C\}, r \in \{1, \ldots, H\} \quad (9)$$

$$u^t_{isrk+1} = u^t_{isrk} + y^t_{isrk} - x^t_{isrk} - v^t_{isrk}, \quad \forall i \in \{1, \ldots, N\}, s \in \{1, \ldots, C\}, r \in \{1, \ldots, H\}, k \in \{1, \ldots, G-1\}, t \in \{1, \ldots, T\} \quad (10)$$

$$u^t_{isr1} = u^{t-1}_{isrG} + y^{t-1}_{isrG} - x^{t-1}_{isrG} - v^{t-1}_{isrG}, \quad \forall i \in \{1, \ldots, N\}, s \in \{1, \ldots, C\}, r \in \{1, \ldots, H\}, t \in \{2, \ldots, T\} \quad (11)$$

$$\sum_{i=1}^{N} \sum_{r=1}^{H} \sum_{k'=k+1}^{G} u^t_{isrk'} + \sum_{i=1}^{N} \sum_{r=1}^{H} \sum_{k'=k+1}^{G} \sum_{t'=t+1}^{T} u^{t'}_{isrk'} \leq G * T\left(1 - \sum_{s=1}^{C} \sum_{r=1}^{H} v^t_{isrk}\right), \quad \forall i \in \{1, \ldots, N\}, k \in \{1, \ldots, G\}, t \in \{1, \ldots, T\} \quad (12)$$

$$\sum_{s=1}^{C} \sum_{r=1}^{H} \sum_{k=1}^{G} \sum_{t=1}^{T} v^t_{isrk} = 1, \forall i \in \{1, \ldots, N\} \quad (13)$$

$$\sum_{i=1}^{N} u^t_{isrk} \leq 1, \forall s \in \{1, \ldots, C\}, r \in \{1, \ldots, H\}, k \in \{1, \ldots, G\}, t \in \{1, \ldots, T\} \quad (14)$$

$$\sum_{s=1}^{C} \sum_{r=1}^{H} u^t_{isrk} \leq 1, \forall i \in \{1, \ldots, N\}, k \in \{1, \ldots, G\}, t \in \{1, \ldots, T\} \quad (15)$$

$$u^t_{isrk}, x^t_{isrk}, y^t_{isrk}, v^t_{isrk} \in \{0, 1\}, \quad \forall i \in \{1, \ldots, N\}, s \in \{1, \ldots, C\}, r \in \{1, \ldots, H\}, k \in \{1, \ldots, G\}, t \in \{1, \ldots, T\} \quad (16)$$

In case only postponements of scheduled truck appointments should be allowed, constraint (2) should be replaced by constraint (17), which is presented below.

$$\sum_{s=1}^{C} \sum_{r=1}^{H} \sum_{k=1}^{G} \sum_{t=1}^{T} (t \cdot v[i, s, r, k, t]) \geq p[i], \forall i \in \{1, \ldots, N\} \quad (17)$$

The objective function (1) here is to minimize the weighted sum of container relocations and schedule deviation given a predefined schedule with truck appointments based on two parameters α and β. If both aspects should be equal, the pair of weight values $\alpha = 0.5$ and $\beta = 0.5$ are suitable parameters. By reducing δ, the port can provide more reliable time slots by reducing the container pickup time deviation from the preferred pickup time submitted by the trucking company. Constraints (2) and (3) ensure that a containers' scheduled retrieval can not exceed a deviation of δ time windows. In constraint (4), assuming that a truck can pick up one container, the queue length at the bay is limited to L trucks. Constraint (5) controls is limiting the number of container moves per time window to the value of the parameter G (Crane capacity). This constraint is also used to define the stage k at which each container move can take place. Constraints (6), (7), and (8) describe the relocation process. Under constraint (6), when relocating a container, at any time the topmost blocking container must be relocated, before any below it. Constraint (7)) ensures that a relocated container will go to a different stack. In constraint (8), when a container is moved from or to a slot, it is either relocated or retrieved. This constraint prevents transitive and cyclic container moves within the bay. Constraints (9), (10), and (11) are used to update the bay layout when containers are moved: Constraint (9) initiates the bay layout before the first container move, constraint (10) updates the bay layout from one stage to the next within a time window, and constraint (11) updates the layout transition from the last stage of time window $t-1$ to the first stage in the next time window t. (Constraints (12)) through (15) are logical constraints. Constraint (12) ensures that if a container is retrieved, it can no longer occupy any slot in the configuration. Constraint (13) guarantees that each container must be retrieved. Constraint (14) states that each slot must be occupied by at most one container; similarly, constraint (15) specifies that a container cannot be in more than one slot. Finally, the constraints in (16) define the binary domain of the decision variables.

B. COMPLEXITY

The complexity of our BRSRP problem follows that of [33], in which the authors prove the BRP to be NP-hard. According to previous studies, a state-of-the-art formulation of the BRP can take hours to derive a feasible relocation plan for a rather small instance [46]. This is also confirmed in our study. The complexity of solving our proposed BRSRP problem, which extends the BRP problem by including the second objective of minimizing schedule deviations, is further increased. Specifically, Figure 5 reveals that the time complexity of the problem further increases when both problems are considered similarly or when the focus is on minimizing schedule deviations. This motivates us to develop a heuristic, which is presented in the following section.

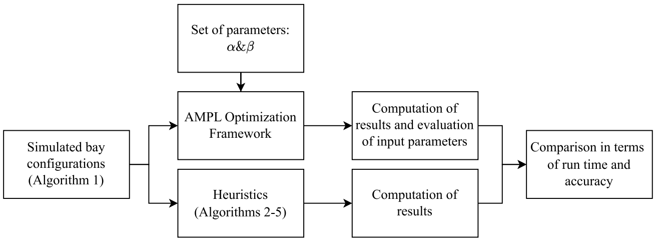


FIGURE 4. Overview of the applied methodology of this paper.

V. SIMULATION SETUP

In this section, we present our simulation procedure, including data generation, and demonstrate our heuristic, consisting of several algorithms, which is able to solve both problems in a time-efficient way without significant loss of accuracy.

Figure 4 presents the overall simulation framework of our paper. All experiments are run on a computer with an Intel Core i7-8565U CPU @ 1.80GHz, 1992Mhz, 4 cores, 8 logics and 16GB RAM. The Gurobi solver was utilized in AMPL to obtain optimal solutions for various problems and parameters outlined in Section IV-A.

Lacking real data of bay configurations (placement of containers in the bay) and collection schedules, we have generated a large number of such random container placements and allocated each a collection time-window. The leftmost box of Figure 4 introduces this data generation process as an integral part of our simulation framework, while Algorithm 1, detailed in the following subsection, describes the process itself. Results presented are averages as, for example in previous similar works we have (e.g., see [47] and references therein). Moreover, in our results we discuss also extreme cases of bay configurations, to delimit the expected performance bounds.

A. INPUT DATA GENERATION

To consider a large number of potential bay configurations, a number of 100 random bay layouts were generated for each simulation. Given a predefined number of containers N, stacks C, tiers H, pick-up appointments per time window L and time windows T, a random bay layout is created by filling the bay (later also referred to as config) from the bottom-left to the top-right whereby each time window t is represented L times as a container in the bay. Finally, the resulting 2D matrix is complemented with the parameter values to obtain a.dat file suitable for AMPL,which will also be used later (and converted back to a Python-readable format) to test and compare with our proposed heuristics.

B. HEURISTIC SOLUTION

The first important decision in the heuristic we propose is to decide which of the target containers from the set of containers belonging to the current time frame to retrieve first, and then to determine the retrieval sequence of the remaining containers.

Algorithm 1 Generate Input Data

```
function GENERATE_PVALUES(N, T, L)
    possible_values ← list(range(1, T + 1)) × L
    Shuffle possible_values to randomize them
    p_values ← first N values of possible_values
    return p_values
end function

function GENERATE_INPUT_DATA(N, C, H, T, G, L, δ)
    input_data ← empty string
    p_values ← GENERATE_PVALUES(N, T, L)
    Add parameter declarations to input_data
    max_slots ← H × C
    for i from 1 to N do
        Add container data to input_data
        if i ≤ max_slots then
            Update container matrix for input_data
        end if
    end for
    Add p_values to input_data
    return input_data
end function
```

Algorithm 2 Determine Retrieval Sequence

```
function DETERMINE_RETRIEVAL_SEQUENCE(config, containers)
Input: 2D Array of container layout, position (row and column) of containers belonging to current time window
Output: Sorted list of containers to be retrieved
    function COUNT_BLOCKING_CONTAINERS(container)
        row, col ← container
        count ← 0
        for i ← 0 to row − 1 do
            if config[i][col] ≠ 0 then
                count ← count + 1
            end if
        end for
        return count
    end function
    return sort(containers, by {count_blocking_containers(x), x[0]})
end function
```

Algorithm 2 performs a sorting that is based on the count of blocking containers first and then, if there are ties, by the row number in ascending order. It ensures that the targeted containers to be picked up with fewer blocking containers come first, and in case of a tie, the one with the lower row number is prioritized. This ensures that the target container with the least number of blocking containers is retrieved, which in turn provides the fastest way to create space for future blocking containers

Algorithm 3 identifies and sorts blocking containers in a way that always the top-most container is retrieved first. This ensures that the gantry crane's requirement to always pick up the top container first is met.

The next critical decision is to decide where to relocate the blocking container(s). Algorithm 4 first evaluates whether there is a stack without another target container to be retrieved within the current time window. If this is the case,

Algorithm 3 Identify and Sort Blocking Containers

```
function DETERMINE_RELOCATION_SEQUENCE(config, row, col)
Input: 2D Array of container layout, position (row and column) of container to be retrieved
Output: Sorted indices of all blocking containers
    blocking_positions ← {}
    for i ← row − 1 downto 0 do
        if config[i][col] ≠ 0 then
            blocking_positions.append([i, col])
        end if
    end for
    return sort(blocking_positions, by {blocking_positions(x), x[0]})   ▷ Sort by the number of blocking positions and then by the row index
end function
```

the best suitable spot for relocation is selected based on the following criteria: (1) Prioritize empty spots that are at the bottom or close to the bottom.

(2) If there are multiple stacks that fulfill (1), take the one with the higher sum of its column entries.

(3) If there is an empty spot in a stack that causes no blocking, (i.e., all column entries are higher than the blocking container to be removed), relocate it there, and if there are multiple, take the lower one. This guarantees that blocking containers are retrieved in a way that future relocations are minimized. However, there are some extreme cases (see Section VI-B) in which all stacks contain a target container, whereby these are at the bottom in the worst-case scenario. In this case we need to find a new empty spot in a different stack. The best empty spot then is the spot which is highest and has as many late scheduled retrievals as possible. This ensures that containers can be retrieved faster and more empty spots are created within the process.

Finally, the pseudocode of our proposed algorithm that incorporates the previously presented algorithms (2, 3, 4) to solve the BRP problem with an emphasize on minimizing container relocations is outlined in Algorithm 5. The proposed algorithms scans systematically through all time windows and aims to retrieve all containers belonging to the current time window based on a restricted number of crane movements G_t and truck pick-up appointments L_t. To avoid cost-and energy intense reschedule operations, the algorithms first identifies all containers belonging to the current time window, and then starts with those that are easiest to access (see Algorithm 2). If a target container is not directly accessible, the one or more blocking containers are first identified and sorted (see Algorithm 3) and subsequently relocated with the aim to generate as little future relocations as possible (see Algorithm 4). If all crane movements or truck pick-up appointments are spent, remaining containers get rescheduled to the next time window. Since delayed containers are often at the top of the stack because crane movements in the previous time window were used to remove blocking containers, these are usually the first to be retrieved in the next time window. We note that in most cases, it is good practice to add an additional time window to ensure that

Algorithm 4 Find Best Spot

```
function FINDBESTSPOT(config, avoid_value, candidate_pairs)
Input: 2D array representing container layout, value to avoid, candidate spots
Output: Optimal spot to remove a blocking container
    best_spot ← None
    lowest_row_index ← −∞
    best_column_index ← None
    for (i, j) ∈ candidate_pairs do
        if config[i, j] ≠ avoid_value and avoid_value is not present in column j of config then
            if i > lowest_row_index                                ▷ Prefer spots closer to the bottom
            or (i = lowest_row_index and sum of column j > sum of column best_column_index ▷ Prefer stacks with late retrievals
            or (i = lowest_row_index and every element below row i in column j exceeds avoid_value) then ▷ Avoid blocking other containers
                lowest_row_index ← i
                best_column_index ← j
                best_spot ← (i, j)
            end if
        end if
    end for
    if best_spot = None then                                       ▷ If all columns contain another target container
        valid_candidates ← {(i, j) | (i, j) ∈ candidate_pairs and j is a new column
        and the spot in the row below is non-empty}
        best_spot ← topmost spot in valid_candidates with maximum late retrievals
    end if
    return best_spot
end function
```

all containers are retrieved. The complexity of our algorithm is polynomial, since it comprises a set of nested loops, without any recursive calls, and is easily verifiable to be $O(H^4 \log(H)GLCN)$.

VI. RESULTS

A. CALCULATION OF OPTIMAL SOLUTIONS AND RUN TIME COMPARISON USING THE OPTIMIZATION MODEL WITH DIFFERENT WEIGHTINGS

Figure 5 shows the comparison of different run times using the optimization model with different weights. The Figure demonstrates that the run time increases as α increases, and that the run time tends to grow exponentially as the number of containers increases. In Figure 5, each of the 7 time windows T has a capacity of 4 crane movements (G) and can serve 2 trucks (L). In case of Figure 5, both pre-and postponements of scheduled container retrieval appointments by truck are allowed. For each number of containers, a set of 50 random bay configurations was generated using Algorithm 1. When viewing the run time results of Tables 3c and 3d,, it is evident that the run time keeps growing exponentially with each added container.

As described in Section V, the effect of varying the values of the α and β parameters for weighting different preferences to minimize either or both relocations and schedule deviations is analyzed using different bay configurations. For each bay configuration, a set of 100 random bay layouts are

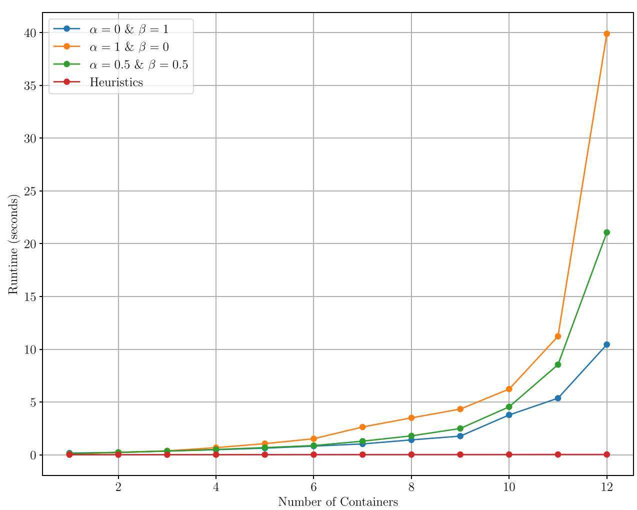


FIGURE 5. run time evaluation for a 4x4 bay layout.

analysed. The resulting 100 simulations, which are presented in each row of the Table 3 and thus span horizontally through both Subtables, are identical to allow comparison of the effects of allowing both pre-and postponements versus only postponements of scheduled truck retrieval appointments. Consequently, in case of Subtables 3(a) and 3(c), both pre- and postponements of scheduled retrieval appointments are enabled, while in the case of Subtables 3(b) and 3(d), only postponements of scheduled truck retrieval appointments

Algorithm 5 Retrieve and Relocate Container Algorithm

```
procedure ContainerRelocationProblem(config, G, L, T)
Input: 2D Array of container layout, Number of crane moves (G), retrievals per time window (L), and time windows (T)
Output: Required number of relocations and schedule deviations, sequence of steps to retrieve all containers
    number_of_relocations ← 0
    number_of_schedule_deviations ← 0
    for t ← 1 to T do
        container_positions ← Indices of all containers belonging to current time window t
        container_positions ← DETERMINE_RETRIEVAL_SEQUENCE(container_positions, config)    ▷ Algorithm 2
        G_t ← G
        L_t ← L
        while G_t > 0 and L_t > 0 and len(container_positions) > 0 do
            target_container ← container_positions[0]
            if target_container is on top or there is no blocking container then
                retrieve target container
                G_t ← G_t − 1
                L_t ← L_t − 1
                container_positions.remove(target_container)
            else
                positions_blocking_element ← DETERMINE_RELOCATION_SEQUENCE(config, indices target_container)    ▷ Algorithm 3)
                for each blocking_container in positions_blocking_element do
                    if G_t < 1 then
                        break
                    end if
                    empty_spots ← Indices of all empty spot
                    best_spot ← FIND_BEST_SPOT(config, t, target_container, empty_spots)    ▷ Algorithm 4
                    relocate blocking_container to the best_spot
                    G_t ← G_t − 1
                    number_of_relocations ← number_of_relocations + 1
                end for
                if G_t > 0 and L_t > 0 then
                    retrieve target container
                    container_positions.remove(target_container)
                    G_t ← G_t − 1
                    L_t ← L_t − 1
                end if
            end if
        end while
        if container_positions ≠ ∅ then
            reschedule all containers in container_positions to time window t + 1
        end if
    end for
    Return number_of_relocations, number_of_schedule_deviations
end procedure
```

are enabled. Both Subtable 3(a) and 3(b) have the same layout, which is represented as a 4x4 square matrix, while Subtable 3(c) and 3(d) represent a larger 5x5 square matrix. For all bay configurations, at least one additional time window is enabled to ensure that all containers can potentially be delayed and to avoid infeasibility of the simulated bay configurations.

The comparison of Subtables 3(a) and 3(b) shows that allowing both advance and postponement of scheduled container pick-up dates has a positive impact on minimizing both the number of required schedule deviations and the number of relocations. This pattern is repeated throughout the comparison of Subtables 3(a) and 3(b). However, this requires a high degree of flexibility for both scheduled

TABLE 3. Comparison of the number of required relocations and schedule deviations, as well as the run time, for different α and β values. In the case of the two tables on the left, both pre- and postponements of scheduled truck retrieval appointments are allowed, while the tables on the right only allow postponements.

Parameters		Run time		Schedule deviations		# of relocations	
α	β	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
1.0	0.0	33.307	56.976	0.9	1.667	8.89	1.476
0.9	0.1	26.854	46.834	0.9	1.667	4.75	1.313
0.8	0.2	23.896	41.352	0.9	1.667	4.75	1.313
0.7	0.3	27.23	40.85	0.9	1.667	4.75	1.313
0.6	0.4	22.707	33.576	0.9	1.667	4.75	1.313
0.5	0.5	18.093	24.936	1.16	1.71	4.49	1.374
0.4	0.6	24.965	31.236	1.46	1.795	4.2	1.463
0.3	0.7	27.146	34.12	4.11	2.174	2.87	1.568
0.2	0.8	23.047	33.032	4.76	2.257	2.69	1.475
0.1	0.9	18.921	28.042	5.76	1.98	2.45	1.438
0.0	1.0	11.322	20.008	8.24	1.621	2.45	1.438
Bay configuration: N=12, C=4, H=4, L=2, G=4, T=7, and $\delta = 1$							

(a) Pre-and postponements of scheduled truck appointments

Parameters		Run time		Schedule deviations		# of relocations	
α	β	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
1.0	0.0	22.322	40.403	1.96	3.025	9.55	1.66
0.9	0.1	22.874	40.638	1.96	3.025	4.79	1.409
0.8	0.2	19.825	31.619	1.96	3.025	4.79	1.409
0.7	0.3	24.179	35.175	1.979	3.018	4.79	1.409
0.6	0.4	27.237	40.543	1.97	3.037	4.77	1.392
0.5	0.5	21.814	33.907	1.97	3.037	4.77	1.392
0.4	0.6	28.735	42.257	2.04	3.038	4.72	1.415
0.3	0.7	31.373	44.679	2.54	3.043	4.47	1.527
0.2	0.8	29.422	41.67	3.94	2.737	3.99	1.586
0.1	0.9	31.679	42.913	4.57	2.341	3.87	1.655
0.0	1.0	21.538	32.044	6.39	1.82	3.87	1.655
Bay configuration: N=12, C=4, H=4, L=2, G=4, T=7, and $\delta = 1$							

(b) Only postponements of scheduled truck appointments

Parameters		Run time		Schedule deviations		# of relocations	
α	β	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
1.0	0.0	446.37	823.427	0.65	1.132	10.68	1.82
0.9	0.1	260.81	650.317	0.65	1.132	5.4	1.602
0.8	0.2	210.692	370.484	0.65	1.132	5.4	1.602
0.7	0.3	249.883	398.946	0.65	1.132	5.4	1.602
0.6	0.4	247.918	372.392	0.65	1.132	5.4	1.602
0.5	0.5	174.287	208.18	0.97	1.344	5.07	1.603
0.4	0.6	207.949	243.173	2.07	1.95	4.1	1.795
0.3	0.7	163.528	198.444	5.29	2.397	2.43	1.559
0.2	0.8	91.655	124.636	6.07	2.319	2.19	1.412
0.1	0.9	74.05	116.588	6.78	2.067	2.03	1.396
0.0	1.0	37.013	57.569	9.61	1.723	2.03	1.396
Bay configuration: N=15, C=5, H=4, L=3, G=6, T=6, and $\delta = 1$							

(c) Pre-and postponements of scheduled truck appointments

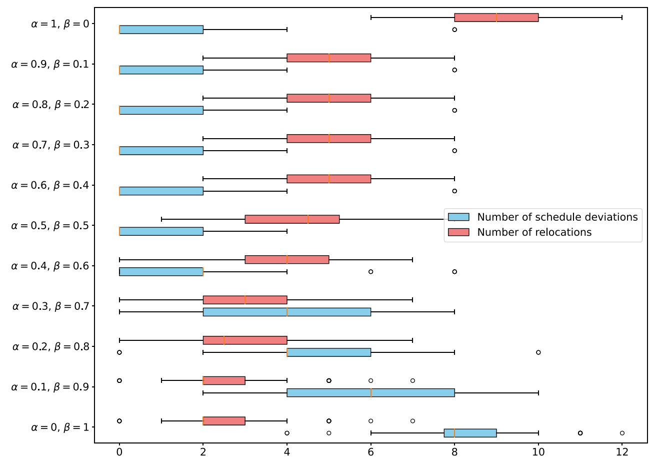
Parameters		Run time		Schedule deviations		# of relocations	
α	β	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
1.0	0.0	396.673	581.525	1.36	2.2	11.2	1.964
0.9	0.1	181.222	263.029	1.36	2.2	5.3	1.514
0.8	0.2	202.49	316.952	1.36	2.2	5.3	1.514
0.7	0.3	228.001	316.332	1.36	2.2	5.3	1.514
0.6	0.4	183.182	219.531	1.36	2.2	5.3	1.514
0.5	0.5	120.971	140.401	1.38	2.196	5.28	1.505
0.4	0.6	199.936	256.826	1.75	2.262	4.98	1.563
0.3	0.7	256.474	670.717	2.94	2.183	4.31	1.756
0.2	0.8	158.955	216.541	3.9	2.047	3.99	1.732
0.1	0.9	174.702	230.492	4.89	2.178	3.78	1.643
0.0	1.0	109.793	143.956	6.86	2.188	3.78	1.643
Bay configuration: N=15, C=5, H=4, L=3, G=6, T=6, and $\delta = 1$							

(d) Only postponements of scheduled truck appointments

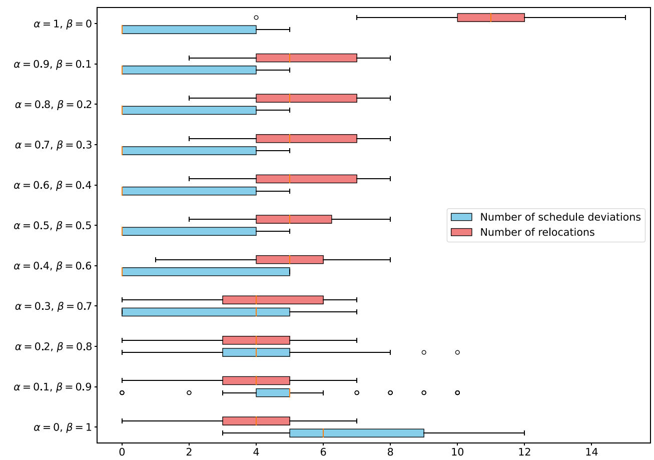
trucks and gate operations, resulting in frequent queue changes.

From the evaluation of Subtables 3(b) and 3(d), it is clear that the run time increases significantly with larger problem instances. In general, it seems that apart from the number of stacked containers (N), G and δ have the biggest impact on the results, as they significantly increase the potential solution space. Therefore, the impact of the parameters G and δ is further explored in Table 4.

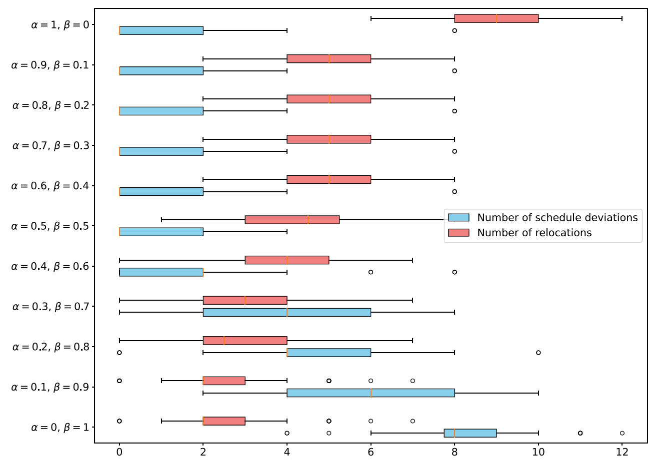
The results in Table 3 reveal that the run time performance of solely utilizing the block relocation minimization (with $\alpha = 0$ and $\beta = 1$ objectives outperforms all other tested combinations. However, just a small increase of α significantly lowers schedule deviations without impacting the number of relocations too much. A similar pattern can be observed when emphasizing solely on schedule deviations (i.e., $\alpha = 1$ and $\beta = 0$), as a small increase in β already significantly decreases the number of required relocations without effecting the number of required schedule deviations. If both problems are of similar relevance, i.e., the port aims to reduce both crane movements to save energy and to avoid queuing to reduce intra-port congestion, the choice of setting



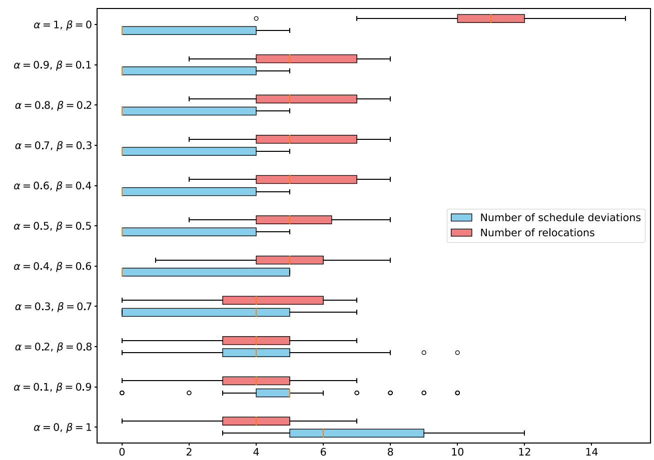
(a) Boxplot for the layout presented in Table 3a



(b) Boxplot for the layout presented in Table 3b



(c) Boxplot for the layout presented in Table 3c



(d) Boxplot for the layout presented in Table 3d

FIGURE 6. **Visualization of the impact of different pairs of parameter values on required relocations and schedule deviations.**

both α and β to 0.5 might be most appropriate. The results presented A more thorough discussion regarding the different use cases is outlined in Section VII.

The results of Table 3 thus demonstrate the immersive potential of jointly minimizing both container relocation moves and truck schedule appointment deviations for increased overall port efficiency.

Figure 6 contains a grouping of boxplots representing the distribution of the required number of container relocations and schedule deviations using the AMPL optimization model for each problem configuration presented in Table 3. It confirms the assumption from Table 3 that joint tackling of both problems is beneficial, since even a small step, such as from 0 to 0.1 in α or β, increases overall system performance. The comparison of the Subfigures 6(a) and 6(b), representing the two cases from Table 3, which allow both pre-and postponements of scheduled truck retrieval appointments, shows that a α value of 0.4 and a β value of 0.6 have the highest joint optimization potential if the goal is to minimize both problems together. The same pattern can also be discovered for Subfigures 6(c) and 6(d), which represent the two cases from Table 3, where only postponements are allowed.

Since both the number of relocations and the number of schedule deviations are highly dependent on the parameters G and δ and their interaction, these two parameters are analyzed in more detail using three different weightings in Table 4. This Table uses the same bay configuration as the Subtables 3(a) and 3(b). The resulting evaluation demonstrates that setting $G = L$, despite allowing δ to grow up to 3, is not sufficient to retrieve all containers in the bay within the desired time interval T, especially if δ is limited to allowing only postponements. Once G is set to $L + 1$ and $\delta >= 2$ or $G = 2 * L$ and delta is $>= 1$ (as in the case of Table 3), all containers can be retrieved within the given time horizon T. It is further remarkable that the solutions of Subtables 4(b) and 4(c) are nearly identical. especially for higher G and δ values, with the solutions of Subtables 4(a) and 4(b) are much more distinct.

TABLE 4. Comparison of the impact of varying δ and G values ($N = 12$, $C = 4$, $H = 4$, $L = 2$, $T = 7$).

G	δ	Pre-and postponements							Only postponements						
		Run time (s)		Schedule deviations		# of relocations		Feasibility rate	Run time (s)		Schedule deviations		# of relocations		Feasibility rate
		Avg	Std Dev	Avg	Std Dev	Avg	Std Dev		Avg	Std Dev	Avg	Std Dev	Avg	Std Dev	
2	1	1.011	0.188	6.333	2.517	1.0	1.0	3/20	0.828	0.287	6.0	4.243	1.5	0.707	2/20
2	2	1.203	0.451	10.267	2.604	0.4	0.632	15/20	1.448	0.84	8.8	1.989	1.7	0.675	10/20
2	3	1.642	0.855	11.8	4.595	0.2	0.41	20/20	1.941	2.175	9.5	2.175	1.0	0.784	14/20
3	1	6.03	5.062	4.882	2.176	2.353	1.057	17/20	8.997	10.06	5.25	2.646	3.312	1.195	16/20
3	2	8.902	8.02	6.25	2.425	1.75	1.209	20/20	36.459	85.602	6.25	3.041	3.1	1.447	20/20
3	3	8.431	7.657	7.5	2.763	1.1	1.021	20/20	25.037	45.904	6.9	2.222	2.3	1.455	20/20
4	1	23.204	30.518	3.9	2.553	3.0	1.522	20/20	40.383	76.928	3.45	3.634	4.05	1.701	20/20
4	2	24.362	30.868	5.35	2.641	2.0	1.451	20/20	47.134	95.401	3.35	2.777	3.55	1.538	20/20
4	3	26.792	28.288	7.2	2.802	1.2	1.196	20/20	49.658	72.636	4.55	3.137	2.9	1.889	20/20

(a) α=0.25, β=0.75

G	δ	Pre-and postponements							Only postponements						
		Run time (s)		Schedule deviations		# of relocations		Feasibility rate	Run time (s)		Schedule deviations		# of relocations		Feasibility rate
		Avg	Std Dev	Avg	Std Dev	Avg	Std Dev		Avg	Std Dev	Avg	Std Dev	Avg	Std Dev	
2	1	0.636	0.036	6.0	3.0	1.333	0.577	3/20	0.696	0.1	6.0	4.243	1.5	0.707	2/20
2	2	1.44	1.016	9.733	2.738	0.733	0.594	15/20	1.328	0.432	8.6	2.459	1.8	0.422	10/20
2	3	2.602	2.134	10.55	4.273	0.85	0.587	20/20	1.653	0.885	8.786	2.547	1.357	0.633	14/20
3	1	5.004	4.186	2.941	2.436	3.353	0.862	17/20	7.248	7.473	4.5	3.055	3.688	1.138	16/20
3	2	8.064	6.641	3.15	2.368	3.35	0.813	20/20	17.199	24.684	5.35	3.498	3.55	1.356	20/20
3	3	10.984	11.572	3.2	2.462	3.25	0.716	20/20	17.4	25.03	4.9	3.042	3.3	1.38	20/20
4	1	17.43	27.374	1.6	2.563	4.1	1.071	20/20	21.866	33.889	2.55	3.845	4.45	1.432	20/20
4	2	27.004	47.705	1.1	1.651	4.1	0.968	20/20	25.015	40.496	2.05	2.874	4.2	1.196	20/20
4	3	31.734	57.379	1.1	1.651	4.1	0.968	20/20	33.277	58.327	2.05	2.874	4.1	1.165	20/20

(b) α=0.50, β=0.50

G	δ	Pre-and postponements							Only postponements						
		Run time (s)		Schedule deviations		# of relocations		Feasibility rate	Run time (s)		Schedule deviations		# of relocations		Feasibility rate
		Avg	Std Dev	Avg	Std Dev	Avg	Std Dev		Avg	Std Dev	Avg	Std Dev	Avg	Std Dev	
2	1	0.875	0.125	6.0	3.0	1.333	0.577	3/20	0.5	0.177	6.0	4.243	1.5	0.707	2/20
2	2	2.55	1.469	9.333	2.769	1.133	0.516	15/20	1.337	0.518	8.6	2.459	1.8	0.422	10/20
2	3	4.493	2.902	10.35	4.234	1.05	0.605	20/20	1.998	1.29	8.643	2.405	1.5	0.65	14/20
3	1	8.37	10.781	2.941	2.436	3.353	0.862	17/20	8.026	9.75	4.5	3.055	3.688	1.138	16/20
3	2	19.898	27.511	3.15	2.368	3.35	0.813	20/20	16.622	26.311	5.35	3.498	3.55	1.356	20/20
3	3	25.491	28.972	3.15	2.368	3.3	0.801	20/20	14.837	17.408	4.9	3.042	3.3	1.38	20/20
4	1	24.691	42.68	1.5	2.585	4.2	1.056	20/20	21.858	29.561	2.55	3.845	4.45	1.432	20/20
4	2	31.735	52.328	1.1	1.651	4.1	0.968	20/20	30.399	45.56	2.05	2.874	4.2	1.196	20/20
4	3	43.823	85.24	1.1	1.651	4.1	0.968	20/20	27.298	37.679	2.05	2.874	4.1	1.165	20/20

(c) α=0.75, β=0.25

B. EXTREME CASES

For all of the following cases, a bay configuration consisting of 20 containers (N) stored in 5 stacks (C) and 5 tiers (H) is assumed. For each time frame (T), a number of 10 crane movements (G) and 5 truck appointments (L) are allowed.

TABLE 5. Comparison of the results of our proposed heuristics with the optimal values obtained using AMPL.

Parameters				Results AMPL (Only postponements)						Results of our proposed heuristic					
N	α	β	Refers to Subtable	Run time (s)		Schedule deviations		# of relocations		Run time (s)		Schedule deviations		# of relocations	
				Avg	Std Dev	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
12	0.4	0.6	3b	28.7	42.2	2.04	3.038	4.72	1.415	0.017	0.006	3.040	3.055	5.080	1.482
15	0.4	0.6	3d	199.9	256.8	1.75	2.262	4.98	1.563	0.019	0.007	1.830	1.965	6.280	2.449

1) CONTAINERS ARE STACKED IN A SEQUENCE READY FOR RETRIEVAL

In the best-case scenario all containers are directly accessible, i.e., the containers are sorted in descending order of removal priority, and no relocations or schedule deviations are necessary if sufficient crane movements and truck appointments are available so that all target containers per time slot can be removed immediately. The run time for the case illustrated in 7 using our proposed heuristics (Algorithm 5) was 0.0128 seconds and neither relocations or schedule deviations were performed.

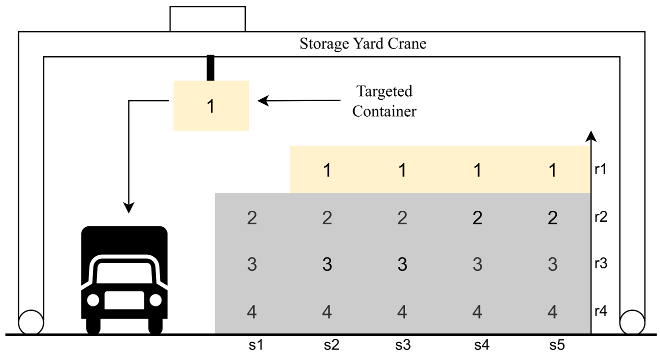


FIGURE 7. Containers are stacked in the sequence in which they are retrieved.

2) CONTAINERS STACKED IN REVERSE ORDER OF RETRIEVAL

In the worst-case scenario, not a single container is directly accessible, as the stacking of the containers is reversed compared to the retrieval sequence and therefore a large number of relocations and/or schedule deviations are required to retrieve all target containers in their given priority. The run time for the case illustrated in 7 using our proposed heuristics (Algorithm 5) was 0.0393 seconds and a total of 14 relocations and 15 schedule deviations were needed.

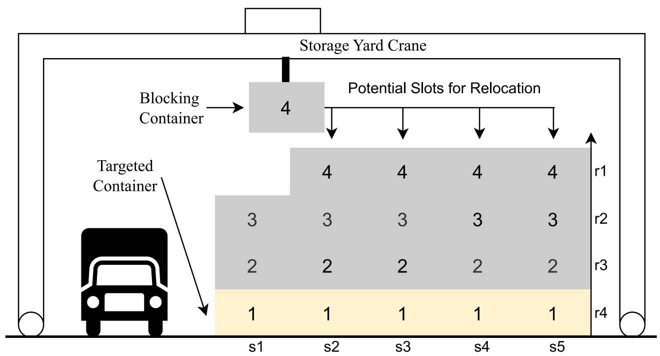


FIGURE 8. Containers are stacked in the opposite sequence to their retrieval sequence.

3) ALL CONTAINERS BELONG TO A SINGLE TIME WINDOW

In case of disruptions or high uncertainties of truck arrivals, there might be a case in which a group of containers is stacked in a bay, with all containers belonging to the same time window. In such a case no crane movements can be deployed once the first trucks arrive in the yard to retrieve the containers. The run time for the case illustrated in 9 using our proposed heuristics (Algorithm 5) was 0.0236 seconds, 0 relocations and 30 schedule deviations were needed.

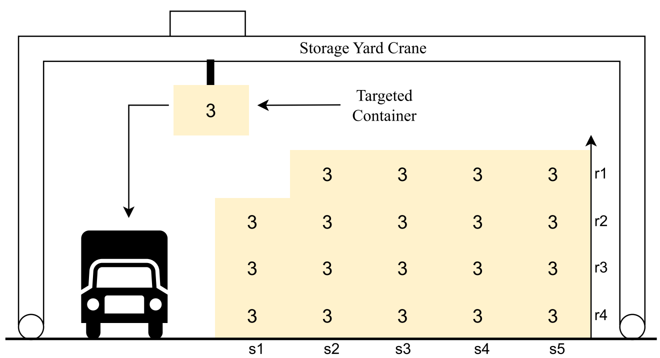
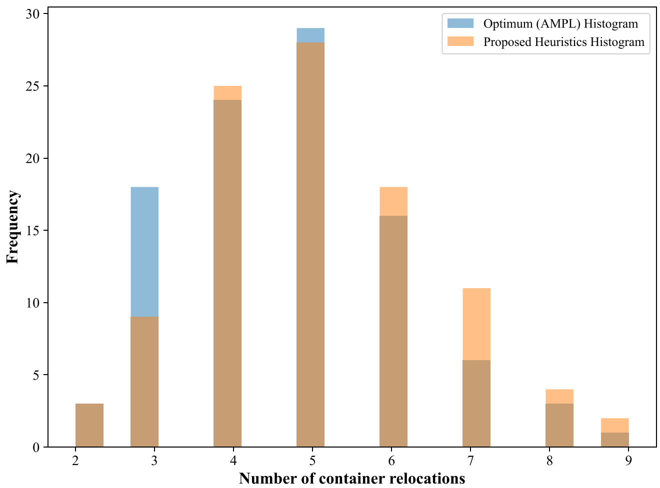


FIGURE 9. All containers are to be retrieved in the same time window.

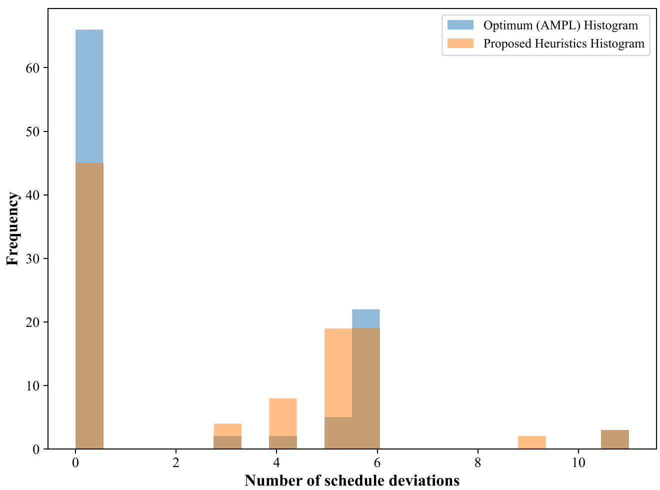
C. COMPARISON TO OUR PROPOSED HEURISTICS

The comparison of Table 5 with Tables 3(b) and 3(d) demonstrates that the heuristics we propose represents a trade-off between keeping schedule deviations tight and reducing container relocation moves. It further reveals that the outcome of our proposed heuristics mirrors a configuration of $\alpha = 0.4$ and $\beta = 0.6$.

The comparison of the number of required relocations and schedule deviations between our optimization model, which calculates the optimal, and our proposed heuristic yields that our proposed heuristic cannot provide the same accuracy as expected, but does not deviate too much from the distribution of the optimal in Figures 10 and 11. This is particularly the case for Figure 10(a), where our proposed heuristic solves the joint optimization of both problems close to the optimum in terms of the number of required relocations. Moreover,

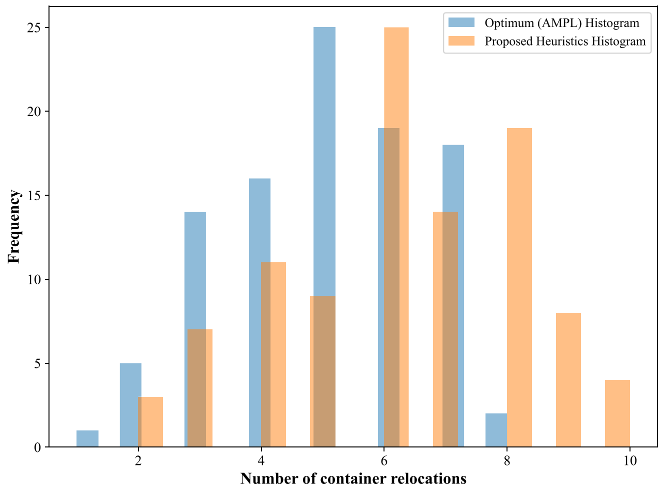


(a) Comparison of the required number of container relocations

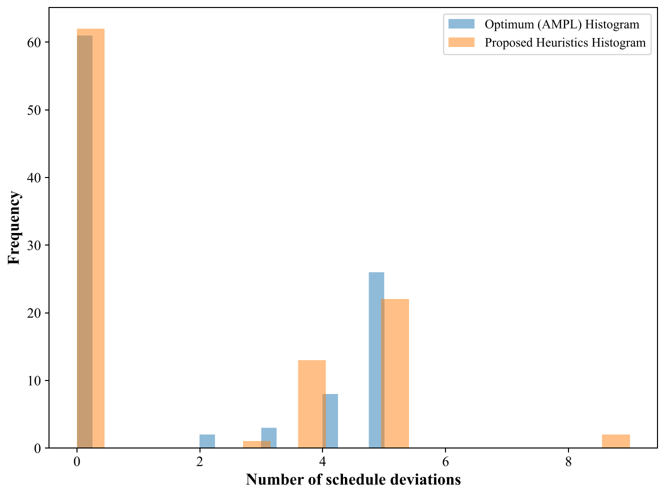


(b) Comparison of the required number of schedule deviations

FIGURE 10. **Comparison between the optimum and our proposed heuristics with $\alpha = 0.4$ and $\beta = 0.6$ for the bay layout outlined in Table 3(b).**



(a) Comparison of the required number of container relocations



(b) Comparison of the required number of schedule deviations

FIGURE 11. **Comparison between the optimum and our proposed heuristics with $\alpha = 0.4$ and $\beta = 0.6$ for the bay layout outlined in Table 3(d).**

when comparing the distribution of required relocations in Subfigures 10(a) and 11(a), it is evident that they are distributed close to normal. Comparing Subfigures 10(b) and 11(b), it is noticeable that the heuristics performed better in minimizing schedule deviations in the case of 11(b), which represents the bay configuration presented in Table 3. This could be explained by the higher number of crane movements per time window compared to the bay configuration of Table 3 used in Figure 10. Our proposed heuristics thus offers a slightly lower level of accuracy, but a much more realistic and practical computation time, as can be seen in Table 5, thus revealing a trade-off between accuracy and computational complexity. For small instances, the differences between the optimal solution and the solution of our proposed heuristic are negligible. Comparing the running time of our proposed heuristics with the one required to obtain the optimum using AMPL, our proposed heuristics is 10521.05 times faster than computing the optimum in the case of the configuration presented in 3(b).

The evaluation of the previous Figure 5 and our expanded Figure 12, representing run time results for larger bay configurations, shows that the run time required to compute the optimal solutions using our optimization model is exponential and thus not capable of solving larger instances in practice, while the run time of our proposed heuristic grows more linear as the input size increases. For Figure 12, a total of 100 random bay configurations were tested for each number of containers and otherwise fixed parameters. It thus demonstrates that even for very large bay configurations, our proposed heuristics still solves large problem sizes within reasonable speed.

VII. DISCUSSION

Recent supply chain studies highlight that optimizing individual components does not necessarily optimize the entire

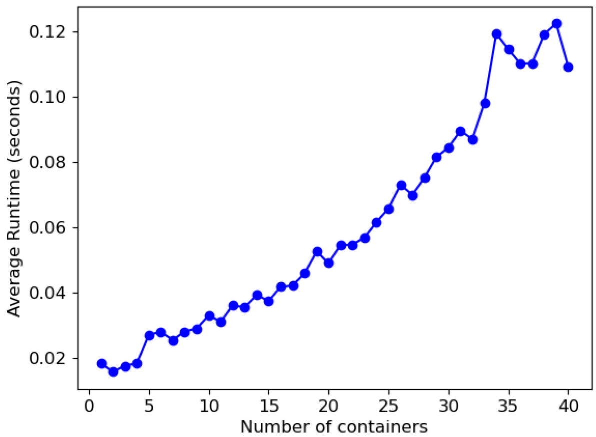


FIGURE 12. Run time development of our proposed heuristics for large bay layouts ($C = 8$, $H = 6$, $L = 5$, $G = 10$, $T = 10$).

supply chain, emphasizing that the collective value is greater than the sum of the parts [48]. Similarly, in the context of ports, overall efficiency depends on the synergy and coordination of the interlinked port processes. This underscores the critical importance of effectively coordinating different processes for optimal performance [49]. Furthermore, as ports are characterized by increasing competition, port logistics service providers need to coordinate their processes with each other based on the use of advanced Industry 4.0 technologies to leverage shared resources to create an integrated and collaborative logistics service chain [50]. The coordination of two major terminal operations, container relocations and retrieval, and truck appointment scheduling, has been therefore the subject of this paper.

The simulations performed show that linking both processes, whether in a balanced fashion, or in favor of one of the processes, leads to a higher overall efficiency, since even small weights leads to efficiency gains of the aspect considered. A total of 11 pairs of α and β values, reflecting the weighting of the problems, were tested to obtain different weighted joint problem combinations to find the optima when both problems are tackled jointly. The different weightings of both problems reflect the varying requirements of different ports, which can vary considerably due to different equipment, operating practices and operational fluctuations. For example, for a port whose yard cranes are already very time- and energy efficient, it may be more important to give greater weight to improving accuracy in truck scheduling.

Finding the optimal relocation and retrieval sequence while considering truck schedules is only suitable for very small instances due to the complexity of the BRP, which is proven to be NP-hard [33], In modern container terminals, containers are typically stored in blocks with a maximum of four tiers (rows). The utilization rate ranges from 70% to 90%, meaning that up to four containers are often stacked on top of each other [51]. However, in ports with above-average throughput, more containers can be stacked on top of each other. Assuming that an average bay consists of 5-10 stacks (columns), the number of containers to be considered is at least 20. For dealing with such realistic instances, a heuristic has been herein proposed that is able to solve the combined problems in a timely manner.

The results of our simulations performed in Section VI show that our proposed algorithm is able to solve both problems, most comparable to an alpha-beta configuration of 0.4 and 0.6 respectively in the optimization model, thus reducing both the number of container movements and schedule deviations. Comparing the results of our algorithm with the solution quality and running time of the optimum obtained by optimization software, there is a large gain in computational speed while the deviation of the solution is comparatively small compared to the optimum. However, this algorithm is not a one-size-fits-all solution, as different terminal operators may require different weighting of the two problems.

Consequently, there is a trade-off between minimising container movements and minimising potential schedule deviations and thus maintaining high reliability. This paper thus complements the existing block relocation literature by examining this trade-off in detail for different configurations.

The need for establishing and characterising such a trade-off is also evident in the literature, both from an efficiency and sustainability perspective, as recent research on energy-aware optimization of port operations shows that small shifts in truck arrival times have a significant impact on port operations and truck emissions [52]. Furthermore, the authors in paper [7] reveal that long truck queues at the gates often limit the efficiency of a container terminal and cause significant air pollution. In addition, we argue that the minimization of the total container moves might have the higher impact in energy savings in the storage yard itself, but from an overall system perspective assuming that the gains in punctuality and reliability lead to more efficient subsequent processes and less congestion. Moreover, we argue that reliable scheduling of external trucks is also crucial for the reliability of the port, and thus also important for the pricing of the services offered by the port. The price that port users are willing to pay depends, among other things, on the available capacity, reliability and overall quality of port services [2].

Thus, this study shows the importance of combining different models as the port is a multi-stakeholder environment where one of the main problems behind its inefficiency is the lack of coordination between stakeholders. A potential solution to address these different problems and stakeholders is digital twins [49], as a digital twin can help improve coordination as it combines multiple models and provides an overview that allows for discussion of trade-offs between different performance variables. Zhao and Goodchild further note in paper [53] that significant improvements could be realized if trucks were equipped with GPS units and location information was shared with the terminal operator along with container details. Provided a high level of collaboration and timely data exchange, digital twins can also serve as a tool to achieve full automation of operations [54].

Future research is needed to explore the trade-off between minimising container movements and truck punctuality and subsequent processes for the whole end-to-end delivery.

VIII. FUTURE WORK

A. FUTURE PROBLEM DIRECTIONS

To integrate the results of this work, which essentially highlight the potential for joint optimization to increase the overall efficiency of the port, into the port's overall system some future research directions can be taken. Although our model reflects the potential for joint optimization of multi-stakeholder and multi-faceted port processes, the current implementation is limited to the yard and hinterland side of the port. Therefore, further research is necessary to investigate the potential impact of integrating the seaside of the port and stacking of incoming containers into our model. This integration could result in reduced crane movements for stacking, relocating, and retrieving containers, possibly taking into account incoming vessel schedules and quay crane capacities. Such a joint coordination and joint optimization of all the processes spanning the whole terminal is required for efficient terminal automation [42].

One assumption that may conflict with actual operations is that all containers within all time windows have the same priority. In reality however, containers in different time windows may have varying priorities, reflected by the willingness of different port customers to pay more. This can result in dynamic pricing applications [55] among the containers to be retrieved to explore the impact of different prices on the prioritized retrieval of different containers. The optimal price for prioritized retrieval could then be derived as a function of the required crane movements and schedule deviations.

B. FURTHER ALGORITHMIC DEVELOPMENT

Although the proposed greedy heuristic could efficiently reflect and solve the configuration with the highest joint potential, which has a weight value of 0.4 for schedule deviations and 0.6 for relocations. Using advanced optimization algorithms more improved and timely algorithms could be obtained. A number of advanced optimization algorithms could be considered for this purpose, such as hybrid heuristics [42], evolutionary algorithms [56], hyper-heuristics [57], swarm intelligence algorithms, such as firework algorithms [58], or polyploid algorithms [59]. Further algorithms designed for multi-objective optimization problems to identify the best trade-off between the conflicting objectives are proposed by Singh et al. in paper [60].

C. REAL DATA AND LEARNING APPROACHES

In this paper, we generated data random bay layouts for each simulation, covering a wide range of bay configurations. By combining these with our extreme cases, we ensure the inclusion of a diverse set of bay layouts, which sufficiently supported our conclusions. Incorporating real data into our model would reflect actual operational scenarios and reveal common container stacking patterns. The use of real data would enable the application of machine learning to detect patterns and perform supervised learning approaches to redetect those patterns in new data, predicting the number of required crane moves or schedule deviations. Alternatively, reinforcement learning could be used to establish a reward function to detect the most efficient operational scheme.

D. MANAGERIAL IMPLICATIONS

This paper highlights the importance of systems integration and data sharing to derive maximum value and benefit from port digitization and the implementation of port community systems. Port authorities considering the development of digital twins must consider the interoperability of different systems and the establishment of a port community system that gathers all the necessary data to analyze and support the entire flow from sea to land and vice versa. To achieve this, it is necessary to address the issue of data sharing in the maritime industry, as currently a lot of data is not accessible and visible to all stakeholders. Here it is necessary to develop an understanding among the involved actors of the value in terms of shorter stops and lower costs that will come with the implementation of combined scheduling through the use of digital twins of higher maturity [49]. To ensure this, we need to start developing a set of good examples that present the business case from the point of view of the different actors. However, this also implies policy implications and the need to move away from the first come first served system used today to a pre-booking system [61]. This requires changes in charter parties as well as in local port regulations.

IX. SUMMARY AND CONCLUSION

The performance of a port is significantly influenced by the efficiency of its storage yard operations. To increase the efficiency of storage yard operations, an intelligent solution to the block relocation problem is required. Although there is literature in place, even most recent solutions consider storage yard operations in isolation, disregarding many interlinked port processes. Building on recent work extending the block relocation problem to consider the scheduling of truck pick-ups, we demonstrated that it is reasonable to link these problems, even if the two problems are not weighed equally. In this paper, we approach the problem from a reliability perspective and argue that there should be a trade-off between reducing container movements and schedule deviations. Our calculation of an optimal solution, addressing jointly the two problems, which belong to two different stakeholders, helps terminal operators understand the efficiency gains of the overall port system. In addition, and given the computational complexity of the problem, our proposed heuristic provides an effective way to combine the two problems for reasonable sizes with a realistic solution time. The comparison of our approach with previous approaches demonstrates that our proposed algorithm can reduce the number of schedule deviations without significantly increasing the number of container moves. The main conclusions of this paper can be thus summarized follows:

- Joint optimization of terminal yard processes, i.e., minimization of required yard crane movements and schedule deviations, taking into account the needs of the various parties involved, i.e., truckers and the terminal operator, leads to overall terminal efficiency.
- Our proposed optimization framework, derived to extract the exact solutions, reveals that even slightly considering a problem instead of excluding it, as in the case of a weighting of 0.1 and 0.9, leads to improved overall terminal efficiency.
- The greatest potential for joint optimization is achieved by applying a weighting of 0.4 and 0.6 for minimizing schedule deviations and container relocations, respectively.
- Obtaining an exact solution is not a viable option when a timely solution, such as in the case of a port digital twin, is required. The proposed greedy heuristic is capable of solving the problem within milliseconds.
- Future research is required to incorporate the seaside of the terminal, applying real data and developing more sophisticated algorithms.

REFERENCES

[1] M. Nijdam and M. van der Horst, "Port definition, concepts and the role of ports in supply chains: Setting the scene," in *Ports and Networks*. London, U.K.: Routledge, 2017, pp. 9–25.

[2] T. Notteboom, A. Pallis, and J.-P. Rodrigue, *Port Economics, Management and Policy*. London, U.K.: Routledge, 2021.

[3] M. Gulić, L. Maglić, T. Krljan, and L. Maglić, "Solving the container relocation problem by using a metaheuristic genetic algorithm," *Appl. Sci.*, vol. 12, no. 15, p. 7397, 2022.

[4] B. W. Wiegmans and E. Louw, "Changing port–city relations at Amsterdam: A new phase at the interface?" *J. Transp. Geogr.*, vol. 19, no. 4, pp. 575–583, 2011.

[5] Y. Keceli, "A simulation model for gate operations in multi-purpose cargo terminals," *Marit. Policy Manag.*, vol. 43, no. 8, pp. 945–958, 2016.

[6] X. Lyu, R. R. Negenborn, X. Shi, and F. Schulte, "A collaborative berth planning approach for disruption recovery," *IEEE Open J. Intell. Transp. Syst.*, vol. 3, pp. 153–164, 2022.

[7] G. Chen, K. Govindan, and Z. Yang, "Managing truck arrivals with time windows to alleviate gate congestion at container terminals," *Int. J. Prod. Econ.*, vol. 141, no. 1, pp. 179–188, 2013.

[8] M. Ashrafi, T. R. Walker, G. M. Magnan, M. Adams, and M. Acciaro, "A review of corporate sustainability drivers in maritime ports: A multi-stakeholder perspective," *Marit. Policy Manag.*, vol. 47, no. 8, pp. 1027–1044, 2020.

[9] M. d. M. da Silva, G. Erdoğan, M. Battarra, and V. Strusevich, "The block retrieval problem," *Eur. J. Oper. Res.*, vol. 265, no. 3, pp. 931–950, 2018.

[10] S. Tanaka and K. Takii, "A faster branch-and-bound algorithm for the block relocation problem," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 1, pp. 181–190, Jan. 2016.

[11] B. A. Weerasinghe, H. N. Perera, and X. Bai, "Optimizing container terminal operations: A systematic review of operations research applications," *Marit. Econ. Logist.*, vol. 26, pp. 1–35, Jun. 2024.

[12] O. Merk, "The port-city interface," in *Ports and Networks: Strategies, Operations and Perspectives*. London, U.K.: Routledge, 2017, p. 90.

[13] A. S. Alamoush, F. Ballini, and A. I. Ölçer, "Ports' technical and operational measures to reduce greenhouse gas emission and improve energy efficiency: A review," *Mar. Pollut. Bull.*, vol. 160, Nov. 2020, Art. no. 111508.

[14] B. Behdani, B. Wiegmans, V. Roso, and H. Haralambides, "Port-hinterland transport and logistics: emerging trends and frontier research," *Marit. Econ. Logist.*, vol. 22, no. 1, pp. 1–25, 2020.

[15] R. Klar, A. Fredriksson, and V. Angelakis, "Digital twins for ports: Derived from smart city and supply chain twinning experience," *IEEE Access*, vol. 11, pp. 71777–71799, 2023.

[16] H. J. Carlo, I. F. Vis, and K. J. Roodbergen, "Storage yard operations in container terminals: Literature overview, trends, and research directions," *Eur. J. Oper. Res.*, vol. 235, no. 2, pp. 412–430, 2014.

[17] M. Caserta, S. Schwarze, and S. Voß, "Container rehandling at maritime container terminals," in *Handbook of Terminal Planning*. New York, NY, USA: Springer, 2011, pp. 247–269.

[18] M. Caserta, S. Schwarze, and S. Voß, "Container rehandling at maritime container terminals: A literature update," in *Handbook of Terminal Planning*. New York, NY, USA: Springer, 2020, pp. 343–382.

[19] D. Kizilay and D. T. Eliiyi, "A comprehensive review of quay crane scheduling, yard operations and integrations thereof in container terminals," *Flex. Services Manuf. J.*, vol. 33, no. 1, pp. 1–42, 2021.

[20] C. Lersteau and W. Shen, "A survey of optimization methods for block relocation and premarshalling problems," *Comput. Ind. Eng.*, vol. 172, Oct. 2022, Art. no. 108529.

[21] S. Tanaka and S. Voß, "An exact approach to the restricted block relocation problem based on a new integer programming formulation," *Eur. J. Oper. Res.*, vol. 296, no. 2, pp. 485–503, 2022.

[22] K. H. Kim and G.-P. Hong, "A heuristic rule for relocating blocks," *Comput. Oper. Res.*, vol. 33, pp. 940–954, Apr. 2006.

[23] M. Ji, W. Guo, H. Zhu, and Y. Yang, "Optimization of loading sequence and rehandling strategy for multi-quay crane operations in container terminals," *Transp. Res. Part E, Logist. Transp. Rev.*, vol. 80, pp. 1–19, Aug. 2015.

[24] R. Dekker, P. Voogd, and E. Van Asperen, "Advanced methods for container stacking," in *Container Terminals and Cargo Systems*. Berlin, Germany: Springer, 2007, pp. 131–154.

[25] A. Azab and H. Morita, "Coordinating truck appointments with container relocations and retrievals in container terminals under partial appointments information," *Transp. Res. Part E, Logist. Transp. Rev.*, vol. 160, Apr. 2022, Art. no. 102673.

[26] D. Ku and T. S. Arthanari, "Container relocation problem with time windows for container departure," *Eur. J. Oper. Res.*, vol. 252, no. 3, pp. 1031–1039, 2016.

[27] T. Bacci, S. Mattia, and P. Ventura, "The realization-independent reallocation heuristic for the stochastic container relocation problem," *Soft Comput.*, vol. 27, no. 7, pp. 4223–4233, 2023.

[28] R. Jovanovic, S. Tanaka, T. Nishi, and S. Voß, "A GRASP approach for solving the blocks relocation problem with stowage plan," *Flex. Services Manuf. J.*, vol. 31, pp. 702–729, Sep. 2019.

[29] A. Kimms and F. Wilschewski, "A new modeling approach for the unrestricted block relocation problem," *OR Spect.*, vol. 45, no. 4, pp. 1071–1111, 2023.

[30] B. Jin and S. Tanaka, "An exact algorithm for the unrestricted container relocation problem with new lower bounds and dominance rules," *Eur. J. Oper. Res.*, vol. 304, no. 2, pp. 494–514, 2023.

[31] R. Ye, R. Ye, and S. Zheng, "Machine learning guides the solution of blocks relocation problem in container terminals," *Transp. Res. Rec.*, vol. 2677, no. 3, pp. 721–737, 2023.

[32] Y. Tang, Z. Ye, Y. Chen, J. Lu, S. Huang, and J. Zhang, "Regulating the imbalance for the container relocation problem: A deep reinforcement learning approach," *Comput. Ind. Eng.*, vol. 191, May 2024, Art. no. 110111.

[33] M. Caserta, S. Schwarze, and S. Voß, "A mathematical formulation and complexity considerations for the blocks relocation problem," *Eur. J. Oper. Res.*, vol. 219, pp. 96–104, May 2012.

[34] A. Azab and H. Morita, "The block relocation problem with appointment scheduling," *Eur. J. Oper. Res.*, vol. 297, pp. 680–694, Mar. 2022.

[35] R. Jovanovic, M. Tuba, and S. Voß, "An efficient ant colony optimization algorithm for the blocks relocation problem," *Eur. J. Oper. Res.*, vol. 274, pp. 78–90, Apr. 2019.

[36] S. Tanaka and S. Voß, "An exact algorithm for the block relocation problem with a stowage plan," *Eur. J. Oper. Res.*, vol. 279, no. 3, pp. 767–781, 2019.

[37] V. Galle, C. Barnhart, and P. Jaillet, "A new binary formulation of the restricted container relocation problem based on a binary encoding of configurations," *Eur. J. Oper. Res.*, vol. 267, no. 2, pp. 467–477, 2018.

[38] Q. Zeng, Y. Feng, and Z. Yang, "Integrated optimization of pickup sequence and container rehandling based on partial truck arrival information," *Comput. Ind. Eng.*, vol. 127, pp. 366–382, Jan. 2019.

[39] Y. Feng, D.-P. Song, D. Li, and Q. Zeng, "The stochastic container relocation problem with flexible service policies," *Transp. Res. Part B, Methodol.*, vol. 141, pp. 116–163, Nov. 2020.

[40] M. Ma, W. Zhao, H. Fan, and Y. Gong, "Collaborative optimization of yard crane deployment and inbound truck arrivals with vessel-dependent time windows," *J. Marine Sci. Eng.*, vol. 10, no. 11, p. 1650, 2022.

[41] H.-P. Hsu, H.-H. Tai, C.-N. Wang, and C.-C. Chou, "Scheduling of collaborative operations of yard cranes and yard trucks for export containers using hybrid approaches," *Adv. Eng. Informat.*, vol. 48, Apr. 2021, Art. no. 101292.

[42] H.-P. Hsu, C.-N. Wang, T. T. T. Nguyen, T.-T. Dang, and Y.-J. Pan, "Hybridizing WOA with PSO for coordinating material handling equipment in an automated container terminal considering energy consumption," *Adv. Eng. Informat.*, vol. 60, Apr. 2024, Art. no. 102410.

[43] B. Xu, X. Liu, Y. Yang, J. Li, and O. Postolache, "Optimization for a multi-constraint truck appointment system considering morning and evening peak congestion," *Sustainability*, vol. 13, no. 3, p. 1181, 2021.

[44] N. Li, G. Chen, M. Ng, W. K. Talley, and Z. Jin, "Optimized appointment scheduling for export container deliveries at marine terminals," *Marit. Policy Manag.*, vol. 47, no. 4, pp. 456–478, 2020.

[45] W. Zhao and A. V. Goodchild, "Using the truck appointment system to improve yard efficiency in container terminals," *Marit. Econ. Logist.*, vol. 15, pp. 101–119, Mar. 2013.

[46] C. Lu, B. Zeng, and S. Liu, "A study on the block relocation problem: Lower bound derivations and strong formulations," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 4, pp. 1829–1853, Oct. 2020.

[47] V. Angelakis, I. Avgouleas, N. Pappas, E. Fitzgerald, and D. Yuan, "Allocation of heterogeneous resources of an IoT device to flexible services," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 691–700, Oct. 2016.

[48] W. Jiang, "An intelligent supply chain information collaboration model based on Internet of Things and big data," *IEEE Access*, vol. 7, pp. 58324–58335, 2019.

[49] R. Klar, N. Arvidsson, and V. Angelakis, "Digital twins' maturity: The need for interoperability," *IEEE Syst. J.*, vol. 18, no. 1, pp. 713–724, Mar. 2024.

[50] R. Amonkar, V. Roy, and D. Patnaik, "Intermodal service supply chain and seaport logistics performance," *Supply Chain Forum, Int. J.*, vol. 22, no. 2, 2021, pp. 171–187.

[51] D. Ambrosino and H. Xie, "Optimization approaches for defining storage strategies in maritime container terminals," *Soft Comput.*, vol. 27, no. 7, pp. 4125–4137, 2023.

[52] Ç. Iris and J. S. L. Lam, "A review of energy efficiency in ports: Operational strategies, technologies and energy management systems," *Renew. Sustain. Energy Rev.*, vol. 112, pp. 170–182, Sep. 2019.

[53] W. Zhao and A. V. Goodchild, "The impact of truck arrival information on container terminal rehandling," *Transp. Res. Part E, Logist. Transp. Rev.*, vol. 46, no. 3, pp. 327–343, 2010.

[54] E. Thonhofer et al., "Infrastructure-based digital twins for cooperative, connected, automated driving and smart road services," *IEEE Open J. Intell. Transp. Syst.*, vol. 4, pp. 311–324, 2023.

[55] Y. Ding, K. Chen, D. Xu, and Q. Zhang, "Dynamic pricing research for container terminal handling charge," *Marit. Policy Manag.*, vol. 48, no. 4, pp. 512–529, 2021.

[56] M. A. Dulebenets, "A diffused memetic optimizer for reactive berth allocation and scheduling at marine container terminals in response to disruptions," *Swarm Evol. Comput.*, vol. 80, Jul. 2023, Art. no. 101334.

[57] E. Singh and N. Pillay, "A study of ant-based pheromone spaces for generation perturbative hyper-heuristics," in *Proc. Genet. Evol. Comput. Conf.*, 2023, pp. 84–92.

[58] M. Chen and Y. Tan, "SF-FWA: A self-adaptive fast fireworks algorithm for effective large-scale optimization," *Swarm Evol. Comput.*, vol. 80, Jul. 2023, Art. no. 101314.

[59] M. A. Dulebenets, "An adaptive polyploid memetic algorithm for scheduling trucks at a cross-docking terminal," *Inf. Sci.*, vol. 565, pp. 390–421, Jul. 2021.

[60] P. Singh, J. Pasha, R. Moses, J. Sobanjo, E. E. Ozguven, and M. A. Dulebenets, "Development of exact and heuristic optimization methods for safety improvement projects at level crossings under conflicting objectives," *Rel. Eng. Syst. Safety*, vol. 220, Apr. 2022, Art. no. 108296.

[61] A. Mubder, "Just-in-time arrival in port calls: Potential and implementation," Ph.D. dissertation, Dept. Sci. Technol., Linköping Univ. Electron. Press, Linköping, Sweden, 2023.

ROBERT KLAR (Graduate Student Member, IEEE) received the M.Sc. degree in transportation and geoinformation technology from the Royal Institute of Technology (KTH), Stockholm, Sweden, in 2021. He is currently pursuing the Ph.D. degree with the Department of Science and Technology, Linköping University, Norrköping, Sweden. He is also employed as a Research Assistant with the Swedish National Road and Transport Research Institute (VTI), Linköping. From February 2021 to August 2021, he worked as a Research Assistant within the Division of Geoinformatics, KTH. His research interests include digital twins with respect to ports and the application of machine learning and optimization toward efficient and sustainable port operations.

ANDERS ANDERSSON received the Ph.D. degree from the Department of Computer and Information Science, Linköping University, Sweden. He is currently a Researcher with the unit Vehicle Systems and Driving Simulation, Swedish National Road and Transport Research Institute, Linköping. His research interests are simulator technology and simulator architectures covering topics, such as digital twins, connected and automated vehicles, hardware-in-the-loop, and distributed real-time systems.

ANNA FREDRIKSSON received the Ph.D. degree from the Chalmers University of Technology, Sweden, and the docent (Habilitation) degree from Linköping University, where she is currently a Professor of Construction logistics and the Head of Research Education with the Department of Science and Technology. She has a research interest in material flow management and production logistics within different industries and leads several projects focusing on decreasing the environmental impact of freight transport and improving efficiency of logistics in general.

VANGELIS ANGELAKIS (Senior Member, IEEE) received the Ph.D. degree from the Department of Computer Science, University of Crete, Greece, in 2008, and the Habilitation (Docent) degree from the Department of Science and Technology, Linköping University, Norrköping, Sweden, in 2015, where he is currently a Professor. He has authored over 100 papers in international journals and peer-reviewed conferences. His research interests, departing from the field of communication systems and performance engineering in the IoT, lie in the area of inclusive and sustainable smart cities solutions design, implementation, and evaluation.