Cyprus University of Technology

Faculty of Engineering and Technology

Doctoral Dissertation

COMPUTATIONAL INTELLIGENCE APPROACHES FOR OPTIMIZING SEASIDE OPERATIONS IN SMART PORTS

Sheraz Aslam

Limassol, October 2022

CYPRUS UNIVERSITY OF TECHNOLOGY

FACULTY OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING, COMPUTER ENGINEERING AND INFORMATICS

Doctoral Dissertation

COMPUTATIONAL INTELLIGENCE APPROACHES FOR OPTIMIZING SEASIDE OPERATIONS IN SMART PORTS

Sheraz Aslam

Limassol, October 2022

Approval Form

Doctoral Dissertation

COMPUTATIONAL INTELLIGENCE APPROACHES FOR OPTIMIZING SEASIDE OPERATIONS IN SMART PORTS

Presented by

Sheraz Aslam

Supervisor: Herodotos Herodotou, Assistant Professor

Signature ____

Member of the committee: Mikael Lind, Professor

Signature ____

Member of the committee: Andreas Andreou, Professor

Signature ____

Cyprus University of Technology

Limassol, October 2022

Copyrights

The approval of the dissertation by the Department of Electrical Engineering, Computer Engineering and Informatics does not imply necessarily the approval by the Department of the views of the writer.

Acknowledgements

First and foremost, I would like to thank Allah Almighty for giving me the strength and confidence to complete this dissertation. Then, I would like to express my deepest appreciation to my research advisors Dr. Herodotos Herodotou and Michalis P. Michaelides for their continuous support throughout my my PhD. and related research. I believe that this dissertation could not have been completed without the support and encouragement of my supervisors. Also, thanks to the invaluable support and patience of Dr. Herodotou, I have become much more comfortable to the new environment. I am deeply indebted to him for his knowledge, thoughts and friendship.

I also gratefully acknowledge the support of the Cyprus University of Technology and the funding source that made my PhD. possible. I was funded by the STEAM project.

I would like to thank my parents and my wife Kainat Mustafa for their constant support, understanding and help whenever I needed it during my PhD studies and research work. I would also like to thank my brothers Shahzad Aslam and Shahbaz Aslam, and my loving sister Arshia Sana. I believe that without their motivation it is not possible to succeed in my life. I am always grateful to them for their encouragement and support.

Finally, I would also like to express my gratitude to my advisor during my Masters studies in COMSATS University Islamabad, Pakistan. He is now a good friend, Professor Dr. Nadeem Javaid. He was the mentor who set me on the research and academia path, and without his advice and support, my career decisions would have been very different.

Sheraz Aslam

Dedicated

To my Parrents Muhammad Aslam and Raj Baigum.

To my wife Kainat Mustafa and my daughter Aizal Fatima.

ABSTRACT

Over the last couple of decades, demand for seaborne containerized trade has increased significantly and it is expected to continue growing over the coming years. As an important node in the maritime industry, a marine container terminal (MCT) should be able to tackle the growing demand for international sea trade. The increasing number of ships and containers creates several challenges to MCTs, such as congestion, long waiting times before ships dock, delayed departures, and high service costs. The berth allocation problem (BAP) and the quay crane assignment problem (QCAP) are two of the most important optimization problems in container terminals at ports worldwide. A BAP concerns allocating berthing positions to arriving ships to reduce total service cost, waiting times, and delays in vessels' departures. The latter concerns assigning optimal number of quay cranes to docked vessels. From both the port operator's and the shipping lines' point of view, minimizing the time a vessel spends at berth and minimizing the total cost of berth operations are considered fundamental objectives with respect to terminal operations.

This dissertation initially focuses on the BAP, with the objective of reducing the total service cost, which includes waiting cost, handling cost, and several penalties, such as a penalty for late departure and a penalty for non-optimal berth allocation. First, the BAP is formulated as a mixed-integer linear programming (MILP) model. Since BAP is an NP-hard problem and cannot be solved by exact optimization methods in a reasonable time, a metaheuristic approach, namely, a cuckoo search algorithm (CSA), is proposed to solve the BAP. To validate the performance of the proposed CSA-based method, we use two benchmark approaches, namely, the genetic algorithm (GA) and the optimal MILP solution. Next, we conduct several experiments using a benchmark data set as well as a randomly-generated larger data set. Simulation results show that the proposed CSA algorithm has higher efficiency in allocating berths within a reasonable computation time than its counterparts.

Furthermore, we extend the study of BAP, which considers a single quay (straight line) for berthing ships, to multiple quays, as found in many ports around the globe. Multi-quay BAP (MQ-BAP) adds the additional dimension of assigning a preferred quay to each arriving ship, rather than just specifying the berthing position and time. Here, we address MQ-BAP with the objective of minimizing the total service cost, which includes minimizing the waiting times and delays in the departure of ships. MQ-BAP is first formulated as a MILP and then solved using three computational intelligence (CI)-based approaches, namely, CSA, GA, and particle swarm optimization (PSO). In addition, the exact MILP method is also implemented for comparison purposes. Several experiments are conducted using real data from the Port of Limassol, Cyprus, which has five quays serving commercial vessel traffic. The comparative analysis and experimental results show that the CSA-based method outperforms the other CI-based methods, while achieving near-optimal results in affordable time for all considered scenarios.

Eventually, this dissertation investigates, for the first time, multi-quay combined BAP and QCAP, and solves it using CI approaches. First, a mathematical model has been developed based on a real port scenario and real constraints. Then, based on the developed model, we solve multi quay combined BAP and QCAP using exact method and CI approaches, i.e., CSA, GA, and PSO. Validation and performance evaluation of the developed modeling framework and the proposed methods are performed through extensive experiments with real data. The real dataset is collected from the Port of Limassol, Cyprus. In addition, the dataset contains data for multiple quays (five), two of which are container terminals and the other three are passenger or general cargo terminals. The experimental results reveal that the exact method can solve the problem only when one week dataset is used; however, our newly adopted CI-based methods for MQ combined (BAP and QCAP) problem are able to solve large instances (i.e., one month) with small computation time.

To summarize, this dissertation develops several CI based methodologies for several BAP formulations (stand-alone BAP, MQ-BAP, and MQ combined BAP and QCAP) in real world environments with several practical constraints. The proposed methods have been tested and evaluated extensively using real data against benchmark approaches. Numerical findings from experiments confirm the effectiveness of the proposed solutions. Therefore, the proposed CI-based methods can serve as promising decision support tools and assist terminal operators while developing berth allocation plans. The latter (MQ combined BAP and QCAP) will also assist port operators with the development of a fully-specified berth schedule, for container ships as well as for other general cargo or passengers ships, to ensure that the ships will be moored and departed in a timely manner.

Keywords: Internet of Ships; Smart Ports; Intelligent Maritime Transportation; Berth

Allocation Problem; Seaside Operational Problems;

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AEA	Adaptive evolutionary algorithm
ALNS	Adaptive large neighborhood search
ABQ	Alternative berthing quay
BAP	Berth allocation problem
BCO	Bee colony optimization
B&C	Branch and cut
BMO	Bird mating optimizer
BP	Berthing position
CCPSO	Chaos cloud particle swarm optimization
CD-BAP	Continuous and dynamic BAP
CI	Computational intelligence
CRO	Chemical reaction optimization
CS	Clustering search
CSA	Cuckoo search algorithm
DC-BAP	Dynamic and continuous BAP
DD-BAP	Discrete and dynamic BAP
DE	Differential evolution
DL	Deep learning
DPCEA	Deterministic parameter control EA
EA	Evolutionary algorithm
ERO-BAP	Expanded robust optimization BAP
ETA	Expected time of arrival
ETD	Expected time of departure
FBS	Filtered beam search

Continued on next page

Continued from previous page

FCFS	First come first serve
GA	Genetic algorithm
GASSR	GA with state-space reduction
GCs	Gantry cranes
GRASP	Greedy randomized adaptive search procedure
GSSP	Generalized set partitioning problem
GVNS	Generalized variable neighborhood search
GWO	Grey wolf optimization
HD-BAP	Hybrid and dynamic BAP
HGA	Hybrid GA
HSA	Hybrid SA
HT	Handling time
IA	Immune algorithm
IG	Iterated greedy
ILS	Iterated local search
ILP	Integer linear programming
IoS	Internet of ships
IP	Integer progrmming
IMT	Intelligent maritime transportation
ITS	Iterated tabu search
ITVs	Internal transport vehicles
LoS	Length of ship
LRVM	Left and right vessel move
MA	Memetic algorithm
MASSR	MA with state-space reduction

Continued on next page

Continued from previous page

MBO	Migrating birds optimization
MCT	marine container terminal
MIP	Mixed integer programming
MILP	Mixed integer linear programming
MINLP	Mixed-integer non-linear programming
ML	Machine learning
MPA	Model predictive allocation
MPC	Model predictive control
MNSGA-II	Modified non-dominated sorting GA II
MUT	Multi-user terminal
MQ-BAP	Multi-quay BAP
NOB	Non optimal berthing
OFs	Objective functions
PBP	Preferred berthing position
PBQ	Preferred berthing quay
PSO	Particle swarm optimization
QC	Quay crane
QCAP	Quay crane allocation problem
QCSP	Quay crane scheduling problem
RCRO-BAP	Risk constrained robust optimization BAP
RO	Robust optimization
RH	Rolling horizon
RTPSO	Random topology PSO
SA	Simulated annealing
SEDA	Sedimentation algorithm

Continued on next page

SEDA+ERH	SEDA estimation & rearrangement heuristic
SAEA	Self-adaptive evolutionary algorithm
SCs	Straddle carriers
SI	Swarm intelligence
SWO	Squeaky wheel optimization
TEUs	Twenty-foot equivalent units
TS	Tabu search
VND	Variable neighborhood descent
VNS	Variable neighborhood search
WC	Waiting cost
WT	Waiting time
YCs	Yard cranes
YCS	Yard crane scheduling
YTs	Yard trucks
YTD	Yard truck dispatching
YTS	Yard truck scheduling

LIST OF CONTRIBUTIONS

1) **Sheraz Aslam**, Michalis P Michaelides, and Herodotos Herodotou. *Internet of ships: A survey on architectures, emerging applications, and challenges*. IEEE Internet of Things journal, 7(10):9714–9727, 2020.

2) Herodotou, Herodotos, **Sheraz Aslam**, Henrik Holm, and Socrates Theodossiou. *Big Maritime Data Management*. In Maritime Informatics, pp. 313-334. Springer, Cham, 2020.

3) Evmides, Nicos, **Sheraz Aslam**, Aris Televantos, Apostolos Karagiannis, Andreas Paraskeva, Michalis Michaelides, and Herodotos Herodotou. *Employing Fuzzy Matching for Cleaning Manual AIS Entries*. In Proceedings of World of Shipping Portugal - An International Research Conference on Maritime Affairs. 2021.

4) **Sheraz Aslam**, Michalis P Michaelides, and Herodotos Herodotou. *Dynamic and continuous berth allocation using cuckoo search optimization*. In Proceedings of the 7th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS), pages 72–81, 2021.

5) **Sheraz Aslam**, Michalis P Michaelides, and Herodotos Herodotou. *Enhanced berth allocation using the cuckoo search algorithm*. SN Computer Science, 3(4):1–15, 2022.

6) **Sheraz Aslam**, Michalis P Michaelides, and Herodotos Herodotou. *Optimizing multi-quay berth allocation using the cuckoo search algorithm*. In Proceedings of the 8th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS), pages 124–133, 2022.

7) **Sheraz Aslam**, Michalis P. Michaelides, and Herodotos Herodotou. *Optimizing Seaside Operations for Intelligent Maritime Transportation: A Survey on Methods, Solutions, and Future Opportunities*. Under review, 2022

8) **Sheraz Aslam**, Michalis P. Michaelides, and Herodotos Herodotou. *Optimizing Berth Allocation Considering Multiple Quays Using Computational Intelligence Approaches*. Under review, 2022

9) **Sheraz Aslam**, Michalis P. Michaelides, and Herodotos Herodotou. *Computational Intelligence for Optimizing Combined Berth and Quay Crane Allocation Considering Multiple Quays*. Under review, 2022

1 Introduction

Sea transportation is considered one of the crucial modes for the delivery of goods around the globe. For instance, around 80% of global trade and 74% of total imports/exports to/from Europe are carried out through the maritime industry and handled by ports worldwide, which is increasing every day [6, 14, 83]. According to a recent report [141], the total number of containers handled per year has increased dramatically and is continuously growing every year. For instance, in 2019, global container throughput reached approximately 802 million twenty-foot equivalent units (TEUs), an addition of 2.3% over the previous year. Since maritime trade is increasing (8.4 billion tons in 2010 to 11.1 billion tons in 2019) [3], the number of marine container terminals (MCTs) and the competition among them, in terms of throughput capacity maximization and vessel turnaround time minimization, is also increasing. Therefore, the MCTs have high importance and are considered one of the major nodes in sea transportation systems. As a critical and integral part of the global transportation network, the MCTs serve the cost-efficient delivery of various products in different markets. Linear shipping companies use mega-ships in order to carry large containers up to 20,000 TEUs [42]. Since the MCTs are of such high value in the maritime industry, there is an exigent need to enhance the operational efficiency of MCTs by minimizing the total turnaround service times of vessels and achieving competitive strategy along with customer satisfaction. Moreover, port authorities always try to optimize MCT operations by employing various strategies for the efficient utilization of all the port resources.

MCT operations can be categorized into three major operational areas, namely seaside, land-side, and yard-side operations, as presented in Figure 1. Among all MCT operations, the seaside operations are the most important as they affect the overall performance of MCTs. Inefficient planning and improper utilization of port resources may create several issues, including congestion, long waiting times, and late departures. For instance, $13,647$ vessels arrived from Jan-Sep 2019 at Port of Shanghai, China, from which almost 57% of vessels arrived late (more than 12 hours) [30]. According to another recent report presented in [140], the average waiting times for vessels from port-to-berth are 2.2, 2.4, and 2.7 hours in Malaysia, Dubai, and China, respectively. Michaelides et al. [112] investigate the factors influencing the various waiting times at the Port of Limassol, Cyprus, both from a quantitative and a qualitative perspective. For shipping, and particularly for short sea shipping, there are obvious and immediate benefits from improving efficiency by supporting all actors involved in the port call process to engage more easily, to give shipping companies, port service providers, and ship agents better information and decision support systems to boost their efficiency and that of their port [100]. Hence, MCTs' operators need to employ suitable strategies and approaches for proper utilization of the port resources and to avoid the aforementioned issues.

One of the major operations at the MCTs is the allocation of incoming vessels to berths, which is known as the *berth allocation problem* (BAP). On the fleet side, the berth allocation schedule establishes berthing times along with berthing positions for arriving vessels with the objectives of achieving reduced costs, waiting times, handling times, and delays in departure. However, on the port side, an efficient berth allocation plan indicates how many ships can be handled in a scheduling period with the objectives of maximum profit and proper utilization of port resources. Berth allocation is considered a heavy cost operation and the ports pay a high penalty when ships start their mooring process late due to congestion, low tides, or any other problem [144]. Figure 3 describes the timing of typical berth operations at the MCT. After berth allocation, an efficient unloading of containers from ships to quay is needed, which is performed with special types of cranes located alongside the quay, known as quay cranes (QC). The allocation of these QCs to moored ships for unloading/loading and the determination of a work plan lead to a further problem named *quay crane assignment problem* (QCAP). The QCAP is often combined with the BAP as it is immediately needed after berth allocation; however, a solution to the QCAP must follow the BAP characteristics, such as length of vessel, berth allocation time, expected departure time, and the total number of containers that need to be loaded or unloaded [103, 98]. Furthermore, most of current studies deal only with the allocation of berths at a single quay (SQ), assuming that it forms one straight line in which vessels can be berthed according to their length and the positions of other vessels. For example, an exact approach to solve SQ-BAP is presented in [147], an evolutionary algorithm in [44], and a metaheuristic-based method in [21]. However, this assumption is not realistic for several ports around the globe, which consist of multiple separate line segments or quays for berthing [51]. For example, the Port of Limassol in Cyprus, has seven continuous berthing quays. Considering multiple quays adds a new dimension to the BAP; the problem of assigning vessels to quays in addition to assigning berthing positions and times for each separate quay.

So motivated from above-discussed challenges, this study focuses on enhancing seaside operations, and more specifically, the *berth allocation problem* (*BAP*), MQ-BAP, and MQ combined BAP and QCAP. The BAP is a well-known problem that aims to assigning berthing positions to arriving vessels at the port in order to minimize or maximize a given objective function (e.g., minimize total waiting time, reduce late departures, or maximize terminal performance). Before dealing with the BAP, it is necessary to understand the problem environment. Based on the current literature, there are two major factors affecting the BAP, i.e., the configuration of quay/wharf and the arrival time of ships. Quays can be configured in three different ways: 1) *continuous* berthing layout, where arriving vessels can be moored at any location along the wharf; 2) *discrete* berthing layout, where the wharf is divided into a fixed number of berths; and 3) *hybrid* berthing layout, where a mix of continuous and discrete berthing layouts is encountered [31]. In terms of vessel arrivals, there are two main types: 1) *static arrivals* where all the vessels are assumed to be at the MCT before berth planning and 2) *dynamic arrivals* meaning that vessels are not at the MCT before berth planning but instead the expected time of arrival (ETA) is known for each vessel. Static arrivals are a simple special case of dynamic arrivals (where all ships are expected to arrive at the same time), while the discrete berthing layout is a basic variation of the more-complex continuous layout scenario. Thus, this study focuses primarily on the continuous berthing layout together with dynamic vessel arrivals and both port environments (i.e., single quay port and multi quay port).

This study enhances the productivity of seaside operations at the smart ports by solving the BAP with the goal of reducing the total service cost of arriving ships, which includes waiting costs, handling costs, and penalties for late departures. We first formulate BAP as a mixed-integer linear programming (MILP) problem subject to some practical constraints and solve it using the cuckoo search algorithm (CSA), a metaheuristic computational intelligence approach. We also implement two benchmark algorithms to verify the effectiveness of our proposed algorithm, namely, genetic algorithm (GA) and an exact approach (MILP). Extensive simulations were performed on two types of data instances; the first one was obtained from the existing literature (benchmark data) and the second dataset was generated based on real-world data (by uniform distribution). The simulation results show that the proposed CSA method outperforms its counterparts in terms of reducing service costs, delays in departures, and pre-docking waiting times within a reasonable computation time.

Furthermore, this dissertation also contributes by developing MILP model for solving MQ-BAP with the objective of minimizing the total service cost, considering several practical constraints, including safety entrance time, safety distance and safety time between berths, preferred berthing quays, and preferred berthing positions for arriving vessels. The developed model also introduces alternative berthing quays (ABQs), for the first time, for vessels to list multiple other preferred berthing quays, with the primary objective of avoiding long waiting time before mooring. Next, three popular metaheuristic-based algorithms, namely, CSA, GA, and particle swarm optimization (PSO), are developed to solve the problem in affordable computation time, since MQ-BAP is NP-hard and cannot be solved efficiently by exact methods. Finally, to validate the performance of the developed methods, several experiments are conducted using real data (with one-week, two-week, and four-week planning horizons), collected from the Port of Limassol, with real port settings. This study also implements the exact MILP method for fair comparison and results indicate the efficacy of the CSA method over counterparts.

Finally, this dissertation focuses on finding the optimal berthing position with berthing time and the best combination of crane assignments for arriving vessels, while considering MQ ports (e.g., Limassol Port). We solve MQ combined BAP and QCAP, for the first time, with the objective of minimizing total turnaround time and total operating cost of ships, where it includes handling cost, waiting cost, penalties for late departure, and penalties due to non-optimal berth/quay assignment. First, an appropriate mathematical model is established for the MQ combined BAP and QCAP. For this purpose, three metaheuristics are developed to solve the problem, namely the CSA, the PSO, and the GA. Numerical experiments demonstrate the superiority of our methods compared to the mathematical methods.

The remainder of the dissertation is organized as follows. Section 2 presents preliminaries on marine container terminal and seaside operational problems. Section 3 reviews recent literature related to the BAP, including discrete BAP, continuous BAP, and hybrid BAP. Literature review related combined BAP and QCAP is presented in Section 4. Section

5 presents contribution 1 of this study, where berth allocation problem is solved using cuckoo search algorithm. Next, Section 6 unfolds 2^{nd} contribution of this dissertation, where MQ-BAP has been solved. Section 7 presents the solution of MQ combined BAP and QCAP. The last section concludes the PhD. dissertation.

2 Preliminary on Marine Container Terminal and Seaside Operations Problems

The marine container terminals (MCTs) have been essential towards low-cost and efficient sea transportation along with economic growth worldwide. The MCTs handle a huge volume of containers per year that has drastically increased in the last couple of decades. For instance, 74% of total international trade is done through MCTs [14] and it is expected to increase in the future [31]. Hence, with the increasing volume of containers, the MCTs are continuously challenged to enhance their productivity by adapting many software and hardware innovations, such as terminal design, goods handling equipment, automatic berthing, and operations research applications. The MCTs can be partitioned into three major areas, namely, seaside, marshaling yard, and landside [64], as depicted in Figure 1. The berth, quay, and waiting areas are included in the seaside, the arriving and departing containers are stored in the yard area, and internal/external transport connects various areas of MCTs [31, 83]. Furthermore, current literature focuses on various MCT operations, which are primarily divided into three major research areas [31, 114, 40, 12]: 1) seaside operations, 2) marshaling yard operations, and 3) landside operations, which are further discussed below. Figure 2 presents different fundamental operational problems of MCT for each of these areas.

2.1 Landside Operations

MCTs offer several services as an intermediator between landside and seaside operations, where loading and unloading of containers are the major seaside operations. As for the landside operations, trains, trucks, or barges are employed to either pick up or deliver containers. The trucks or trains enter the MCT through specific terminal gates and wait in a dedicated area until loaded on or loaded off. Furthermore, the MCTs can have various interfaces for landside operations, e.g., rail terminals for trains, transfer points where trucks are loaded or unloaded, and barges service center [53]. The major problems in the landside operations include: straddle carriers (SCs) operation optimization [133], truck appointment optimization problem in order to avoid truck congestion and reduce truck turnaround time [161, 118, 138, 162, 4], double cycling (i.e., loading and unloading simultaneously to enhance productivity) scheduling on the landside [160], route and sched-

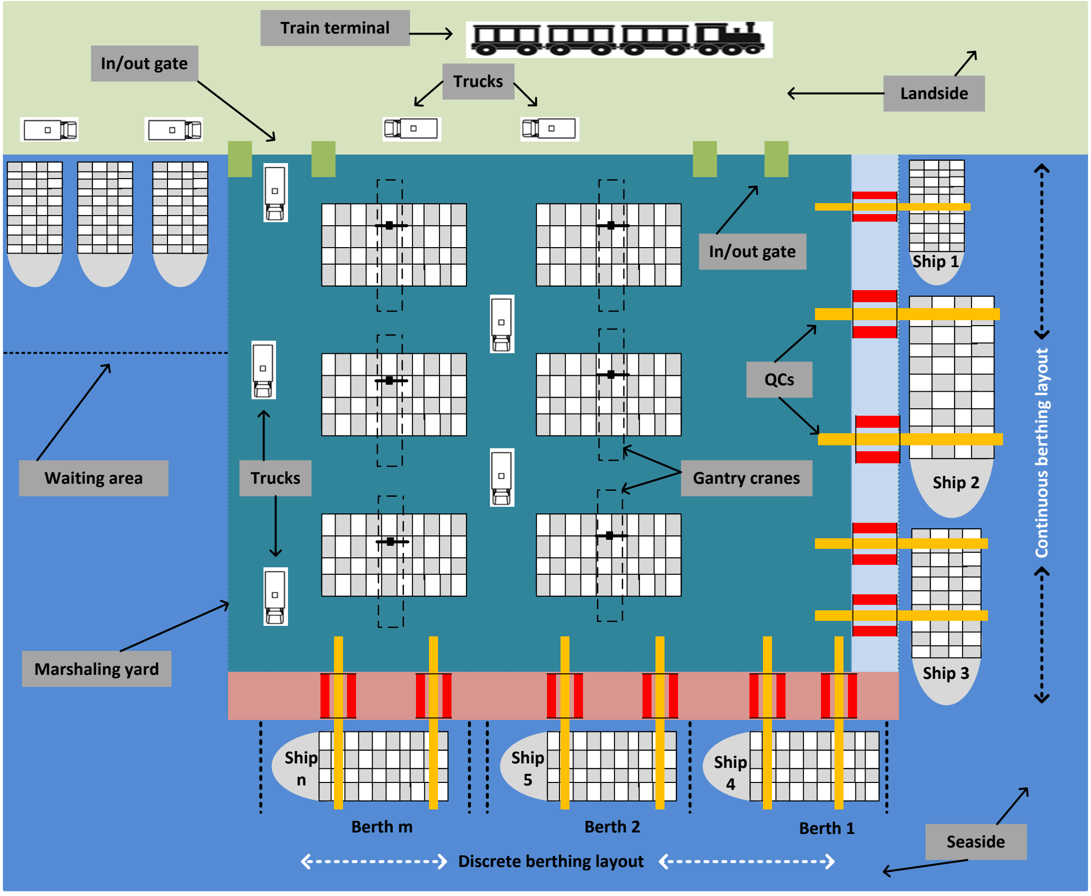


Figure 1: An illustration of the different areas of a marine container terminal.

ule planning of rail-mounted gantry cranes for efficient exchange of containers between the terminal and road [52], scheduling appointments of container trucks [154, 50], and internal transport vehicle selection to avoid delays and enhance terminal performance [123].

2.2 Marshaling Yard Operations

The major operations in the marshaling yard include vehicle routing [67, 120, 121], transport vehicle dispatching [158, 159], yard cranes scheduling (YCS) and yard truck scheduling (YTS) [68, 85, 106, 107], yard cranes and yard truck utilization [129, 155], storage blocks allocation [66, 95, 111, 143], reshuffling of containers, and traffic control [83].

Typically, all containers are stored at the MCT for a particular time period before delivering them to trucks or trains for inland transportation or loading them to vessels for water transportation. In the marshaling yard operations, yard cranes (YCs), also known as gantry cranes (GCs), are employed to handle unloaded containers. The YCs, are responsible for placing all delivered containers in the yard blocks. Furthermore, housekeeping operations

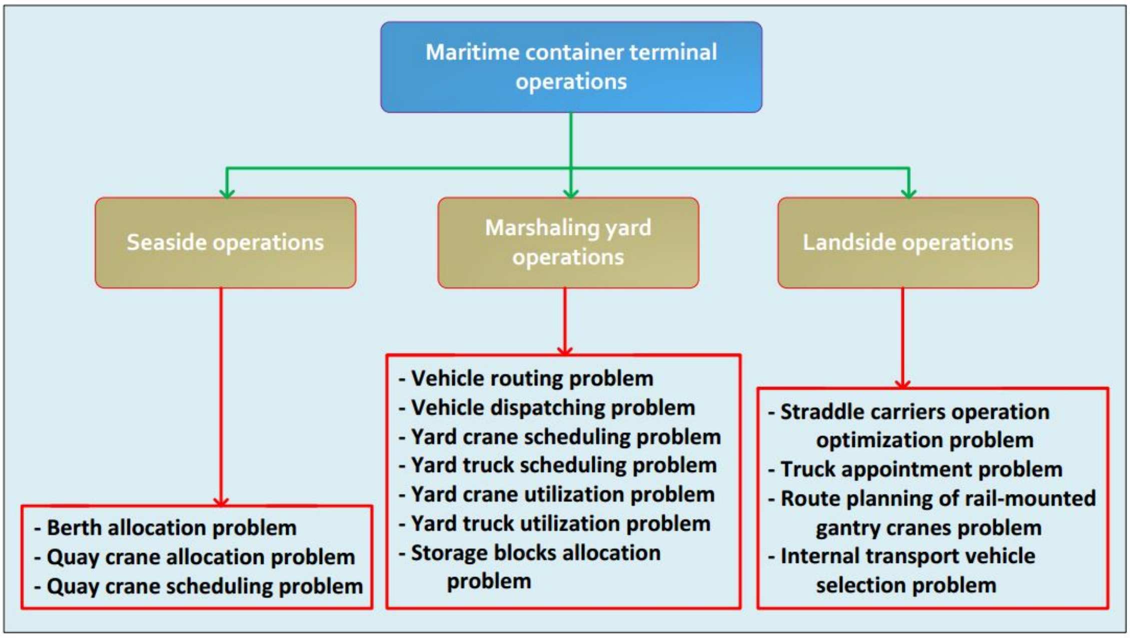


Figure 2: Primary operational problems considered under the marine container terminal umbrella.

(e.g., container relocation and pre-marshaling) are also performed through YCs. Therefore, efficient YCS is key to the enhanced performance of the MCTs. In addition, yard trucks (YTs) are employed to transport the inbound and outbound containers within the terminal. Hence, YT dispatching, scheduling, and routing operations affect the traffic congestion inside the terminal and the overall progress of operations in MCTs.

2.3 Seaside Operations

Seaside operations typically include loading and unloading of containers from arrived vessels at the port, by employing on-shore quay cranes. These containers are then transferred to the marshaling yard area by using internal transport vehicles (ITV), e.g., automated lifting vehicles, automated guided vehicles, yard trucks, etc. The seaside operations phase three fundamental tactical problems, namely, BAP, QCAP, and QC scheduling problem (QCSP) [58]. In the seaside operations, berth allocation is considered one of the major operations and an available berthing position can be allocated to an arrived ship based on various ship and berth characteristics. For instance, the vessels arrival pattern, berthing layout, and ships handling time are the major considerations for vessel berthing, as discussed in Section 2.4.

The QCAP deals with assigning QCs to berthed vessels, while QCSP deals with how the

QCs assigned to a particular vessel will be used for loading/unloading containers from that vessel [58]. The ship's cargo along with loading and unloading information can be collected from a stowage plan. Based on this information, QCs are assigned to vessels. Concerning QCs specifications, the QC assignment and QC scheduling operations can be classified in various ways. The first classification refers to how QCs handle loading/unloading tasks, where each QC has particular constraints while operating. All the containers that are unloaded or loaded to a ship belong to a specific bay (i.e., a space in the vessel to store containers), each corresponding to a task [104, 126]. The number of bays is used to determine the maximum number of QCs that can be assigned to a vessel and work in parallel [126, 33, 82].

The second classification is based on QCs restrictions. Based on the current literature, the QCs restrictions include safety distance between two QCs, movement limitations, and interference among QCs [27]. The QCs are rubber-tired at some MCTs and rail-mounted on others [48, 22]. The rubber-tired QCs (also known as moveable QCs) can move freely and cross each other, while the latter cannot [10]. Hence, there is a need for some non-crossing constraints for models that consider rail-mounted QCs. Furthermore, the constraints related to safety distance are also considered in order to avoid interference with other QCs. The last classification involves initial positions of QCs, ready times of QCs, and availability and unavailability of QCs, as discussed in [126].

2.4 Introduction to Operational Problems of Seaside

In this section, we introduce the two major problems of seaside operations (i.e., the BAP and the QCAP), which have been extensively investigated in the last two decades.

2.4.1 BAP

Allocating available berthing slots to arriving vessels, based on ship properties (e.g., dimensions, draft, etc.), berth characteristics (e.g., length, depth, etc.), and various constraints, is considered the first decision problem, known as BAP. A few studies also refer to BAP as the berth scheduling problem (BSP); however, we have used the more popular term the BAP throughout this survey. The goal of the BAP is to specify which berthing position is suitable for which arriving ship while considering the various types of con-

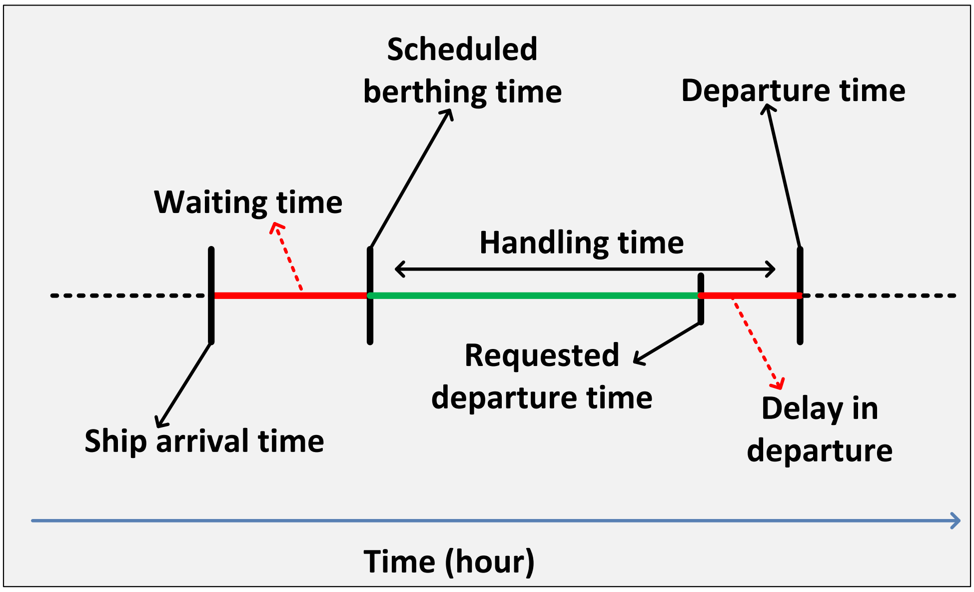


Figure 3: Timing of berth operations at a marine container terminal.

straints, both physical and operational. Figure 3 describes the timing of typical berth operations at the MCT. For developing solutions for a typical BAP, the possible inputs include: expected time of arrival (ETA) for ships, preferred berthing position (PBP), length of ship (LoS), number of containers loaded on each ship, handling time, cranes productivity, and expected time of departure (ETD). The typical outputs of the BAP include: scheduled berthing position for each ship, berthing time for each ship, and scheduled departure times. BAPs can be classified on the bases of two key aspects, namely berthing layout and arrival times [24, 31].

2.4.1.1 Berthing Layout Several berthing layouts are assumed in the literature in order to provide feasible berthing positions to arriving ships, which are explained below.

Discrete layout: In a discrete berthing layout, the wharf is divided into various sections, called berths, and only a single ship can be moored at any berth at a particular time period [43, 24]. It is important to note that the extra length of the berth is wasted if the assigned ship's length is less than the length of the berth. Furthermore, the quay is partitioned either based on the quay construction or divided in order to ease the planning problem, as presented in Figure 4 (a).

Continuous layout: In a continuous berthing layout, ships can be moored at any arbitrary location along the wharf, i.e., the quay is not partitioned into a discrete number of berths.

The quay can be better utilized because of no partitioning into a discrete number of berths. However, berth planning for continuous berthing layout is more complex than a discrete berthing layout due to the much higher number of available berthing positions. A typical structure of a continuous berthing layout is depicted in Figure 4 (b).

Hybrid layout: Hybrid berthing layout is a combination of both discrete and continuous berthing layouts, where the wharf is partitioned into a number of berths. However, a large ship can occupy more than one berths, and sometimes two small ships can be moored at a single berthing slot, as depicted in Figure 4 (c). Furthermore, indented berths are also possible when two opposing berths exist, which can be utilized to unload/load one large ship from both sides [79]. An indented berth is considered a form of a hybrid berth since a large ship uses two discrete berths.

2.4.1.2 Vessel Arrivals As for the ships' arrivals, the BAP can be distinguished as static arrival and dynamic arrival.

Static arrival: In static ship arrival, no arrival times are given against incoming ships. Instead, it is assumed that all ships have already arrived at the MCT, towed to the waiting area, and can be moored immediately based on berth planning.

Dynamic arrival: In dynamic ship arrival, ships are not assumed to be present at the MCT during the planning horizon. Instead, the estimated time of arrival (ETA) for each ship is provided to the MCT for the sake of better berth planning. A ship can arrive before its ETA. It is important to note that for the purposes of planning, the ship cannot be moored before its expected arrival time.

In this study, we only consider the dynamic arrival case since (i) it is not practical for all arriving vessels to be at the terminal before berth planning as in the static arrival case, and (ii) static arrival is a special case of dynamic arrival where all vessels are expected to arrive in 0 minutes.

2.4.1.3 Performance Measure The performance measure is an important aspect of the BAP and it explains the objective functions (OFs) to be maximized or minimized, such as throughput maximization, cost reduction, etc. Based on current literature, most studies use the OF that aims to minimize the total time vessels spend at the MCT. Other OFs are

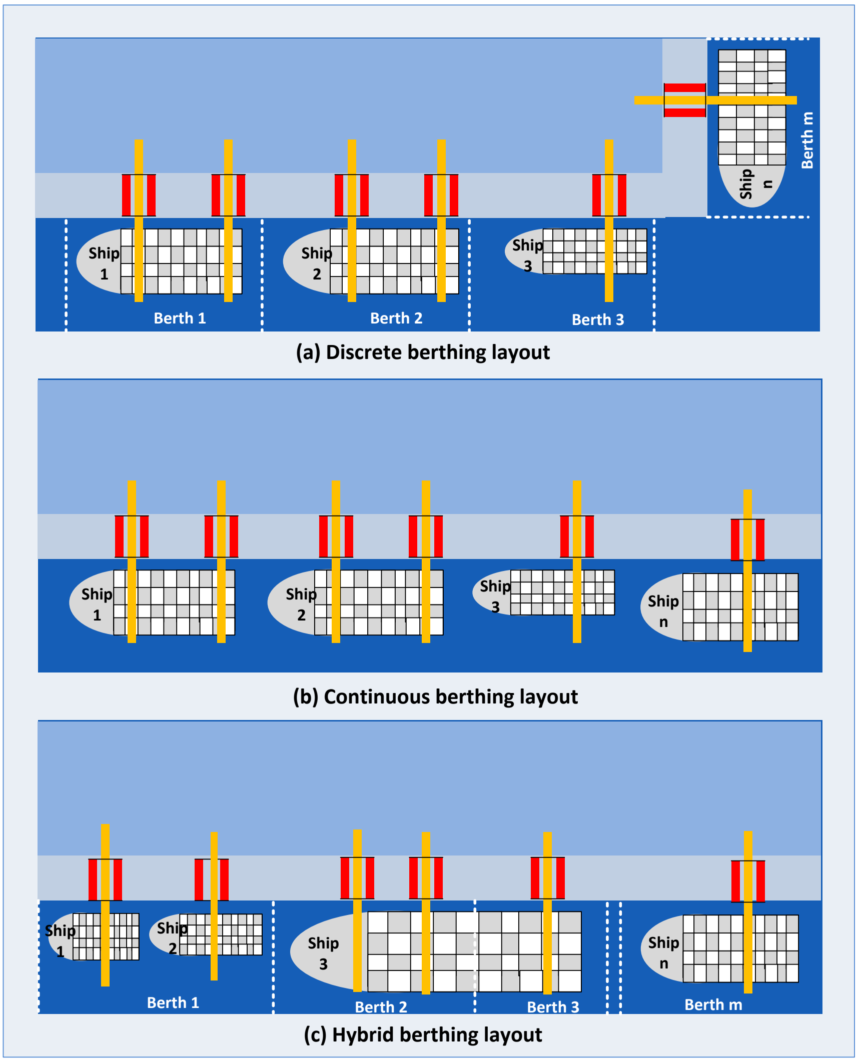


Figure 4: (a) Discrete, (b) Continuous, and (c) Hybrid berthing layouts.

associated with the minimization of the total weighted handling time [136, 73, 28], the total waiting time [6, 81, 125], and the total services cost for all the arriving ships at the MCT [43, 78].

2.4.2 QCAP

The important decisions that need to be made immediately after berth allocation concern the type and number of handling equipment to be employed for loading/unloading the containers to/from vessels. Quay cranes (QCs) are employed at the MCTs to load/unload containers from vessels and are typically manned because automation in this process creates multiple challenges, such as inaccurate positioning of containers [142]. The cranes contain trolleys that can move along the QC arm to transfer the containers from transport vehicles to vessels and vice-versa. A trolley is equipped with the spreader, a pick-up device, that is used to pick the containers. Furthermore, loading and unloading operations on a single ship can also be done simultaneously, i.e., one crane can be employed for loading and the other for unloading [142]. Hence, the number of QCs allocated to vessels is a crucial decision for arriving vessels.

Since the major objective of several studies is to minimize vessels' waiting times and late departures, it is necessary to perform the process of unloading/loading very efficiently in order to meet customers' demands. For unloading/loading vessels, there exist two types of QCs, movable (dynamic) and non-movable (static). The first type of cranes can shift from one berth to another berth during the process of loading/unloading cargo from the currently assigned ships, which makes their use more flexible [103]. The latter type of QCs cannot move from one berth to another before completion of the process on currently assigned ships.

3 Literature Review on BAP

This section deals with the latest efforts of the research community to cope with various types of stand-alone BAP, namely, discrete dynamic BAP, continuous dynamic BAP, and hybrid dynamic BAP, summarized in Table 2.

3.1 Discrete and Dynamic BAP

This section reviews recent literature on discrete and dynamic BAP (DD-BAP), which considers discrete berthing layout and dynamic vessel arrivals as described in sections 2.4.1.1, 2.4.1.2, respectively.

A heuristic-based approach is presented in [137] to deal with discrete and dynamic BAP. The key objectives of this study are twofold: minimization of waiting time of serving vessels at MCT and minimization of total handling time. The problem is modeled as a mixed-integer linear programming (MILP) model and solved by a particle swarm optimization (PSO) algorithm. The proposed method is tested with several datasets of different sizes and compared with state-of-the-art methods. Results indicate the higher performance over counterparts in terms of better berth allocation with low computation time.

Simrin *et al.* employed genetic algorithm (GA) to solve the same BAP [131]. The main aim of this study is to alleviate total vessels' service time that they have divided into two parts, (i) waiting time and (ii) handling time. Unlike other works, this study assumes that each berth has different handling productivity and showed that GA can handle much larger problem sizes compared to a CPLEX-based approach. The waiting time of vessels is the time that vessels spend at MCT before servicing, where handling time is the time that vessels spend between the start servicing until the departure of ships.

The authors of [69] proposed a heuristic-based solution, where they also employed GA for optimal berth allocation. The authors construct a mixed-integer programming (MIP) model that considers discrete berthing layout along with dynamic vessel arrivals and the key objective of this study is to minimize total weighted late departures and workload in night times. Simulations have been carried out for one and two weeks in order to validate the proposed GA-based solution of BAP.

The work presented in [124] proposed an enhanced differential evolution (DE) algorithm-based solution for DD-BAP, where the enhanced DE uses game theory to control the selection of mutation operator. The objectives of this work are twofold: minimization of late departure penalties and alleviation of handling time that spends a vessel at MCT. GSSP, TS, CS, and PSO are also employed to solve the same problem and results denote that enhanced DE outperforms over counterparts in terms of both objectives.

Another DD-BAP is discussed in [13], where a novel EA, namely bird mating optimizer (BMO) algorithm is developed. The problem is modeled as a vehicle routing problem with time windows and solved by the BMO algorithm. The authors of the study [13] also implement PSO and exact approaches for the sake of comparison. Results demonstrate the effectiveness of the BMO method in terms of minimum turnaround time.

An approach called POPMUSIC is proposed in [92] to solve DD-BAP. The proposed method is partially meta-heuristic (and inspired by Tabu search) and partially mathematical, which solves the problem and allocates berths to arriving vessels in a reasonable time. Experimental results show the efficacy of the proposed method over counterparts.

Another study [109] exploits a heuristic-based adaptive large neighborhood search (ALNS) algorithm in order to solve DD-BAP. The BAP is formulated as an MILP problem and the key objective is to achieve minimum cost that occurred due to late departures within minimum computational time. They also used priorities for various vessels based on different factors, e.g., total load, a vessel belongs to a specific linear company, etc. In this way, vessels with higher priority get served and departed sooner than low priority ships. Eventually, the proposed model is compared with existing models and it is affirmed from comparisons that the newly developed method provides an efficient and reliable solution within affordable computation time.

A Memetic algorithm (MA)-based method is adopted in [45] to solve DD-BAP, where the BAP is formulated as a nonlinear MIP model. This study also proposes a new policy, where demand can be shifted from normal MCT to an external MCT at an extra cost. A large number of simulations have been performed to affirm the productiveness of the proposed MA-based approach. Results from simulations indicate the efficacy of the newly developed algorithm in terms of its key objective to minimize handling cost.

Another research presented in [46] proposes a self-adaptive EA (SAEA), which, unlike other EAs, employs a self-adaptive parameter control strategy for efficient planning of berth allocation. The primary objectives are to minimize the total weighted turnaround time of all the docked vessels as well as delays in departures. The problem is also formulated as an MILP model and extensive experiments have been carried out to affirm the performance of the proposed SAEA approach in terms of reduced total turnaround time and delays in departures over counterparts. Compared algorithms include standard EA, deterministic parameter control EA (DPCEA), and adaptive EA (AEA).

The authors of [128] also investigate the discrete BAP along with stochastic vessels arrivals. The primary objective of this study is to mitigate the total turnaround time of vessels that they spend between arrival and departure. For optimal berth allocation, various proactive and reactive approaches are developed, a meta-heuristic iterated tabu search (ITS) is proposed as a proactive method for berth allocation, while stochastic dynamic programming method as a reactive approach is modeled for real-time ship arrivals. Here it is important to note that the study also considers uncertainty in ships' arrival. Simulation results indicate that the ITS has higher efficiency in terms of minimum turnaround time with affordable computation time.

Wang *et al.* developed a novel meta-heuristic based approach to solve DD-BAP, where the proposed method combines the nature-inspired Lèvy Flight random walk with local search [144]. The primary goals of this study are twofold: minimize the total cost of ships' handling cost and provide optimal berth allocation of arriving vessels while considering a multi-tidal planning horizon. They also perform a comparative study to investigate the performance of the proposed Lèvy Flight based algorithm. Results demonstrate the effectiveness of the proposed method in terms of the study's goals while comparing it with state-of-the-art approaches, i.e., PSO, CPLEX (exact approach), and iterated greedy (IG) heuristic algorithms.

The research presented in [20] also deals with the DD-BAP using a novel version of GA, namely GA with state-space reductions (GASSR) for efficient berth allocation, with the objectives of minimizing total service time and maximizing the quay occupation simultaneously. A large number of experiments have been performed on small-scale and large-scale problem instances. Results show the effectiveness of the proposed GASSR over counterparts, i.e., standard GA, standard MA, and MA with state-space reductions (MASSR) in terms of optimal berth allocation and lower computational complexity.

The work presented in [42] focuses on dynamic BAP, where waiting time along with fuel consumption of vessels when they are in the anchorage area is also investigated. The problem is first formulated as an MILP model. In their proposed model, the fuel cost is associated with the waiting times of arriving vessels. The key reason for including fuel cost/ consumed fuel is to address the sustainability aspects of the BAP. Later, this study proposes a chemical reaction optimization (CRO) algorithm to solve the BAP and real-time instances are employed for experiments. Eventually, a comparative study has been taken into account in order to validate the newly proposed CRO algorithm, where GA, block-based GA, and PSO methods are considered as benchmarks. Results show the efficacy of the proposed CRO method in terms of efficient utilization of MCT resources.

Sheikholeslami *et al.* in [130] investigates dynamic and discrete BAP, where the key objective is to mitigate the late departure of vessels. They utilized a real-time dataset from the Port of Shahid Rajaee situated in Iran, where tide effects are also investigated during implementations. The problem is modeled as an MILP model and solved by the exact approach. Experimental results guarantee that late departures are reduced while implementing the proposed solution.

3.2 Continuous and Dynamic BAP

In the continuous and dynamic BAP (CD-BAP) formulation, continuous berthing layout (Section 2.4.1.1) and dynamic vessel arrivals (Section 2.4.1.2) are considered.

Frojan *et al.* in [51] consider three wharves with the length of 800, 600, and 100 meters, and 12 vessels that arrive dynamically for mooring. To tackle this CD-BAP with multiple quays, an integer linear model has been developed to describe the elements of the problem and interactions, and then the problem is solved with GA. Furthermore, the authors perform extensive simulations by employing several real time instances as well as past datasets used in the existing literature. Results demonstrate that the proposed method provides a high quality solution in terms of total operational cost that includes waiting and late departure costs.

The study presented in [125] applies an evolutionary method, namely a differential evolution (DE) algorithm. The authors also explore the impact of DE's user-defined parameters on the solution of the problem and conduct a statistical analysis for establishing the optimal values for DE. Finally, extensive simulations have been carried to affirm the productiveness of the proposed DE based solution, and results are compared with several state-of-the-art approaches, i.e., mixed integer programming (MIP), simulated annealing (SA), immune algorithm (IA), and greedy randomized adaptive search procedure (GRASP) algorithms.

The work presented in [11] discusses CD-BAP, where the major objective is to mitigate the late departures of vessels by efficient berth allocation. For near-optimal berth position searching, this study develops a hybrid of GA and branch and cut (B&C) methods, which assigns the best berthing location based on vessels' arrival and departure times and other constraints. Furthermore, the proposed method is tested on both small and large datasets, and compared with standard GA and CPLEX methods. Results indicate the effectiveness of the proposed solution to the BAP over counterparts.

The authors of [49] also deals with the CD-BAP in order to avoid delays in vessels' departures. The authors also consider tidal constraints during the berth allocations. The CD-BAP is formulated in three different ways, i.e., standard MILP, MILP based on sequence variables (S-MILP), and time-indexed variables-based MILP (TI-MILP). This study also develops datasets based on real-time characteristics of ports, which can be used in the future as benchmark instances. Subsequently, these three models are solved on a CPLEX solver and simulation results demonstrate that the time-indexed-based MILP achieves higher efficiency with lower computation time.

Another study investigates the CD-BAP in [148], where the primary objective is to achieve a robust berth allocation strategy along with minimum turnaround time. A heuristic-based grey wolf optimization (GWO) algorithm is developed to solve the BAP. Furthermore, this study also considers uncertainties in vessels' arrival and operational time of vessels, and the proposed GWO based method shows efficiency in solving the BAP with uncertainties over existing solutions such as GA and CPLEX.

The authors of [76] also examine the CD-BAP intending to reduce the total time that vessels stay at the port for loading/unloading operations. They transformed constrained single-objective BAP to unconstrained multi-objective BAP model by converting constraints' violation as objectives. Next, the multi-objective continuous BAP is solved by modified non-dominated sorting genetic algorithm II (MNSGA-II). Furthermore, the newly proposed unconstrained multi-objective model along with the proposed method is tested on benchmark instances. The results indicate the effectiveness of the proposed model and MNSGA-II method over counterparts in terms of minimum handling time of arriving vessels at the port.

A genetic algorithm (GA) based solution of the CD-BAP is presented in [34]. The key concerns of this work are to attain port efficiency and reduce late departures and in this way, the penalty cost will be reduced. The GA-based solution of the BAP is tested on real-time instances and results show the effectiveness of the proposed method.

Xu *et al.* developed a hybrid SA-based heuristic method to deal with the BAP while considering traffic limitations in the navigation channel [151]. The primary aim of this work is to propose cost-efficient berth allocation while enhancing the performance of MCT. The problem is formulated using the MILP model and then solved by the hybrid SA (HSA) method that combines SA and reheat treatment methods. In order to validate the newly proposed method, real time instances from two container terminals of Tianjin, China are used and CPLEX and Greedy methods are also applied for comparative study. Results demonstrate the effectiveness of the proposed HSA method in terms of lower cost and MCT efficiency.

Another study presented in [99] examines the CD-BAP where the major objective is to minimize total weighted handling time along with deviation cost that occurs by not assigning vessels to their preferred berthing positions. Different variants of the SA algorithm are proposed to solve the BAP and then mathematical and other heuristics are also implemented for comparison purposes. They test the proposed method on several datasets including small and large instances, and results indicate the effectiveness of the proposed SA method in terms of the above-mentioned primary objectives.

Another study [113] focuses on the CD-BAP with the aim of reducing total waiting cost, handling costs, and late departures penalties. An SA-based hybrid algorithm is developed to solve the problem, where one algorithm deals with berthing positions, and the other algorithm determines berthing times. Eventually, several experiments were performed and results are compared with existing methods. It is validated from comparison, the proposed method provides efficient berth allocation within acceptable computational time.

The work presented in [157] deals with the BAP where the continuous wharf is assumed and various priorities are taken into account, e.g., priori berthing of VIP customers. The objective of this work is to improve robustness and attain minimum handling cost. A heuristic-based SA is developed and tested on data instances taken from past literature. Results denote that the proposed method is able to find the optimal solutions with minimum cost and maximum robustness.

Song *et al.* also focus on the CD-BAP while considering time-varying water depths [132]. The primary objective of this study is to mitigate the total turnaround time of arriving vessels. They develop improved DE-based solution and implement on small and medium scale instances. It is affirmed from simulation results that the proposed DE-based method is appropriate for optimal berth allocation over compared algorithms, i.e., simple DE, GA, Tabu search, and integer programming.

Liu *et al.* focus on the CD-BAP [102], where the mooring schedule of arriving vessels at continuous berthing layout is provided by employing the two-stage robust optimization (RO) method. Furthermore, this study also deals with uncertainty in the problem, which includes uncertainty in vessels' arrival and handling time. Extensive simulations have been carried out to affirm the productiveness of newly proposed approaches, i.e., expanded RO-BAP (ERO-BAP) and risk constrained RO-BAP (RCRO-BAP).

A fuzzy logic-based solution is developed in [117] to deal with CD-BAP, where the primary objective is to minimize vessels' waiting time. In addition, this study also models uncertainty in vessels' arrivals. They performed simulations on randomly generated data and results show the productiveness of the proposed method.

3.3 Hybrid and Dynamic BAP

This section presents a detailed review of current works that investigate the BAP with hybrids berthing layout and dynamic vessels' arrival as discussed in sections 2.4.1.1, 2.4.1.2, respectively.

Umang *et al.* studied the HD-BAP with the objective of minimizing total service time of arriving vessels [139]. The problem is first modeled as an ILP problem and then it is solved by the exact approach as well as by a metaheuristic method, namely squeaky wheel optimization (SWO). The SWO method works on the principle of a construct, analyze, and prioritize, where, at each iteration, possible solutions are constructed and analyzed, and results are used to build a new priority order to attain new solutions in the next iteration. In order to affirm the productiveness of the proposed SWO algorithm, several experiments are performed on real-time data taken from the container terminal at SAQR port, UAE. Results are also compared with other methods, i.e., MILP, FCFS, and generalized set partitioning problem (GSPP), to show the efficacy of the proposed method.

A study presented in [86] examines the hybrid BAP and an EA is proposed to solve the problem. The objective of this study is to minimize the total cost, which includes handling cost, waiting cost, and cost occurred due to delays in departures. In order to check the productiveness of the proposed EA method, several simulations have been performed and results show the efficiency of the proposed algorithm over counterparts.

The authors of [41] investigate the hybrid BAP where the key objective is to mitigate total cost that includes waiting and handling costs. They developed a heuristic-based deterministic variant of variable neighborhood search (VNS) named variable neighborhood descent (VND). In order to test the performance of the proposed VND method, simulations have been performed on two datasets and results show the effectiveness of the proposed algorithm over other meta-heuristics, i.e., bee colony optimization (BCO) and EA.

The authors of [84] investigate the BAP with the objective of minimum vessels' handling cost, where hybrid berthing layout is taken into account. They developed two methods to solve the hybrid BAP, namely sedimentation algorithm (SEDA) and SEDA with an estimation & rearrangement heuristic (SEDA+ERH). The first method is an exact combinatorial optimization algorithm and the latter method employs a heuristic as a pre-processing step to alleviate the computational complexity. The proposed methods are tested on three different instances taken from the literature. Experimental results denote the effectiveness of SEDA+ERH in terms of minimum cost in affordable computation time.

Imai *et al.* also deal with a novel variant of BAP, where two or more vessels can be moored at a single berth if the length of vessels is less than the length of a certain discrete berth [71], where they formulated their problem as an integer linear programming problem for simple calculations. The major objective of this study is to provide an efficient solution in terms of higher berth productivity for feeder ships and mega-container ships. Furthermore, GA is employed to solve this problem and various experiments have been carried out in order to validate the effectiveness of the developed method.

Kovavc *et al.* extend their previous work on the hybrid BAP ([41]) and proposed a new method namely general variable neighborhood search (GVNS) [87]. The problem is formulated as MILP and then solved by GVNS along with other three meta-heuristics that were proposed in their previous work, i.e., EA, BCO, and VND. The proposed method and other compared methods are tested on randomly generated datasets and results denote that the newly proposed GVNS outperformed EA, BCO, and VND in terms of computation times while maintaining the superior quality of solutions.

Azza *et al.* developed a meta-heuristic based bat-inspired algorithm to deal with the hybrid BAP [19]. The major objective of this work is to reduce vessels' stay time at the terminal. Furthermore, they performed extensive simulations to validate the efficacy of the proposed algorithm, and the results are compared with the CPLEX solver. The results from simulations are evident that the proposed method is efficient over counterparts in terms of providing good solutions in minimum computation time.

Another study also deals with the BAP while assuming various irregular berthing layouts [39]. The primary objective is to minimize the total stay times of vessels at the port. Basically, they solved discrete and dynamic BAP; however, due to considering irregular berthing layouts, it becomes hybrid BAP, where one vessel can take two berths if its length exceeds one berth. In order to solve this type of problem, first, an exact method based MILP is developed for small instances and then a heuristic-based iterated local search (ILS) is developed for large scale problems. For simulations, they employ data from a tank terminal and the results from experiments indicate the effectiveness of the proposed method.

The work presented in [42] examines two variants of BAP, i.e., the DD-BAP and hybrid dynamic BAP. In their proposed hybrid BAP formulations, a large vessel can take more than one berths and two or more vessels can take one berth if they are small. In this study, the problem is first formulated as a mixed-integer non-linear programming (MINLP) model, then it is solved by the CRO algorithm. The primary objectives of this work are to alleviate handling and fuel costs. Extensive simulations are carried out to affirm the productiveness of the proposed CRO based method and results guarantee its higher performance over counterparts.

Bouzekri *et al.* developed an integer linear programming (ILP) based solution for the hybrid BAP [25], where they tested their proposed method on a real-time dataset taken from the port of Jorf Lasfar, Morocco. Extensive simulations have been carried out and results denote that the ILP based method can solve the hybrid BAP considering up to 40 vessels within reasonable computation time.

The HD-BAP is also investigated in [47], where a hybrid GA (HGA) algorithm is developed. In order to affirm the productiveness of the proposed HGA method, several simulations have been conducted on various datasets, i.e., small, medium, and large datasets, where 10 vessels with 2 berths, 30 vessels with 7 berths, and 50 vessels with 12 berths are taken into account. Simulation results show the effectiveness of the proposed HGA method over CPLEX in terms of minimum computation time.

Table 2: Summary of current literature related to BAP. [RD: random data, RPD: real port data]

Ref.	**Pub. year**	**Developed method**	**Compared method(s)**	**Employed dataset**
DD-BAP				
[137]	2014	PSO	GSSP, LP, and clustering search	[36], RPD
[131]	2015	GA	CPLEX	RD
[69]	2015	GA	–	RD
[92]	2016	POPMUSIC	PSO	[31, 91]
[109]	2016	ALNS	PSO, CS, GSPP, GRASP, SA, CPLEX, and TS	[36, 91]
[45]	2018	MA	FCFS and EA	[31, 70]
[46]	2018	SAEA	Standard EA, AEA, DPCEA	[57, 31, 70]
[128]	2019	ITS	TS, and stochastic DP	[36]
[144]	2019	Lèvy Flight	PSO, CPLEX, and IG	[150, 90]
[20]	2020	GASSR	Standard GA, standard MA, and MASSR	RPD
[42]	2020	CRO	GA, block-based GA, PSO, and exact approach	RPD
[130]	2020	MILP	–	RPD
[42]	2022	AACS	ACO and exact method	[43]
CD-BAP				
[51]	2015	GA	CPLEX, GRASP, TS, and SBS	[115]

Continued on next page

Table 2 – Continued from previous page

Ref.	Pub. year	Developed method	Compared method(s)	Employed dataset
[125]	2016	DE	SA, MIP, IA, and GRASP	[59]
[11]	2016	Hybrid GA and B&C	Standard GA and CPLEX	RD
[49]	2017	TI-MILP	MILP and S-MILP	RPD [2]
[148]	2017	GWO	CPLEX and GA	[116]
[76]	2017	MNSGA-II	GA, NSGA-II-III, and ALNS	[55]
[34]	2017	GA	–	RPD
[113]	2018	SA	GA and CPLEX	[110]
[99]	2018	SA	GRASP, TS, SBS, and GA	[93, 36]
[157]	2018	SA	–	RD
[132]	2019	IDE	DE, GA, TS, and IP	[56]
[102]	2020	ERO, RCRO	MILP and S-MILP	[116]
[117]	2020	Fuzzy logic	–	RD
[60]	2021	PSO	MILP	RPD
[17]	2022	CSA	MILP and GA	[125]
HD-BAP				
[139]	2014	SWO	MILP, FCFS, and GSPP	RPD
[86]	2015	EA	CPLEX	[32]
[41]	2015	VND	EA and BCO	[32]
[84]	2016	SEDA and SEDA+ERH	CPLEX	[56]
[87]	2018	GVNS	EA, VND, BCO, and CPLEX	[32]
[19]	2018	Bat-inspired	CPLEX	[36, 72]
[39]	2019	ILS	MILP	RPD
[42]	2020	CRO	GA, block-based GA, and PSO	RPD
[25]	2020	ILP	–	RPD [26]
[47]	2020	HGA	CPLEX	RPD

4 Literature Review on Combined BAP with QCAP

This section examines in detail current studies that deal with the combined BAP and QCAP, i.e., discrete and dynamic BAP with QCAP, continuous and dynamic BAP with QCAP, and hybrid and dynamic BAP with QCAP, summarized in Table 3.

4.1 Discrete and Dynamic BAP with QCAP

This section reviews discrete and dynamic BAP, while the QCAP is also taken into account.

Lalla *et al.* examine the combined BAP and QCAP in [89] for reducing the total weighted service time of ships and propose a new heuristic method named migrating birds optimization (MBO). The MBO is a newly developed nature-inspired technique that is based on the V-formation flight of migrating birds. Furthermore, the proposed MBO method is tested on five datasets, with different numbers of vessels and berths, taken from recent literature [36]. Simulation results show that MBO beats existing approaches, i.e., CPLEX and PSO, in terms of lower computation times.

A study presented in [28] solves the BAP along with the QCAP by proposing a model predictive control (MPC)-based method, where the primary concerns of this work are to reduce total handling and waiting costs. In order to affirm the productiveness of the proposed MPC-based method for berth and QC allocation, several experiments are performed on a real-time dataset taken from Indonesian Seaport. The results from simulations validate the performance of the proposed technique over FCFS and a density-based strategy.

A multi-objective berth and QC allocation model is developed in [94], where the major objectives are to ensure the earliest departure time of vessels and enhance terminal efficiency. The problem is formulated as an MIP model and solved by the newly-developed chaos cloud PSO (CCPSO) method. A comparison study was also carried out to verify the productiveness of the proposed optimization algorithm. Experimental results indicate the effectiveness of the proposed method over GA in terms of the earliest departure times.

Lu *et al.* developed improved NSGA-II for solving the combined BAP and QCAP in [101], where discrete berthing layout is assumed. In this work, the authors also consider uncertainties in ships' arrival times and container handling times. Additionally, movable

QCs (that can move to other berths/ships before completing the process of an assigned ship) are examined to enhance the flexibility of the MCT. The problem is formulated as a non-linear MIP problem and then solved by improved NSGA-II. Several experiments are carried out to affirm the benefits over standard NSGA-II.

The paper presented in [149] investigates both the BAP and the QCAP problems of MCT while employing discrete berth allocation. The primary objective of this study is to mitigate the total handling costs. First, the problem is formulated as an MIP model that also deals with several disruptions in vessels' handling, i.e., deviation of ships' arrivals, uncertainties in vessels' unloading/loading times, and failure of QCs or other handling equipment. Then, a heuristic-based rolling horizon (RH) algorithm is developed to find a feasible solution under disruptions, which is further tested on real-time instances. The results from experiments affirm the efficacy of the proposed reactive method over the proactive technique.

Another study presented in [96] investigates the same problem and develops a stochastic ILP (S-ILP) model to solve it. This study also considers uncertainties in QCs maintenance activities and the primary aim of this work is to minimize total turnaround time. In order to verify the productiveness of the proposed model, the authors perform simulations on 90 instances and the results verify the proposed model in terms of higher efficiency and accuracy.

The paper presented in [65] investigates the discrete and dynamic combined BAP and QCAP. In order to make the problem more realistic, this study considers variable QCs assignment that makes MCT more flexible. Furthermore, three hybrids of GA are developed to solve the problem, where three different mutations are employed for three algorithms, i.e., swap mutation, thoros mutation, and thoras mutation. Finally, extensive simulations have been carried out to validate the proposed methods. The results from simulations affirm the productiveness of proposed techniques over GA.

Abou *et al.* also investigated a similar problem with the objective of minimizing the total service time of arriving vessels [5]. This study considers both QCs allocations, i.e., static and dynamic allocations. Furthermore, the problem is mathematically formulated and an exact solution is proposed. For experimental purposes, the authors perform a case study on Abu Dhabi's container terminal and the proposed solution is implemented on the same terminal. The results from experiments indicate that the proposed method attained decreased service times over the current operational approach.

The authors of [28] extend their work and solve the combined berth and QC allocation problem in [29], where they proposed a dynamic modeling framework that is based on discrete event systems (DESs), which explains the mooring procedure with several discrete berthing slots and multiple QCs. After dynamic modeling, a new algorithm, namely model predictive allocation (MPA), which is based on the MPC principle, is developed to solve the integrated problem. Extensive simulations have been performed on real-time data taken from Tanjung Priuk port, Indonesia, to verify the effectiveness of the proposed method and results compared with FCFS, GA, and PSO methods. Results show the efficacy of the proposed method over counterparts.

The work presented in [35] also deals with the combined BAP and QCAP. Their formulation allows the reassignment of ships to other terminals in a multi-user terminal. The primary objective of this study is to reduce the total service cost and this work claims that unnecessary movements of QCs reduce MCT efficiency. To tackle this issue, heuristic-based GRASP and filtered beam search (FBS) are developed. Furthermore, real-time datasets from Busan Terminal, Republic of Korea, are employed for experiments in order to validate the productiveness of the proposed method.

In [146], an exact method, namely a decomposition algorithm, is developed to deal with combined BAP and QCAP, where the basic objective is to minimize the total cost incurred due to deviations from departure times and berth positions. In addition, the authors also consider two uncertainty factors, namely the increase/decrease in the number of containers and the late arrival of ships. They conduct several experiments with randomly generated data and the results confirm the effectiveness of the decomposition method.

4.2 Continuous and Dynamic BAP with QCAP

In this section, current studies on another form of BAP and QCAP are investigated, where continuous berthing layout and dynamic vessels arrivals are taken into account.

A two-phase model of BAP and QCAP is developed in [62], where the major aims are to improve port resources savings and reduce QCs movements to enhance MCT efficiency. As a first step, BAP is solved with a metaheuristic-based PSO algorithm and then QCAP is addressed by a CPLEX solver. In order to confirm the productiveness of proposed solutions for the combined problem, experiments are performed on real-time datasets that were adapted from the Ningbo Beilun container terminal, China.

The work presented in [74] also cope with the combined problem. For increasing handling productivity of the MCT, an ALNS-based heuristic is proposed to mitigate overall cost, which includes penalty cost due to late departures and QCs assignment cost. The output of the ALNS method is compared with GA, TS, and SWO algorithms, where comparison analysis demonstrates the efficacy of the proposed method.

Correcher *et al.* studied the continuous BAP and QCAP with time-invariant crane assignment in [38], where the primary objective was to reduce total handling cost along with vessel service times. They proposed biased random-key GA with memetic improvement and local search to solve the problem and the proposed method is tested on several real-time instances to verify its performance. Experimental results demonstrate that the biased random-key GA provides optimal solution considering up to 40 vessels; however, it can also provide near-optimal solutions considering up to 100 ships within reasonable computation time.

[145] consider carbon emission policies in order to reduce total carbon emission at the port along with the primary objectives of the study, i.e., reduction in penalty costs and operating costs. Basically, total operating and carbon emission costs both depend on the operating hours of QCs and unnecessary operations of QCs because inefficient berth and cranes allocation lead to high carbon and handling cost. In order to solve this joint problem, a branch and bound algorithm was developed, which was tested on several real-time instances, taken from [32]. The results from experiments affirm the benefits of the proposed method over the CPLEX solver.

The authors of [6] investigate the combined BAP and QCAP. First, the problem is mathematically formulated based on relative position formulation and then solved by two approaches, the exact approach and using the RH method. In addition, this work considers several uncertainties and various inequality constraints. In order to assure the effectiveness of developed methods, experiments are performed on real-time instances extracted from a multi-user terminal that was primarily used for bulk cargo operations. The results from experiments denote that the exact approach easily solves small-scale instances and

RH is a more suitable algorithm for large-scale instances.

An MIP model is developed in [156] to deal with the BAP and the QCAP simultaneously, where the GA and its variant, improved GA (IGA), are developed for minimizing total service cost, which includes handling cost, waiting cost, and QCs conversion cost. In order to validate the proposed method, several experiments are carried out and results show the efficacy of IGA over standard GA and CPLEX.

Correcher *et al.* also deal with the same problem in [37]. The continuous BAP with QCAP is modeled as an MILP problem, where arriving vessels can be moored at any location throughout the quay. Next, the problem is solved following an iterated local search (ILS) approach. Simulation results denote that the proposed method efficiently solves the problem considering 50 vessels in a week.

Zheng *et al.* also studied the same problem while considering QCs maintenance [163]. The problem is formulated as the ILP model and solved by the exact approach; however, it can solve the problem considering up to only 18 vessels. Then, this study develops improved GA and a novel heuristic named left and right vessel move (LRVM) algorithm. Furthermore, this study is also compared with the previous work presented in [6] while ignoring QCs maintenance constraints. Results from simulations affirm the productiveness of the proposed method over counterparts.

A combined BAP and QCAP is investigated in [108], where the primary objective is to enhance the performance of MCT by efficiently allocating berths and QCs to arriving ships. The authors of this study developed a novel variant of the PSO algorithm, namely random topology PSO (RTPSO). Unlike standard PSO, the proposed RTPSO works on two basic rules while assuming there are k neighborhoods: each particle is connected to itself and it is also linked to $k-1$ particle. The problem is formulated as an MIP model and several tide constraints are also considered in the formulations. In addition, several simulations on self-generated instances are performed to validate the performance of the proposed RTPSO method.

The authors of the study presented in [97] investigated a new bi-objective optimization model of BAP and QCAP while considering QCs maintenance activities. The objectives of this work are twofold: reducing the total turnaround time of serving ships and minimizing the total penalty costs of QCs maintenance tardiness and earliness. The problem is first formulated as an ILP model and then it is solved by a heuristic-based two-phase iterative method (IM). The authors tested their newly designed method on real-time and randomly generated instances to affirm their productiveness and effectiveness.

Krimi *et al.* studied the same problem in [88] while considering several uncertainties, e.g., bad weather conditions, QCs maintenance activities, etc. In order to improve the performance of the container terminal located at a port in Morocco, the joint problem is formulated as an MIP model. Later, it is solved with a GVNS-based heuristic algorithm. Furthermore, various experiments are performed on a real dataset taken from the practical problem of the port in Morocco. Experimental results demonstrate that the compared method, a CPLEX solver, takes higher computation time and in many cases cannot solve the problem in a given time frame. However, the proposed GVNS guarantees optimal or near-optimal solutions in reasonable computation time.

In [135], a search-based (remove & reinsert) heuristic (SBH) is developed for the combined BAP and QCAP, where the authors' goal is to reduce the costs involved in container handling. The problem is formulated as a MIP that can handle both fixed and flexible departure time settings. To confirm the effectiveness of the proposed method, several simulations are performed with small, medium, and large data instances.

4.3 Hybrid and Dynamic BAP with QCAP

This section examines the recent status of literature that deals with one of the most challenging problem of seaside operations, i.e., the combined hybrid and dynamic BAP with QCAP [139, 75].

A study presented in [8] examines the combined hybrid BAP and QCAP. Its objective is to minimize the total service times of arriving vessels. A GAMS solver is employed to solve the modeled integrated problem with an exact solution. Data instances are generated randomly that assume 3 ships with 4 berthing positions. Simulation results affirm its effectiveness in terms of optimal berth and QCs allocations.

Alnaqabi *et al.* have studied the combined problem of BAP and QCAP [9] while assuming a fixed number of QCs at each berthing position and a safety distance between adjacent QCs. The ultimate objective of this work is to reduce the total processing time of ships. The problem is modeled as MILP and then solved by a GAMS solver. Simulations are also carried out to validate the productiveness of the proposed solution and results show that the proposed method gave an optimal solution in reasonable computation time.

The authors of [134] also deal with the combined hybrid BAP and QCAP, where hybrid berthing layout and dynamic vessel arrivals are assumed. Furthermore, the non-crossing constraints of QCs are assumed in this study. Finally, a simulated annealing (SA) based heuristic algorithm is developed to minimize late departures of vessels. In order to test the proposed SA-based approach, simulations are performed on a randomly generated dataset that includes 40 vessels, 7 berths, and 18 QCs. Simulation results indicate that the proposed heuristic-based can easily solve large-scale instances.

Table 3: Summary of current literature related to combined BAP and QCAP. [RD: random data, RPD: real port data]

Ref.	Pub. year	Developed method	Compared method(s)	Employed dataset
DD-BAP+QCAP				
[89]	2015	MBO	CPLEX and PSO	[36]
[28]	2015	MPC	FCFS & density-based strategy	RPD
[94]	2017	CCPSO	GA	RD
[101]	2018	Improved NSGA-II	Standard NSGA-II	RPD.
[149]	2018	RH method	CPLEX and greedy	RD, [148]
[96]	2019	S-ILP	–	RD
[5]	2019	Exact model	Existing operational practices	RD
[65]	2019	HGAs	Standard GA	RD
[29]	2019	MPA	FCFS & density-based strategy, GA, and hybrid PSO	RD, [28]
[35]	2020	GRASP, FBS	Iterative approach	RPD
[146]	2021	Decomposition Algorithm	CPLEX	RD
CD-BAP+QCAP				
[62]	2015	PSO	CPLEX	RPD
[74]	2017	ALNS	GA, TS, and SWO	[110]
[38]	2017	GA	Discrete DE	[116]

Continued on next page

Table 3 – Continued from previous page

Ref.	Pub. year	Developed method	Compared method(s)	Employed dataset
[6]	2018	RH method	Exact approach	RD
[156]	2019	IGA	Standard GA, CPLEX	–
[37]	2019	ILS	MILP and B&C	[116]
[108]	2020	RTPSO	PSO and exact approach	RD
[97]	2020	Two-phase IM	CPLEX	RPD
[88]	2020	GVNS	CPLEX	RD
[135]	2021	SBH	MIP	[116, 110]
HD-BAP+QCAP				
[8]	2016	GAMS	–	RD
[9]	2016	GAMS	–	RD
[134]	2019	SA method	–	RD

5 Enhanced Berth Allocation using the Cuckoo Search Algorithm

5.1 Motivation and Problem Statement

There exist several studies that deal with well-known problem, namely BAP, using different approaches, i.e., mathematical and metaheuristic/ evolutionary approaches [69, 124, 131, 148]. Since BAP is an NP-hard problem, current researches try to use heuristic algorithms that provide near-optimal solutions in an affordable computation time [17]. This is because, depending on the needs of container terminals, a small amount of computation time is required, which makes metaheuristics an advantageous way to solve BAP. As mentioned earlier, several metaheuristics and evolutionary approaches have been developed in this area to solve BAP, e.g., GA, PSO, GWO, etc. In this paper, for the first time (to the best of our knowledge), a novel cuckoo search algorithm (CSA) is proposed to solve the dynamic and continuous berth allocation problem (DC-BAP). The CSA was first introduced in [15] (part of this thesis) for solving the dynamic and continuous BAP (DC-BAP). In this study, we extend the BAP mathematical formulation by considering penalty for non-optimal berth position, safety time interval between consecutive berth arrivals, and account for smaller time intervals (30-minutes time interval), which offer better, more fine-grained berth allocation decisions.

5.2 Problem Formulation

This section first describes in detail the BAP considered in this work, followed by a mathematical formulation as a mixed-integer linear programming problem. Table 4 lists all abbreviations and notations used in this section and throughout the paper.

In the dynamic and continuous berth allocation problem, the MCT has one or more continuous berthing layouts of known lengths that serve vessels arriving at different points in time (i.e., in a dynamic fashion). Let $B = \{1,2,...,M\}$ denote the set of all possible berthing positions on the wharf of the port. Typically, the BAP considers a particular time period of vessel arrivals, such as the next 48 hours. Hence, time is modeled as a set of time intervals $T = \{1,2,...,K\}$ that can represent some time duration of interest (e.g., an hour or a 30-minute interval). Finally, let $S = \{1,2,...,N\}$ denote the set of ships arriving at the terminal. For each ship, the estimated time of arrival (ETA), the preferred berthing position (PBP), the ship's length, and the estimated (or requested) time of departure (ETD) are known in advance.

In the ideal scenario, as soon as a vessel arrives at the MCT, it should be moored at its preferred berthing position. If the MCT cannot serve the vessel at the time of arrival, the vessel must be towed to the waiting area of the terminal, as shown in Figure 1, and/or berth at a non-optimal berthing position. In the first scenario, the number of ships in the waiting area increases, causing congestion and navigational challenges to arise at the seaside of the terminal. In this case, the MCT incurs an extra waiting cost WC_s against the ship s for the duration of s' waiting time (e.g., calculated in EURO/hour).

Once the ships are moored at their assigned berthing position, the quay cranes (QCs) start working in order to load/unload containers. Container handling resources (e.g., number of QCs, gantry cranes) are allocated to ships based on the handling rate that is negotiated between the MCT operator and the shipping company. The handling time for ship s at the assigned berthing position is calculated based on the total number of containers loaded on that ship and the requested handling productivity. Note that this study adopts a dataset for implementation with precomputed handling times for all arriving vessels. However, the handling productivity is reduced if the vessel is assigned to a berth position other than its preferred berthing position (PBP) [24, 23]. The PBP typically depends on vessel characteristics such as the vessel length or vessel load as well as port-related considerations such as the number of available quay cranes of the berthing area allocated to a particular ship. Hence, the major cause of handling productivity reduction is the increased loading/unloading and transfer time of containers from the assigned (suboptimal) berth to storage.

Finally, each ship s specifies its own estimated (or requested) time of departure ETD_s and the MCT is supposed to complete the tasks (loading/unloading) of s before the ETD_s, $\forall s \in S$. Otherwise, the MCT is liable to pay a late departure penalty cost LDC_s for the duration of the delay (e.g., calculated in EURO/hour) to the shipping companies. Overall, the aim of the MCT is to minimize the total waiting, handling, and late departure costs for all arriving vessels at the port.

5.2.1 Mathematical Formulation

Before disclosing the mathematical formulation of BAP, we list the assumptions that are considered in our work.

- The total number of arriving ships at the planning horizon is known.
- Each berth position is able to handle only one vessel at a particular time.
- A ship takes consecutive time intervals until loading/ unloading completes (i.e., no shifting).
- The ETA and ETD for each vessel are known and will not change.
- Estimated processing time for each vessel is known.
- Each ship has a preferred berthing position and it is known.
- Each ship can berth and be served at any berthing position.
- All berths are idle at the start of the time horizon.
- The length of the wharf is known.

The total processing cost of a vessel s that is scheduled for berthing at position BP_s at time BT_s includes a waiting cost, a handling cost, and late departure penalty, expressed by the following function:

$$\begin{aligned} Cost(s, BP_s, BT_s) = & WT_s \cdot WC_s \\ & + HT_s \cdot (HC_s + |BP_s - PBP_s| \cdot NBC_s) \\ & + LDT_s \cdot LDC_s \end{aligned} \tag{1}$$

The first term in Equation (1), $WT_s \cdot WC_s$, represents the waiting cost when a vessel has to wait for berthing. The waiting time WT_s of vessel s is calculated as the difference between the berthing time BT_s and the planned time of arrival ETA_s:

$$WT_s = BT_s - ETA_s, \quad \forall\, s \in S \tag{2}$$

Table 4: Mathematical notation used in Section 5.2.1

Name	Explanation	Name	Explanation
Notations			
BP_s	Berthing position of ship s	BT_s	Berthing time of s
ETA_s	Estimated time of arrival of s	ETD_s	Estimated time of departure of s
HC_s	Handling cost of s per time period	HT_s	Handling time of s
L_b	Length of berth b (only for discrete berthing layout)	L_s	Length of ship s
LDC_s	Late departure cost of s per time period	LDT_s	Late departure time of s
NBC_s	Non-optimal berthing cost of s	PBP_s	Preferred berthing position of s
SET	Safety entrance time	W	Length of wharf
WC_s	Waiting cost of s per time period	WT_s	Waiting time of s
Indices			
$b \in B = \{1,2,...,M\}$	Berthing position		
$s \in S = \{1,2,...,N\}$	Individual ship		
$t \in T = \{1,2,...,K\}$	Single time period		

The second term ($HT_s \cdot (HC_s + |BP_s - PBP_s| \cdot NBC_s)$) in Equation (1) corresponds to the total handling cost for loading or unloading containers, which is proportional to the handling time. Basically, the handling time HT_s of any ship s depends on the total volume of containers to be loaded or unloaded on the vessel, the number of quay cranes available at this berth, and the average handling productivity of the cranes. However, in this study, we assume handling time as an input for solving BAP, as considered in [125]. Furthermore, unlike our previous work [15], this study introduces a new aspect of the handling cost, which will penalize the total service cost based on the absolute difference between the assigned berthing position BP_s and the preferred berthing position PBP_s for the entire duration of handling that a ship spends at the non-optimal position. The factor NBC_s defines the penalty cost for ship s against non-optimal berthing position per unit (meter) distance. In order to NBC_s to be comparable to the handling cost HC_s, we set NBC_s as the ratio of HC_s to the length of the wharf W.

The final term in Equation (1), $LDT_s \cdot LDC_s$, computes the late departure penalty when a vessel departs after its estimated time of departure. The delayed departure time LDT_s of vessel s (if any) is calculated as the difference between the time s completes its operations and the estimated time of departure ETD_s.

$$LDT_s = max\{BT_s + HT_s - ETD_s, 0\}, \quad \forall\, s \in S \tag{3}$$

The goal of the dynamic and continuous berth allocation problem is to find the optimal berthing positions and times for all vessels such that the total processing cost is minimized, as shown by the following objective function:

$$\textbf{minimize} \sum_{s \in S} \sum_{b \in B} \sum_{t \in T} x_{sbt} \cdot Cost(s, BP_s, BT_s) \tag{4}$$

subject to the following set of **constraints**:

$$x_{sbt} \in \{0,\ 1\}, \qquad \forall\, s \in S,\ b \in B,\ t \in T \tag{5}$$

$$\sum_{b \in B} \sum_{t \in T} x_{sbt} = 1, \qquad \forall\, s \in S \tag{6}$$

$$BTs \geq ETA_s, \qquad \forall\, s \in S \tag{7}$$

$$|BT_s - BT_{s'}| \geq SET, \qquad \forall\, s, s' \in S \tag{8}$$

$$BP_s + L_s \leq W, \qquad \forall\, s \in S \tag{9}$$

$$\sum_{s' \neq s \,\in\, S} \; \sum_{b = BP_s - L_{s'} + 1}^{BP_s + L_s} \; \sum_{t = BT_s - HT_{s'} + 1}^{BT_s + HT_s} x_{s'bt} = 0, \quad \forall\, s \in S \tag{10}$$

In Equation (5), the variable x_{sbt} is 1 if vessel s is assigned to berthing position b at berthing time t, and 0 otherwise. Constraint (6) ensures that each arrived ship at the MCT will be assigned at a particular berthing position only once during the planning time. Constraint (7) warrants that the scheduled berthing time BT_s of ship s must always be later

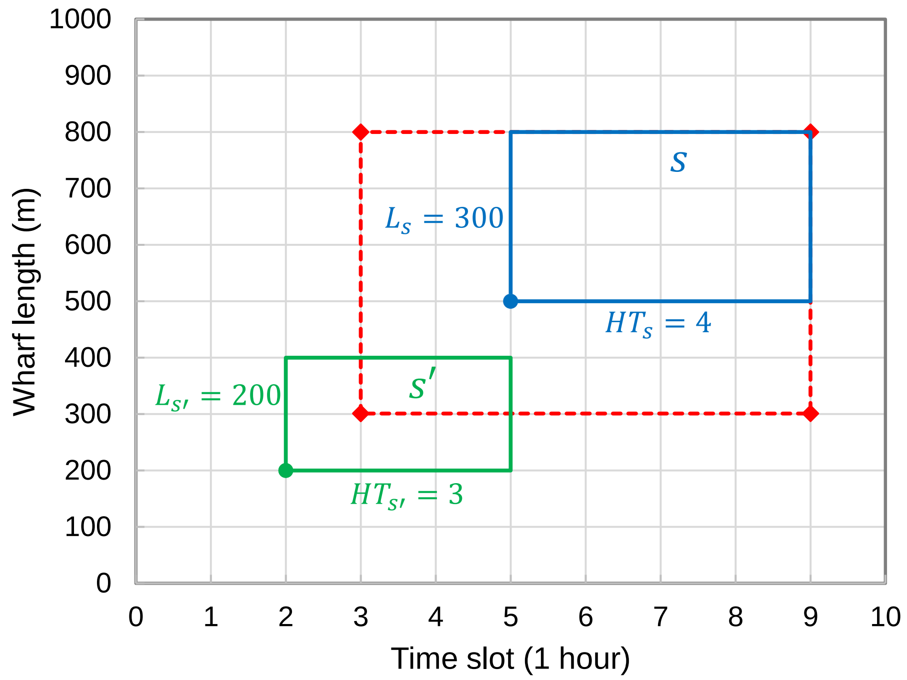


Figure 5: Berth allocation example for ship s (blue square) with $BP_s = 500$ and $BT_s = 5$ and ship s' (green square) with $BP_{s'} = 200$ and $BT_{s'} = 2$. The red dotted square shows the solution area that is not valid for ship s' according to constraint (10).

than or equal to its planned time of arrival ETA_s. Constraint (8) ensures a minimum safety entrance time between berthing times of any two ships, s and s', since most ports will only berth one ship at a time due to physical constraints at the port's entrance. Constraint (9) guarantees that the berthing position BP_s of ship s plus its length L_s will always be less than or equal to the total length W of the wharf. Finally, constraint (10) ensures that no two ships can share (part of) the same berth during the handling times of the two ships. For instance, suppose a ship s is planned to be berthed at time $5h$, has handling time equal to $4h$, utilizes berthing position $500m$, and its length is $300m$ as shown in Figure 5. According to constraint (10), no other ship can use berthing positions from $500m$ to $800m$ (as length of ship s is $300m$) in the time interval $5h$ to $9h$. In addition, a second ship s' with length $200m$ and handling time $3h$ cannot use the berthing positions from $301m$ to $800m$ in the time interval $3h$ to $9h$ as it would overlap with ship s. Visually, this constraint ensures that the two rectangles denoting the time intervals and berthing positions allocated to the two vessels shown in Figure 5 can never overlap.

As discussed earlier in Section 2, the case of the the discrete berthing layout is a basic variation of the continuous layout scenario. If the berthing layout is discrete, then each berthing position b represents a specific berth. The objective function shown in Equation

4 remains the same and constraints (5)–(8) still apply, whereas constraints (9) and (10) are respectively replaced by the following two constraints:

$$L_s \leq L_b, \qquad \forall\, s \in S,\ b = BP_s \tag{11}$$

$$\sum_{s' \neq s \,\in\, S} \; \sum_{t=BT_s-HT_{s'}+1}^{BT_s+HT_s} x_{s'bt} = 0, \quad \forall\, s \in S,\ b = BP_s \tag{12}$$

Specifically, constraint (11) ensures that the length L_s of ship s is less than or equal to the length L_b of berth b that is assigned to ship s. Finally, constraint (12) guarantees that a berth is never assigned to two ships during the same time intervals.

5.3 Proposed and Benchmark Approaches

In this section, we uncover the details of the proposed method (i.e., CSA) and state-of-the-art heuristic-based popular approach, i.e., GA.

5.3.1 Proposed Cuckoo Search-based Method

CSA is a swarm-based metaheuristic optimization algorithm that was developed by Yang et al. [152]. The CSA emulates the breeding behavior of some cuckoo species, which have a fascinating reproduction mechanism. In particular, some cuckoos lay their eggs in nests of other birds (often nests of other species' nests), where they may discard eggs of other birds in order to enhance the hatching ratio of their own eggs. Then, the host birds take care of cuckoo eggs as they presume that the eggs belong to them. Nonetheless, sometimes the host birds distinguish between their own eggs and the alien eggs. Accordingly, either the discovered alien eggs are thrown out of the current nest or new nests are built in new locations. Inspired by this particular mechanism of laying eggs by the cuckoo birds, the following three standard rules are adopted to employ CSA for optimization problems [152]:

1) each cuckoo lays one egg at a time at a randomly chosen nest;

2) the best nests with high-quality eggs will not be removed and will be carried over to the next generation;

3) the quantity of host nests is fixed and the egg dumped by a cuckoo is discovered by a host bird with a probability $p_\alpha \in (0,1)$.

In this study, each nest denotes a solution set that includes the berthing times and berthing positions for all arriving vessels. An egg represents either a berthing position or time, while a cuckoo egg represents a new (and better) berthing position or time. The total number of host nests reflects the total search space at each iteration of the algorithm. In this work, 100 host nests are considered and each nest contains $2N$ eggs, where N is the total number of vessels. Hence, the total number of eggs in a nest is double the total number of arriving vessels. Overall, the high-level goal of the algorithm is to use cuckoo eggs (better solutions) to replace not-so-good eggs in the nests.

Algorithm 1 Cuckoo Search Algorithm for BAP

1: $X[1..k]$ = Generate initial population of host nests
2: **for** $t = 1$ to max number of iterations **do**
3:     **for** $i = 1$ to k **do**
4:         $x_{new} = X[i] + \alpha \oplus Levy(\lambda)$
5:         **if** (fitness(x_{new}) $<$ fitness($X[i]$)) **then**
6:             $X[i] = x_{new}$
7:         **end if**
8:     **end for**
9:     **for** $i = 1$ to k **do**
10:         **if** $(rand(0,1) < p_a)$ **then**
11:             $X[i]$ = Generate new host nest
12:         **end if**
13:     **end for**
14:     x_{best} = Find nest with lowest fitness value in X
15: **end for**

Algorithm 1 presents the pseudocode of the Cuckoo Search Algorithm, which begins with a randomly distributed initial population of k=100 host nests over the search space (line #1). In each iteration of the algorithm, the reproduction step is performed first, where new solutions are generated by replacing some existing eggs with cuckoo eggs in randomly selected nests (lines #3-8). The rationale for the egg replacements is that if a cuckoo egg is very similar to a host egg, then this egg has lesser chances to be discovered. Thus, a random walk is performed through Lévy flights in order to generate new nests (i.e., new solutions).

$$X_i^{(t+1)} = X_i^{(t)} + \alpha \oplus Levy(\lambda), \tag{13}$$

where t denotes the current iteration number, X_i the solution for nest i, and α ($\alpha > 0$) the step size. The $\oplus$ operation denotes entry-wise multiplication. A random walk in Lévy flights is performed from a Lévy distribution with a scale parameter λ [127]. The primary aim of performing random steps is to increase the possibility of finding the global solution instead of becoming stuck in a local optimum. A new solution replaces a current solution if its fitness score is lower than the fitness score of the current solution (lines #5-7). The fitness of each possible solution is evaluated using the objective function of the BAP presented in Equation (4) and accounts for the total processing cost, which includes waiting, handling, and late departure penalty costs. In addition, it is possible for some cuckoo eggs to be discovered by host birds with a discovering probability p_α. In our work, p_α is set to 0.45 as reported in [152]. In this case, the nests with the discovered cuckoo eggs are abandoned and new ones are built; as a result, the exploration of the search space is enhanced (lines #9-13). Finally, the best solution across all iterations is kept (line #14). The above steps repeat until either the total number of iterations is reached (which equals 100 in this work) or there has been no fitness improvement for some iterations.

5.3.2 Genetic Algorithm

GA is one of the most popular algorithms from the metaheuristic family and is based on the evolution process in natural systems, i.e., Darwin's principles of survival of the fittest individuals [63]. Since GA has a high convergence rate compared to most metaheuristics, it can solve big and high complexity problems relatively quickly. GA is a population-based approach that finds a better solution by managing a population that contains various possible solutions, which are revised generation to generation by employing several genetic operators, including selection, crossover, and mutation. The complete working procedure of GA consists of the following six steps [119].

Initial population generation: A random population of different possible solutions (chromosomes) is generated at the first step. A single solution is known as a gene, a solution set is known as a chromosome, and all solution sets form a population.

Fitness evaluation: The fitness values of all solutions are evaluated to demonstrate the goodness of solutions.

Selection: The selection phase aims to select a couple of fittest individuals (parents), which are used for the next generation. There are more chances for the selection of individuals for reproduction having the best fitness value. A few of the fittest individuals are selected as parents for the next generation.

Crossover: Crossover is employed to produce offsprings, where, a child adopts one portion of its characteristics from one parent and the other part from the second parent.

Mutation: A mutation operator is also applied to some portion of the solutions in the new generation to avoid premature convergence and maintain diversity in the population. In addition, it also ensures that the probability of any solution is never zero.

Termination of algorithm: The fitness values of the new population are calculated and the same process repeats until conditions of termination are met. Termination conditions include maximum available computation time, maximum iterations, and a maximum number of generations.

Table 5: One-hour interval dataset employed for experiments [125]

Ship #	ETA	HT	ETD	PBP	LoS
1	4	3	8	778	128
2	5	5	11	1416	113
3	10	5	15	957	334
4	4	2	8	1437	423
5	13	3	16	362	173
6	15	1	18	1015	391
7	11	3	15	434	338
8	6	2	9	1008	140
9	9	1	11	1043	302
10	2	2	5	102	194

5.4 Experimental Results

In this section, we present the settings, datasets, and results of our extensive berth allocation simulations. In addition to the CSA method, we implemented a popular population-based heuristic method (i.e., GA) proposed in the recent literature [126], as well as the exact MILP approach. The implemented algorithms are coded in MATLAB 2019b on a Windows 10 PC with COREi7 processor and 8GB RAM. The number of iteration for both metaheuristics (CSA and GA) is set to 100.

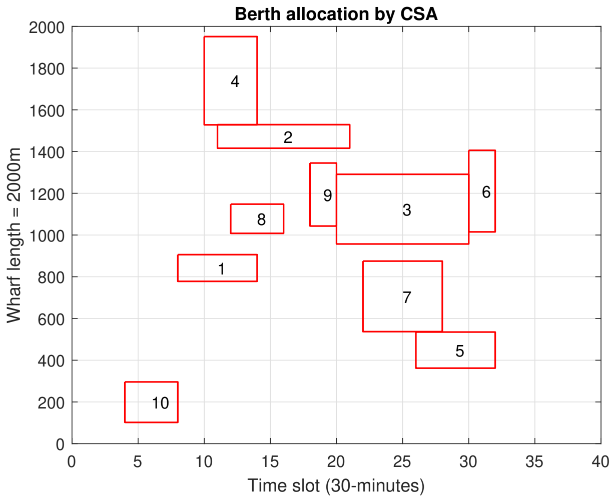
For our experiments, two data sets were used. The first problem dataset is taken from

Şahin et al. [125] and shown in Table 5. The dataset contains a number of arriving ships in a day along with the estimated arrival time, handling time, estimated departure time, preferred berth position, and length for each arriving ship. This dataset was adapted to generate more ships within the same planning horizon of 1 day. For instance, the minimum and maximum handling times for any ship are 2 hours and 5 hours, respectively (see column #3 in Table 5); so, we generate randomly handling times for all ships in between 2 and 5 hours; the same policy is used for the other parameters. In order to stress-test the algorithms and also to test other planning horizons, a second larger dataset was generated randomly by employing uniform distribution with realistic settings for all parameters, and a planning horizon of up to 1 week.

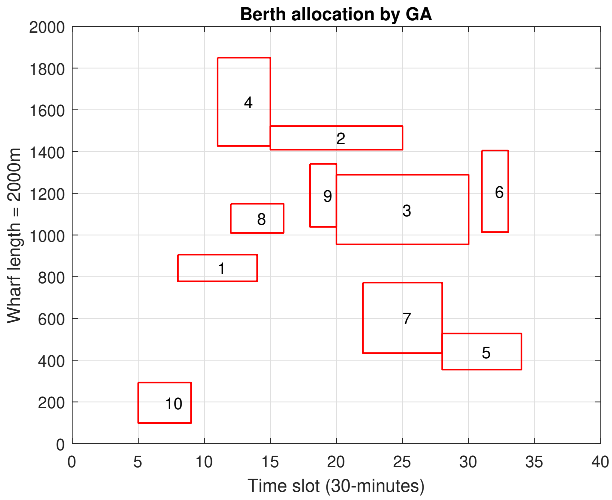
The quay is continuous and has a length of $2000m$ in both datasets. For the cost calculations of the different algorithms, we have used the values 10, 5, and 5 Euros per hour handling cost, waiting cost, and late departure cost, respectively [125]. When a vessel is moored at a berth position other than its optimal berthing position, a penalty based on the absolute difference between the assigned berth position and the optimal berthing position is added to the handling cost. The non-optimal berthing cost (NBC) is calculated as follows: $NBC = HC_s/W = 0.005$ Euros per meter.

Furthermore, unlike our previous [15] and many other studies (e.g., [125, 126]), we use a time interval of less than an hour for the experiments, i.e., 30 minutes, to make the problem more practical and to reduce the time loss in the simulations. For example, if a ship takes 5 hours and 30 minutes to load and unload, a time interval of 1 hour wastes 30 minutes since the algorithms work with a time interval of 1 hour and cannot schedule a ship before the next hour, i.e., 6. A more fine-grained time interval of 30 minutes avoids this situation. The same problem instances were used in the case of the 30-minute time intervals; however, all times in Table 5 are multiplied by two to generate 30-minute values since the original data contains values based on 1-hour time interval.

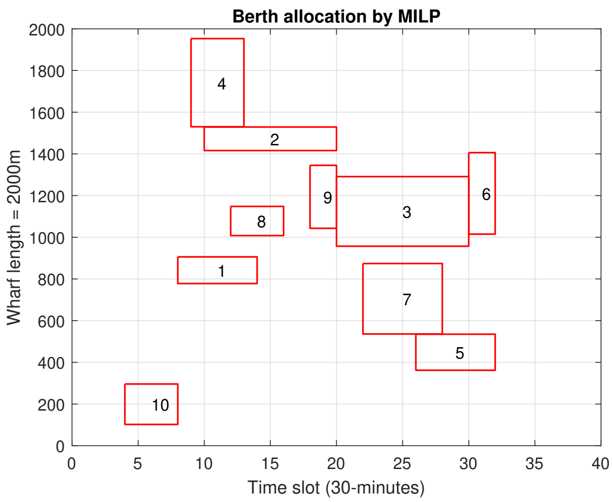
Next, we present simulation results when our proposed and benchmark methods are implemented considering a 30-minutes time interval. When we implement the 30-minute time interval, this study considers a new constraint, shown in Equation (8), which guarantees a minimum and realistic safety entrance time (SET) of one 30-minute time interval between the berthing times of any two vessels. This constraint is yet another benefit de-



(a)



(b)



(c)

Figure 6: Berth allocation solution, when we consider 30-minutes time interval, generated by (a) CSA, (b) GA, and (c) MILP.

rived from using a smaller time interval as using a 1-hour SET would be too long and wasteful. The results presented in Figure 6 show the three solutions for berth assignment when we consider a 30-minute time interval and the ten arriving ships shown in Table 5. All three approaches allocate ships to the available berth positions and time slots based on the primary objective of this study, which is to minimize the total processing cost, as shown in Equation (4). In Figure 6, the vertical axis shows the berth positions, while the horizontal axis shows time divided into 30-minute time intervals. Each rectangle in this figure denotes the berthing periods and berthing positions assigned to an arriving vessel. The label within a rectangle indicates the ship index. Unlike the previous study, a safety entrance time can be noticed as no two ships berth at the same time. By comparing the sub-figures, it becomes evident that the three different algorithms offer three different solutions to the problem, with some vessels berthed at different positions and/or berthing times. For example, the estimated arrival time of ship 5 is the 26th time slot (30-minutes each time slot). The berthing time proposed by MILP and CSA is the 26th slot, while GA chose the 28th slot because of the berth placement of ship 7. Hence, ship 5 does not have to wait before berthing when using the CSA and MILP approaches. On the contrary, ship 5 has to wait for 1 hour (2 slots of 30-minutes) if GA is employed.

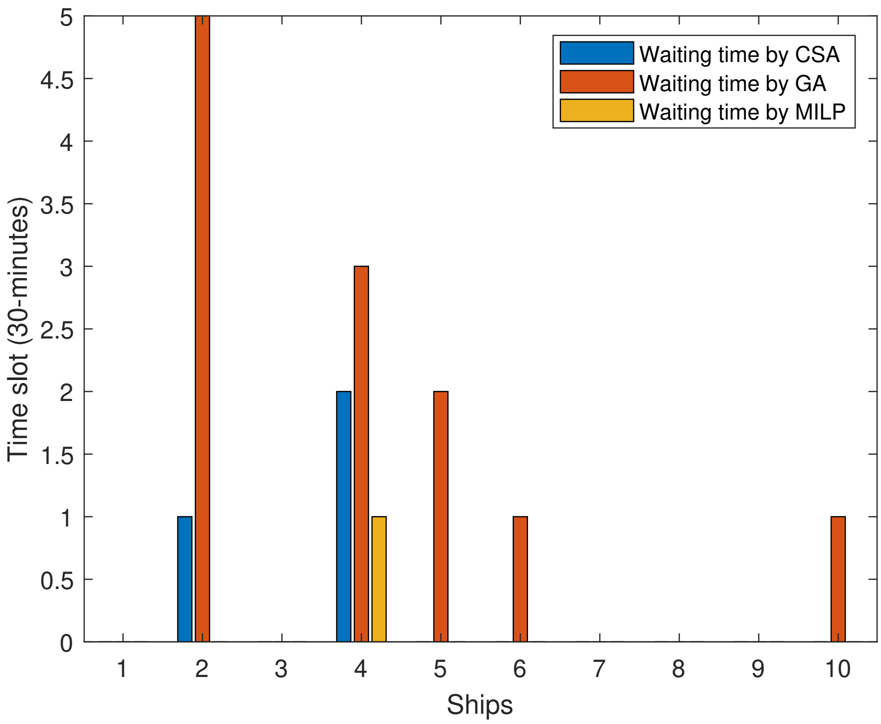


Figure 7: Ships' waiting times when using CSA, GA, and MILP approaches

Figure 7 shows the waiting times incurred by the ten ships when using the three approaches. From Figure 7 it can be seen that only one vessel has to wait (for 30-minutes only) when MILP is used. However, our newly proposed CSA method provides a solution with a maximum waiting time of two time periods (30-minutes each) for any ship and only two ships have to wait before berthing. In contrast, when using GA, five ships have to wait for optimal berthing and the maximum waiting time for ship 2 is five time slots (i.e., 2.5 hours). Such waiting times are very high and in turn can cause late departures. From these results, we can conclude that the CSA method outperforms GA in terms of reducing the waiting times.

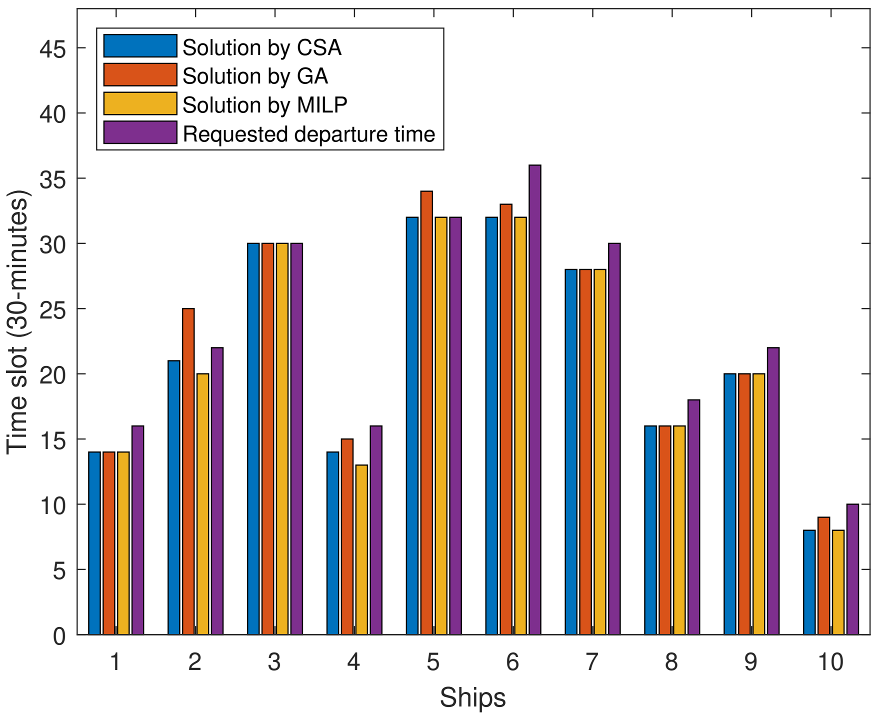


Figure 8: Requested and planned departure times for each ship using CSA, GA, and MILP

Figure 8 shows the requested departure times of all arriving ships and the proposed departure times by our developed and compared algorithms, i.e., CSA, GA, and MILP. From this figure, it can be seen that no vessel departs late using either MILP or our proposed CSA approach. On the other hand, with GA, two ships depart late: ships 2 and 5 are three slots and two slots late, respectively. Once again, we conclude that CSA shows higher performance in minimizing late departures compared to GA.

The results presented in Figure 9 show the absolute difference between the optimal berthing

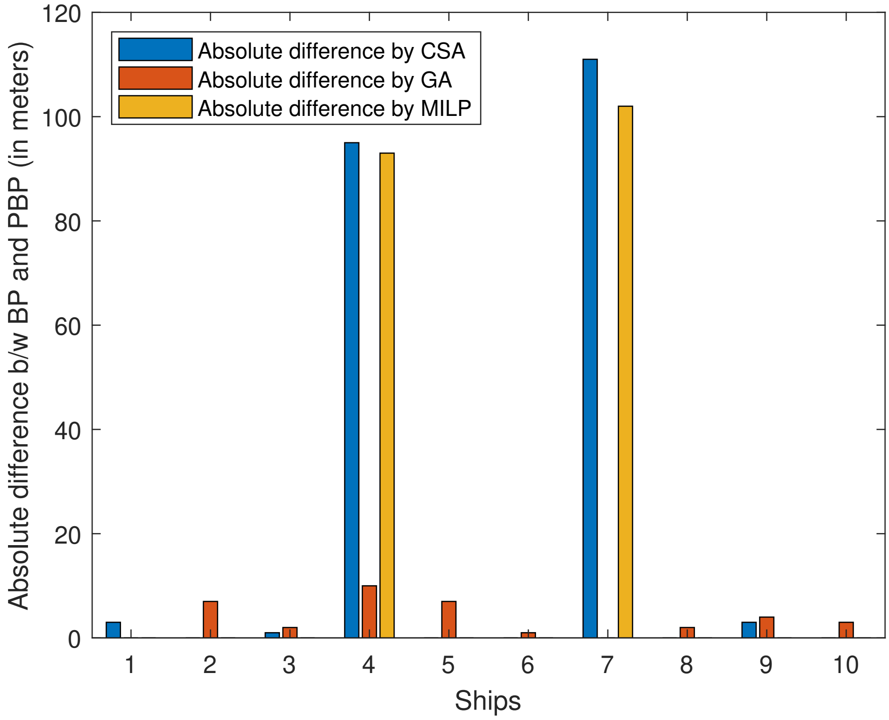


Figure 9: Absolute difference between the berthing position and the preferred berthing position of the ten ships provided by CSA, GA, and MILP

positions and berthing positions of the 10 ships provided by the implemented algorithms, i.e., CSA, GA, and MILP. We can observe from this figure that MILP always provides optimal berthing positions except for only two ships, namely, ships 4 and 7. CSA's behavior in terms of berth placement is similar to MILP as it also uses non-optimal berthing positions for ships 4 and 7, plus some very small deviations for three more ships. On the contrary, 8 out of the 10 ships are moored to other than optimal berthing positions when using GA, albeit with small differences. Even though ships 4 and 7 berth at (near) optimal positions with GA, they cause major waiting times and departure delays for ships 2 and 5, respectively, leading to higher-cost solutions. Overall, by considering all results presented so far, it is evident that it is not possible to achieve both zero waiting times and optimal berthing positions for all ships from Table 5. The best solutions comes from delaying ship 4 by 30 minutes (due to the safety entrance time constraint) and using non-optimal berthing positions from ships 4 and 7 in order to avoid delays for other ships. In this regard, the solution provided by CSA is much closer to the optimal MILP solution compared to the GA solution.

For comparison purposes, the total processing cost along with the computation times for

Table 6: Comparative analysis in terms of cost and computation time when using data instances from [125] with 30-minute time intervals.

Method		CSA		GA		MILP	
No.	No. ships	Cost (€)	Time (s)	Cost (€)	Time (s)	Cost (€)	Time (s)
1	10	302	0.94	347	0.21	288	207.26
2	15	434	1.26	470	0.30	395	318.84
3	20	595	1.43	620	0.21	535	404.59
4	25	680	2.01	688	0.57	615	530.96
5	30	827	3.08	832	0.98	725	644.01

all implemented algorithms, i.e., CSA, GA, and MILP, are shown in Table 6. In addition, we have also varied the number of arriving ships to study the scalability of the proposed method. We tested the three approaches on multiple instances, where 10-30 ships and a 30-minutes time interval are considered, while all other parameters are the same, i.e., the length of the quay, the arrival pattern of the ships, and the berth layout. It can be seen from Table 6 that MILP provides the optimal solution in terms of minimum processing cost in all cases; however, our proposed CSA-based algorithm also provides a near-optimal solution in all cases when we compare it with the alternative GA approach. For example, the results of the first instance show that MILP achieves the lowest cost of 285 Euros, as expected, CSA also has a lower cost as compared to GA, which is 322 Euros; however, total processing cost with the GA is significantly higher (347 Euros). A similar pattern is observed for the other instances; the solutions proposed by CSA are only slightly costlier (ranging from 4.9% to 14.1%) than the optimal MILP solution, while the solutions of GA are more expensive (ranging from 11.9% to 20.5%). Furthermore, Table 6 also presents the computational time of all three algorithms when using 30-minute time interval. It can be noticed from this table that GA has the minimum computational time and MILP has higher computational time. Our proposed CSA-based method takes only slightly higher time compared to GA (up to 3 seconds vs. 1 second) but provides better cost reduction. Compared to CSA, the MILP is orders of magniture slower and takes on average 250× more computational time to solve the berth allocation problem.

In order to stress-test the algorithms, we also generated a larger synthetic dataset with 20 uniformly random instances (10-100 vessels), 10 with 1-hour interval and 10 with a 30-minute interval. In addition, we increased the planning period from one day to one week

Table 7: Comparative analysis in terms of cost and computational time when using randomly generated data instances with 30-minute time intervals.

Method		**CSA**		**GA**		**MILP**	
No.	No. ships	Cost (€)	Time (s)	Cost (€)	Time (s)	Cost (€)	Time (s)
1	10	355	1.16	380	0.40	305	195.25
2	20	702	1.71	722	0.42	620	333.29
3	30	1005	2.83	1020	0.56	895	557.70
4	40	1322	4.99	1785	0.89	1080	719.98
5	50	1730	6.92	1792	1.56	1460	957.05
6	60	2102	5.07	2137	1.22	–	–
7	70	2527	6.49	2597	1.91	–	–
8	80	2990	1.29	3060	2.42	–	–
9	90	3092	17.89	3215	4.94	–	–
10	100	3635	12.82	3692	4.07	–	–

when the number of ships is greater than 40. Table 7 shows a comparative analysis when using the data instances that were randomly generated with 30-minute time intervals. The overall trends in terms of cost are the same as before: MILP offers the lowest cost, followed closely by CSA (with 12.3%–22.4% higher cost), while GA lead to the highest cost with up to 65.3% worse cost than the minimum. In terms of computation times, both GA and CSA scale well with very low running times, less than 5 and 18 seconds, respectively. MILP, on the other hand, leads to much longer running times (3-16 minutes) that are about *two orders of magnitude* higher compared to CSA and grow super-linearly. To make matters worse, when the number of ships in the problem set becomes greater than 50, the MILP runs out of memory and thus it is unable to solve large problem instances. Overall, CSA is a very efficient approach that leads to near-optimal solutions in terms of total processing costs.

To further illustrate the better performance of CSA and to highlight the statistical significance of the compared approaches, we also perform a statistical test, namely ANOVA, on twenty randomly generated data instances. Table 8 presents the results of the ANOVA test, where SS is the sum of squares within groups, df is the degrees of freedom, MS is the mean square, F is the test statistic (that is the ratio of between and within group variances), and F crit is the critical F value. The P-value of the ANOVA test is 0.00030774, which is less than 0.5 (standard value for significance) and shows that the differences among the results from the different methods are statistically significant.

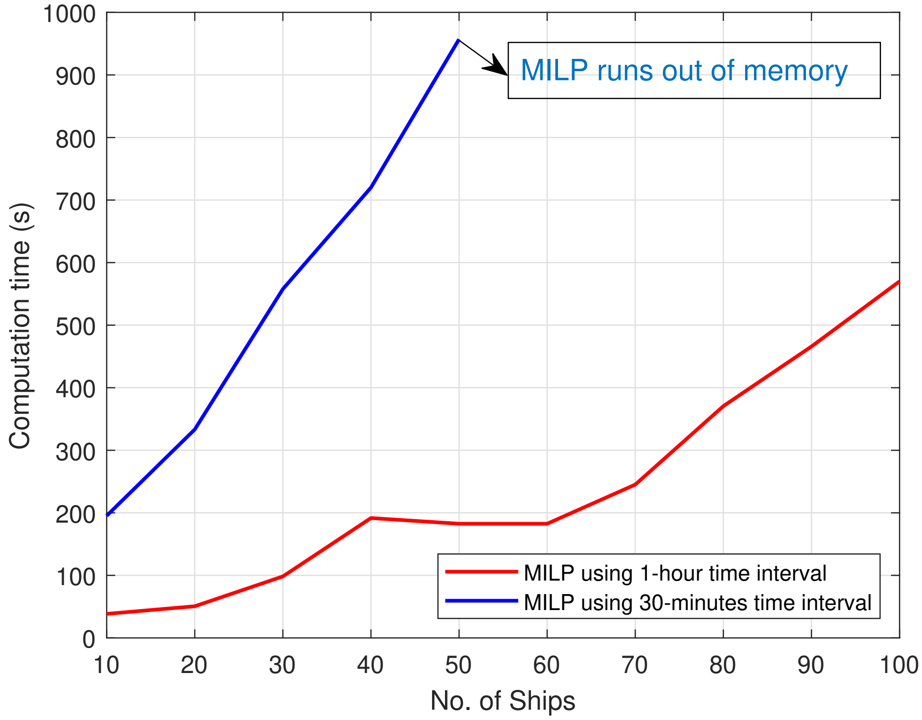


Figure 10: Computation time of MILP using random data (10-100 ships) when we consider both time intervals, i.e., 1-hour and 30-minutes

Table 8: ANOVA test results considering twenty random data instances considering 20 arriving ships. [SS=sum of squares within groups; df=degrees of freedom; MS=mean square; F=test statistic; and F crit=the critical F value]

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	286272.4	2	143136.2	9.35004035	0.00030774	3.15884272
Within Groups	872591.3	57	15308.6			
Total	1158863.7	59				

Finally, Figure 10 shows the computation time of MILP when using a 1-hour or a 30-minute time interval as we increase the number of ships in the problem dataset. In both scenarios, the computation time grows super-linearly, with the 30-minute time interval case growing much more aggressively, even though the number of intervals is only double compared to the 1-hour interval case. As previously mentioned, MILP runs out of memory and cannot solve problem instances with more than 50 cases when using 30-minute intervals. Therefore, MILP cannot be used for larger, more realistic problem sizes; an observation that has been reported previously in other studies, such as requiring over 100 hours of CPU time for real-world instances [126]. Such times are certainly not acceptable in the context of MCT operations.

5.5 Summary

This section focuses on the berth allocation problem with dynamic ship arrivals, where a metaheuristic-based CSA is proposed to solve the BAP. In addition, we implemented two benchmark methods for comparison, a well-known metaheuristic GA and an MILP. Unlike existing studies, and to make the problem more practical, a fine-grained time interval of 30 minutes is used in this study along with other practical constraints such as a safety time distance between berths. The proposed and compared approaches are implemented on multiple data instances generated from a benchmark dataset. In addition, randomly (uniformly) generated data instances with up to 100 vessels and a planning horizon up to a week are used for experiments to test the flexibility and scalability of the proposed method. The results show that our proposed algorithm (CSA) has higher efficiency in terms of minimum processing cost for all incoming ships compared to GA. Compared to MILP, our proposed CSA algorithm provides a near-optimal solution at a fraction of the computation time. Moreover, when we implement all algorithms on large datasets, the MILP algorithm runs out of memory and cannot provide an optimal solution in reasonable time. Overall, CSA beats GA in terms of processing cost and outperforms MILP in terms of computation time, and thus provides near-optimal solutions in affordable computation times.

In the next section of this thesis, we present the solution of BAP, considering multiple quays in a single terminal, rather than just one quay (as in this chapter). Implementing CI methods in different environments and with different sized data instances will help to verify their scalability and productiveness.

6 Optimizing Berth Allocation Considering Multiple Quays Using Computational Intelligence Approaches

6.1 Motivation and Problem Statement

Numerous studies deal only with the allocation of berths at a single quay, assuming that it forms one straight line in which vessels can be berthed according to their length and the positions of other vessels. For example, an exact approach to solve single quay BAP is presented in [147], an evolutionary algorithm in [44], and a metaheuristic-based method in [21]. However, this assumption is not realistic for several ports around the globe, which consist of multiple separate line segments or quays for berthing [51]. For example, the Port of Limassol in Cyprus, has seven continuous berthing quays, as depicted in Figure 12. Considering multiple quays adds a new dimension to the BAP; the problem of assigning vessels to quays in addition to assigning berthing positions and times for each separate quay. This requires a multiple space–time representation, as can be seen in Figure 11. On the contrary, very few studies propose solutions for the Multi-Quay BAP (MQ-BAP) [51, 88, 61]. These studies often make some unrealistic assumptions, do not consider practical constraints, are limited to solving small scale problems, and cannot provide optimal solutions for medium/large scale problems. These limitations motivate us to consider a formulation with several practical constraints and to propose a solution that can solve real port problems, which have been ignored in the current literature. Therefore, this section extends single quay BAP to the case of MQ-BAP, proposing a solution for ports having multiple quays. Real data is used from the Port of Limassol, the largest port in Cyprus, to validate our method. The port has seven continuous quays, but only five of them are used for commercial purposes.

6.2 Problem Explanation and Formulation

In contrast to existing studies, and to make the problem more practical, this work considers MCT with multiple quays (having both continuous and discrete berthing layouts) to berth arriving vessels. A continuous quay consists of a section of the berth line and arriving ships can be moored at any berthing position along the berth line. The length of the continuous berth line is known in advance. A discrete quay considers a section of the berth divided into berth segments. In the discrete berthing layout, only one ship can be

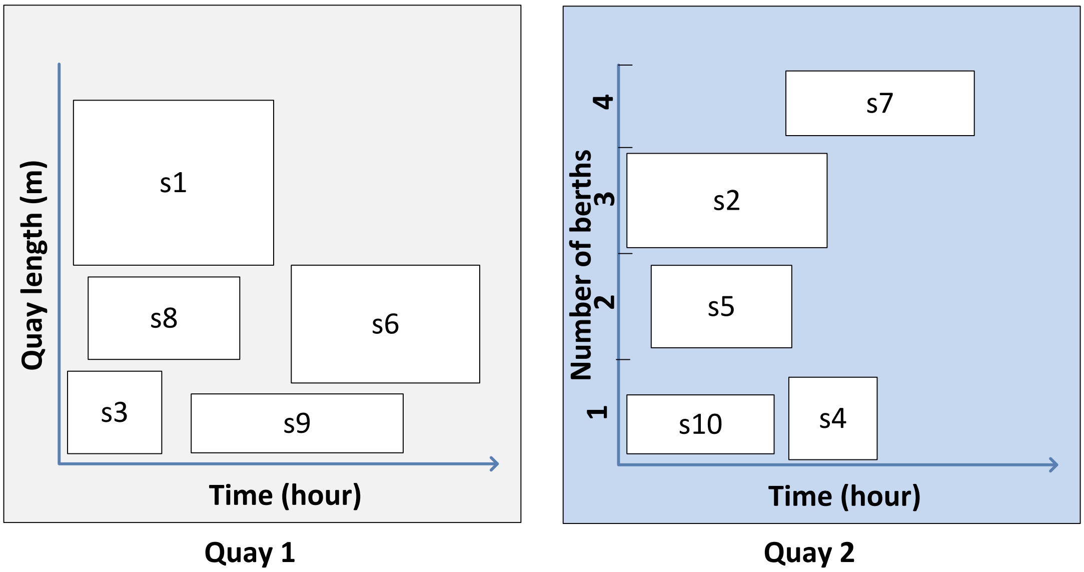


Figure 11: BAP solution with two berthing quays (Quay 1: continuous and Quay 2: discrete) and 10 arriving ships. Each rectangle denotes a ship, whose height (y-dimension) shows the ship's length and whose width (x-dimension) is the handling time of ship.

moored at a single berth segment in a single time interval. Note that time is discretized in a set of time intervals (e.g., 30- or 60-minute intervals) for planning. Finally, there is a set of arriving ships, with each ship having multiple known characteristics, such as length of ship (LoS), expected time of arrival (ETA), expected time of departure (ETD), handling time (HT), preferred berthing position (PBP), and preferred berthing quay (PBQ).

The objective of this study is to determine the berthing time, berthing quay, and berthing position or segment (depending whether it is a continuous or discrete quay, respectively), for arriving ships in order to reduce the total cost associated with the berthing process. The cost against a ship includes handling cost, waiting cost, and penalty costs due to late departures, allocation of a non-optimal berthing (NOB) position, and non-optimal berthing quay. The handling cost includes the cost of loading and unloading containers and depends on the handling time of the ship. The waiting cost is calculated based on the waiting time, which is the difference between ETA and the berthing time, while the late departure penalty cost depends on the late departure time that is defined as the difference between the task finishing time and the ETD of each ship. The penalty cost due to non-optimal berth allocation is incurred when the ship is moored at a location other than its PBP, since more resources are needed to move containers over a longer distance. Another penalty cost is added if the ship is moored to a quay other than the preferred one.

Figure 12: A satellite view of the Port of Limassol, Cyprus illustrating its seven berthing quays (taken from [7]). Note: $*$ with quays indicates that these are used for commercial purposes only.

6.2.1 Assumptions of this study

The problem under consideration and the solution are based on the following assumptions.

- The number of incoming ships in the planning period is known;
- When a vessel starts operations at any quay, it cannot be interrupted until loading/unloading is completed;
- Berths from any quay become available immediately after a ship completes its tasks;
- The length of a continuous quay and the number of berths available at a discrete quay are known;
- The ETA and ETD for each vessel are known;
- The estimated turnaround time for each vessel is known;
- Each vessel has a PBQ, a PBP, and ABQs that are known in advance;
- All berths are assumed to be free at the beginning of the time horizon ($t = 0$);
- The processing speed is the same for all QCs and it is known;

- Handling and waiting costs per hour for all vessels are known;
- Penalty costs for late departure, non-optimal berth allocation, and non-optimal quay allocation are known and assumed to be the same for all arriving vessels.

6.2.2 Mathematical formulation

The total processing cost of a ship s that is scheduled for berthing at position B_s of particular quay Q_s at time T_s^b includes a waiting cost, a handling cost, and a penalty for late departure, expressed by the following function:

$$\begin{aligned} Cost(s, Q_s, B_s, T_s^b) = & \; T_s^w \cdot C_s^w \\ & + T_s^h \cdot [C_s^h + f(s, Q_s, B_s)] \\ & + T_s^d \cdot C_s^d \end{aligned} \tag{14}$$

The first term in Equation (14), $T_s^w \cdot C_s^w$, shows the waiting cost when a ship s has to wait for mooring. The waiting time T_s^w of ship s is calculated as the difference between the ETA T_s^{ea} and berthing time T_s^b of ship s, illustrated in Figure 13:

$$T_s^w = T_s^b - T_s^{ea}, \quad \forall\, s \in S \tag{15}$$

The second term in Equation (14), $T_s^h \cdot [C_s^h + f(s, Q_s, B_s)]$ presents the total processing cost of ship s that was incurred by unloading and loading containers from/to ship s. Similar to previous studies (e.g., [16]), this work considers the handling time of each ship to be an input to the problem. However, calculating handling time of each ship s is fairly straightforward and reported in other studies such as [136]. In particular, the handling time T_s^h depends on the total volume (TEUs) $Load_s$ to be loaded/unloaded on the ship and the number of assigned cranes N_s^{qc} along with the average handling productivity of cranes HP_s^{qc} assigned to that particular ship.

$$T_s^h = \frac{Load_s}{N_s^{qc} \cdot HP_s^{qc}}, \quad \forall\, s \in S \tag{16}$$

Without loss of generality, this work also introduces the penalty function $f(s, Q_s, B_s)$,

Table 9: Nomenclature and notations used in Section 6.2

Name	Explanation	Name	Explanation
Acronyms			
ABQ	Alternative berthing quay	BAP	Berth allocation problem
CSA	Cuckoo search algorithm	ETA	Expected time of arrival
ETD	Expected time of departure	GA	Genetic algorithm
HT	Handling time	LoS	Length of ship
MCT	Maritime container terminal	MQ-BAP	Multi-quay BAP
NOB	Non-optimal berthing	PBP	Preferred berthing position
PBQ	Preferred berthing quay	PSO	Particle swarm optimization
QCs	Quay cranes		
Notations			
B_s	Berthing position of ship s	C_s^d	Penalty cost for late departure (per hour) of ship s
C_s^h	Handling cost per time unit (hour) of ship s	C_s^{nob}	Penalty cost for NOB position per m of ship s
C_s^{noq}	Penalty cost for NOB quay of ship s	C_s^w	Waiting cost per time unit (hour) of ship s
HP_s^{qc}	Handling productivity of QCs assigned to ship s	$Load_s$	Total load (in TEUs) on ships s
L_b	Length of a berth segment b (in a discrete quay)	L_q	Length of a (continuous) quay q
L_s	Length of ship s	N_s^{qc}	Number of quay cranes assigned to ship s
Q_s	Berthing quay of ship s	SD	Safety distance (in meters) between two ships
SE	Safety entrance time between two ships	ST	Safety time between two ships during berthing
T_s^{ad}	Actual departure time of ship s	T_s^b	Berthing time of ship s
T_s^d	Late departure time of ship s	T_s^{ea}	Expected arrival time of ship s
T_s^{ed}	Expected departure time of ship s	T_s^h	Handling time of ship s
T_s^w	Waiting time of ship s		
Indices			
$s \in S$	A ship s from a set of arriving ships S	$q \in Q$	A quay q from a set of continuous and discrete quays Q
$b \in B_q$	A berth position or segment b from a set of available berth positions/segments B_q in a continuous or discrete quay q, respectively	$t \in T$	A time interval t from a set of time intervals T

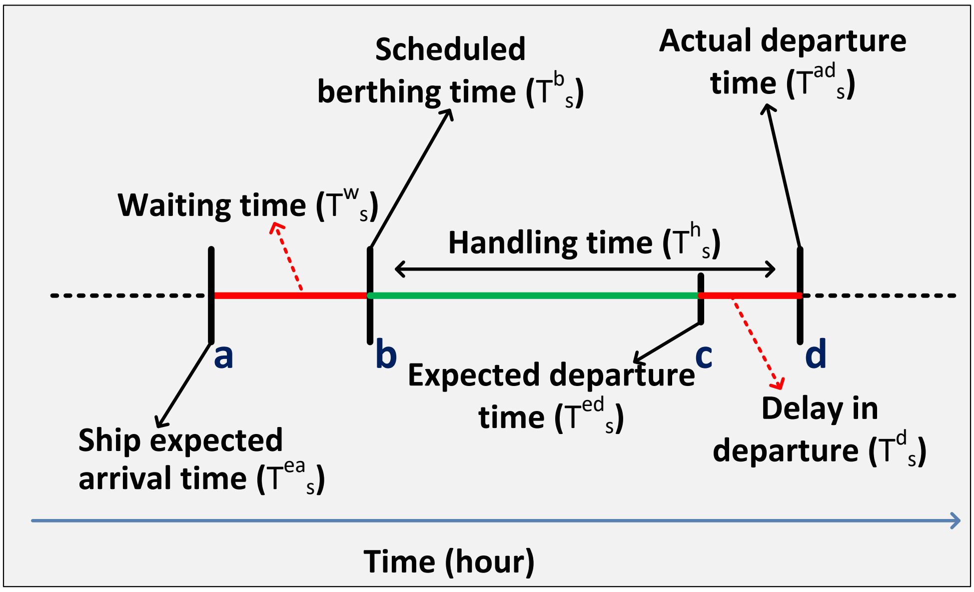


Figure 13: An illustration of the berthing timeline (waiting and late departure times)

which will penalize the handling cost based on non-optimal berth allocation of ship s. Unlike our previous study and to make the model more realistic, this work calculates penalty based on the absolute difference between the assigned berthing position B_s and the preferred berthing position PBP_s (if assigned to its PBQ), and takes into account ABQ_s as:

$$f(s, Q_s, B_s) = \begin{cases} |PBP_s - B_s| \cdot C_s^{nob} & \text{, if } Q_s = PBQ_s \\ C_s^{noq} & \text{, if } Q_s \in ABQ_s \\ \infty & \text{, otherwise} \end{cases} \tag{17}$$

The final term $T_s^d \cdot C_s^d$ in Equation (14) calculates the late departure penalty cost against ship s when it departs after the ETD. The delayed departure time T_s^d of ship s (if any) is computed as the difference between the actual departure time T_s^{ad} and the expected time of departure T_s^{ed}, as depicted in Figure 13.

$$T_s^d = max\{T_s^{ad} - T_s^{ed}, 0\}, \quad \forall\, s \in S \tag{18}$$

where, T_s^{ad} can be calculated as,

$$T_s^{ad} = T_s^b + T_s^h, \quad \forall\, s \in S \tag{19}$$

The primary objective of the multi-quay berth allocation problem is to allocate optimal quays and berthing positions along with berthing times to arriving ships such that the total processing cost (that includes waiting cost, handling cost, and various penalties) can be minimized, as presented by the following objective function:

$$\textbf{minimize} \sum_{s\,\in\,S}\ \sum_{q\,\in\,Q}\ \sum_{b\,\in\,B_q}\ \sum_{t\,\in\,T} Cost(s,q,b,t)\cdot x_{sqbt} \tag{20}$$

subject to several constraints that are presented below.

General constraints:

$$x_{sqbt} \in \{0,\ 1\}, \forall\, s \in S,\ q \in Q,\ b \in B_q,\ t \in T \tag{21}$$

The variable x_{sqbt} is 1 if the ship s is scheduled at position b of quay q at time t, and 0 otherwise.

$$\sum_{q\,\in\,Q}\ \sum_{b\,\in\,B_q}\ \sum_{t\,\in\,T} x_{sqbt} = 1,\ \forall\, s \in S \tag{22}$$

This constraint ensures that each ship may moor only once during the time t at the mooring position b of the quay q.

$$T_s^b \geq T_s^{ea}, \quad \forall\, s \in S. \tag{23}$$

The constraint specifies that the proposed berthing time T_s^b for a given ship s must always be equal to or later than its expected time of arrival T_s^{ea}.

$$T_s^b - T_j^b \geq SE \quad \forall\, s \neq j \in S \tag{24}$$

This condition guarantees a minimum safety entrance time SE between any two consecutive berthing operations.

Constraints for continuous berthing layout:

$$\sum_{j \neq s \in S} \sum_{b=B_s-L_j-SD+1}^{B_s+L_s+SD} \sum_{t=T_s^b-T_j^h-ST+1}^{T_s^b+T_s^h+ST} x_{jqbt} = 0, \quad \forall s, j \in S, q = Q_s \tag{25}$$

This is an overlap avoidance constraint that does not allow two vessels to share (part of) the same berth positions during their handling times. Suppose a vessel s is scheduled to berth at time $5h$, has a processing time of $3h$, uses a berthing position $500m$, and is $300m$ long. Thus, this constraint ensures that no other ship can use the berths from $500m$ to $800m$ (since the length of ship s is $300m$) in the time interval $5h$ to $8h$. Visually, this constraint ensures that two rectangles (denoting the time intervals and the berths assigned to the ships) shown in Figure 14 can never overlap. In addition, this constraint is also responsible for maintaining the safety distance SD and safety time ST between two ships to avoid any danger during berthing.

$$B_s + L_s \leq L_q, \quad \forall s \in S, \tag{26}$$

This constraint ensures that the length L_s of any ship s plus its berthing position B_s must be less than or equal to the length L_q of the quay q, where s is planned to be berthed.

Constraints for discrete berthing layout:

$$\sum_{j \neq s \in S} \sum_{t=T_s^b-T_j^h-ST+1}^{T_s^b+T_s^h+ST} x_{jqbt} = 0, \quad \forall s, j \in S, q = Q_s, b = B_s \tag{27}$$

This is a restriction to avoid overlap in the case of a discrete berthing layout that does not allow two vessels to use the same berth at the same time. Suppose a ship s is scheduled to berth at time $10h$, has a processing time of $5h$, and uses the berth numbered 4. Thus, no other ship j can use berthing 4 in time slots between $10h$ to $15h$ (since ship s needs 5 hours to complete its tasks). Furthermore, this constraint is responsible to ensure safety time ST between any two ships s and j.

$$L_b \geq L_s, \quad \forall s \in S, \; b = B_s \tag{28}$$

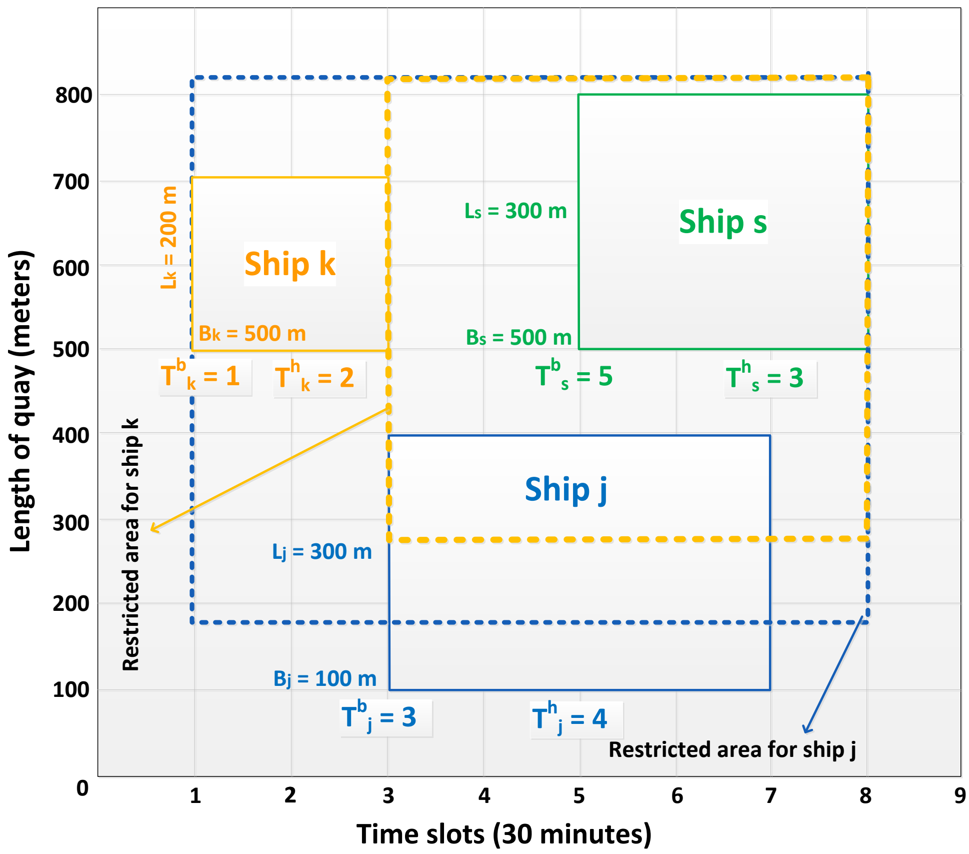


Figure 14: An Illustration of overlapping constraint with three arriving ships (ship s, ship j, and ship k) with different berthing times, berthing positions, and lengths. This figure shows the restricted areas for ships j and k (using dotted boxed) to avoid overlap with ship s, the already scheduled.

This constraint ensures that a berth b assigned to any vessel s must be at least as long as the vessel itself.

Table 10: Example data for 28 ships that arrived during the first week of March 2018 at the Port of Limassol, Cyprus

Ship	ETA (d \ t)	HT (min.)	ETD (d \ t)	PBQ	ABQ	PBP (m)	LoS
1	1\04:00	919	1\22:30	Ro-Ro Quay	Container Quay	240	194
2	1\05:30	1490	2\06:50	East Quay	–	276	139
3	1\14:00	1285	2\12:50	West Quay	North Quay	84	84
4	1\15:00	5700	5\14:03	East Quay	–	51	89

Continued on next page

Table 10 – Continued from previous page

Ship	ETA (d \ t)	HT (min.)	ETD (d \ t)	PBQ	ABQ	PBP (m)	LoS
5	1\17:00	5970	5\21:00	West Quay	North Quay	314	190
6	2\04:30	470	2\13:50	Ro-Ro Quay	Container Quay	138	159
7	2\05:00	168	2\09:30	Container Quay	Ro-Ro Quay	571	196
8	2\08:00	440	2\15:55	North Quay	West Quay	53	155
9	3\04:00	905	3\20:50	Ro-Ro Quay	Container Quay	31	175
10	3\03:30	1331	4\06:15	Container Quay	Ro-Ro Quay	389	277
11	3\07:30	1870	4\14:55	East Quay	–	358	162
12	3\12:30	640	3\22:40	West Quay	North Quay	34	88
13	3\23:00	295	4\05:00	Ro-Ro Quay	Container Quay	162	133
14	5\05:00	825	5\19:00	West Quay	North Quay	208	90
15	5\05:30	635	5\16:30	North Quay	West Quay	190	121
16	5\08:30	315	5\13:15	East Quay	–	267	178
17	5\17:30	1290	6\20:50	Ro-Ro Quay	Container Quay	96	129
18	5\16:00	455	6\00:25	North Quay	West Quay	112	84
19	5\20:00	614	6\09:35	Container Quay	Ro-Ro Quay	125	294
20	6\03:30	937	6\21:25	Ro-Ro Quay	Container Quay	269	122
21	6\04:30	425	6\12:00	West Quay	North Quay	35	102
22	6\05:30	635	6\16:30	North Quay	West Quay	128	87
23	6\06:30	705	6\18:05	West Quay	North Quay	113	84
24	6\07:30	1750	7\12:50	East Quay	–	207	130
25	6\12:00	1070	7\10:15	Container Quay	Ro-Ro Quay	260	217
26	6\14:00	705	7\02:05	West Quay	North Quay	219	88

Continued on next page

Table 10 – Continued from previous page

Ship	ETA (d \ t)	HT (min.)	ETD (d \ t)	PBQ	ABQ	PBP (m)	LoS
27	7\05:30	455	7\13:05	West Quay	North Quay	364	121
28	7\09:30	335	7\15:25	North Quay	West Quay	7	155

6.3 Developed CI-based Approaches

In this section, we present the implementation of three popular metaheuristic approaches, namely, CSA, GA, and PSO.

6.3.1 Cuckoo Search Algorithm

CSA is a metaheuristic optimization algorithm developed by [152]. The CSA is inspired by the breeding mechanism of some cuckoo species, which are fascinating because of their beautiful sounds and aggressive reproduction mechanism. Some cuckoos lay their eggs in communal nests of other species, where they try to remove the eggs of other birds in order to improve the hatching probability of their own eggs. Then, other birds, probably from other species, known as host birds take care of cuckoo eggs. However, if the host birds realize that some eggs do not belong to them, then the cuckoo eggs are disposed or current nests are destroyed and built elsewhere. In particular, some cuckoo species (e.g., new world brood-parasitic Tapera) specialize in the mimicry of the pattern or color of eggs and they lay their eggs in nests of relevant species in order to reduce the probability of theirs eggs being thrown or destroyed [54]. Overall, the CSA works based on the behavior of cuckoos for laying eggs and adopts three idealized rules [152]:

1) each cuckoo bird dumps only one egg at a time in a random nest;

2) the best nests having high quality eggs are kept and used for the next generation;

3) the number of host nests is fixed and the egg laid by a cuckoo is detected by a host bird with probability $p_\alpha \in (0,1)$.

The mapping of CSA to MQ-BAP is as follows. A single nest shows a set of possible solutions containing the berthing times, quays, and positions of all arriving ships, as shown in Figure 15. An egg in a nest denotes either a berthing time or a berthing quay or a berthing position in that quay for an arriving ship, whereas, a cuckoo egg shows a novel (or better) solution (i.e., a berthing time or quay or position). Hence, each nest includes $3N$ eggs, where N shows the number of ships that have arrived at a given time. Therefore, the total number of eggs in a nest is three times the total number of ships arriving. This is because we need three solutions for each ship (i.e., its berthing quay, position, and time). The total search space of the problem at each iteration is reflected by the total number of host nests, which is fixed (100 host nests are assumed in this study). The overall goal of the algorithm is to use cuckoo eggs (better solutions) to replace the not so good eggs in the various nests.

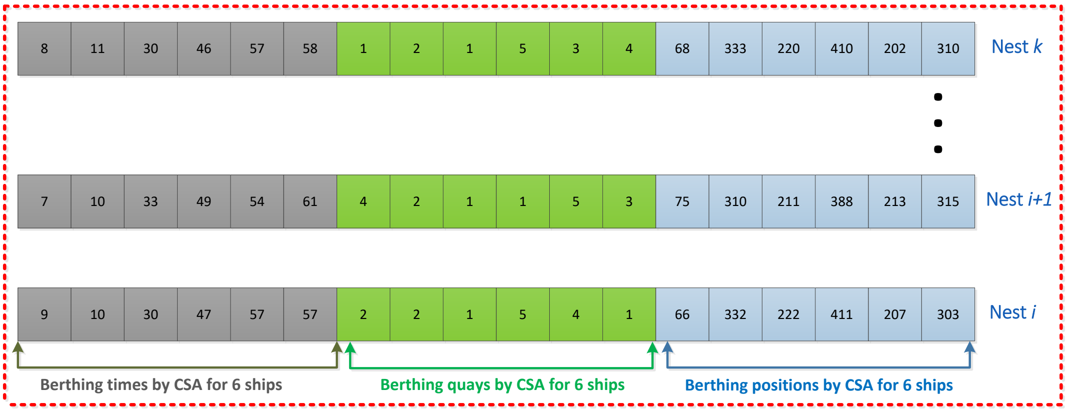


Figure 15: Solution representation by CSA considering six arriving vessels

6.3.2 Genetic Algorithm

The Genetic Algorithm (GA) is a well-known population-based metaheuristic algorithm (also known as a global search algorithm). It is inspired by the theory of biological evolution developed by [63]. GA is famous in the family of metaheuristics due to its high convergence rate, and therefore it can solve various types of optimization problems. Since there is a high probability to survive in fitter organisms, GA follows the concept of survival of fittest [77]. To find an optimal solution, GA generates a random population and updates it using iterative genetic operators, i.e., chromosome representation, selection, crossover, and mutation.

The complete working mechanism of GA is described next. First, a random population of n chromosomes (possible solutions) is initialized, where each chromosome is generated. A single solution is called a gene, a solution set is called a chromosome, and all solution sets form a population. Next, the fitness of all chromosomes (solutions) is calculated using the objective function of this study. The crossover cr is performed on two parents using crossover probability cr_p to produce an offspring o. Then, mutation m is applied with probability of m_p to offspring o to produce a new offspring o'. The new offspring o' is included in the entire population to avoid the algorithm getting stuck in local optima and ensure diversity in new solutions. The fitness values of the new population are calculated and the same steps (selection, crossover, and mutation) will be repeated until the termination conditions are met.

6.3.3 Particle Swarm Optimization

The Particle Swarm Optimization (PSO) algorithm also belongs in the metaheuristic family, it is proposed by [80], and it works on the basis of behaviours of social animals. The PSO algorithm employs a swarm of particles that traverse a multi-dimensional search space to find optima. Each particle is a possible solution and altered by experiences of its own and neighbors. Furthermore, each particle is associated with a position vector and a velocity vector and updates its position based on the velocity vector as well as its previous experiences.

The main steps in PSO are given next. A random population of particles (solutions), based on the problem dimension, is initialized, where a random position vector and a random velocity vector are assigned to each particle (solution). A random population of possible solutions (particles) is generated and random velocity and position vectors are assigned to each particle. The population size depends on the problem dimension. The fitness of all particles is computed following the objective of the study and the best particle with the fittest objective value is selected. Then, the position and velocity of all particles are revised based on previous values along with some model parameters. The fitness evaluation and vector updates repeat until the termination criteria are met.

6.4 Computational Experiments

This section first presents experimental settings, details about data instances, and experimental results using metaheuristic-based and mathematical-based approaches. We conduct experiments for planning horizons of one week, two weeks, and four weeks, with 28 ships arriving for loading and unloading in one week (1st week of March 2019), 68 ships arriving in two weeks, and 138 ships arriving in four weeks (March 2019). We conduct the experiments on an Intel Core i7 2.4 GHz computer system with 16 GB RAM. All compared algorithms are developed in MATLAB on the same computer system and tested on the same datasets.

As for the multi-quay data instances, this study uses real data collected from the Port of Limassol, Cyprus. The Limassol port has five commercial continuous quays with different lengths: Container/Ro-Ro Quay: 450m, Container Quay: 800m, East Quay: 480m, West Quay: 770m, and North Quay: 430m. For each ship, the ETA, HT, ETD, PBQ, ABQ. PBP, and LoS are known. It is important to note that the real data do not include PBPs and ABQs for arriving vessels. Hence, we added PBPs randomly, as shown in the 7th column of Table 5. We also assign up to one ABQ for each vessel based on the port characteristics (e.g., availability of cranes, passenger boarding bridges) and ship type (e.g., container ship, passenger ship). In particular, the Container/Ro-Ro Quay is ABQ for ships having Container Quay as PBQ and vice versa and the West Quay is ABQ for ships with North Quay as PBQ and vice versa. There is no ABQ for ships with PBQ East Quay as this is the only quay that can handle passenger vessels. The real world data is collected through an online system developed for the STEAM Project [1].

6.4.1 Experimental Results

To verify the efficiency and effectiveness of the proposed approaches, this section discusses the results obtained from several experiments. CSA, GA, PSO, and MILP, have all been implemented to perform a comparative analysis. For the CSA implementation, we set the number of host nests to 100, discovery rate to 0.45, and total iterations to 1000. For GA, population size is 100, crossover rate is 0.10%, mutation rate is 90%, and total iterations are 1000. Regarding PSO, inertia weight, local learning coefficient, global learning coefficient, and population size are set to 1, 1.5, 2.0, and 1000, respectively.

Figure 16 shows the solutions for berth allocation developed by the four implemented algorithms, i.e., CSA, GA, PSO, and MILP. In this figure, each rectangle represents a ship, with the x-axis indicating the berthing time and the y-axis indicating the berthing position for a given ship. The blue colored rectangles show the ships moored at their preferred berth. The red rectangles, on the other hand, show the ships that moored at their ABQs instead. Based on the cost models, the implemented methods move a ship to an ABQ if the waiting time before the optimal berth is long, resulting in a delayed departure for the ships. A particular vessel may also be moored at an ABQ if its PBP is unavailable for a long time and a NOB position causes high costs or delays in delivering containers to the designated storage area. From this figure, it can also be seen that there is a safety distance and a safety time between any two vessels during berthing, which are subject to the following constraints. The safety time is set to one time slot (30 minutes) and the safety distance between two vessels is set to 10 meters. By respecting the overlapping constraint visualized in Figure 14), none of the solutions allow for any scheduling overlapping between any two ships. For instance, in order to avoid overlapping, ship 21 (highlighted with red color in Figure 16) is moored at its ABQ (North Quay) instead of its PBQ (West Quay) using CSA and PSO methods. However, GA and MILP shift ship 23 to its ABQ instead to avoid overlapping.

Figure 17 shows the mean difference (and standard error) between the proposed berthing times by the various algorithms and the optimal berthing times for the three scenarios (i.e., one week, two weeks, and three weeks). From this figure, it can be seen that there is no difference in case of MILP, i.e., it assigns ships always at optimal berthing times, for the one-week scenario. However, it is important to note that MILP can only solve the problem for one week planning horizon and it runs out of memory for the other two scenarios (i.e., two weeks and four weeks). On the other hand, when we compare CSA with the other heuristic methods, we observe that CSA provides a near-optimal solution (lowest mean difference) in all tested scenarios. The mean difference between proposed and optimal berthing times using CSA method are 0.32, 0.54, and 0.69 interval for one-week, two-weeks, and four-weeks, respectively. On the contrary, PSO performs better than GA in the cases of one week and two weeks, but worse in the four-weeks case. The highest mean difference are 0.39 (using GA), 1.38 (using GA), and 1.69 (using PSO) intervals in one-week, two-weeks, and four-weeks scenarios, respectively.

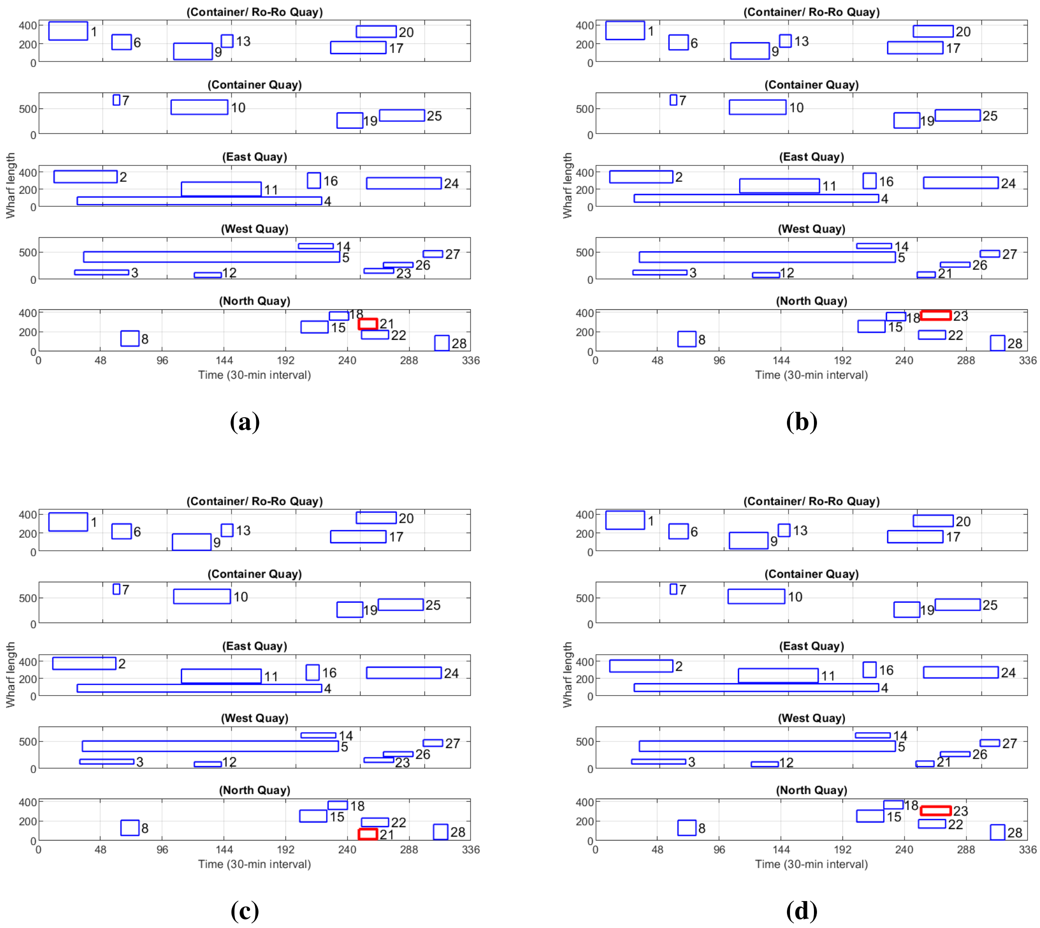


Figure 16: Berth allocation solutions for ships arriving over one week planning horizon, generated by (a) CSA, (b) GA, (c) PSO, and (d) MILP.

Figure 18 shows the mean and standard error for non-optimal berthing costs (in Euro) for all arriving vessels. This figure presents the results for all four algorithms and the three considered scenarios, i.e., one week, two weeks, and four weeks. It can be noted from the figure that the minimum NOB cost is achieved using MILP, closely followed by CSA; however, MILP can only solve the problem for one week planning horizon. Furthermore, when we compare the CSA-based solution with other heuristic methods, it can be noticed that CSA has the lowest NOB mean cost and standard error, in all tested scenarios. On the other hand, GA performs better compared to PSO in 1-week and 2-weeks scenarios but worst in the 4-weeks scenario. Finally, as the planning period grows from one week to four weeks, the performance of GA and PSO worsens at a much higher rate that CSA, showcasing the robustness of the CSA approach to handle longer planning periods.

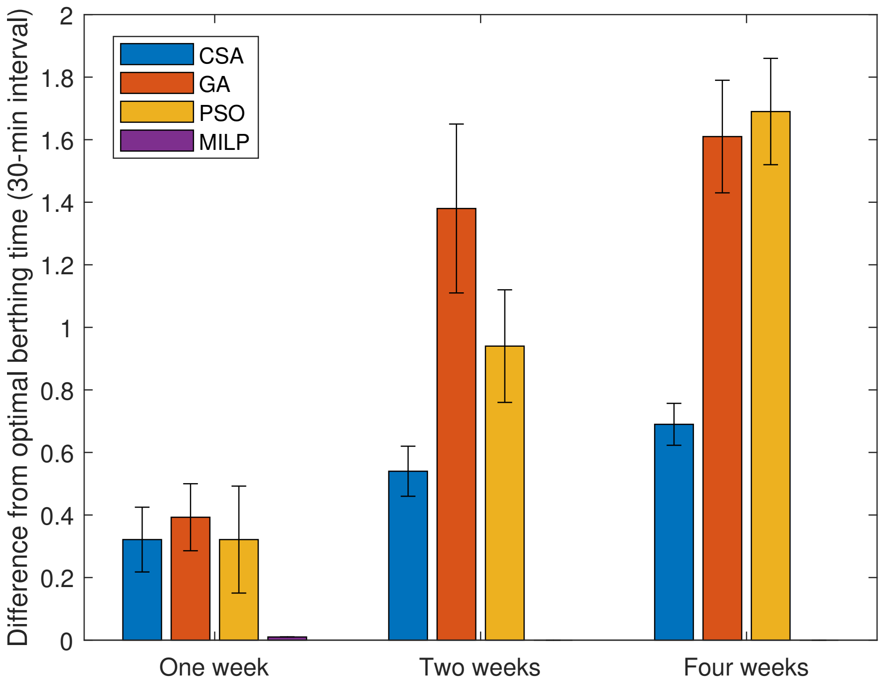


Figure 17: Mean difference between optimal and proposed berthing time for all vessels per method. MILP obtains zero difference for one week and is not able to run for two weeks and four weeks.

Figure 19 shows the total service cost incurred for all arriving ships in the planning horizon of one week, two weeks, and one month (four weeks). From this figure, it can be seen that the total service cost (in Euro) for the planning horizon of one week is minimal when MILP is used (i.e., 11005). In contrast, CSA has the lowest cost (11095) compared to the other two heuristic methods (i.e., GA and PSO). The highest service cost for a planning

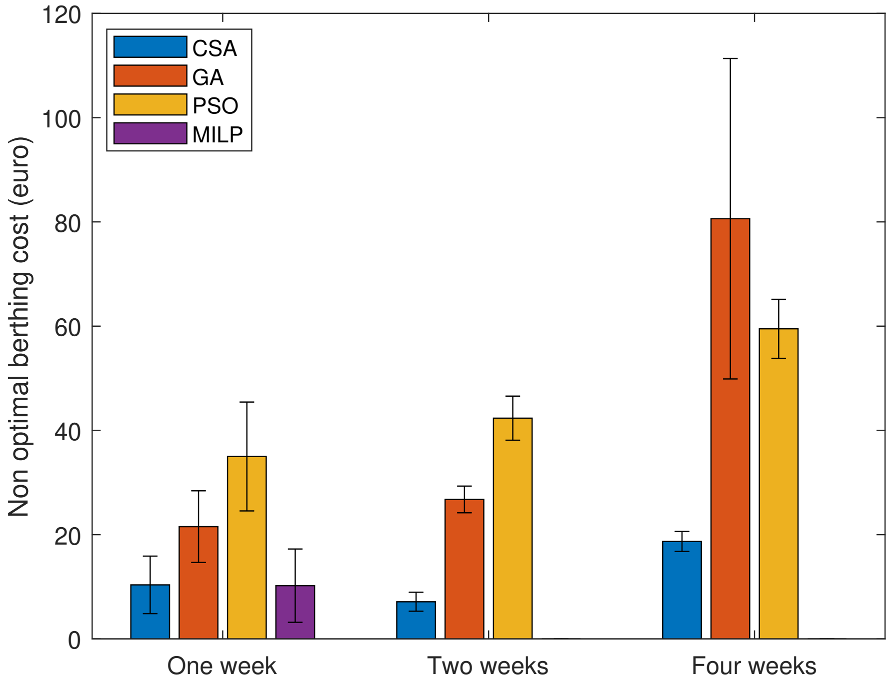


Figure 18: Non-optimal berthing (NOB) cost for all vessels per method. MILP is not able to run for two weeks and four weeks.

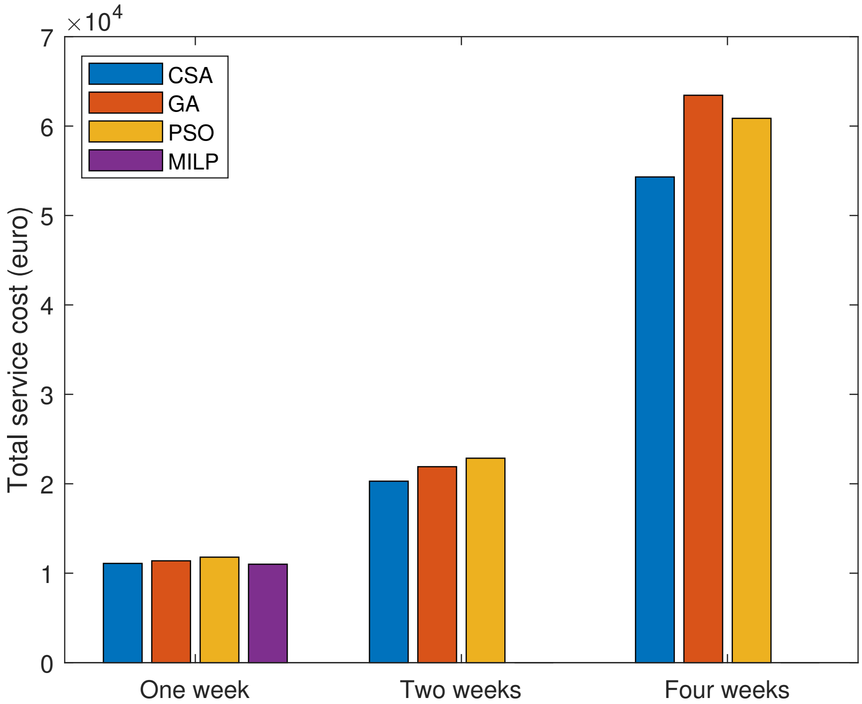


Figure 19: Total service cost for arriving vessels in one, two, and three weeks, for each approach. MILP is not able to run for two weeks and four weeks.

horizon of one week is 11795 when PSO is used. As the planning horizon increases to two and four weeks, the CSA approach is able to achieve the lowest service cost, while the difference between the CSA and the other two metaheuristic approaches increases.

For a more in-depth comparative analysis, Table 11 shows the different costs (i.e., waiting cost, non-optimal berth allocation cost, late departure cost, normal handling cost, and total service cost) along with the computation times for different planning horizons using all four implemented algorithms. We conducted experiments with different data instances to validate the productivity of the proposed CSA-based method. From this table, we can see that the MILP method has the highest computation time of 776 seconds (~13 minutes) for the planning horizon of one week. The MILP method provides an optimal solution with the lowest total service cost (and lowest individual costs). However, if we increase the planning horizon, and therefore the number of arriving vessels, from 1-week to 2-weeks or 4-weeks, MILP cannot solve the problem and runs out of memory. The NOB cost includes penalty costs when a vessel s is moored at a location other than its PBP or at a quay other than its PBQ. As shown in Equation 17, the penalty is calculated based on the absolute difference between the proposed and preferred berthing positions, i.e., $|PBP_s - B_s| \cdot C_s^{nob}$, where C_s^{nob} is equal to 5 Euros per meter. If a vessel berths at its ABQ instead of its PBQ, an additional fixed penalty of 50 Euros is charged; however, berthing a vessel at a location other than ABQ and PBQ incurs an infinite penalty (and thus is not possible). From Table 11, it can be seen that the penalty for NOB is lowest when MILP is used (235 Euros), closely followed by CSA with 280 Euros. PSO has the highest penalty cost of 980 Euros and for GA it is 560 Euros. In the case of a 1-week scheduling period, GA takes the least computation time of 18.95 seconds, closely followed by CSA with 21.91 seconds, while PSO takes 73.59 seconds. Furthermore, if we run experiments for 2-weeks and 4-weeks scenarios, it can be seen from the Table 11 that CSA always provides an optimal solution (minimum total service cost) within the least computation time, even less than GA and PSO. In the case of two weeks and four weeks, the CSA takes 70.60 and 133.27 seconds, respectively. In contrast, GA and PSO take 332.96 and 223.33 seconds for two weeks and 768.81 and 642.95 seconds for four weeks, respectively. From the above analysis, we can conclude that the newly developed CSA-based solution for MQ-BAP is more efficient and always provides a near-optimal solution.

Table 11: Comparative analysis of all methods while using data for 1-4 weeks (March 2018). All costs are in Euro.[WC: waiting cost; NOBC: non optimal berthing cost; LDC: late departure cost; NHC: normal handling cost; TSC: total service cost; CT (computational time)]

	One week (28 ships)				**Two weeks (68 ships)**				**Four weeks (168 ships)**			
	CSA	GA	PSO	MILP	CSA	GA	PSO	MILP	CSA	GA	PSO	MILP
WC	45	55	45	0	185	470	320	–	480	1110	1165	–
NOBC	280	560	980	235	485	1450	2880	–	2580	11125	8210	–
LDC	0	0	0	0	0	0	40	–	80	40	320	–
NHC	10770	10770	10770	10770	19620	19620	19620	–	51170	51170	51170	–
TSC	11095	11385	11795	11005	20290	21540	22860	–	54310	62445	60865	–
CT(s)	21.91	18.95	73.59	775.77	70.60	332.96	223.33	–	133.27	768.81	642.95	–

6.5 Summary

In this section of thesis, we deal with a special variant of the continuous BAP, namely the MQ-BAP, where more than one quay is available for mooring the arriving ships. The MQ-BAP is formulated as a mixed-integer linear problem and then solved using exact and metaheuristic methods, with the main objective of minimizing the total service cost while reducing the waiting time before berthing and the delayed departures of the ships. We considered the case of the Port of Limassol and used real data collected from the same port. Moreover, this study also considers several practical constraints of the port and introduces the new concept of alternative berthing quay (ABQ). The purpose of ABQ is to reduce long waiting times of vessels. For example, if a vessel's preferred position is occupied for a long time, the vessel can be moored at the nearest mooring quay instead. We have conducted several experiments to corroborate our model and verify the effectiveness of our proposed metaheuristic algorithms (i.e., CSA, GA, and PSO) over an exact method (i.e., MILP). Results reveal that the exact method can only solve the problem for a one-week planning horizon with high computation time. In contrast, the CSA-based method is able to solve all tested scenarios and beats the other metaheuristic methods (GA and PSO) in terms of minimum service cost and low computation time.

It is important to mention that the proposed methods are dynamic and able to solve any port environment with continuous or discrete berthing quays, and are already tested on benchmark data from different port (see Section 5). For the methods to work on a particular port, we only need to specify the necessary settings of a particular port, such as the number of quays, the number of cranes, the productivity of cranes, some additional constraints, etc.

The next section of this thesis will solve the combined BAP with quay crane allocation problem, considering a port environment with multiple quays. In addition, the next section also considers the entire port operation (including container and passenger/ special purpose terminals) instead of considering only container terminals.

7 Optimizing Combined Berth and Quay Crane Allocation Considering Multiple Quays

7.1 Motivation and Problem Statement

In the current literature, there are many studies dealing with stand-alone BAP [15, 17, 23]. Nowadays, however, there is an increasing trend to solve BAP and QCAP simultaneously, since the number of cranes (and which cranes in case of different handling productivity) assigned to a ship determines the berthing time of the vessels [146, 122]. Most of the current studies consider only a single quay, while they look at stand-alone BAP or combined BAP-QCAP [122]. There are very few studies dealing with terminals with multiple quays, especially terminals with different quays, e.g., container and passenger terminals. For example, in a study presented in [51], a solution for multiple quays is proposed for BAP; however, in this study, the total length of the quay is divided equally between two quays and random data are used for experiments. In addition, practical constraints are not considered. A recent study [88] also solves multi-quay BAP and concludes that the proposed method does not always provide an optimal solution and is sometimes 40% away from the optimal solution. In another work, [61] proposes a fuzzy-based solution, but as the authors acknowledge, the proposed method provides an optimal solution when only up to 10 vessels are considered. We have found only one research paper that addresses combined BAP and QCAP considering multiple quays, which employs fuzzy logic to solve the problem [105]. However, the authors of that study conclude that their approach is feasible only for small instances and suggest the use of metaheuristics for solving medium and large size problems. Based on the above discussion, this study solves a combined BAP-QCAP problem for a real port, considering several heterogeneous quays (i.e., with container, general cargo, and passenger quays).

7.2 Problem Explanation and Formulation

The combined BAP and QCAP is an optimization problem in which the objective is to allocate available berths and available quay cranes (QCs) across time to incoming ships to perform unloading/loading operations with the goal of minimizing the total handling cost. In this study, we consider a realistic setup of a port with multiple quays and a number of cranes available at each quay. Moreover, all quays follow a continuous berthing

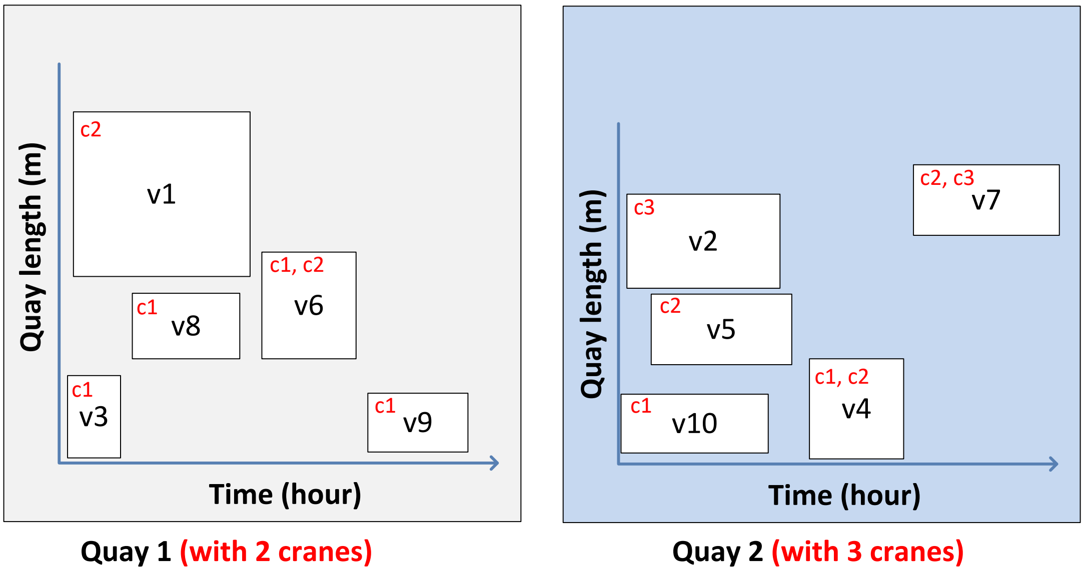


Figure 20: Combined BAP and QCAP solution with two berthing quays (both are continuous) and 10 arriving ships. Each rectangle denotes a ship, whose height (y-dimension) shows the ship's length and whose width (x-dimension) is the handling time of ship. Quay 1 has two cranes and quay 2 has three cranes, which are allocated to the ships as shown in the upper left corner of each rectangle.

layout, and arriving vessels can be moored anywhere on the quay. Vessels are arriving in a dynamic fashion; however, their expected arrival times are known in advance. The installed QCs perform loading and unloading operations with some average productivity, which can be different for each QC. For the handling cost calculation of each vessel, this study considers waiting cost, total service cost, late departure cost, and non-optimal berthing cost due to the allocation of a non-optimal position and/or quay. The assumptions we consider listed in Table 12.

7.2.1 Multi-Quay Combined BAP and QCAP Formulation

The primary objective of the multi-quay combined BAP and QCAP is to allocate optimal berthing position at preferred quay, berthing times, and QCs to arriving vessels in order to reduce the total handling cost (that includes waiting cost, service cost, and various penalties), as presented in the following cost function:

Table 12: Assumptions of the formulation in Section 7.2

- Number of ships, quay, and lengths of quays are known
- ETA and RDT for vessels are known
- Planning horizon is known and divided into equal time intervals
- Each berthing position can serve only one vessel at any time
- Vessels will take consecutive positions at the quay during service
- PBQ, PBP, and ABQ of each vessel are known
- All the berthing positions of all quays are assumed free at the beginning of the planning period
- The vessel cannot change its berthing position during loading/ unloading
- All the costs are also known
- All the penalties are known and the same for all arriving vessels

$$\begin{aligned} Cost\ (v, Q_v, BP_v, {}_v, BT_v) = HT_v \cdot [C_v^h + f(v, Q_v, BP_v)] \\ + WT_v \cdot C_v^w + LDT_v \cdot C_v^d \end{aligned} \tag{29}$$

The first term in the above Equation calculates the total handling cost, which depends on handling time (HT_v), handling cost (C_v^h), and a penalty (f) due to non-optimal quay and/or non-optimal berth assignment. This study, unlike our previous work in [18], does not consider handling time as an input but it is calculated based on the total load on the vessel $Load_v$ and the handling productivity (per hour) of each of the cranes $({}_v)$ that are assigned to the vessel v as described below.

$$HT_v = \frac{Load_v}{\sum_{c\,\in\,{}_v} HP_q^c}, \quad \forall\, v \in V, q = Q_v \tag{30}$$

There is a set of cranes $C(q) = \{c_1, c_2, c_3, ...c_n\}$ that are installed at each quay q. Regarding crane allocation to arriving vessels, there are several combinations of cranes possible, depending on the vessels' length and total load. For instance, if there is a large vessel with a heavy load, it can use multiple cranes, and vice versa. All possible combinations of crane assignments to a particular vessel are present in the power set of $C(q)$:

Table 13: Mathematical notation used in Section 7.2

Name	Explanation	Name	Explanation
AQ_v	Alternative quay for vessel v	AT_v	Expected arrival time of vessel v
BP_v	Planned berthing position of vessel v	BT_v	Planned berthing time of vessel v
$\mathbb{C}(q)$	Power set of cranes set C_q	C_v^h	Per hour handling cost of vessel v
C_v^w	Per hour waiting cost of vessel v	C_v^{ld}	Per hour late departure cost of vessel v
C_v^{nob}	Penalty of vessel v for non-optimal berthing position	C_v^{noq}	Penalty of vessel v for non-optimal berthing quay
c^{min}	Minimum berthing position served by crane c	c^{max}	Maximum berthing position served by crane c
$_v$	Set of cranes assigned to vessel v (presented in binary encoding)	FT_v	Task finishing time of vessel v
HT_v	Handling time of vessel v	HP_q^c	Handling productivity of crane c located on quay q
L_v	Length of vessel v	L_q	Length of quay q
LDT_v	Late departure time of vessel v	$Load_v$	Total load of vessel v
SC_q^c	Service cost per hour of crane c located on quay q	PBP_v	Preferred berthing position of vessel v
PQ_v	Preferred quay of vessel v	RDT_v	Requested departure time of vessel v
WT_v	Waiting time of vessel v		
Indices			
V	Set of arriving ships; $v \in V$ a ship		
T	Set of time periods (planing horizon); $t \in T$ a time period		
Q	Set of quays; $q \in Q$ a Quay		
$B(q)$	Set of available berth positions on quay $q \in Q$; $b \in B(q)$ a berth position		
$C(q)$	Set of quay cranes on quay $q \in Q$; $c \in C_v$ a crane; $_v$ is a subset of power set $\mathbb{C}(q)$		

$$\begin{aligned}\mathbb{C}_q = \Big\{ & \{\}, \{c_1\}, \{c_2\}, \ldots, \{c_n\}, \\ & \{c_1, c_2\}, \{c_1, c_3\}, \ldots, \{c_{n-1}, c_n\}, \\ & \{c_1, c_2, c_3\}, \ldots, \{c_1, c_2, \ldots, c_n\} \Big\}\end{aligned} \quad (31)$$

The set of cranes allocated to a vessel v is one element of the power set $\mathbb{C}_q$. To simplify the formulation, we encode each element of $\mathbb{C}_q$ as a number by using a binary interpretation.

In particular, each QC corresponds to a binary digit in the number (from right to left); if the digit is 1, the QC is allocated, while if it is 0, the QC is not allocated. For example, the element $\{c_1\}$ is encoded as 0001, $\{c_2\}$ is encoded as 0010, $\{c_1, c_2\}$ is encoded as 0011, and so on. The number of digits used in the encoding equals the number n of available QCs. Furthermore, the empty set (with encoding 0000) is also included for when a vessel is berthed to a terminal that does not have quay cranes, such as a passenger or Ro-Ro quay. In summary, $_v$ is a number, whose binary interpretation represents the set of cranes assigned to vessel v, and can take numbers between 0 and $2^n - 1$.

The per hour handling cost C_v^h charged to a vessel depends on the per hour service cost of the set of cranes $_v$ assigned to vessel v. Total service cost can be calculated as:

$$C_v^h = \sum_{c \in {}_v} SC_q^c, \quad \forall\, v \in V, q = Q_v \tag{32}$$

The last part of the first term, namely $f(v, Q_v, BP_v)$ in Equation 29, calculates the penalty cost due to non-optimal berth position and/or non-optimal quay assignment, if any, as shown in Equation 33. Moreover, this penalty is multiplied with the per hour handling cost C_v^h of vessel v. Hence, if any vessel needs four hours to perform its operations and it is moored at a non-optimal position or quay, the penalty is charged for all (four) time intervals.

$$f(v, Q_v, BP_v) = \begin{cases} |PBP_v - BP_v| \cdot C_v^{nob} & \text{, if } Q_v = PBQ_v \\ C_v^{noq} & \text{, if } Q_v \in AQ_v \\ \infty & \text{, otherwise.} \end{cases} \tag{33}$$

The second term in Equation 29 ($WT_v \cdot C_v^w$) calculates total waiting cost and it depends on the total waiting time WT_v of vessel v and the per hour waiting cost C_v^w. The waiting time WT_v of any vessel v is the difference between berthing time BT_v and arrival time AT_v, as calculated in Equation 34.

$$WT_v = BT_v - AT_v, \quad \forall\, v \in V \tag{34}$$

The last expression in Equation 29 ($LDT_v \cdot C_v^d$) calculates the penalty cost due to late departures, which is based on the late departure time LDT_v and the per hour penalty for late departure C_v^{ld}. The late departure time LDT_v can be calculated as:

$$LDT_v = max(0,\ FT_v - RDT_v), \quad \forall\, v \in V \tag{35}$$

where, RDT_v is the requested departure time for vessel v and FT_v is the finishing time of $v's$ operations (i.e., loading and unloading), as calculated below.

$$FT_v = BT_v + HT_v, \quad \forall\, v \in V \tag{36}$$

The fundamental **objective** of this study is to solve the combined BAP and QCAP in a multi-quay environment while reducing the total cost that includes handling cost, waiting cost, and several penalties. The objective function of our study is presented in Equation 37.

$$\textbf{minimize} \sum_{v\,\in\,V}\ \sum_{q\,\in\,Q}\ \sum_{b\,\in\,B_q}\ \sum_{\in\,\mathbb{C}(q)}\ \sum_{t\,\in\,T} Cost(v,q,b,,t)\cdot x_{vqbt} \tag{37}$$

The objective function is subject to several constraints that are presented below.

$$x_{vqbt} \in \{0,1\}, \forall\, v \in V, q \in Q, b \in B_q, \in \mathbb{C}_q, t \in T \tag{38}$$

$$\sum_{q\,\in\,Q}\ \sum_{b\,\in\,B_q}\ \sum_{\in\,\mathbb{C}_q}\ \sum_{t\,\in\,T} x_{vqbt} = 1, \quad \forall\, v \in V \tag{39}$$

$$BT_v \geq AT_v, \qquad \forall\, v \in V \tag{40}$$

$$BT_v - BT_u \geq SE \qquad \forall\, v \neq u \in V \tag{41}$$

$$BP_v + L_v \leq L_q, \qquad\qquad \forall\, v \in V,\ q \in Q_v \tag{42}$$

$$\sum_{v \neq u\, \in V} \sum_{b = BP_v - L_u - SD}^{BP_v + L_v + SD} \sum_{\in \mathbb{C}_q} \sum_{t = BT_v - HT_u - ST + 1}^{BT_v + HT_v + ST - 1} x_{uqbt} = 0, \qquad \forall\, v \neq u \in V,\ q = Q_v = Q_u \tag{43}$$

$$\sum_{v \neq u\, \in V} \sum_{b\, \in B_q} \sum_{\substack{\in \mathbb{C}_q \\ \&\, _v \neq 0}} \sum_{t = BT_v - HT_u - ST + 1}^{BT_v + HT_v + ST - 1} x_{uqbt} = 0, \qquad \forall\, v \neq u \in V,\ q = Q_v = Q_u \tag{44}$$

$$c^{min} < BP_v + L_v \ \&\ BP_v < c^{max},\ \ \forall\, v \in V,\ c \in {}_v \tag{45}$$

The variable x_{vqbt} shown in Constraint (38) is 1 if the vessel v is scheduled at position b of quay q at time t to be served by cranes , and 0 otherwise. Constraint (39) guarantees that each arriving vessel must be berthed only once during the time t at the mooring position b of the quay q. The constraint (40) specifies that the proposed berthing time BT_v for a given vessel v must always be equal to or later than its expected time of arrival AT_v. The condition in (41) ensures a minimum safety entrance time SE between any two consecutive berthing operations. Constraint (42) ensures the length of vessel v plus its berthing position must not cross the length of quay q, where it is moored. Constraint (43) restricts two vessels from overlap during mooring, both in terms of berthing positions as well as berthing times. Furthermore, it also ensures safety distance and safety time between the berthing of two ships. A graphical presentation of constraint (43) is presented in Figure 21. Constraint (44) restricts the set of cranes that is assigned to vessel u to not contain any of the cranes $\underset{\approx}{\succsim}$ allocated to another vessel v during the same time period. Checking if two sets of cranes have common cranes can be easily done using the 'bitwise and' (&) operation due to our binary representation of crane sets. A pictorial presentation of constraint (44) is presented in Figure 22. Finally, constraint (45) ensures that any crane c assigned to vessel v can reach the vessel by checking that there is an overlap between the

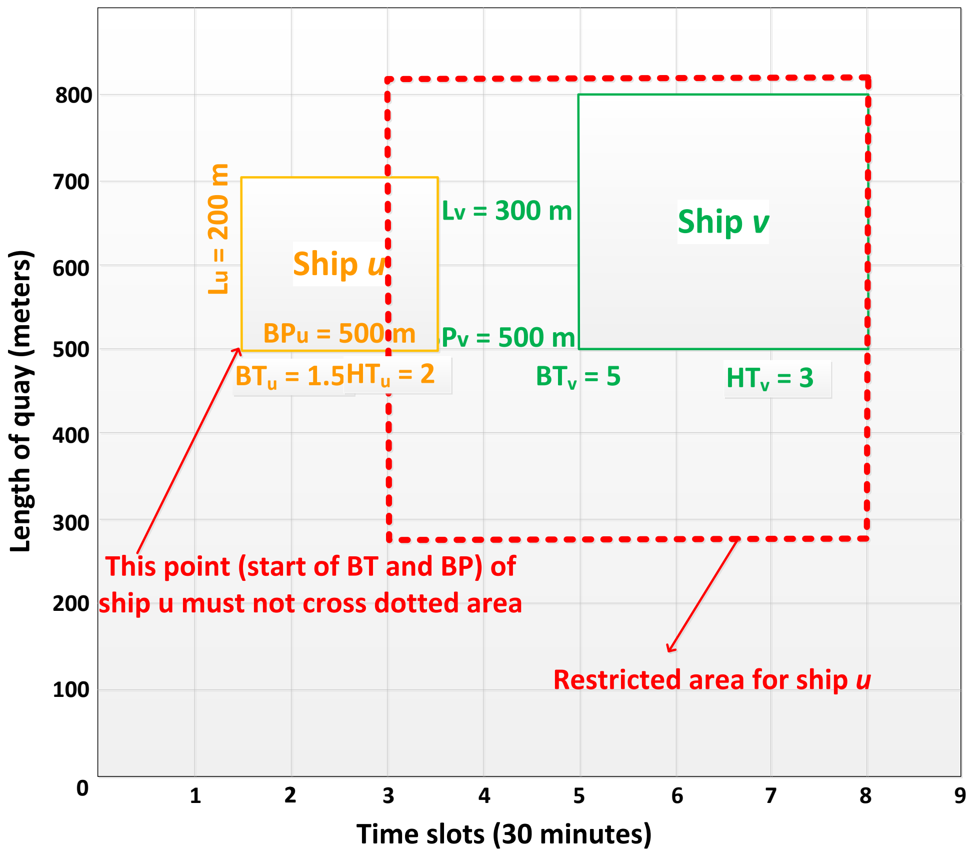


Figure 21: An Illustration of overlapping constraint with two arriving ships (ship v and ship u) with different berthing times, berthing positions, and lengths. This Figure shows the restricted areas for ship *u* (using dotted boxed) to avoid overlap with ship v, the already scheduled.

minimum and maximum berthing positions served by c and the quay positions occupied by v.

Table 14: Example data for 28 ships that arrived during the first week of March 2018 at the Port of Limassol, Cyprus

Ship	ETA (d \ t)	HT (min.)	ETD (d \ t)	PBQ	ABQ	PBP (m)	LoS
1	1\04:00	9827	1\22:30	Ro-Ro Quay	Container Quay	240	194
2	1\05:30	–	2\06:50	East Quay	–	276	139
3	1\14:00	–	2\12:50	West Quay	North Quay	84	84
4	1\15:00	–	5\14:03	East Quay	–	51	89

Continued on next page

Table 14 – Continued from previous page

Ship	ETA (d \ t)	HT (min.)	ETD (d \ t)	PBQ	ABQ	PBP (m)	LoS
5	1\17:00	–	5\21:00	West Quay	North Quay	314	190
6	2\04:30	4914	2\13:50	Ro-Ro Quay	Container Quay	138	159
7	2\05:00	10209	2\09:30	Container Quay	Ro-Ro Quay	571	196
8	2\08:00	–	2\15:55	North Quay	West Quay	53	155
9	3\04:00	10714	3\20:50	Ro-Ro Quay	Container Quay	31	175
10	3\03:30	33235	4\06:15	Container Quay	Ro-Ro Quay	389	277
11	3\07:30	–	4\14:55	East Quay	–	358	162
12	3\12:30	–	3\22:40	West Quay	North Quay	34	88
13	3\23:00	3557	4\05:00	Ro-Ro Quay	Container Quay	162	133
14	5\05:00	–	5\19:00	West Quay	North Quay	208	90
15	5\05:30	–	5\16:30	North Quay	West Quay	190	121
16	5\08:30	–	5\13:15	East Quay	–	267	178
17	5\17:30	3097	6\20:50	Ro-Ro Quay	Container Quay	96	129
18	5\16:00	–	6\00:25	North Quay	West Quay	112	84
19	5\20:00	16832	6 \09:35	Container Quay	Ro-Ro Quay	125	294
20	6\03:30	3998	6\21:25	Ro-Ro Quay	Container Quay	269	122
21	6\04:30	–	6\12:00	West Quay	North Quay	35	102
22	6\05:30	–	6\16:30	North Quay	West Quay	128	87
23	6\06:30	–	6\18:05	West Quay	North Quay	113	84
24	6\07:30	–	7\12:50	East Quay	–	207	130
25	6\12:00	16058	7\10:15	Container Quay	Ro-Ro Quay	260	217
26	6\14:00	–	7\02:05	West Quay	North Quay	219	88
27	7\05:30	–	7\13:05	West Quay	North Quay	364	121
28	7\09:30	–	7\15:25	North Quay	West Quay	7	155

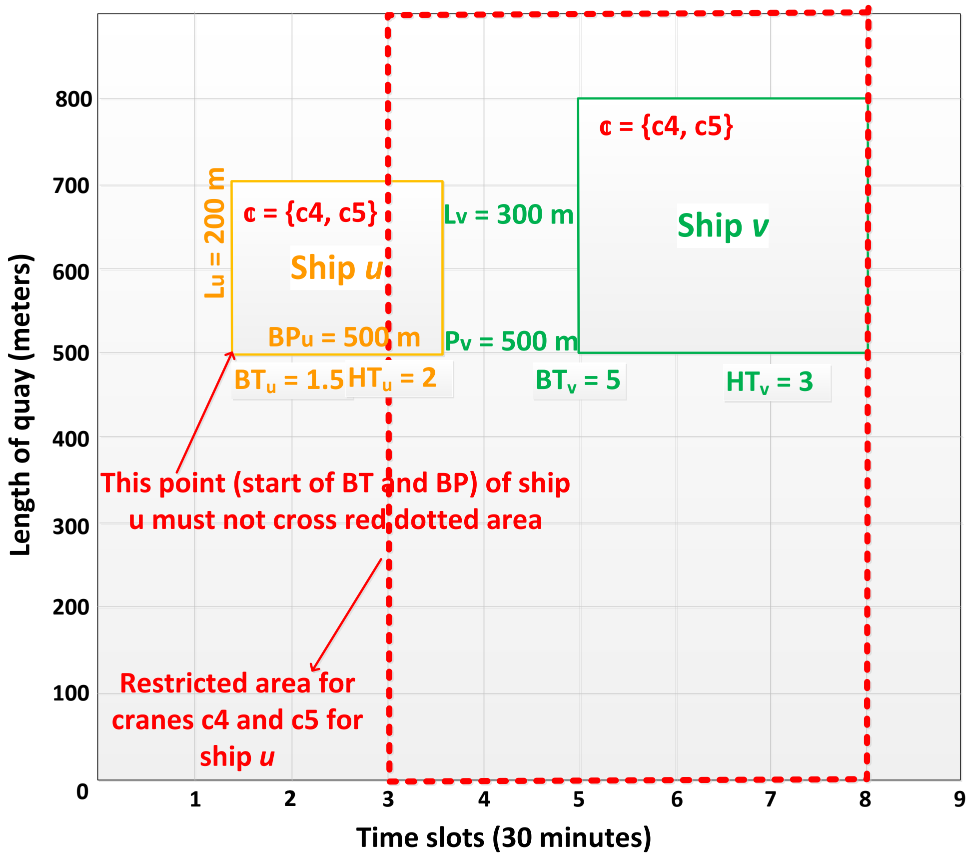


Figure 22: An illustration of Constraint 44. The cranes c_4 and c_5 assigned to ship v cannot be assigned to ship u if ship u is scheduled to be berthed in the restricted area marked with the red dotted box.

7.3 Proposed Methodology

In this section, we disclose the proposed methods for multi quay combined BAP and QCAP, which includes three CI methods, CSA, GA, and PSO.

7.3.1 Cuckoo Search Algorithm

The cuckoo search algorithm is a relatively new nature-inspired optimization method proposed by [152] that has proven efficient in solving several global optimization problems. CSA is based on the basic rules of breeding parasitism of some cuckoo species and then extended by the so-called Levy flights [24] instead of a simple isotropic random walk [153]. Some cuckoo birds follow an aggressive production strategy of laying eggs in communal nests and possibly removing eggs from other birds (host birds) to maximize the hatching probability of their own eggs. When host birds discover the cuckoo eggs, hosts either discard or abandon the eggs and build new nests. Overall, the CSA operates on the basis of cuckoo reproductive behavior and adopts three idealized rules [152]:

1) each cuckoo bird dumps only one egg at a time in a random nest;

2) the best nests having high quality eggs are kept and used for the next generation;

3) the number of host nests is fixed and the egg laid by a cuckoo is detected by a host bird with probability $p_\alpha \in (0,1)$.

The mapping of CSA to multi quay combined BAP and QCAP is as follows. A single nest shows a set of possible solutions containing the berthing times, quays, positions, and possible set of assigned cranes for all arriving ships, as shown in Figure 23. An egg in a nest denotes either a berthing time or a berthing quay or a berthing position in that quay for an arriving ship or a possible set of cranes (represented as a single number as explained in Section 7.2.1), whereas, a cuckoo egg shows a novel (or better) solution (i.e., a berthing time or quay or position or set of cranes). Hence, each nest includes $4N$ eggs, where N shows the number of ships that have arrived at a given time. Therefore, the total number of eggs in a nest is four times the total number of arriving ships. The total search space of the problem at each iteration is reflected by the total number of host nests, which is fixed (100 host nests are assumed in this study). The overall goal of the algorithm is to use cuckoo eggs (better solutions) to replace the not so good eggs in the various nests, while satisfying all constraints of the problem.

7.3.2 Genetic Algorithm

The genetic algorithm is an evolution-based algorithm developed from the law of evolution in the ecological world. It is also known as a population-based method that explores the concept of survival of the fittest [77]. After the first population is generated, it evolves better and better approximate solutions from generation to generation. In each generation, the individual is selected based on the fitness of different individuals in a particular problem domain. Then the individuals are combined and crossed and vary by the genetic operators in natural genetics, and then a new population is generated, which is a new solution set. Chromosome representation, selection, crossover, mutation, and fitness function calculation are the key elements of GA.

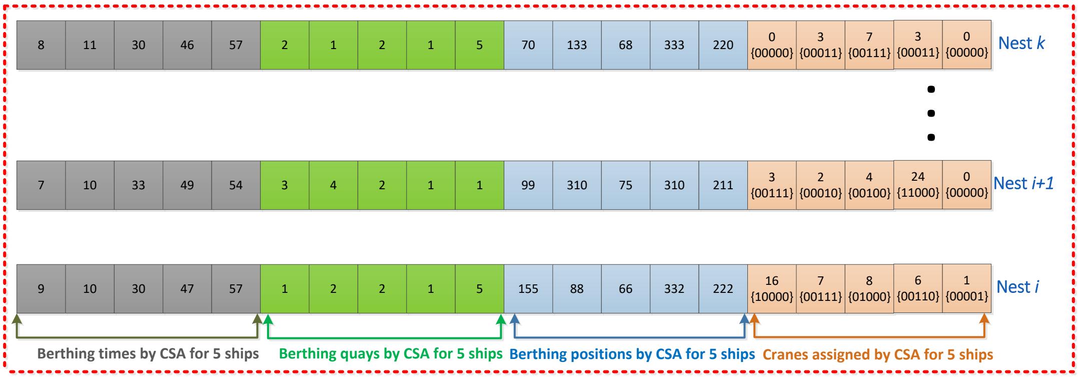


Figure 23: Solution representation by CSA considering five arriving vessels (last part of solution contains number of cranes and it maybe 0, 1, 2, and maximum of 3 in our study). Please note that decimal numbers (0,1,2,3) shows total number of cranes and binary encoding demonstrates specific cranes.

The working mechanism of GA is as follows. A random population of n chromosomes is generated. The fitness value of each chromosome is calculated using the objective function, and the best chromosome is selected as the local best chromosome. Two chromosomes, $C1$ and $C2$, are randomly selected from the population, and crossover with probability C_p is applied to $C1$ and $C2$ to generate an offspring o. Then, mutation with probability M_p is performed on the offspring o to generate o'. The new offspring o' is placed into a new population. The entire steps are repeated until the maximum number of iterations is reached.

7.3.3 Particle Swarm Optimization

The particle swarm optimization algorithm is a swarm-based metaheuristic global optimization method that has attracted much attention in the last two decades. The PSO is capable of solving large and complex problems that cannot be addressed by traditional methods. The PSO follows the behavior and social cooperation of flocks of birds and schools of fish and borrows heavily from the evolutionary behavior of these organisms. In PSO, all possible solutions are represented by particles (birds) in a search space, and each particle has its fitness value based on the objective function that to be optimized. Each particle also has a velocity that controls how the particles fly. The particles fly in the search space and follow/consider the current optima to find a local optimum. At each iteration, the local optimum is updated by the global optimum based on the optimal objective function. The general operation of PSO is as follows. 1) Generate random particles (solutions) in the search space. 2) Evaluate and compare the fitness of all particles. 3) Update the best particle (solution set). It is important to note that in this study, each solution set contains the mooring time, the quay, the position on that quay, and the set of cranes. 4) Update the velocities and positions of the particles to generate new solutions. Steps 2-4 are repeated until the maximum number of iterations is reached.

7.4 Experimental Setting and Results

In this section, we first present a case study from the Port of Limassol, Cyprus, and then show experimental results. A real-world case study (with real data, real port settings, real constraints, and real arriving ships) is used to efficiently test the performance of the proposed CI based approaches. We conduct experiments with different data instances, e.g., with one week, two weeks, and four weeks of data. The newly developed methods for multi-quay combined BAP and QCAP have been implemented in MATLAB R2021b and the number of iterations for all metaheuristics (CSA, PSO, and GA) was set to 1000. All experiments are performed using a Windows 10 computer system with 3.4 GHz Core i7 and 16 GB RAM.

7.4.1 A Case Study

This study deals with the case of a real Port, located in the city of Limassol, the largest port of the island (Cyprus). In the Port of Limassol, there are five commercial berthing quays, all of which are continuous. With a continuous quay, the whole quay is not divided into several berths, but rather the arriving ships can moor anywhere on the quay. All quays are of in different lengths: Container Quay: 800m, Ro-Ro Quay: 450m, West Quay: 770m, North Quay: 430m, and East Quay: 480m. The Container Quay serves only container ships, the Ro-Ro Quay serves both container and Roll-on/roll-off ships, the West and North Quays serve general cargo traffic, and the East Quay serves only passenger vessels.

There are five cranes installed at the container Quay, and each crane has a different productivity: two cranes with a maximum productivity of 40 containers per hour (red colored cranes), two cranes with a maximum productivity of 35 containers per hour (blue colored), and the last crane has a productivity of 22 containers per hour (white colored). It should be noted here that all five cranes never reach maximum productivity due to manpower issues, traffic problems, and some other technical challenges. So the average productivity of the first four cranes is 25 containers per hour; however, the last crane achieves an average productivity of 20 containers per hour. There are also two cranes installed at the Ro-Ro Quay with an average productivity of 20 containers per hour. In addition, all installed cranes at both (Container and Ro-Ro) Quays are moveable but within a certain range and cannot cross each other.

In our experiments, we use data from one week, which includes a total of 28 arriving ships in the first week of March 2018. Then, we extend our experiments by using data from two and four weeks, which contain more than 70 and 130 ships, respectively. For each vessel, the ETA, ETD, PBQ, ABQ, PBP, LoS, and total load are known in advance (Table 14 shows data for one week). It is important to note that the real data do not include PBPs and ABQs for arriving ships. Therefore, we added PBPs randomly, as shown in the 7th column of Table 14. We also allocate up to one ABQ (see column 6 of Table 14) for each vessel based on port characteristics (e.g., availability of cranes, passenger boarding bridges) and ship type (e.g., container ship, passenger ship).

7.4.2 Results and Discussions

In this section, we present the experimental results obtained by applying three CI algorithms and mathematical method. As for the datasets, we use real data collected in the Port of Limassol, Cyprus (one week's data is given in Table 14). Figure 24 shows the solutions proposed by all four algorithms for the allocation of berths and quay cranes. The rectangles in this figure show each ship, with the x-axis indicating the berthing time and the y-axis indicating the berthing position of each ship. The number in front of the rectangle shows the ship index and the text in green color shows the assigned set of cranes to each vessel. In addition, ships with blue rectangle indicate that they are moored at their PBQ, while ships moored in ABQ are colored red. Vessels are moored in ABQ when there is a long waiting time before the optimal berth assignment, which may result in delayed departures. Therefore, the ship is moored at ABQ to avoid high overall service cost.

From Figure 24 it can be seen that vessel 21 is moored at North Quay instead of West Quay when CSA, GA, and PSO are used. MILP, on the other hand, moves ship 23 to

ABQ instead of PBQ. Here it is important to note that there are only two quays where QCs are installed and assigned by all algorithms, i.e., CSA, GA, PSO, and MILP. All other quays are passenger/general cargo quays and no cranes are installed on these quays. In the case of the container quays, the total operating time of the vessels is calculated based on the number of cranes used and their productivity. However, in the case of the other three quays, the total operating time of the vessels is considered as input. In a week, four ships arrive at the container quay and all of them are assigned the optimal number of cranes using all the implemented algorithms. It should also be noted that the cranes can only move within a certain range from their location. Furthermore, the cranes cannot cross each other and all have different handling productivity rates.

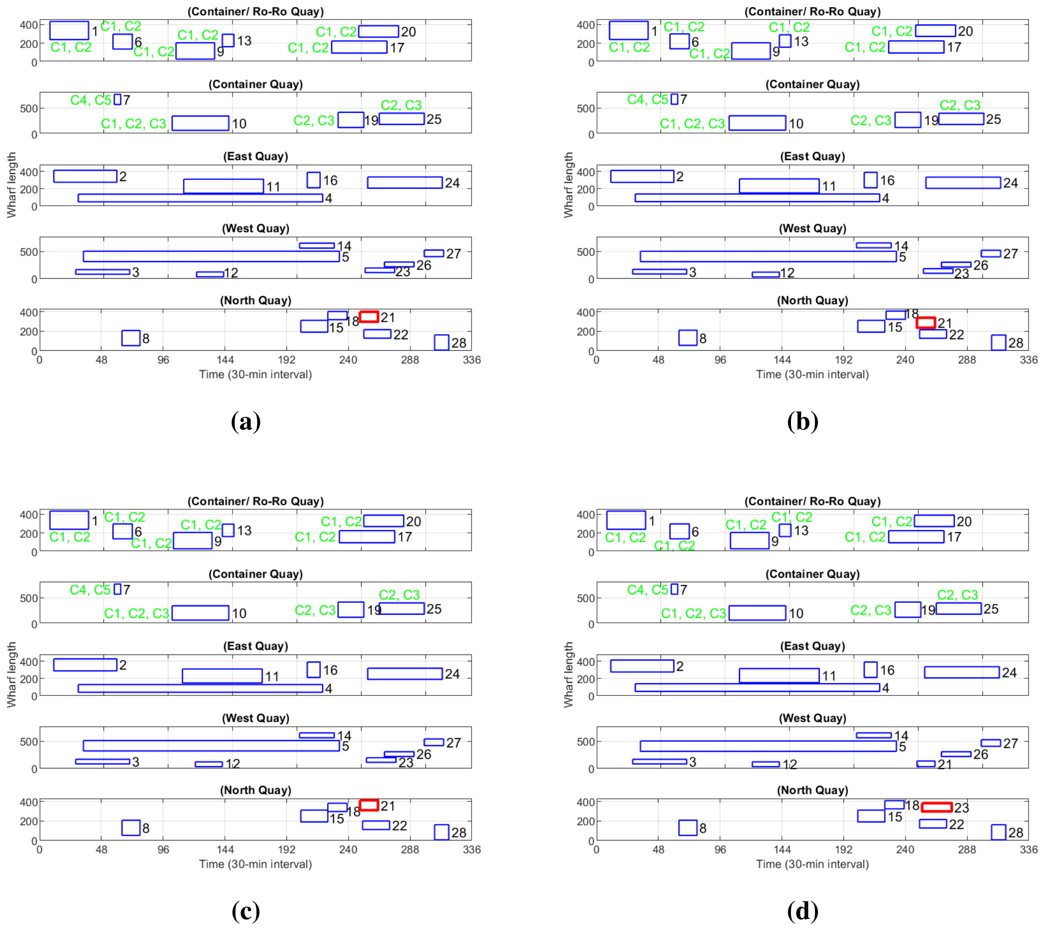


Figure 24: Berth allocation solutions by the four compared approaches for ships arriving over one week planning horizon, generated by (a) CSA, (b) GA, (c) PSO, and (d) MILP.

The results presented in Figure 25 show the mean difference and standard error between the optimal berthing times and the times proposed by different implemented CI methods, i.e., CSA, GA, PSO, and MILP. In this figure, it can be seen that there is no mean difference and standard error when MILP is used for a planning horizon of one week. However, MILP was only able to solve a one-week planning horizon and got stuck at 2 and 4 weeks planning horizons and ran out of memory. On the other hand, CSA shows superiority over the other two metaheuristics, i.e., GA and PSO, in all three scenarios. Moreover, GA performs well in case of one week and four weeks scenario, compared to PSO, while PSO shows a smaller mean difference and error in the case of the two-weeks scenario.

Figure 26 shows the mean difference between optimal and non-optimal berthing positions. It can also be seen that the MILP has the smallest difference in case of one week case; however, it cannot solve the other two cases (two weeks and four weeks). On the other hand, CSA again performs well in case of one-week and two-weeks scenarios compared to the other methods, and beats PSO in case of 4 weeks. However, GA has a lower mean difference in four-week scenario compared to PSO and CSA. In the case of the two weeks scenario, GA performs well and outperforms PSO.

The results presented in Figure 27 show the mean difference between optimal and non-optimal berthing cost that occurred due to allocation of vessels to locations other than the PBPs. Again, we can clearly see that CSA has the smallest mean difference after MILP in the case of one and two weeks scenarios; GA, on the other hand, has the smallest difference in the case of the four weeks scenario. We can conclude from the above discussions that CSA performs well overall compared to GA and PSO; however, MILP shows better results, but only in the case of one-week case study.

Figure 28 shows the total service cost (in euros) for all arriving ships using four different algorithms, i.e., CSA, GA, PSO, and MILP, while considering three different scenarios, i.e., one week, two weeks, and four weeks. In case of one week case, MILP shows minimum cost that is 10860, closely followed by CSA which is 10955 (only 0.99% away from the optimum). In the other two cases, MILP cannot solve the problem and runs out of memory. However, CSA always performs better than GA and PSO in two and four week scenarios. Moreover, in case of one week, PSO has the lowest cost compared to GA, on the other hand, GA beats PSO in terms of minimum service cost for all arriving ships in other two cases (two weeks and four weeks).

To show a more in-depth comparison of the different CI methods and the exact MILP method, Table 15 presents the different costs associated with the total service costs and

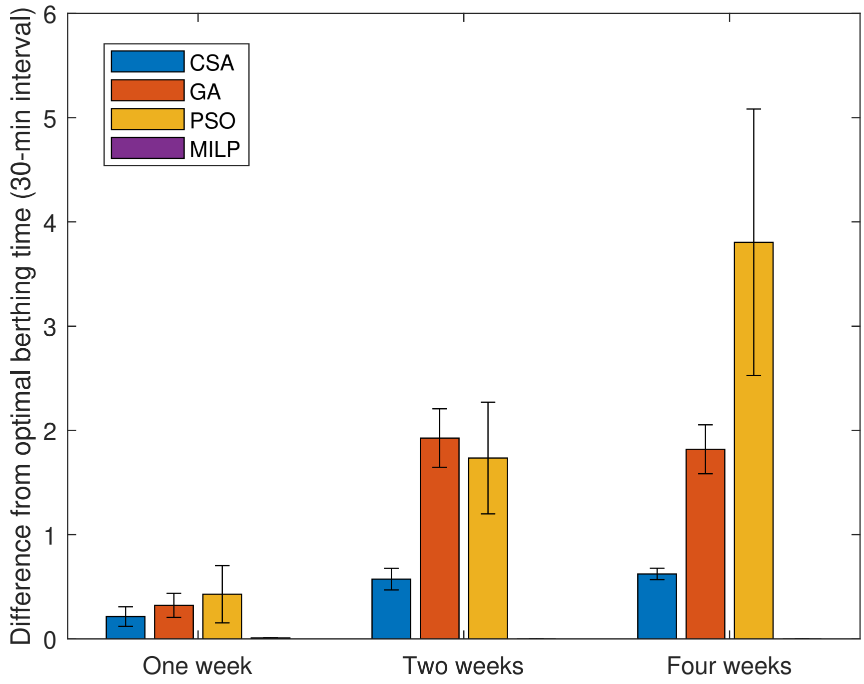


Figure 25: Mean difference and standard error between berthing time by four implemented methods and optimal berthing time three different scenarios (1-week, 2-weeks, and 4-weeks)

computation times of the different algorithms in different scenarios, i.e., one week, two weeks, and four weeks. Waiting costs are incurred when a vessel v has to waits before the optimal berth allocation, while NOB costs are included in the total service cost when a vessel v is moored at a berth position other than its preferred berth position or at ABQ instead of PBQ. NOB is added based on the absolute difference between the optimal berth position and the assigned position (by any algorithm), as described in Equation 33. However, a fixed penalty is added in case of berth allocation at ABQ. Furthermore, to avoid the berthing of vessels to quays other than ABQ or PBQ, a penalty of infinity is added (see Equation 33). From this table, it can be seen that MILP has a minimum total cost (10860) with 0 waiting cost and 0 cost for late departures. However, it provides an optimal solution at the cost of computation time, which is 912.55 seconds (more than 15 minutes) for a one-week scenario. In the experiments for two weeks and four weeks planning periods, MILP cannot solve the problem and runs out of memory. On the other hand, total service cost of CSA is minimum (that is 10955), closely followed by MILP, compared to other CI methods (i.e., GA and PSO) in case of one week scenario. Then, PSO beats GA in term of minimum total service cost for one week case study. Furthermore, when we run

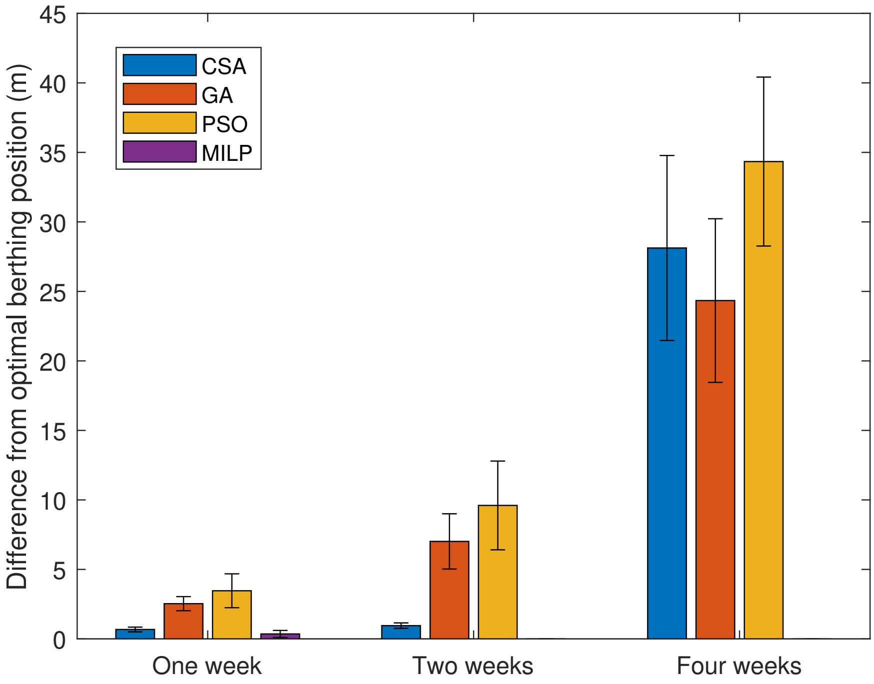


Figure 26: Mean difference and standard error between optimal and non-optimal berthing position using four CI methods when implemented in three different scenarios (1-week, 2-weeks, and 4-weeks)

experiments for two weeks, GA again show supremacy over GA and PSO in terms of total service cost and individual costs (i.e., waiting cost, NOB cost, and late departure cost). However, it can be noticed that GA beats PSO this time in case of two weeks scenario. Eventually, when we run experiments for 4 weeks (one month), we have noticed that NOB cost of all algorithms is increased; in this case, GA achieves minimum as compared to PSO and CSA. The total service cost by GA is 66260 that is minimum compared to other methods in case of four weeks. However, CSA also achieves minimum cost, that is 66895, compared to PSO (75310). Finally, as for computation time, this is an important parameter while developing algorithms, CSA solves all the scenarios in minimum computation times that are 84.73, 336.16, and 388.20 seconds for one week, two weeks, and four weeks, respectively. In contrast, GA and PSO take 94.67 and 316.23 seconds for one week, 226.57 and 534.80 seconds for two weeks, and 2663.13 and 2777.40 seconds for four weeks, respectively. From the above comparative analysis, it can be concluded that the newly developed CSA-based method for multi quay combined BAP and QCAP is more efficient and always provides a near-optimal solution within affordable computation time.

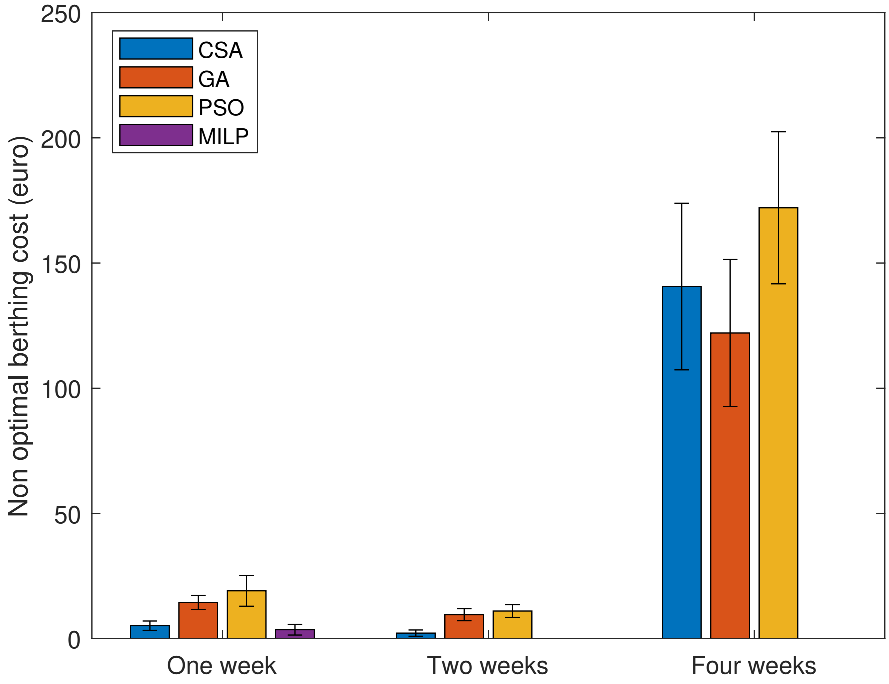


Figure 27: Mean difference and standard error between non optimal berthing cost using CI methods when implemented in three different scenarios (1-week, 2-weeks, and 4-weeks)

Table 15: Comparative analysis of all methods while using data for 1-4 weeks (March 2018). All costs are in Euro.[WC: waiting cost; NOBC: non optimal berthing cost; LDC: late departure cost; NHC: normal handling cost; TSC: total service cost; CT (computational time)]

	One week (28 ships)				**Two weeks (68 ships)**				**Four weeks (168 ships)**			
	CSA	GA	PSO	MILP	CSA	GA	PSO	MILP	CSA	GA	PSO	MILP
WC	30	45	60	0	195	655	590	–	430	1255	2625	–
NOBC	145	405	535	100	150	650	750	–	19405	16845	23745	–
LDC	0	0	0	0	240	680	900	–	1620	2720	3500	–
TSC	10955	11210	11355	10860	20530	23990	25125	–	66895	66260	75310	–
CT(s)	84.73	94.67	316.23	912.55	336.16	226.57	534.80	–	388.2	2663.1	2777.4	–

7.5 Summary

This section investigates multi quay berth allocation problem and quay crane allocation problem in a real-world scenario with the objective of minimizing total service cost for ar-

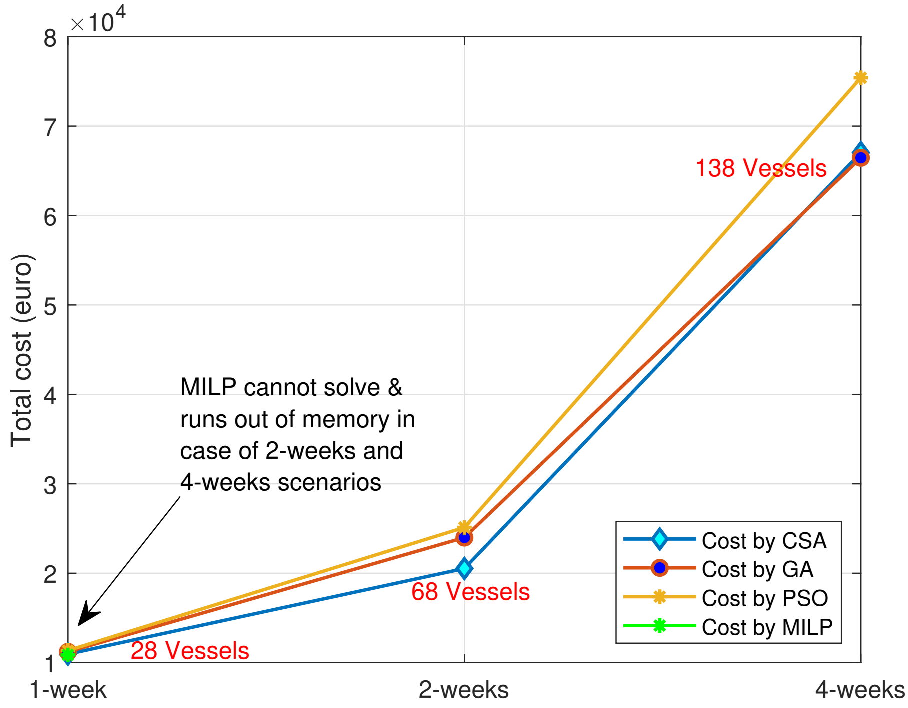


Figure 28: Total cost of all algorithms when implemented in three different scenarios (1-week, 2-weeks, and 4-weeks)

riving vessels. Total service cost includes handling cost, waiting cost, and various penalties. To solve multi quay combined BAP and QCAP, a MILP model is developed and solved using exact and CI methods. We implement the CSA, GA, and PSO, for the first time for multi quay combined BAP and QCAP. To validate the methods we developed for multi quay combined BAP and QCAP, we test them on real data collected in the port Port of Limassol, Cyprus. We use three different scenarios, i.e., one week, two weeks, and four weeks, to verify the scalability of developed approaches. Simulation results confirm the productivity of the CI-based methods over the exact method (MILP), since the latter can only solve a one week scenario and requires a lot of computation time (912 seconds). In contrast, the CSA method solves a one week scenario in only 84.73 seconds and the achieved objective value (10955) is only 0.99% away from the optimal solution (10860 euro). Moreover, the CSA-based solution also beats the other two methods, i.e., GA and PSO, in terms of objective value and computation time.

Based on extensive experiments, we can conclude that the exact method for multi quay combined BAP and QCAP studied in this paper tends to require too much computational effort to be of practical use, nor can it solve the problem in affordable computational time. On the contrary, CI based approaches, especially CSA, are able to offer superior quality results (close to the optimal solution) in a short computation time. Moreover, even for large data instances, the computation time of our CSA remains below 400 seconds (about 6 minutes). This gives the opportunity to evaluate the berthing and crane allocation plans more quickly in the dynamic environment of large container terminals, allowing the berth planner to more efficiently handle adjustments due to sudden schedule changes or disruptions.

Even though the experimental results presented are specific to the Port of Limassol, the problem formulation and all developed approaches can be used for solving the combined multi-quay BAP and QCAP for any other port environment with continuous or discrete berths. The main requirement for doing so is to specify the settings of a particular port, such as the number of quays, the number of cranes, the productivity of cranes, some additional constraints, etc.

CONCLUSIONS

Over the last couple of decades, demand for seaborne containerized trade has increased significantly and it is expected to continue growing over the coming years. As an important node in the maritime industry, a marine container terminal (MCT) should be able to tackle the growing demand for sea trade. Due to the increased number of ships that can arrive simultaneously at an MCT combined with inefficient berth allocation procedures, there are often undesirable situations when the ships have to stay in waiting queues and delay both their berthing and departure.

Therefore, the objective of this dissertation is to enhance berth planning in MCTs to increase the overall efficiency of the port. Since BAP has NP-hard complexity, the exact optimization algorithms cannot solve the real-size problem in a reasonable computation time (which has been proved by our work and the current literature). The latter justifies the application of metaheuristic algorithms to solve BAP. Therefore, this study models and presents solutions to stand-alone single quay BAP, multi quay (MQ) BAP, and MQ combined BAP and QCAP by proposing computational intelligence (CI) methods, i.e., cuckoo search algorithm (CSA), genetic algorithm (GA), and particle swarm optimization (PSO).

In our first contribution, this study focuses on the berth allocation problem with dynamic ship arrivals to improve port efficiency in terms of reducing the total handling cost and late departures, where a CI-based CSA is proposed to solve the BAP. In addition, we implemented two benchmark methods for comparison, a well-known metaheuristic GA and an exact approach (MILP). Unlike existing studies, and to make the problem more practical, a fine-grained time interval of 30 minutes is used in this study along with other practical constraints such as a safety time distance between berths. The proposed and compared approaches are implemented on multiple data instances generated from a benchmark dataset. In addition, randomly (uniformly) generated data instances with up to 100 vessels and a planning horizon up to a week are used for experiments to test the flexibility and scalability of the proposed method. The results show that our proposed algorithm (CSA) has higher efficiency in terms of minimum processing cost for all incoming ships compared to GA. Compared to MILP, our proposed CSA algorithm provides a near-optimal solution at a fraction of the computation time. Moreover, when we implement all algorithms on large datasets, the MILP algorithm runs out of memory and cannot provide an optimal solution in reasonable computation time. Overall, CSA beats GA in terms of processing cost and outperforms MILP in terms of computation time, and thus provides near-optimal solutions in affordable computation times.

Furthermore, this dissertation also deals with a special variant of the continuous BAP, namely the MQ-BAP, where more than one quay is available for mooring the arriving ships. The MQ-BAP is formulated as a mixed-integer linear problem and then solved using exact and CI methods, with the main objective of minimizing the total service cost while reducing the waiting time before berthing and the delayed departures of the ships. We considered the case of the Port of Limassol and used real data collected from the same port. Moreover, this study also considers several practical constraints of the port and introduces the new concept of alternative berthing quay (ABQ). The purpose of ABQ is to reduce long waiting times of vessels. We have conducted several experiments to corroborate our model and verify the effectiveness of our proposed CI algorithms (i.e., CSA, GA, and PSO) over an exact method (i.e., MILP). Results reveal that the exact method can only solve the problem for a one-week planning horizon with high computation time. In contrast, the CSA-based method is able to solve all tested scenarios and beats the other metaheuristic methods (GA and PSO) in terms of minimum service cost and low computation time.

Finally, this study investigates MQ combined BAP and QCAP in a real-world scenario with the objective of minimizing total service cost for arriving vessels. Total service cost includes handling cost, waiting cost, and various penalties. To solve MQ combined BAP and QCAP, a MILP model is developed and solved using exact and CI methods. We implement the CSA, GA, and PSO, for the first time for MQ combined BAP and QCAP. To validate the methods we developed for MQ combined BAP and QCAP, we test them on real data collected in the port Port of Limassol, Cyprus. We use three different scenarios, i.e., one week, two weeks, and four weeks, to verify the scalability of developed approaches. Simulation results confirm the productivity of the CI-based methods over the exact method (MILP), since the latter can only solve a one week scenario and requires a lot of computation time (912 seconds). In contrast, the CSA method solves a one week scenario in only 84.73 seconds and the achieved objective value (10955) is only 0.99% away from the optimal solution (10860 euro). Moreover, the CSA-based solution also beats the other two methods, i.e., GA and PSO, in terms of objective value and computation time.

In conclusion, this dissertation examined the impact of seaports, particularly MCTs, and analyzed routine operations of MCTs. Based on an extensive literature review and analysis, it was observed that the allocation of berths and quay cranes can significantly affect the performance of ports/ MCTs. Therefore, several innovative solutions have been proposed in this dissertation for several real port scenarios. The proposed solutions include (1) CI-based approaches for stand-alone BAP, (2) CI-based methods for MQ-BAP under real port conditions (Port of Limassol), and (3) CI-based solutions for MQ combined BAP and QCAP considering the whole port (container and passenger/general cargo terminals). In addition, the performance of all proposed solutions was verified by extensive experiments and by comparisons with benchmark approaches. The results show that all proposed methods can assist the port/ MCT operators in developing a cost-effective plan for berth and crane allocation.

We plan to extend the modeling to include a hybrid berthing layout that includes both discrete and continuous berthing layouts simultaneously. In addition, we plan to use our proposed CI based approaches for simultaneous berth allocation, quay crane scheduling, and container storage in the marshaling yard. Considering the uncertain breakdown of QCs is also another track for our future research. We also aim to integrate several uncertainties in BAP and QCAP, due to different weather conditions, maintenance activities at the quays, maintenance of cranes, and storage management of containers at the marshaling yard. Decarbonization of maritime activities is also part of our future research. Furthermore, we also plan to consider several real port factor while solving BAP, e.g., waiting time outside the port, connected cranes, and connected berths, etc. Finally, our immediate research direction is to consider the men power while solving stand-alone BAP or combined BAP and QCAP.

REFERENCES

[1] Sea Traffic Management in the Eastern Mediterranean, November 2019. https://steam.cut.ac.cy/steamimpacts/.

[2] Real-time data of a dry bulk terminal in Newcastle, Australia, December 2020. http://users.monash.edu/~andrease/Downloads.

[3] Maritime Transport: Trends of containerized trade 2020, April 2021. https://stats.unctad.org/handbook/MaritimeTransport/WorldSeaborneTrade.html.

[4] Ahmed M Abdelmagid, Mohamed S Gheith, and Amr B Eltawil. A binary integer programming formulation and solution for truck appointment scheduling and reducing truck turnaround time in container terminals. In *2020 IEEE 7th International Conference on Industrial Engineering and Applications (ICIEA)*, pages 149–153. IEEE, 2020.

[5] Omar Abou Kasm, Ali Diabat, and TCE Cheng. The integrated berth allocation, quay crane assignment and scheduling problem: mathematical formulations and a case study. *Annals of Operations Research*, pages 1–27, 2019.

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

[7] Map of the Port of Limassol, Cyprus by Ship Tracking Intelligence Platform, Jan 2022. https://ais.cut.ac.cy.

[8] Alia Aljasmi, Reem Al Kaabi, and Noura Al Hassani. A static-hybrid berth allocation problem with multi ship crane scheduling. In *2016 7th International Conference on Information, Intelligence, Systems & Applications (IISA)*, pages 1–5. IEEE, 2016.

[9] Bushra Alnaqbi, Hanan Alrubaiai, and Shaikha Al Alawi. Combination of a dynamic-hybrid berth allocation problem with a quay crane scheduling problem. In *2016 7th International Conference on Information, Intelligence, Systems & Applications (IISA)*, pages 1–8. IEEE, 2016.

[10] Ghazwan Alsoufi, Xinan Yang, and Abdellah Salhi. An evolutionary approach to solving a new integrated quay crane assignment and quay crane scheduling mathematical model. *European Journal of Operational Research*, 2016.

[11] Ghazwan Alsoufi, Xinan Yang, and Abdellah Salhi. Robust berth allocation using a hybrid approach combining branch-and-cut and the genetic algorithm. In *International workshop on hybrid metaheuristics*, pages 187–201. Springer, 2016.

[12] Carlos Arango, Pablo Cortés, Jesús Muñuzuri, and Luis Onieva. Berth allocation planning in seville inland port by simulation and optimisation. *Advanced Engineering Informatics*, 25(3):452–461, 2011.

[13] Anas Arram, Mohd Zakree Ahmad Nazri, Masri Ayob, and Ahmad Abunadi. Bird mating optimizer for discrete berth allocation problem. In *2015 International Conference on Electrical Engineering and Informatics (ICEEI)*, pages 450–455. IEEE, 2015.

[14] Sheraz Aslam, Michalis P Michaelides, and Herodotos Herodotou. Internet of ships: A survey on architectures, emerging applications, and challenges. *IEEE Internet of Things journal*, 7(10):9714–9727, 2020.

[15] Sheraz Aslam, Michalis P Michaelides, and Herodotos Herodotou. Dynamic and continuous berth allocation using cuckoo search optimization. In *7th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS)*, pages 72–81, 2021.

[16] Sheraz Aslam, Michalis P Michaelides, and Herodotos Herodotou. Dynamic and continuous berth allocation using cuckoo search optimization. In *Proceedings of the 7th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS)*, pages 72–81, 2021.

[17] Sheraz Aslam, Michalis P Michaelides, and Herodotos Herodotou. Enhanced berth allocation using the cuckoo search algorithm. *SN Computer Science*, 3(4):1–15, 2022.

[18] Sheraz Aslam, Michalis P Michaelides, and Herodotos Herodotou. Optimizing multi-quay berth allocation using the cuckoo search algorithm. In *Proceedings of*

the 8th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS), pages 124–133, 2022.

[19] Lajjam Azza et al. Solving the hybrid berth allocation problem using a bat-inspired algorithm. In *2018 4th International Conference on Optimization and Applications (ICOA)*, pages 1–6. IEEE, 2018.

[20] Eduardo Tadeu Bacalhau, Luciana Casacio, and Anibal Tavares de Azevedo. New hybrid genetic algorithms to solve dynamic berth allocation problem. *Expert Systems with Applications*, page 114198, 2020.

[21] Eduardo Tadeu Bacalhau, Luciana Casacio, and Anibal Tavares de Azevedo. New hybrid genetic algorithms to solve dynamic berth allocation problem. *Expert Systems with Applications*, 167:114198, 2021.

[22] Saeed Behjati and Nasim Nahavandi. A mathematical model and grouping imperialist competitive algorithm for integrated quay crane and yard truck scheduling problem with non-crossing constraint. *IJE TRANSACTIONS A: Basics*, 32(10), 2019.

[23] Christian Bierwirth and Frank Meisel. A follow-up survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, 244(3):675–689, 2015.

[24] Bierwirth, Christian and Meisel, Frank. A survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, 202(3):615–627, 2010.

[25] Hamza Bouzekri, Gülgün Alpan, and Vincent Giard. A dynamic hybrid berth allocation problem with routing constraints in bulk ports. In *IFIP International Conference on Advances in Production Management Systems*, pages 250–258. Springer, 2020.

[26] Hamza Bouzekri, Gülgün Alpan, and Vincent Giard. A Dynamic Hybrid Berth Allocation Problem with Routing Constraints in Bulk Ports: Data sets and results. *Mendeley Data*, December 2020. doi:10.17632/bzmgbjttv3.2.

[27] Nils Boysen, Dirk Briskorn, and Frank Meisel. A generalized classification scheme for crane scheduling with interference. *European Journal of Operational Research*, 258(1):343–357, 2017.

[28] RT Cahyono, EJ Flonk, and B Jayawardhana. Dynamic berth and quay crane allocation for multiple berth positions and quay cranes. In *2015 European Control Conference (ECC)*, pages 3262–3267. IEEE, 2015.

[29] Rully Tri Cahyono, Engel Jacob Flonk, and Bayu Jayawardhana. Discrete-event systems modeling and the model predictive allocation algorithm for integrated berth and quay crane allocation. *IEEE Transactions on Intelligent Transportation Systems*, 21(3):1321–1331, 2019.

[30] Vessels arrivals at China Ports, May 2019. https://www.cargosmart.ai/en/blog/vessels-arrive-at-top-china-ports-with-shorter-delays-in-2019/.

[31] Héctor J Carlo, Iris FA Vis, and Kees Jan Roodbergen. Seaside operations in container terminals: literature overview, trends, and research directions. *Flexible Services and Manufacturing Journal*, 27(2-3):224–262, 2015.

[32] Daofang Chang, Zuhua Jiang, Wei Yan, and Junliang He. Integrating berth allocation and quay crane assignments. *Transportation Research Part E: Logistics and Transportation Review*, 46(6):975–990, 2010.

[33] Jiang Hang Chen, Der-Horng Lee, and Mark Goh. An effective mathematical formulation for the unidirectional cluster-based quay crane scheduling problem. *European Journal of Operational Research*, 232(1):198–208, 2014.

[34] Leilei Chen and Youfang Huang. A dynamic continuous berth allocation method based on genetic algorithm. In *2017 3rd IEEE International Conference on Control Science and Systems Engineering (ICCSSE)*, pages 770–773. IEEE, 2017.

[35] Sung Won Cho, Hyun Ji Park, and Chulung Lee. An integrated method for berth allocation and quay crane assignment to allow for reassignment of vessels to other terminals. *Maritime Economics & Logistics*, pages 1–31, 2020.

[36] Jean-François Cordeau, Gilbert Laporte, Pasquale Legato, and Luigi Moccia. Models and tabu search heuristics for the berth-allocation problem. *Transportation science*, 39(4):526–538, 2005.

[37] Juan F Correcher, Ramon Alvarez-Valdes, and Jose M Tamarit. New exact methods for the time-invariant berth allocation and quay crane assignment problem. *European Journal of Operational Research*, 275(1):80–92, 2019.

[38] Juan Francisco Correcher and Ramon Alvarez-Valdes. A biased random-key genetic algorithm for the time-invariant berth allocation and quay crane assignment problem. *Expert Systems with Applications*, 89:112–128, 2017.

[39] Juan Francisco Correcher, Thomas Van den Bossche, Ramon Alvarez-Valdes, and Greet Vanden Berghe. The berth allocation problem in terminals with irregular layouts. *European Journal of Operational Research*, 272(3):1096–1108, 2019.

[40] Ali Dadashi, Maxim A Dulebenets, Mihalis M Golias, and Abdolreza Sheikholeslami. A novel continuous berth scheduling model at multiple marine container terminals with tidal considerations. *Maritime Business Review*, 2017.

[41] Tatjana Davidović, Nataša Kovač, and Zorica Stanimirović. Vns-based approach to minimum cost hybrid berth allocation problem. In *Proc. XLII International Symposium on Operations Research, SYMOPIS*, pages 237–240, 2015.

[42] Arijit De, Saurabh Pratap, Akhilesh Kumar, and MK Tiwari. A hybrid dynamic berth allocation planning problem with fuel costs considerations for container terminal port using chemical reaction optimization approach. *Annals of Operations Research*, 290(1):783–811, 2020.

[43] Maxim A Dulebenets. Application of evolutionary computation for berth scheduling at marine container terminals: Parameter tuning versus parameter control. *IEEE Transactions on Intelligent Transportation Systems*, 19(1):25–37, 2017.

[44] Maxim A Dulebenets. An adaptive island evolutionary algorithm for the berth scheduling problem. *Memetic Computing*, 12(1):51–72, 2020.

[45] Maxim A Dulebenets, Mihalis M Golias, and Sabya Mishra. A collaborative agreement for berth allocation under excessive demand. *Engineering Applications of Artificial Intelligence*, 69:76–92, 2018.

[46] Maxim A Dulebenets, Masoud Kavoosi, Olumide Abioye, and Junayed Pasha. A self-adaptive evolutionary algorithm for the berth scheduling problem: towards efficient parameter control. *Algorithms*, 11(7):100, 2018.

[47] Issam El Hammouti, Azza Lajjam, Mohamed El Merouani, and Yassine Tabaa. An evolutionary algorithm approach to solve the hybrid berth allocation problem. In *Embedded Systems and Artificial Intelligence*, pages 339–347. Springer, 2020.

[48] Simon Emde. Optimally scheduling interfering and non-interfering cranes. *Naval Research Logistics (NRL)*, 64(6):476–489, 2017.

[49] Andreas T Ernst, Ceyda Oğuz, Gaurav Singh, and Gita Taherkhani. Mathematical models for the berth allocation problem in dry bulk terminals. *Journal of Scheduling*, 20(5):459–473, 2017.

[50] Houming Fan, Xiaoxue Ren, Zhenfeng Guo, and Yang Li. Truck scheduling problem considering carbon emissions under truck appointment system. *Sustainability*, 11(22):6256, 2019.

[51] Pablo Frojan, Juan Francisco Correcher, Ramon Alvarez-Valdes, Gerasimos Koulouris, and Jose Manuel Tamarit. The continuous berth allocation problem in a container terminal with multiple quays. *Expert Systems with Applications*, 42(21):7356–7366, 2015.

[52] Gary Froyland, Thorsten Koch, Nicole Megow, Emily Duane, and Howard Wren. Optimizing the landside operation of a container terminal. *OR spectrum*, 30(1):53–75, 2008.

[53] Bart Gadeyne and Pieter Verhamme. Optimizing maritime container terminal operations. *Ghent University, Ghent, Belgium*, 2011.

[54] Amir Hossein Gandomi, Xin-She Yang, and Amir Hossein Alavi. Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Engineering with computers*, 29(1):17–35, 2013.

[55] SR Seyedalizadeh Ganji, A Babazadeh, and N Arabshahi. Analysis of the continuous berth allocation problem in container ports using a genetic algorithm. *Journal of marine science and technology*, 15(4):408–416, 2010.

[56] Giovanni Giallombardo, Luigi Moccia, Matteo Salani, and Ilaria Vacca. Modeling and solving the tactical berth allocation problem. *Transportation Research Part B: Methodological*, 44(2):232–245, 2010.

[57] Mihalis M Golias, Georgios K Saharidis, Maria Boile, Sotirios Theofanis, and Marianthi G Ierapetritou. The berth allocation problem: Optimizing vessel arrival time. *Maritime Economics & Logistics*, 11(4):358–377, 2009.

[58] Neven Grubisic, Tomislav Krljan, and Livia Maglic. The optimization process for seaside operations at medium-sized container terminals with a multi-quay layout. *Journal of Marine Science and Engineering*, 8(11):891, 2020.

[59] Yongpei Guan and Raymond K Cheung. The berth allocation problem: models and solution methods. *Or Spectrum*, 26(1):75–92, 2004.

[60] Liming Guo, Jun Wang, and Jianfeng Zheng. Berth allocation problem with uncertain vessel handling times considering weather conditions. *Computers & Industrial Engineering*, 158:107417, 2021.

[61] Flabio Gutierrez, Edwar Lujan, Rafael Asmat, and Edmundo Vergara. Fully fuzzy linear programming model for the berth allocation problem with two quays. In *Uncertainty Management with Fuzzy and Rough Sets*, pages 87–113. Springer, 2019.

[62] XiaoLong Han, Xing Gong, and Jungbok Jo. A new continuous berth allocation and quay crane assignment model in container terminal. *Computers & Industrial Engineering*, 89:15–22, 2015.

[63] John Henry Holland et al. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[64] Hsien-Pin Hsu and Tai-Lin Chiang. An improved shuffled frog-leaping algorithm for solving the dynamic and continuous berth allocation problem (dcbap). *Applied Sciences*, 9(21):4682, 2019.

[65] Hsien-Pin Hsu, Tai-Lin Chiang, Chia-Nan Wang, Hsin-Pin Fu, and Chien-Chang Chou. A hybrid ga with variable quay crane assignment for solving berth allocation problem and quay crane assignment problem simultaneously. *Sustainability*, 11(7):2018, 2019.

[66] Hongtao Hu, Xiazhong Chen, Tingsong Wang, and Ye Zhang. A three-stage decomposition method for the joint vehicle dispatching and storage allocation problem in automated container terminals. *Computers & Industrial Engineering*, 129:90–101, 2019.

[67] Qu Hu, Xiaojie Luan, Francesco Corman, and Gabriel Lodewijks. A tabu search algorithm for inter-terminal container transport. *IFAC-PapersOnLine*, 49(3):413–418, 2016.

[68] Xiaoyuan Hu, Jianquan Guo, and Yue Zhang. Optimal strategies for the yard truck scheduling in container terminal with the consideration of container clusters. *Computers & Industrial Engineering*, 137:106083, 2019.

[69] Zhi-Hua Hu. Multi-objective genetic algorithm for berth allocation problem considering daytime preference. *Computers & Industrial Engineering*, 89:2–14, 2015.

[70] Akio Imai, Hsieh Chia Chen, Etsuko Nishimura, and Stratos Papadimitriou. The simultaneous berth and quay crane allocation problem. *Transportation Research Part E: Logistics and Transportation Review*, 44(5):900–920, 2008.

[71] Akio Imai, Etsuko Nishimura, Masahiro Hattori, and Stratos Papadimitriou. Berth allocation at indented berths for mega-containerships. *European Journal of Operational Research*, 179(2):579–593, 2007.

[72] Akio Imai, Etsuko Nishimura, and Stratos Papadimitriou. The dynamic berth allocation problem for a container port. *Transportation Research Part B: Methodological*, 35(4):401–417, 2001.

[73] Akio Imai, Etsuko Nishimura, and Stratos Papadimitriou. Berthing ships at a multi-user container terminal with a limited quay capacity. *Transportation Research Part E: Logistics and Transportation Review*, 44(1):136–151, 2008.

[74] Çağatay Iris, Dario Pacino, and Stefan Ropke. Improved formulations and an adaptive large neighborhood search heuristic for the integrated berth allocation and quay crane assignment problem. *Transportation Research Part E: Logistics and Transportation Review*, 105:123–147, 2017.

[75] Çağatay Iris, Dario Pacino, Stefan Ropke, and Allan Larsen. Integrated berth allocation and quay crane assignment problem: Set partitioning models and computational results. *Transportation Research Part E: Logistics and Transportation Review*, 81:75–97, 2015.

[76] Bin Ji, Xiaohui Yuan, and Yanbin Yuan. Modified nsga-ii for solving continuous berth allocation problem: Using multiobjective constraint-handling strategy. *IEEE transactions on cybernetics*, 47(9):2885–2895, 2017.

[77] Sourabh Katoch, Sumit Singh Chauhan, and Vijay Kumar. A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, 80(5):8091–8126, 2021.

[78] M Kavoosi, MA Dulebenets, O Abioye, J Pasha, O Theophilus, H Wang, R Kampmann, and M Mikijeljević. Berth scheduling at marine container terminals: a universal island-based metaheuristic approach. *Maritime Business Review*, pages 1–37, 2019.

[79] Masoud Kavoosi. *Innovative Metaheuristic Algorithms for Efficient Berth Scheduling at Marine Container Terminals*. PhD thesis, Florida State University, 2019.

[80] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-International Conference on Neural Networks*, volume 4, pages 1942–1948. IEEE, 1995.

[81] Kap Hwan Kim and Kyung Chan Moon. Berth scheduling by simulated annealing. *Transportation Research Part B: Methodological*, 37(6):541–560, 2003.

[82] Kap Hwan Kim and Young-Man Park. A crane scheduling method for port container terminals. *European Journal of operational research*, 156(3):752–768, 2004.

[83] Damla Kizilay, Deniz Türsel Eliiyi, et al. A comprehensive review of quay crane scheduling, yard operations and integrations thereof in container terminals. *Flexible Services and Manufacturing Journal*, pages 1–42, 2020.

[84] Stevan Kordić, Tatjana Davidović, Nataša Kovač, and Branislav Dragović. Combinatorial approach to exactly solving discrete and hybrid berth allocation problem. *Applied Mathematical Modelling*, 40(21-22):8952–8973, 2016.

[85] Sanae Kouismi. A discrete time heuristic for storage and scheduling unloading operations in container terminal under capacity and non-interference constraints. *International Journal of Applied Engineering Research*, 12(24):15018–15026, 2017.

[86] Nataša Kovač, Tatjana Davidović, and Zorica Stanimirović. Evolutionary algorithm for the minimum cost hybrid berth allocation problem. In *2015 6th International Conference on Information, Intelligence, Systems and Applications (IISA)*, pages 1–6. IEEE, 2015.

[87] Nataša Kovač, Zorica Stanimirović, and Tatjana Davidović. Metaheuristic approaches for the minimum cost hybrid berth allocation problem. In *Modeling, computing and data handling methodologies for maritime transportation*, pages 1–47. Springer, 2018.

[88] Issam Krimi, Raca Todosijević, Rachid Benmansour, Mustapha Ratli, Abdessamad Ait El Cadi, and Afaf Aloullal. Modelling and solving the multi-quays berth allocation and crane assignment problem with availability constraints. *Journal of Global Optimization*, pages 1–25, 2020.

[89] Eduardo Lalla-Ruiz, Christopher Expósito-Izquierdo, Jesica De Armas, Belén Melián-Batista, and J Marcos Moreno-Vega. Migrating birds optimization for the seaside problems at maritime container terminals. *Journal of Applied Mathematics*, 2015, 2015.

[90] Eduardo Lalla-Ruiz, Christopher Expósito-Izquierdo, Belén Melián-Batista, and J Marcos Moreno-Vega. A set-partitioning-based model for the berth allocation problem under time-dependent limitations. *European Journal of Operational Research*, 250(3):1001–1012, 2016.

[91] Eduardo Lalla-Ruiz, BeléN MeliáN-Batista, and J Marcos Moreno-Vega. Artificial intelligence hybrid heuristic based on tabu search for the dynamic berth allocation problem. *Engineering Applications of Artificial Intelligence*, 25(6):1132–1141, 2012.

[92] Eduardo Lalla-Ruiz and Stefan Voss. Popmusic as a matheuristic for the berth allocation problem. *Annals of Mathematics and Artificial Intelligence*, 76(1-2):173–189, 2016.

[93] Der-Horng Lee, Jiang Hang Chen, and Jin Xin Cao. The continuous berth allocation problem: A greedy randomized adaptive search solution. *Transportation Research Part E: Logistics and Transportation Review*, 46(6):1017–1029, 2010.

[94] Ming-Wei Li, Wei-Chiang Hong, Jing Geng, and Jianlun Wang. Berth and quay crane coordinated scheduling using multi-objective chaos cloud particle swarm optimization algorithm. *Neural computing and Applications*, 28(11):3163–3182, 2017.

[95] Yanjing Li, Xiaoning Zhu, Li Wang, and Xi Chen. Stowage plan based slot optimal allocation in rail-water container terminal. *Journal of Control Science and Engineering*, 2017, 2017.

[96] Ying Li, Feng Chu, Feifeng Zheng, and Imed Kacem. Integrated berth allocation and quay crane assignment with uncertain maintenance activities. In *2019 International Conference on Industrial Engineering and Systems Management (IESM)*, pages 01–06. IEEE, 2019.

[97] Ying Li, Feng Chu, Feifeng Zheng, and Ming Liu. A bi-objective optimization for integrated berth allocation and quay crane assignment with preventive maintenance activities. *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[98] Chengji Liang, Jianquan Guo, and Yang Yang. Multi-objective hybrid genetic algorithm for quay crane dynamic assignment in berth allocation planning. *Journal of Intelligent Manufacturing*, 22(3):471–479, 2011.

[99] Shih-Wei Lin, Ching-Jung Ting, and Kun-Chih Wu. Simulated annealing with different vessel assignment strategies for the continuous berth allocation problem. *Flexible Services and Manufacturing Journal*, 30(4):740–763, 2018.

[100] Mikael Lind, MP Michaelides, R Ward, H Herodotou, and RT Watson. Boosting data-sharing to improve short sea shipping performance: Evidence from limassol port calls analysis. Technical report, Article No. 35 [UNCTAD Transport and Trade Facilitation Newsletter No. 82 - Second Quarter 2019], 2019. https://unctad.org/en/pages/newsdetails.aspx?OriginalVersionID=2102.

[101] Aijun Liu, Haiyang Liu, Sang-Bing Tsai, Hui Lu, Xiao Zhang, and Jiangtao Wang. Using a hybrid model on joint scheduling of berths and quay cranes—from a sustainable perspective. *Sustainability*, 10(6):1959, 2018.

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

[103] Zhi-qiang Lu, Li-feng Xi, et al. A proactive approach for simultaneous berth and quay crane scheduling problem with stochastic arrival and handling time. *European Journal of Operational Research*, 207(3):1327–1340, 2010.

[104] Zhiqiang Lu, Xiaole Han, Lifeng Xi, and Alan L Erera. A heuristic for the quay crane scheduling problem based on contiguous bay crane operations. *Computers & Operations Research*, 39(12):2915–2928, 2012.

[105] Edwar Lujan, Edmundo Vergara, Jose Rodriguez-Melquiades, Miguel Jiménez-Carrión, Carlos Sabino-Escobar, and Flabio Gutierrez. A fuzzy optimization model for the berth allocation problem and quay crane allocation problem (bap+ qcap) with n quays. *Journal of Marine Science and Engineering*, 9(2):152, 2021.

[106] Jiabin Luo, Yue Wu, and André Bergsten Mendes. Modelling of integrated vehicle scheduling and container storage problems in unloading process at an automated container terminal. *Computers & Industrial Engineering*, 94:32–44, 2016.

[107] Tian Luo, Daofang Chang, and Yinping Gao. Optimization of gantry crane scheduling in container sea-rail intermodal transport yard. *Mathematical Problems in Engineering*, 2018, 2018.

[108] Amirsalar Malekahmadi, Mehdi Alinaghian, Seyed Reza Hejazi, and Mohammad Ali Assl Saidipour. Integrated continuous berth allocation and quay crane assignment and scheduling problem with time-dependent physical constraints in container terminals. *Computers & Industrial Engineering*, page 106672, 2020.

[109] Geraldo Regis Mauri, Glaydston Mattos Ribeiro, Luiz Antonio Nogueira Lorena, and Gilbert Laporte. An adaptive large neighborhood search for the discrete and continuous berth allocation problem. *Computers & Operations Research*, 70:140–154, 2016.

[110] Frank Meisel and Christian Bierwirth. Heuristics for the integration of crane productivity in the berth allocation problem. *Transportation Research Part E: Logistics and Transportation Review*, 45(1):196–209, 2009.

[111] Xia Mengjue, Zhao Ning, and Mi Weijian. Storage allocation in automated container terminals: the upper level. *Polish Maritime Research*, 23(s1):160–174, 2016.

[112] Michalis P Michaelides, Herodotos Herodotou, Mikael Lind, and Richard T Watson. Port-2-port communication enhancing short sea shipping performance: The case study of cyprus and the eastern mediterranean. *Sustainability*, 11(7):1912, 2019.

[113] Mohammad Mohammadi and Kamran Forghani. Solving a stochastic berth allocation problem using a hybrid sequence pair-based simulated annealing algorithm. *Engineering Optimization*, 2018.

[114] Tatsushi Nishi, Tatsuya Okura, Eduardo Lalla-Ruiz, and Stefan Voß. A dynamic programming-based matheuristic for the dynamic berth allocation problem. *Annals of operations research*, 286(1):391–410, 2020.

[115] KT Park and Kap Hwan Kim. Berth scheduling for container terminals by using a sub-gradient optimization technique. *Journal of the operational research society*, 53(9):1054–1062, 2002.

[116] Young-Man Park and Kap Hwan Kim. A scheduling method for berth and quay cranes. In *Container terminals and automated transport systems*, pages 159–181. Springer, 2005.

[117] Boris Pérez-Cañedo, José Luis Verdegay, Alejandro Rosete, and Eduardo René Concepción-Morales. Fully fuzzy multi-objective berth allocation problem. In *International Conference on Hybrid Artificial Intelligence Systems*, pages 261–272. Springer, 2020.

[118] Mai-Ha Phan and Kap Hwan Kim. Collaborative truck scheduling and appointments for trucking companies and container terminals. *Transportation Research Part B: Methodological*, 86:37–50, 2016.

[119] Andrey Popov. Genetic algorithms for optimization. *User Manual, Hamburg*, 2013, 2005.

[120] Ines Rekik and Sabeur Elkosantini. A multi agent system for the online container stacking in seaport terminals. *Journal of Computational Science*, 35:12–24, 2019.

[121] Ines Rekik, Sabeur Elkosantini, and Habib Chabchoub. A case based heuristic for container stacking in seaport terminals. *Advanced Engineering Informatics*, 38:658–669, 2018.

[122] Filipe Rodrigues and Agostinho Agra. Berth allocation and quay crane assignment/scheduling problem under uncertainty: a survey. *European Journal of Operational Research*, 2022.

[123] Debjit Roy, Jan-Kees van Ommeren, MBM de Koster, and Amir Gharehgozli. Effect of service priority and resource synchronization choices on landside terminal queues: Exact analysis and approximations. *Social Science Research Network (SSRN)*, 2020.

[124] Nasser R Sabar, Siang Yew Chong, and Graham Kendall. A hybrid differential evolution algorithm–game theory for the berth allocation problem. In *Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems-Volume 2*, pages 77–87. Springer, 2015.

[125] Cenk Şahin and Yusuf Kuvvetli. Differential evolution based meta-heuristic algorithm for dynamic continuous berth allocation problem. *Applied Mathematical Modelling*, 40(23-24):10679–10688, 2016.

[126] Abdellah Salhi, Ghazwan Alsoufi, and Xinan Yang. An evolutionary approach to a combined mixed integer programming model of seaside operations as arise in container ports. *Annals of operations research*, 272(1-2):69–98, 2019.

[127] Sarangthem Sanajaoba and Eugene Fernandez. Maiden application of cuckoo search algorithm for optimal sizing of a remote hybrid renewable energy system. *Renewable energy*, 96:1–10, 2016.

[128] Xavier Schepler, Nabil Absi, Dominique Feillet, and Eric Sanlaville. The stochastic discrete berth allocation problem. *EURO Journal on Transportation and Logistics*, 8(4):363–396, 2019.

[129] Mei Sha, Tao Zhang, Ying Lan, Xin Zhou, Tianbao Qin, Dayong Yu, and Kai Chen. Scheduling optimization of yard cranes with minimal energy consumption at container terminals. *Computers & Industrial Engineering*, 113:704–713, 2017.

[130] Abdolreza Sheikholeslami, Mehdi Mardani, Ehsan Ayazi, and Hesam Arefkhani. A dynamic and discrete berth allocation problem in container terminals considering tide effects. *Iranian Journal of Science and Technology, Transactions of Civil Engineering*, 44(1):369–376, 2020.

[131] Ahmed Simrin and Ali Diabat. The dynamic berth allocation problem: A linearized formulation. *RAIRO-Operations Research*, 49(3):473–494, 2015.

[132] Yujian Song, Jiantong Zhang, Ming Liu, and Chengbin Chu. The berth allocation optimisation with the consideration of time-varying water depths. *International Journal of Production Research*, 57(2):488–516, 2019.

[133] Francesc Soriguera, Domenec Espinet, and Francesc Robusté. Optimization of internal transport cycle in a marine container terminal managed by straddle carriers. *Transportation research record*, 2033(1):21–30, 2007.

[134] Gokcecicek Tasoglu and Gokalp Yildiz. Simulated annealing based simulation optimization method for solving integrated berth allocation and quay crane scheduling problems. *Simulation Modelling Practice and Theory*, 97:101948, 2019.

[135] Emmanouil Thanos, Tulio Toffolo, Haroldo Gambini Santos, Wim Vancroonenburg, and Greet Vanden Berghe. The tactical berth allocation problem with time-variant specific quay crane assignments. *Computers & Industrial Engineering*, 155:107168, 2021.

[136] Sotirios Theofanis, Maria Boile, and Mihalis Golias. An optimization based genetic algorithm heuristic for the berth allocation problem. In *2007 IEEE Congress on Evolutionary Computation*, pages 4439–4445. IEEE, 2007.

[137] Ching-Jung Ting, Kun-Chih Wu, and Hao Chou. Particle swarm optimization algorithm for the berth allocation problem. *Expert Systems with Applications*, 41(4):1543–1550, 2014.

[138] Mohammad Torkjazi, Nathan Huynh, and Samaneh Shiri. Truck appointment systems considering impact to drayage truck tours. *Transportation Research Part E: Logistics and Transportation Review*, 116:208–228, 2018.

[139] Nitish Umang, Michel Bierlaire, and Ilaria Vacca. Exact and heuristic methods to solve the berth allocation problem in bulk ports. *Transportation Research Part E: Logistics and Transportation Review*, 54:14–31, 2013.

[140] UNCTAD: Ports in 2017, June 2017. https://unctad.org/system/files/official-document/rmt2018ch4_en.pdf.

[141] Global container throughput 2012-2021, December 2020. https://www.statista.com/statistics/913398/container-throughput-worldwide/.

[142] Iris FA Vis and Rene De Koster. Transshipment of containers at a container terminal: An overview. *European journal of operational research*, 147(1):1–16, 2003.

[143] Andre Voß, Gabriel Guckenbiehl, Holger Schütt, and Tobias Buer. An online storage strategy with dynamic bay reservations for container terminals. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8. IEEE, 2016.

[144] Ran Wang, Trung Thanh Nguyen, Changhe Li, Ian Jenkinson, Zaili Yang, and Shayan Kavakeb. Optimising discrete dynamic berth allocations in seaports using a levy flight based meta-heuristic. *Swarm and evolutionary computation*, 44:1003–1017, 2019.

[145] Tingsong Wang, Xinchang Wang, and Qiang Meng. Joint berth allocation and quay crane assignment under different carbon taxation policies. *Transportation Research Part B: Methodological*, 117:18–36, 2018.

[146] Xi Xiang and Changchun Liu. An almost robust optimization model for integrated berth allocation and quay crane assignment problem. *Omega*, page 102455, 2021.

[147] Xi Xiang and Changchun Liu. An expanded robust optimisation approach for the berth allocation problem considering uncertain operation time. *Omega*, 103:102444, 2021.

[148] Xi Xiang, Changchun Liu, and Lixin Miao. A bi-objective robust model for berth allocation scheduling under uncertainty. *Transportation Research Part E: Logistics and Transportation Review*, 106:294–319, 2017.

[149] Xi Xiang, Changchun Liu, and Lixin Miao. Reactive strategy for discrete berth allocation and quay crane assignment problems under uncertainty. *Computers & Industrial Engineering*, 126:196–216, 2018.

[150] Dongsheng Xu, Chung-Lun Li, and Joseph Y-T Leung. Berth allocation with time-dependent physical limitations on vessels. *European Journal of Operational Research*, 216(1):47–56, 2012.

[151] Ya Xu, Kelei Xue, and Yuquan Du. Berth scheduling problem considering traffic limitations in the navigation channel. *Sustainability*, 10(12):4795, 2018.

[152] Xin-She Yang and Suash Deb. Cuckoo Search via Lévy Flights. In *World Congress on Nature & Biologically Inspired Computing (NaBIC)*, pages 210–214. IEEE, 2009.

[153] Xin-She Yang and Suash Deb. Cuckoo search: recent advances and applications. *Neural Computing and applications*, 24(1):169–174, 2014.

[154] Sanghyuk Yi, Bernd Scholz-Reiter, Taehoon Kim, and Kap Hwan Kim. Scheduling appointments for container truck arrivals considering their effects on congestion. *Flexible Services and Manufacturing Journal*, 31(3):730–762, 2019.

[155] Dayong Yu, Dong Li, Mei Sha, and Dali Zhang. Carbon-efficient deployment of electric rubber-tyred gantry cranes in container terminals with workload uncertainty. *European Journal of Operational Research*, 275(2):552–569, 2019.

[156] Tian Yu, Zhou Qiang, and Zhu Benfei. A genetic algorithm based on spatiotemporal conflict between continuous berth-allocation and time-varying specific crane assignment. *Engineering Optimization*, 51(3):390–411, 2019.

[157] Wang Yuping, Hao Yangyang, Zhang Yuanhui, Huang Youfang, and Gu Tianyi. Berth allocation optimization with priority based on simulated annealing algorithm. *Journal of Engineering Science & Technology Review*, 11(1), 2018.

[158] Radhia Zaghdoud, Khaled Mesghouni, Simon Collart Dutilleul, Kamel Zidi, and Khaled Ghedira. A hybrid method for assigning containers to agvs in container terminal. *IFAC-PapersOnLine*, 49(3):96–103, 2016.

[159] Qingcheng Zeng, Yuanjun Feng, and Zhongzhen Yang. Integrated optimization of pickup sequence and container rehandling based on partial truck arrival information. *Computers & Industrial Engineering*, 127:366–382, 2019.

[160] Rui Zhang, Zhihong Jin, Yu Ma, and Weixin Luan. Optimization for two-stage double-cycle operations in container terminals. *Computers & Industrial Engineering*, 83:316–326, 2015.

[161] Xiaoju Zhang, Qingcheng Zeng, and Wenhao Chen. Optimization model for truck appointment in container terminals. *Procedia-Social and Behavioral Sciences*, 96:1938–1947, 2013.

[162] Xiaoju Zhang, Qingcheng Zeng, and Zhongzhen Yang. Optimization of truck appointments in container terminals. *Maritime Economics & Logistics*, 21(1):125–145, 2019.

[163] Feifeng Zheng, Ying Li, Feng Chu, Ming Liu, and Yinfeng Xu. Integrated berth allocation and quay crane assignment with maintenance activities. *International Journal of Production Research*, 57(11):3478–3503, 2019.