Journal of
Marine Science and Engineering

MDPI

Article

Optimizing Multi-Quay Combined Berth and Quay Crane Allocation Using Computational Intelligence

Sheraz Aslam, **Michalis P. Michaelides** and **Herodotos Herodotou** *

Department of Electrical Engineering, Computer Engineering and Informatics, Cyprus University of Technology, Limassol 3036, Cyprus; sheraz.aslam@cut.ac.cy (S.A.); michalis.michaelides@cut.ac.cy (M.P.M.)
* Correspondence: herodotos.herodotou@cut.ac.cy

Abstract: The significant increase in international seaborne trade volumes over the last several years is pushing port operators to improve the efficiency of terminal processes and reduce vessel turnaround time. Toward this direction, this study investigates and solves the combined berth allocation problem (BAP) and quay crane allocation problem (QCAP) in a multi-quay (MQ) setting using computational intelligence (CI) approaches. First, the study develops a mathematical model representing a real port environment and then adapts the cuckoo search algorithm (CSA) for the first time in this setup. The CSA is inspired by nature by following the basic rules of breeding parasitism of some cuckoo species that lay eggs in other birds' nests. For comparison purposes, we implement two baseline approaches, first come first serve and exact MILP, and two CI approaches, particle swarm optimization (PSO) and genetic algorithm (GA), that are typically used to solve such complex or NP-hard problems. Performance assessment is carried out via a comprehensive series of experiments using real-world data. Experimental findings show that the MILP method can address the problems only when a small dataset is employed. In contrast, the newly adapted CSA can solve larger instances of MQ BAP and QCAP within significantly reduced computation times.

Keywords: berth allocation problem; quay crane allocation problem; multi-quay BAP and QCAP; smart ports; computational intelligence methods; cuckoo search algorithm

check for updates

Citation: Aslam, S.; Michaelides, M.P.; Herodotou, H. Optimizing Multi-Quay Combined Berth and Quay Crane Allocation Using Computational Intelligence. *J. Mar. Sci. Eng.* **2024**, *12*, 1567. https://doi.org/10.3390/jmse12091567

Academic Editor: Xinqiang Chen

Received: 30 July 2024
Revised: 26 August 2024
Accepted: 31 August 2024
Published: 6 September 2024

1. Introduction

Seaports play an essential role in the world economy as more than 80% of world trade is carried by sea [1]. With increased demand, congested ports are facing significant challenges related to resource scarcity and several scheduling activities, including the berth allocation problem (BAP) and the quay crane allocation problem (QCAP) [2]. In 2021, the global average container schedule delays doubled, while the median turnaround time for container ships increased by 13.7% [1]. In [3], the authors investigate several factors causing waiting times from both quantitative and qualitative perspectives. Similar problems have attracted considerable attention from academia and industry, highlighting the need to optimize maritime operations in ports [4].

Berths and quay cranes (QCs) are considered two of the primary resources of marine ports, and their efficient use can help reduce the total turnaround time of vessels [5]. The primary operations in ports are divided into three main areas: seaside, marshaling yard, and landside operations. The first involves loading and unloading containers onto/from vessels using quay cranes and other terminal resources. Inbound containers are stored in the marshaling yard. Finally, landside operations include activities related to dispatching containers to their destinations using trucks or trains [6]. A single or multiple berth lines are used to berth arriving vessels, and quay cranes (QCs) are used to perform loading and unloading operations. However, berths and QCs are bottleneck resources in ports due to the limited coastal environment and complexity of port activities [7]. Therefore, good planning and proper coordination between them can improve terminal productivity.

The BAP identifies berthing positions and berthing times for arriving vessels based on various factors, such as expected time of arrival (ETA), handling time or total load, requested time of departure (RTD), etc. In contrast, QCAP deals with the appropriate allocation of cranes based on the BAP solution and availability of cranes, since BAP and QCAP are interdependent problems [8]. In the current literature, there are many studies dealing with the stand-alone BAP [9–11]. 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 [12,13]. There are very few studies dealing with terminals with multi-quays (MQs), especially terminals with multi-purpose quays, e.g., container and passenger terminals [14–16]. MQs introduce an additional layer of complexity to BAP, involving the allocation of ships to quays in addition to assigning berth positions and berthing times to ships for each individual quay. In MQ terminal environments, the big challenges include sharing resources, placing vessels at the correct position of the desired quay, and coordinating among multiple quays. Considering MQs, especially multi-purpose quays, while solving the combined BAP and QCAP further increases the problem's complexity. For example, in a study presented in [14], a solution for MQs is proposed for BAP; however, in that 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 [15] also solves MQ 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, ref. [16] 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 study that addresses combined BAP and QCAP considering MQs, which employs fuzzy logic to solve the problem [17]. However, its authors conclude that their approach is feasible only for small instances and suggest the use of metaheuristics for solving larger problem instances. Furthermore, none of the existing studies consider terminal cooperation, i.e., the same-purpose terminals/quays (e.g., container terminals) can cooperate with each other by re-scheduling vessels from busy quays to idle quays to avoid congestion problems.

The aforementioned limitations of previous studies and the practical importance of the problem (i.e., multi-quay environment) motivate us to develop an efficient computational intelligence (CI) approach for solving the combined BAP and QCAP problem for a real port setting while considering multiple heterogeneous quays (e.g., container, general cargo, and passenger quays). This study focuses on finding the optimal berthing quay, position, and time along with the best assignment of cranes for each arriving vessel at the port. To the best of our knowledge, this is the first study that deals with MQ BAP and QCAP for multi-purpose terminals. The primary contributions are outlined below:

- Develop an MILP-based mathematical model for the MQ BAP and QCAP for minimizing the total service cost while considering multiple heterogeneous quays with real-world settings and constraints;
- Develop a cooperation model between quays, for the first time in an MQ BAP and QCAP setting, to share berthing positions with the fundamental objective of reducing congestion and total service costs at MCTs.
- Adapt and implement the cuckoo search algorithm, for the first time for this setup, as well as two other state-of-the-art CI methods, i.e., genetic algorithm and particle swarm optimization, for comparison purposes.
- Evaluate the effectiveness of the proposed approach against state-of-the-art CI methods and two baseline approaches on real data from the Port of Limassol, Cyprus.

An initial version of the mathematical model and preliminary results were presented in [18]. Compared to our prior study, this work presents (i) the complete mathematical model, including the quay crane encoding and spatiotemporal overlapping constraints; (ii) the adaptation of the GA, PSO, and FCFS algorithms for the MQ BAP and QCAP; and

(iii) new experimental analysis considering up to one month of real data as well as new synthetic data.

The remaining of the study is organized as follows: Section 2 presents the literature review. The problem description and mathematical formulation are discussed in Section 3. Section 4 discloses the developed computational intelligence methods. Section 5 shows the experimental setting and the results of the developed methods. Section 6 provides an in-depth discussion of the results along with managerial insights. Finally, Section 7 concludes the study.

2. Literature Review

This section reviews the most relevant recent studies that use CI approaches to solve the stand-alone BAP, the combined BAP and QCAP, as well as the MQ problems. A more in-depth literature review can be found in [6].

Study [19] solved the BAP with the objective of reducing the number of late departures at the port of Shahid Rajaee Shallow, Iran. To solve the problem, a GA-based solution was developed. The authors of [20,21] also addressed BAP with the goal of optimal berth allocation, taking into account the uncertainty of ship arrivals, and proposed heuristic-based solutions to solve the problem. Numerous studies have focused on the stand-alone BAP and suggested genetic algorithm (GA)-based solutions for optimizing berth allocation decisions, such as [11,19,20]. The authors of [11] proposed an exact method to solve stand-alone BAP with a continuous berthing layout. However, their model cannot solve the problem with large data instances due to its large time complexity. Then, a GA-based solution was developed to deal with large data instances. The work in [22] also solved the BAP with the objective of reducing the late departure of vessels by employing a hybrid of GA and the branch-and-cut (B&C) method, which assigned the best berthing location based on vessel arrival and departure times and other constraints. The authors of [23] developed a differential evolution (DE)-based solution to minimize the service costs of arriving ships at the terminal. A study [24] also solved the BAP, reducing the service time of docked ships using a greedy randomized adaptive search method (GRASP). The authors of [25,26] addressed the BAP to avoid delays in the departure of ships. The authors also considered tidal restrictions when assigning berths and developed datasets based on real-time port characteristics that can be used as benchmark instances in the future. Xu et al. developed a hybrid simulated annealing (SA)-based heuristic method while considering traffic constraints in a navigation channel [27] for improving the performance of maritime container terminals. The problem was solved using the hybrid SA (HSA) method, which combined SA and reheat treatment methods. To validate the proposed method, real-time instances of two container terminals in Tianjin, China, were used, and CPLEX and greedy methods were employed for a comparative study. More recently, [9] proposed a solution for the BAP using the cuckoo search algorithm (CSA) with the goal of reducing the total service cost, which includes handling costs, waiting costs, and various penalties, such as for late departure and non-optimal berth assignment.

For the combined (single-quay) BAP with QCAP, Li et al. proposed an exact solution to minimize the total turnaround time [28] while considering the maintenance activities at the quay. Another study also proposed an exact method for reducing the total turnaround time of arriving ships at Khalifa Port, Abu Dhabi [29]. The study [30] provided a GA-based solution for the same problem. The authors proposed three variants of GA and performed several experiments to compare those variants. The authors in [31] considered carbon emission policies to reduce total carbon emissions at the port along with the primary objectives of the study, i.e., reducing penalty costs and operating costs. To address this joint problem, a branch-and-bound algorithm was developed and tested on multiple real-time instances taken from [32]. Zheng et al. study the same problem while considering QC maintenance [2]. The problem was formulated as an integer linear programming model and then solved using the exact approach; however, the developed method could solve the problem considering up to only 18 vessels. Next, the study developed an improved GA and a new heuristic named left and right vessel move (LRVM) algorithm.

The aforementioned studies assume that the MCT consists of only one continuous quay, but it is unrealistic around the globe, where berthing is distributed across multiple distinct quays [14]. For instance, the Port of Limassol, Cyprus, features seven quays, as shown in Figure 1. The authors of [14] proposed a solution for an MQ stand-alone BAP, consisting of an MILP-based mathematical model and a GA-based solution. In another work, ref. [16] proposed a fuzzy-based solution but, as the authors acknowledge, the proposed method only provides an optimal solution when only up to 10 vessels are considered. A recent study [15] solved MQ-BAP using heuristics based on a general variable neighborhood search and concluded that the proposed method does not always provide an optimal solution and is sometimes 40% away from the optimal solution. Furthermore, in [33], the authors of the current study addressed the MQ-BAP for total cost reduction using CSA.

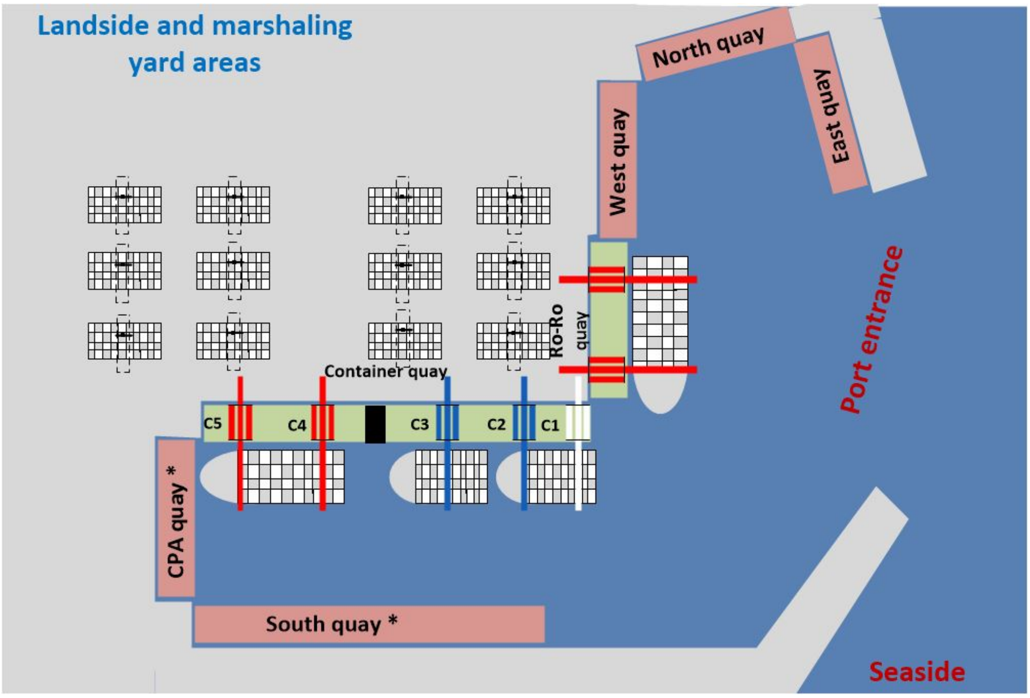


Figure 1. Port of Limassol structure showing its seven berthing quays. The Container and Ro-Ro Quay have 5 and 2 installed cranes, respectively. Note: quays with ∗ indicate that these are not used for commercial purposes.

As for the combined BAP with QCAP when considering MQs, the authors of [17] proposed a fuzzy-based solution. Based on experiments, they concluded that the proposed method can only solve small-scale problems (considering only two quays) and suggested that the medium and large-scale problems can only be solved using metaheuristic approaches. Therefore, motivated by the above discussion, we propose a solution for combined BAP and QCAP that considers multi-purpose MQs and solves a realistic scenario using CI-based methods.

3. Problem Description and Formulations

This section first describes the problem that is considered in this study, and then it discloses its mathematical formulation. The mathematical notation is listed on Table 1.

Table 1. Mathematical notations. Note that time is discretized into small intervals and categorical variables (e.g., vessels, quays) are mapped into unique integer values.

Name	Type	Explanation
Parameters		
ABQ_v	Set of Integers	Alternative (preferred) berthing quays of vessel v
AT_v	Integer	Expected arrival time of vessel v
C_v^h	Continuous	Handling cost per time interval for vessel v
C_v^w	Continuous	Waiting cost per time interval for vessel v
C_v^{ld}	Continuous	Late departure cost per time interval for vessel v
C_v^{nob}	Continuous	Penalty cost for non-optimal berthing position for vessel v
C_v^{noq}	Continuous	Penalty cost for non-optimal berthing quay for vessel v
c^{min}	Integer	Minimum berthing position served by crane c
c^{max}	Integer	Maximum berthing position served by crane c
HP_q^c	Continuous	Handling productivity of crane c located on quay q
L_q	Integer	Length of quay q
L_v	Integer	Length of vessel v
$Load_v$	Integer	Total load of vessel v
SC_q^c	Continuous	Service cost per time interval of crane c located on quay q
SD	Integer	Safety berthing distance between vessels
SE	Integer	Safety entrance time between vessels
ST	Integer	Safety berthing time between vessels
PBP_v	Integer	Preferred berthing position at the preferred berthing quay of vessel v
PBQ_v	Integer	Preferred berthing quay of vessel v
RDT_v	Integer	Requested departure time of vessel v
Decision Variables		
BP_v	Integer	Planned berthing position at the planned berthing quay of vessel v
BQ_v	Integer	Planned berthing quay of vessel v
BT_v	Integer	Planned berthing time of vessel v
k_v	Integer	Set of cranes assigned to vessel v (encoded in binary form)
x_{vqbkt}	Binary	1, if the vessel v is assigned at position b of quay q at time t to be served by cranes encoded in k, and 0 otherwise
Auxiliary Variables		
D_v	Integer	Deviation time for vessel v if it is berthed to a position other than PBP_v
FT_v	Integer	Finishing time of loading/unloading operations of vessel v
HT_v	Integer	Handling time of vessel v
LDT_v	Integer	Late departure time of vessel v
WT_v	Integer	Waiting time of vessel v
Sets and Indices		
V	Set of Integers	Set of arriving vessels; $v \in V$ a vessel
Q	Set of Integers	Set of berthing quays; $q \in Q$ a quay
$B(q)$	Set of Integers	Set of berth positions on $q \in Q$; $b \in B(q)$ a berth position
$C(q)$	Set of Integers	Set of quay cranes on quay $q \in Q$; $c \in C(q)$ a crane
$K(q)$	Set of Integers	Power set of cranes set $C(q)$; $k \in K(q)$ represents a subset of cranes from $C(q)$ encoded as an integer in a binary form
T	Set of Integers	Set of time intervals (planning horizon); $t \in T$ a time interval

3.1. Problem Explanation

The combined BAP and QCAP is an optimization problem in which the objective is to allocate available quays, berths, and quay cranes (QCs) across time to incoming ships to perform unloading/loading operations with the goal of minimizing the total service cost. In this study, we consider a realistic setup of a port with multiple heterogeneous quays and several cranes (with different productivity) available at each quay, which cannot move once they are assigned to a particular vessel. Notably, cranes cannot move from one quay to another because of different quay types and settings. However, cranes can move along the same quay, but each crane has specific, designated positions where it operates. For example, Crane 1 at a container quay can work within the range from 0 to 450 m. The assignment of cranes to a vessel also depends on the vessel's length and load. Moreover, all quays follow a continuous berthing 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 total service cost calculation of each vessel, this study considers waiting cost, handling cost, late departure cost, and non-optimal berthing cost.

3.2. MQ BAP and QCAP Formulation

The primary objective of the MQ BAP and QCAP is to allocate a berthing position at the preferred quay, a berthing time, and QCs to each arriving vessel in order to minimize the total service cost, which includes handling cost, waiting cost, and late departure penalty, as presented in the following cost function:

$$\begin{aligned} Cost\ (v, BQ_v, BP_v, k_v, BT_v) = HT_v \cdot [C_v^h + f(v, BQ_v, BP_v)] \\ + WT_v \cdot C_v^w + LDT_v \cdot C_v^{ld}. \end{aligned} \quad (1)$$

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 [33], does not consider handling time HT_v as an input, but it is calculated based on the total load on the vessel $Load_v$ and the handling productivity (per time interval) of each of the cranes (encoded in k_v) that are assigned to vessel v. Furthermore, if any vessel is moored to a location other than the preferred position or quay, a deviation time for vessel v is added to the total handling time [34], as described below:

$$HT_v = \left\lceil \frac{Load_v}{\sum_{c \in k_v} HP_q^c} \right\rceil + D_v, \quad \forall\ v \in V, q = BQ_v \quad (2)$$

There is a set of cranes $C(q) = \{c_1, c_2, c_3, \dots c_{n_q}\}$ 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, a large vessel with a heavy load can use multiple cranes, whereas a small vessel may use only one crane. All possible combinations of crane assignments to a particular vessel are present in the power set of $C(q)$:

$$\begin{aligned} K(q) = \Big\{ &\{\}, \{c_1\}, \{c_2\}, \dots, \{c_{n_q}\}, \\ &\{c_1, c_2\}, \{c_1, c_3\}, \dots, \{c_{n-1}, c_{n_q}\}, \\ &\{c_1, c_2, c_3\}, \dots, \{c_1, c_2, \dots, c_{n_q}\} \Big\} \end{aligned} \quad (3)$$

The set of cranes allocated to a vessel v is one element of the power set $K(q)$. To simplify the formulation, we encode each element of $K(q)$ as a number by using a binary representation. 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; if it is 0, the QC is not allocated. For example, assuming 4 cranes, the element $\{c_1\}$ is encoded as 0001, $\{c_2\}$ as 0010, $\{c_1, c_2\}$ as 0011, and so on. The number of digits used in the encoding equals the number n_q of available QCs on quay q. The empty set (with encoding 0000) is also included when a vessel is berthed to a terminal that does not have quay cranes, such as a passenger or general cargo quay. In summary, k_v is a number whose binary representation shows the set of cranes assigned to v and can take numbers between 0 and $2^{n_q} - 1$.

The handling cost (per time interval) C_v^h charged to a vessel depends on the service cost per time interval of each crane assigned to vessel v (per the set of cranes k_v), and it can be calculated as

$$C_v^h = \sum_{c \in k_v} SC_q^c, \quad \forall\ v \in V, q = BQ_v \quad (4)$$

The last part of the first term, namely $f(v, BQ_v, BP_v)$ in Equation (1), calculates penalty cost because of a possible non-optimal berthing (NOB) position and/or non-optimal quay (NOQ) assignment:

$$f(v, BQ_v, BP_v) = \begin{cases} |PBP_v - BP_v| \cdot C_v^{nob} & \text{, if } BQ_v = PBQ_v \\ C_v^{noq} & \text{, if } BQ_v \in ABQ_v \\ \infty & \text{, otherwise.} \end{cases} \quad (5)$$

In particular, if a ship v is berthed to its preferred berthing quay PBQ_v, the penalty cost is proportional to the distance between its preferred berthing position PBP_v and its planned berthing position BP_v on the quay. Otherwise, if v is berthed to one of its alternative berthing quays ABQ_v, then the penalty cost is set to equal the penalty for non-optimal berthing quay assignment C_v^{noq}. In all other cases, the penalty cost is set to infinity to prohibit the assignment of v to any quay other than PBQ_v or ABQ_v. Moreover, this penalty is multiplied by the handling time HT_v of vessel v. Hence, if a vessel needs four hours to perform its operations and it is moored at a non-optimal position, the penalty is charged for all (four) hours.

The second term in Equation (1) ($WT_v \cdot C_v^w$) calculates the total waiting cost and it depends on the total waiting time WT_v of vessel v and the waiting cost (per time interval) 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 below.

$$WT_v = BT_v - AT_v, \quad \forall\, v \in V \quad (6)$$

The last expression in Equation (1) ($LDT_v \cdot C_v^{ld}$) calculates the penalty cost due to late departures, which is based on the late departure time LDT_v and the penalty (per time interval) for late departure C_v^{ld}. The late departure time LDT_v is calculated as

$$LDT_v = max(0,\ FT_v - RDT_v), \quad \forall\, v \in V \quad (7)$$

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.

$$FT_v = BT_v + HT_v, \quad \forall\, v \in V \quad (8)$$

The fundamental **objective** is to solve the combined BAP and QCAP in an MQ setting while reducing the total service cost as defined above. The objective function is given by

$$\textbf{minimize} \sum_{v \in V} \sum_{q \in Q} \sum_{b \in B(q)} \sum_{k \in K(q)} \sum_{t \in T} Cost(v, q, b, k, t) \cdot x_{vqbkt} \quad (9)$$

subject to the following constraints (Equations (10)–(17)).

$$x_{vqbkt} \in \{0, 1\}, \forall\, v \in V, q \in Q, b \in B(q), k \in K(q), t \in T \quad (10)$$

$$\sum_{q \in Q} \sum_{b \in B(q)} \sum_{k \in K(q)} \sum_{t \in T} x_{vqbkt} = 1, \quad \forall\, v \in V \quad (11)$$

$$BT_v \geq AT_v, \quad \forall\, v \in V \quad (12)$$

$$BT_v - BT_u \geq SE \quad \forall\, v \neq u \in V \quad (13)$$

$$BP_v + L_v \leq L_q, \quad \forall\, v \in V,\ q = BQ_v \quad (14)$$

$$\sum_{v \neq u \in V} \sum_{b=BP_v-L_u-SD}^{BP_v+L_v+SD} \sum_{k \in K(q)} \sum_{t=BT_v-HT_u-ST+1}^{BT_v+HT_v+ST-1} x_{uqbkt} = 0, \quad \forall\, v \neq u \in V,\ q = BQ_v = BQ_u \tag{15}$$

$$\sum_{v \neq u \in V} \sum_{b \in B(q)} \sum_{\substack{k \in K(q) \\ k \,\&\, k_v \neq 0}} \sum_{t=BT_v-HT_u-ST+1}^{BT_v+HT_v+ST-1} x_{uqbkt} = 0, \quad \forall\, v \neq u \in V,\ q = BQ_v = BQ_u \tag{16}$$

$$c^{min} < BP_v + L_v \ \&\ BP_v < c^{max}, \quad \forall\, v \in V,\ c \in k_v \tag{17}$$

The variable x_{vqbkt} shown in constraint (10) is 1 when the vessel v is assigned at position b of quay q at time t to be served by cranes encoded in k, and it is 0 otherwise. Constraint (11) guarantees that each arriving vessel is scheduled for berthing only once in time t at position b belonging to quay q. Constraint (12) stipulates that the planned berthing time BT_v for a particular vessel v must be later or equal to its estimated arrival time AT_v. Constraint (13) ensures a safety entrance time difference between any two vessels. Constraint (14) ensures the length of vessel v plus its berthing position must not cross the length of quay q, where it is moored. Constraint (15) restricts two vessels from overlapping during mooring both in terms of berthing positions as well as berthing times. It also ensures the safety berthing distance SD and safety berthing time ST during the berthing of any two ships. A graphical presentation of constraint (15) is presented in Figure 2. Constraint (16) restricts the set of cranes k that is assigned to vessel u not to contain any of the cranes k_v allocated to another vessel v during the same time period. Whether two sets of cranes have common cranes can be easily checked using the 'bitwise AND' (&) operation due to our binary representation of crane sets. A pictorial representation of constraint (16) is illustrated in Figure 3. Finally, constraint (17) ensures that any crane c assigned to vessel v can reach the vessel by checking that there is an overlap between the minimum and maximum berthing positions served by c and the quay positions occupied by v.

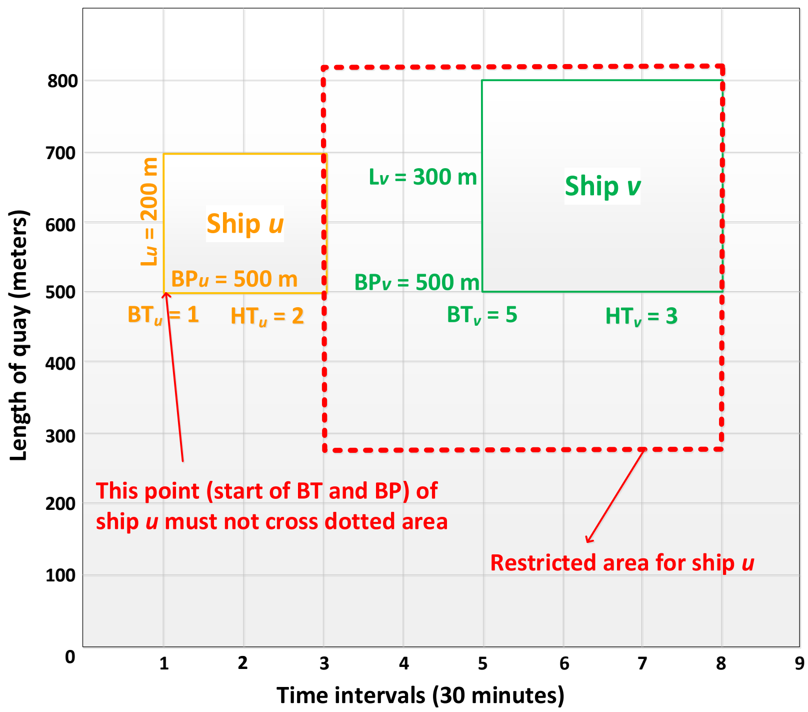


Figure 2. An illustration of the spatiotemporal constraint (15) featuring two arriving vessels (v and u) with different berthing times, berthing positions, and lengths. The red dotted box indicates the restricted area for vessel u to avoid overlap with an already scheduled vessel v.

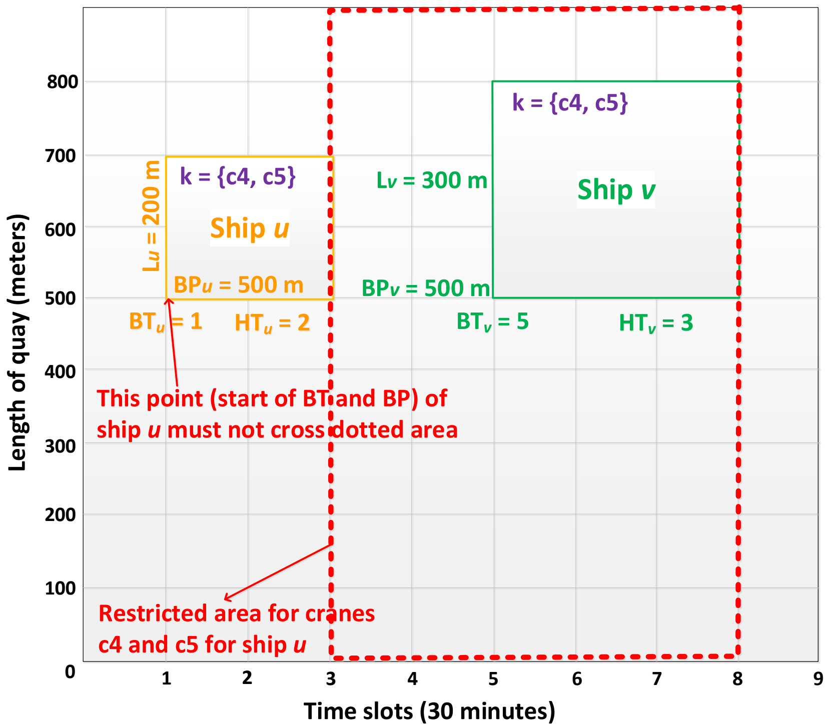


Figure 3. An illustration of constraint (16). The cranes c_4 and c_5 assigned to ship v cannot be assigned to ship u if ship u is scheduled to be berthed in the restricted area marked with the red dotted box.

The unique aspects of our problem formulation compared to previous BAP and/or QCAP formulations are (i) the encoding and consideration of any possible set of heterogeneous (in terms of productivity and service cost) quay cranes to arriving vessels; (ii) the incorporation of multi-purpose heterogeneous quays in the model and the consideration of alternative preferred quays per arriving vessel; and (iii) the incorporation of practical time constraints (e.g., safety berthing distance and time) into the overlap constraints (Equations (15) and (16)) to ensure the correct assignment of berths and cranes to arriving vessels.

4. Proposed CI Methods

This section discloses the various methods implemented for MQ BAP and QCAP.

4.1. Cuckoo Search Algorithm

The cuckoo search algorithm (CSA) is a nature-inspired optimization method proposed by [35] that has proven efficient in solving several global optimization problems. The CSA is based on the basic rules of breeding parasitism of some cuckoo species and then extended by the so-called Levy flights instead of simple isotropic random walks [36]. Some cuckoo birds follow an aggressive production strategy of laying eggs in communal nests and possibly removing eggs from other (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 is inspired by the reproductive behavior of cuckoo birds and follows three idealized principles [35]:

1. Each cuckoo bird deposits a single egg in a random nest;
2. The nests with the highest-quality eggs are preserved and utilized for the next generation;
3. The quantity of host nests is constant and the cuckoo egg is detected by a host bird with probability $p_\alpha \in (0, 1)$.

CSA mapping to MQ BAP and QCAP: Each nest represents a set of possible solutions with berthing times, quays, positions, and a possible set of assigned cranes for all arriving vessels. Each egg in a nest represents either a berthing time or a berthing quay or a berthing position in that quay or a possible set of cranes (expressed as a single number as explained in Section 3.2). A cuckoo egg represents a new (perhaps better) solution (i.e., a berthing time or quay or position or set of cranes). Thus, each nest contains $4N$ eggs, where N is the number of vessels scheduled to arrive at a given time window. The problem's search space at each iteration is determined by a fixed number of host nests, which in this study is set at 100 host nests. The primary aim of the algorithm, outlined in Algorithm 1, is to employ cuckoo eggs (superior solutions) to replace the suboptimal eggs within different nests while ensuring that the various constraints are met. The CSA starts with an initial population of m host nests (line #1). These initial host nests will be attracted by the cuckoos with eggs using random Levy flights to lay the eggs, generating new solutions (lines #3–4). The new nest quality is evaluated and will replace the old host nests if it has a lower fitness score (lines #5–8). If the host bird discovers the egg with some probability $p_\alpha \in (0,1)$, the host abandons the nest and builds a new one (lines #9–11). The above process repeats until a termination criterion is met, such as reaching a maximum number of iterations.

Algorithm 1 CSA for MQ BAP and QCAP

1: $P[1..m]$ = Initialize random population of $m = 100$ host nests
  (each nest contains $4N$ eggs, where N = number of vessels)
2: **for** $t = 1$ to max. iterations **do** ▷ Termination criterion
3:   **for** $i = 1$ to m **do**
4:     $X_{new} = P[i] + \alpha \oplus Levy(\lambda)$ ▷ Generate new solution
5:     $Cost_{prev}$ = Calculate fitness cost of $P[i]$ using Equation (1)
6:     $Cost_{new}$ = Calculate fitness cost of X_{new} using Equation (1)
7:     **if** $Cost_{new} < Cost_{prev}$ **then**
8:       $P[i] = X_{new}$ ▷ Found better solution
9:   **for** $i = 1$ to m **do**
10:     **if** $(rand(0,1) < p_a)$ **then**
11:       $P[i]$ = New host nest is generated (old nest is abandoned)
12: X_{best} = Find best host nest with minimum cost in P

4.2. Genetic Algorithm

The genetic algorithm (GA) 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 [37]. 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 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.

GA mapping to MQ BAP and QCAP: Algorithm 2 shows the working procedure of GA when adopted to the MQ BAP and QCAP. A random population P of m chromosomes is generated, and each chromosome represents a possible solution set for arriving vessels (line #1). The number of chromosomes equals the population size, which is set to 100. A chromosome consists of genes, each representing a single solution, i.e., berthing time or berthing quay or berthing position at the assigned quay or set of assigned cranes. Hence, the number of chromosome genes equals $4N$, where N is the number of vessels arriving in a planning horizon. The fitness value of each chromosome is computed using the objective function (Equation (9)), and the best chromosome with minimum objective value is selected as the local best chromosome (line #2). A proportion of the fittest population from P is selected to start a new generation P' (line #4). Two chromosomes (parents), C_1 and C_2, are randomly selected from the population (line #6), and a crossover with probability P_c is applied to C_1 and C_2 to generate offspring (line #7). During crossover, some of the two parents' single solutions (genes) are exchanged among themselves to generate the offspring. Next, a mutation with probability P_m is performed on the offspring O to generate a new offspring O', where some of the single solutions (genes) of O are replaced (line #8). The new offspring are placed into the new population to avoid local optima (line #9). The above steps are repeated after replacing the old population with the new population until the maximum number of iterations is reached.

Algorithm 2 GA for MQ BAP and QCAP

1: $P[1..m]$ = Initialize random population of $m = 100$ chromosomes
    (each chromosome contains $4N$ genes, where N = number of vessels)
2: Evaluate initial population P using fitness cost function (Equation (1))
3: **for** $t = 1$ to max. iterations **do** ▷ Termination criterion
4:   $P'[1..k]$ = Get the k fittest chromosomes from P
5:   **for** $i = k + 1$ to m **do**
6:     $[C_1, C_2]$ = Select a pair of parents from P using roulette wheel selection
7:     O = Perform multi-point crossover using $[C_1, C_2]$ (crossover rate is 0.90)
8:     O' = Perform mutation on O (mutation rate is 0.10)
9:     $P'[i] = O'$
10:   Evaluate new population P' using fitness cost function (Equation (1))
11:   $P = P'$
12: X_{best} = Find best chromosome with minimum cost in P

4.3. Particle Swarm Optimization

The particle swarm optimization (PSO) 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 traditional methods cannot address. The PSO follows the behavior and social cooperation of flocks of birds 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 is 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 objective function.

PSO mapping to MQ BAP and QCAP: The working procedure of PSO is described in Algorithm 3. First, m random particles are generated in the search space ($m = 100$ in our study), where each particle represents a solution set with $4N$ dimensions, where N is the number of vessels (line #1). Each dimension represents either berthing time or berthing quay or berthing position on the assigned quay or set of assigned cranes. The fitness of all particles (solution sets) is evaluated using Equation (1) to identify the best position (dimensions) for each particle and for the entire swarm (line #2). Next, the velocities and positions (dimensions) of the particles are updated by taking into consideration the local and global best positions in order to generate new positions that move toward the globally best solution and avoid local optima (lines #5–11). The above process is repeated until the maximum number of iterations is reached.

Algorithm 3 PSO for MQ BAP and QCAP

$P[1..m]$ = Initialize random population of $m = 100$ particles
    (each particle position/velocity contains $4N$ dimensions, N = num vessels)
Evaluate initial population P using fitness cost function (Equation (1))
for $t = 1$ to max. iterations **do** ▷ Termination criterion
    for $i = 1$ to m **do**
        Compute new velocity of particle $P[i]$
        Compute new position of particle $P[i]$ using its new velocity
        $Cost_{new}$ = calculate fitness cost of $P[i]$ using Equation (1)
        if $Cost_{new} < P[i]$'s best local cost **then**
            Update $P[i]$'s best local solution
        if $Cost_{new} <$ swarm's best global cost **then**
            Update swarm's best global solution
X_{best} = swarm's best global solution

4.4. First Come First Serve (FCFS)

This study also implements a first come first serve (FCFS) strategy to address the MQ-BAP and QCAP, as adopted in [38] for the BAP. The FCFS method proposes the solution of MQ BAP and QCAP solely based on the sequence of vessel arrivals. In this approach, the first vessel to arrive is given priority for its preferred position on its preferred berthing quay, and it is allocated the maximum available cranes that can serve the given vessel. If upon arrival there are no berthing slots available, the vessel must wait until one becomes vacant. It is important to note that vessels can also be moored at a non-preferred berth position or ABQ in case of high waiting times, for example, when the waiting time is expected to be higher than the vessel's handling time.

5. 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 test the performance of the proposed approach. We perform experiments with different data instances, i.e., with one, two, and four weeks of data. All approaches (i.e., PSO, GA, MILP, FCFS, and the newly adopted CSA) are implemented in MATLAB to conduct a comparative study. For MILP, we employed the solver-based optimization approach and utilized the *intlinprog* solver provided by MATLAB. For the CSA, host nests are 100 with a discovery rate of 0.45, and total iterations are set to 1000. For the GA, population size is set to 100, crossover rate to 0.10%, mutation rate to 90%, and total iterations to 1000 (similar to [22]). For PSO, the inertia weight is set to 1, local learning coefficient to 1.5, global learning coefficient to 2.0, and total iterations to 1000 (similar to [39]). All experiments are performed using a Windows 10 computer system with 3.4 GHz Core i7 and 16 GB RAM.

5.1. A Case Study at the Port of Limassol

This study deals with the case of a real port, located in the city of Limassol, which is the largest port of the island (Cyprus). In the Port of Limassol, there are five commercial berthing quays, all of which are continuous (and hence, arriving ships can berth anywhere on the quay). All quays are of different lengths: Container Quay: 800 m, Ro-Ro Quay: 450 m, West Quay: 770 m, North Quay: 430 m, and East Quay: 480 m. 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.

The Container Quay is divided into two parts, as shown in Figure 1. On one side, there are two cranes, and on the other side, there are three cranes for loading and unloading. There is also a dead space between the two parts of the quay that cannot be used for berthing for safety reasons (see the black-colored part of the quay in Figure 1). The five cranes installed at the Container Quay have different productivity: two cranes have a maximum productivity of 40 containers per hour (red-colored cranes), two cranes have a maximum productivity of 35 containers per hour (blue-colored), and the last crane has a productivity of 22 containers per hour (white-colored), as depicted in Figure 1. It should be noted that the cranes rarely reach maximum productivity due to manpower issues, traffic problems, and other technical challenges. Hence, the average productivity is utilized, which for the first four cranes is 25 containers per hour, while 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. Further details about all cranes, including working locations, are presented in Table 2.

Table 2. Working locations, maximum productivity, and average productivity of cranes at Container Quay and Ro-Ro Quay.

Quay	Crane #	Locations (m)	Productivity max / avg (cont/hour)
Container	Crane 1 (white)	1–100	22 / 20
Container	Crane 2 (blue)	50–275	35 / 25
Container	Crane 3 (blue)	225–450	35 / 25
Container	Crane 4 (red)	470–700	40 / 25
Container	Crane 5 (red)	550–800	40 / 25
Ro-Ro	Crane 1	1–300	25 / 20
Ro-Ro	Crane 2	200–450	25 / 20

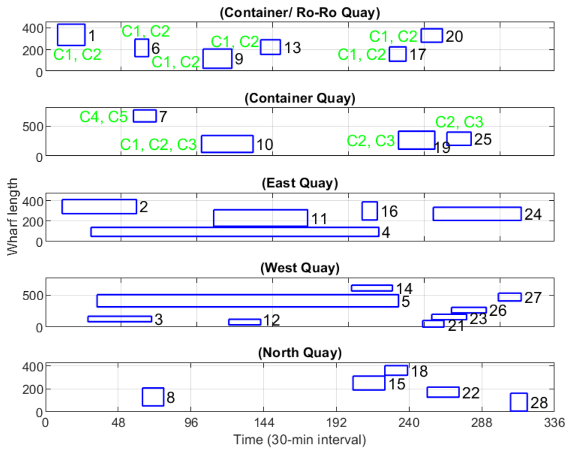
In our experiments, we use data from one week, two weeks, and four weeks, which contain 28, 68, and 168 ships, respectively, arriving in March 2018. Table 3 shows example data. The real data do not include ABQs and PBPs for incoming ships. Thus, we generated random PBPs, as listed in Table 3. We also allocate up to one ABQ (listed in Table 3) for each vessel based on vessel type (e.g., passenger or container vessel) and/or characteristics (e.g., presence of cranes or passenger boarding bridges).

Table 3. Example data for 10 ships that arrived during the first week of March 2018 at the Port of Limassol, Cyprus.

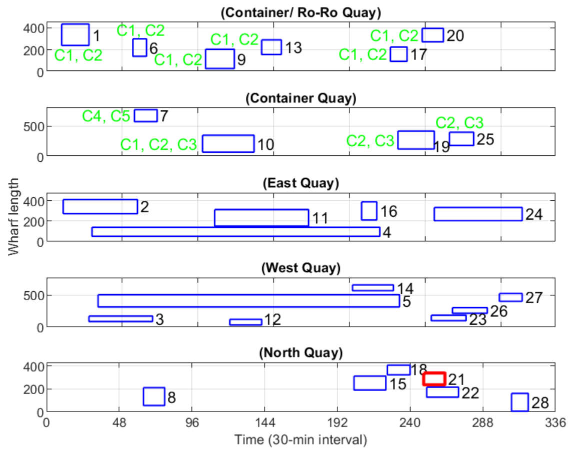
Ship #	ETA (d\t)	HT (min)	ETD (d\t)	PBQ	ABQ	PBP	LoS (m)
1	1\04:00	919	1\22:30	Ro-Ro	Container	240	194
2	1\05:30	1490	2\06:50	East	–	276	139
3	1\14:00	1285	2\12:50	West	North	84	84
4	1\15:00	5700	5\14:03	East	–	51	89
5	1\17:00	5970	5\21:00	West	North	314	190
6	2\04:30	470	2\13:50	Ro-Ro	Container	138	159
7	2\05:00	168	2\09:30	Container	Ro-Ro	571	196
8	2\08:00	440	2\15:55	North	West	53	155
9	3\04:00	905	3\20:50	Ro-Ro	Container	31	175
10	3\03:30	1331	4\06:15	Container	Ro-Ro	389	277

5.2. Results and Discussion

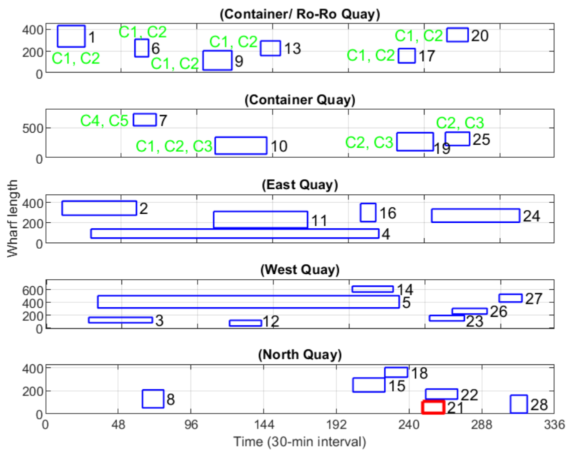
This section shows results obtained by applying the three CI algorithms and the exact MILP method over the real dataset obtained from the Port of Limassol, Cyprus (example data are given in Table 3). Figure 4 shows the solutions proposed by our CSA and other two heuristics along with MILP approaches for the allocation of berths and quay cranes to the 28 arriving ships for the one-week planning horizon. The rectangles in this figure show each vessel, the x-axis indicates the berthing time, and the y-axis indicates the berthing quay and position of each vessel. 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 rectangles 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. From Figure 4, it can be seen that vessel 23 is berthed at North Quay instead of West Quay when MILP, GA, and PSO are used. On the other hand, CSA places vessel 23 at its PBQ but at the expense of placing it far from its preferred berthing position (PBP), increasing the total service cost compared to the other methods; however, moving any vessel to ABQ will incur penalty due to mooring at the non-preferred quay. Here, it is important to note that there are only two quays where QCs are installed and assigned by all algorithms, i.e., Container and Ro-Ro. 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 handling 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.



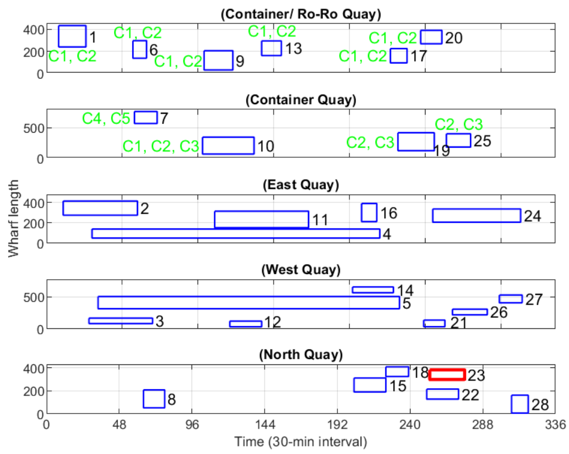
(**a**) Solution by CSA



(**b**) Solution by GA



(**c**) Solution by PSO



(**d**) Solution by MILP

Figure 4. Berth and quay crane scheduling solutions by our proposed CSA and compared approaches.

The results presented in Figure 5 show the mean difference and standard error between the optimal berthing times and the times proposed by the different methods. 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 the one-week planning horizon and became stuck at the two-week and four-week planning horizons and ran out of memory. On the other hand, the CSA shows superiority over the other three approaches (i.e., the GA, PSO, and FCFS) in all three scenarios. Moreover, there is a high mean difference and standard error using the FCFS method due to the frequent long waiting times induced by this approach. Moreover, the GA performs well in the case of one-week and four-week scenarios compared to PSO, while PSO shows a smaller mean difference and error in the case of the two-week scenario.

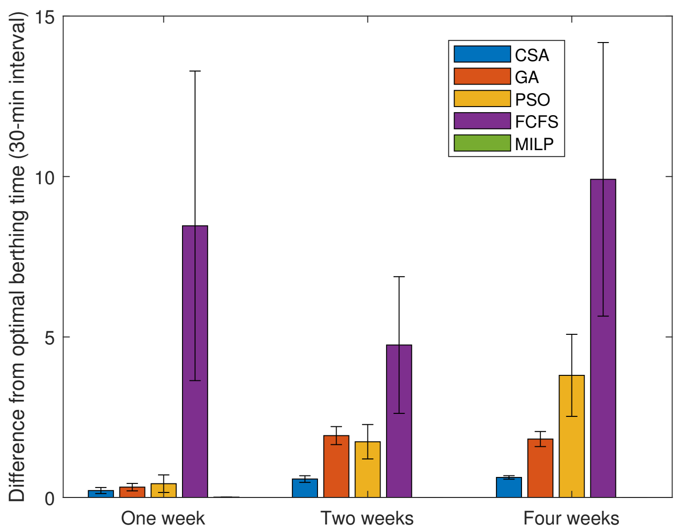


Figure 5. Mean difference and standard error between berthing time by five implemented methods and optimal berthing time for three different scenarios (1, 2, and 4 weeks).

Figure 6 shows the mean difference between optimal and non-optimal berthing positions. It can also be seen that the MILP and FCFS have the smallest difference in the case of one week; however, MILP cannot solve the other two cases. On the contrary, the CSA again performs well in the one-week and two-week scenarios compared to the GA and PSO. However, FCFS has a lower mean difference in the two and four-week scenarios compared to other methods. Furthermore, in all cases, the GA performs well and outperforms PSO.

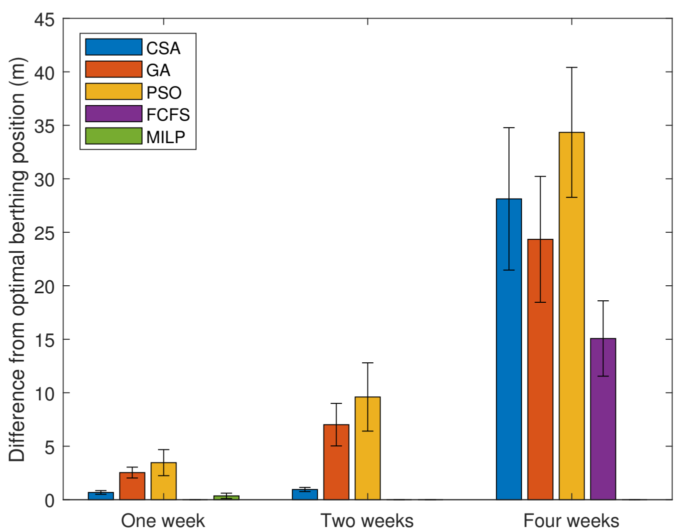


Figure 6. Mean difference and standard error between planned and preferred berthing position by five implemented methods for three different scenarios (1, 2, and 4 weeks).

The results presented in Figure 7 show the mean difference between optimal and non-optimal berthing costs that occurred due to the allocation of vessels to locations other than the PBPs. Again, we can clearly see that the CSA has the smallest mean difference after MILP and FCFS in the case of one- and two-week scenarios; FCFS, on the other hand, has the smallest difference in the case of the four-week scenario. We can conclude from the above discussions that the CSA performs well overall compared to the GA, FCFS and PSO; however, MILP shows better results but only in the case of a week case study.

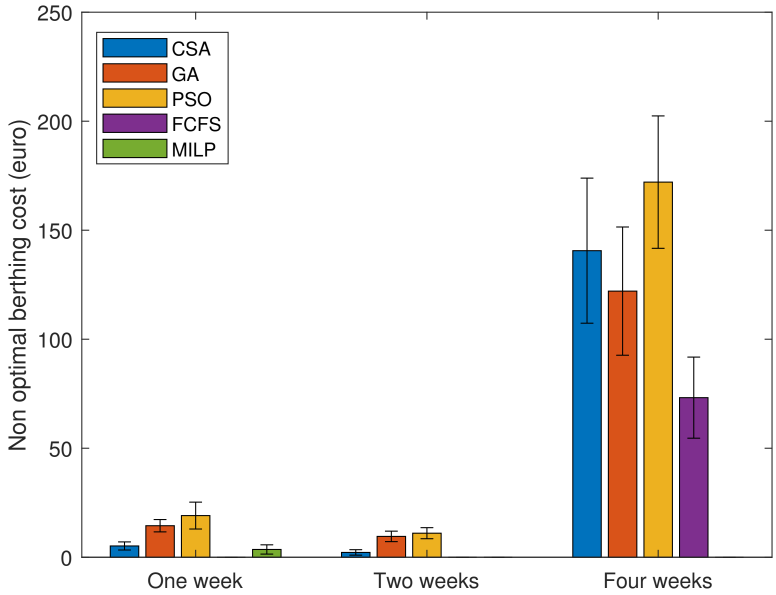


Figure 7. Mean difference and standard error between optimal and non-optimal berthing cost by five implemented methods for three different scenarios (1, 2, and 4 weeks).

To show a more in-depth comparison of the compared methods, Table 4 presents the different costs associated with the total service costs and 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 wait before the optimal berth allocation, while non-optimal berthing (NOB) costs are included in the total service cost when a vessel v is berthed at a position other than its PBP or at an 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 (5). However, a fixed penalty is added in case of berth allocation at ABQ. From this table, it can be seen that MILP has a minimum total cost (10,090) with 0 waiting cost. Nonetheless, it provides an optimal solution at the expense of computation time, which is 912.55 s (more than 15 min) for the one-week scenario. In the experiments for the two-week and four-week planning periods, MILP cannot solve the problem and runs out of memory. On the other hand, the total service cost of CSA (10,320) is close to the cost of MILP and lower compared to other CI methods (i.e., GA and PSO) in the case of the one-week scenario. The CSA cost performance is followed by the GA and then PSO for the same scenario.

Moreover, when we run experiments for two weeks, the CSA again shows supremacy over the GA, PSO, and FCFS in terms of total service cost. Furthermore, FCFS shows 0 or minimum NOB cost compared to other methods, however, at the expense of high waiting cost. Eventually, when we run experiments for four weeks, we noticed that the NOB cost of all algorithms is increased; in this case, the CSA again achieves minimum cost, after FCFS, compared to PSO and the GA. The total service cost by the CSA is 63,875 and is minimal compared to other methods in case of four weeks. However, the GA follows the CSA and achieves a slightly higher cost of 64,975, which is much better compared to PSO (72,290) and FCFS (73,950). Finally, regarding computation time, FCFS has the minimum computation time in all cases; however, the total service cost is very high and 15.77–52.27% higher than our proposed CSA method. However, after FCFS, the CSA solves all the scenarios in minimum computation times that are 84.73, 336.16, and 388.20 s for one week, two weeks, and four weeks, respectively. In contrast, the GA and PSO take 94.67 and 316.23 s for one week, 226.57 and 534.80 s for two weeks, and 2663.13 and 2777.40 s for four weeks, respectively. Based on the aforementioned comparative analysis, it can be concluded that the proposed CSA-based approach for MQ BAP and QCAP delivers a near-optimal solution while maintaining good computational efficiency.

Table 4. Comparative analysis with data spanning a period of 1–4 weeks (March 2018) for all methods. All costs are in Euros. % Deviation is calculated with reference to the total service cost of the CSA approach.

Scenarios:	One Week (28 Ships)					Two Weeks (68 Ships)					Four Weeks (168 Ships)				
Algorithms:	**CSA**	**GA**	**PSO**	**FCFS**	**MILP**	**CSA**	**GA**	**PSO**	**FCFS**	**MILP**	**CSA**	**GA**	**PSO**	**FCFS**	**MILP**
Waiting cost	50	45	75	1185	0	195	655	590	1615	–	430	1255	2625	6840	–
NOB cost	250	405	535	0	100	150	650	750	0	–	19,405	16,845	23,745	10,310	–
Late departure cost	140	120	120	4660	120	240	680	900	6440	–	1620	2720	3500	14,380	–
Normal handling cost	9870	9870	9870	9870	9870	17,220	17,220	17,220	17,220	–	42,420	42,420	42,420	42,420	–
Total service cost	10,320	10,440	10,700	15,715	10,090	17,805	19,205	19,460	25,275	–	63,875	64,975	72,290	73,950	–
% Deviation from CSA	–	1.16	3.68	52.27	−2.22	–	7.86	9.29	41.95	–	–	1.72	13.17	15.77	–
Computation time (sec)	84.73	94.67	316.23	0.20	912.55	336.16	226.57	534.80	3.40	–	388.20	2663.13	2777.40	12.70	–

To further evaluate the proposed method, we conduct additional experiments using randomly generated (uniform) data instances, following the previous literature [14,22,25,40]. The data instances are generated considering various scenarios, including 1 to 7 days for the planning horizon, one to several quays of different lengths, and 10 to 60 arriving vessels. The results from these random data instances, shown in Table 5, further confirm the effectiveness of the proposed method in solving the MQ BAP and QCAP. For example, in cases with 10, 20, and 30 vessels, the CSA is only 2.68%, 0.98%, and 0.41% away from the optimal solution provided by MILP. However, when the number of vessels exceeds 30, the exact method is unable to solve the problem due to memory limitations, whereas the CSA continues to provide (near) optimal solutions within a reasonable computation time. Additionally, when comparing the proposed CSA with other heuristics, the results clearly show that the CSA consistently outperforms PSO, while GA performs better in terms of computation time albeit at the cost of higher total service costs—ranging from a minimum of 2.04% higher for 10 vessels to a maximum of 27.02% higher for 15 vessels.

Table 5. Sensitivity analysis for all approaches with uniform random data (10–60 vessels, 1–7 days, and 1–5 quays).

No. of Ships	Days	No. of Quays	Service Cost (Euro)				Computation Time (Sec.)			
			CSA	**GA**	**PSO**	**MILP**	**CSA**	**GA**	**PSO**	**MILP**
10	1	1	4912	5412	4961	4898	38.15	30.92	30.86	65.54
		2	2956	2996	3606	2850	32.47	9.01	30.86	39.44
		3	9215	9191	9165	8920	44.32	6.74	27.50	26.48
		4	3772	3810	3960	3764	30.15	7.27	28.23	27.11
		5	8773	8825	8837	8400	23.98	5.85	24.68	23.52
Avg deviation from CSA (%)			**–**	**2.04**	**3.04**	**−2.68**	**–**	**−64.63**	**−15.75**	**7.70**
15	1	1	7893	10551	8067	7820	52.11	25.12	51.36	190.67
		2	6086	8327	7350	5970	47.86	14.46	41.65	82.07
		3	5580	6155	11,815	5540	48.08	13.79	41.84	66.02
		4	4813	5547	4984	4800	42.58	11.36	37.53	76.75
		5	5936	7920	6960	5880	35.57	9.45	38.42	82.74

Table 5. *Cont.*

No. of Ships	Days	No. of Quays	Service Cost (Euro)				Computation Time (Sec.)			
			CSA	GA	PSO	MILP	CSA	GA	PSO	MILP
Avg deviation from CSA (%)			–	**27.02**	**29.25**	**−0.98**	–	**67.17**	**6.73**	**120.44**
20	2	1	8503	10,600	10,355	8200	67.05	41.20	78.48	420.86
		2	8270	9643	10,020	8010	61.18	24.06	54.32	207.45
		3	8353	9392	8990	8310	54.49	17.57	55.54	196.60
		4	7518	9882	10296	7480	53.94	16.86	55.20	147.11
		5	6600	8788	12,616	6420	44.53	12.77	48.84	148.01
Avg deviation from CSA (%)			–	**25.20**	**35.50**	**−0.41**	–	**−59.94**	**4.14**	**298.95**
30	2	1	17810	14540	23,832	–	112.27	78.15	135.55	–
		2	19,478	24,862	28,219	–	92.93	33.27	90.50	–
		3	11,922	17,000	14,838	–	81.15	28.42	85.23	–
		4	9103	12,556	9278	–	74.37	25.94	75.51	–
		5	9702	13,634	12,330	–	66.56	28.22	75.23	–
Avg deviation from CSA (%)			–	**21.43**	**30.11**	–	–	**−54.59**	**8.13**	–
60	7	1	38,712	40,420	40,095	–	221.58	292.86	402.24	–
		2	29,584	39,353	30,976	–	207.59	101.89	203.08	–
		3	24,786	20,919	33,542	–	188.23	87.80	199.86	–
		4	16,142	21,995	24,246	–	155.33	89.93	202.66	–
		5	23,750	27,149	27,973	–	151.25	86.38	202.97	–
Avg deviation from CSA (%)			–	**12.68**	**17.94**	–	–	**−28.69**	**31.04**	–

6. Managerial Insights

While this study primarily presents a solution to the operational-level challenges posed by MQ BAP and QCAP, aiming to minimize the total service cost and reduce computation time, it is also able to offer valuable insights to terminal managers and policymakers. Based on the model formulation and the experimental results, the following observations are obtained.

First, the penalties are less when all algorithms are tested for one-week and two-week scenarios (especially for penalty due to non-optimal berthing position), as can be seen from Table 4. The mean difference and standard error are also low when we implement for up to the two-week case, as depicted in Figures 5–7. Therefore, to achieve better results at the Port of Limassol, this study suggests scheduling berth and quay crane assignments one or two weeks in advance using our proposed methodology. In case of changes in the ship arrival schedule, the methodology is able to adjust new or missing vessels, as computation time is low (as shown in Table 4).

Second, in case of any conflict among vessels' arrival/ departure times and positions, the CI-based methodology can help berth planners to assign a berth quay and position near the assigned storage area in the marshaling yard (as determined by the vessel's PBQ, ABQ, and PBP). With this approach, it becomes possible to minimize both the cost and time associated with container/cargo transfers. Ultimately, with the proposed approach, container terminals can maximize their productivity to its fullest potential.

Third, investing in additional QCs may be worthwhile for ports, as ships will be handled faster and at a lower cost if the proposed model is implemented. Our proposed methodology can quantify the gains by re-executing the berth and QC assignment using hypothetical QCs (before investment). In addition, the latest technologies can further help in reducing waiting times, total stay time of vessels, and different penalties during regular processes.

7. Conclusions

This study investigates the MQ BAP and QCAP in a real-world scenario with the objective of minimizing the total service cost for arriving vessels. To solve this problem, an MILP model is developed and solved using both exact and computational intelligence (CI) methods. We implement the cuckoo search algorithm (CSA), genetic algorithm (GA), particle swarm optimization (PSO), and first come first serve (FCFS) for MQ BAP and QCAP. To validate the methods, we test them on real data collected at 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 the developed approaches. Simulation results confirm the effectiveness of our proposed CSA over the compared methods. Furthermore, the MILP can only solve a one-week scenario and requires a lot of computation time (912 s). In contrast, the CSA method solves the one-week scenario in only 84.73 s, and the achieved objective value (10320 Euro) is only 1% away from the optimal solution (10090 Euro). Moreover, the CSA-based solution also performs better than 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 MQ BAP and QCAP tends to require too much computational effort to be of practical use and cannot solve the problem in large instances. On the contrary, CI-based approaches, especially the CSA, are able to offer solutions close to the optimal ones in a short computation time. Moreover, even for large data instances, the computation time of the CSA solution remains below 400 s (about 6 min). This presents 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 and make management decisions.

Author Contributions: Conceptualization, M.P.M. and H.H.; Data curation, S.A. and H.H.; Formal analysis, S.A.; Funding acquisition, M.P.M. and H.H.; Methodology, M.P.M., S.A. and H.H.; Software, S.A.; Supervision, M.P.M. and H.H.; Validation, M.P.M. and S.A.; Visualization, S.A. and H.H.; Writing—original draft, S.A., M.P.M. and H.H.; Writing—review and editing, S.A., M.P.M. and H.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the European Union's Horizon Europe program for Research and Innovation through the aerOS project under Grant No. 101069732 as well as by the European Regional Development Fund and the Republic of Cyprus through the Cyprus Research and Innovation Foundation (MDigi-I: STRATEGIC INFRASTRUCTURES/1222/0113).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data can be provided on request to corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ABQ	alternative berthing quay
BAP	berth allocation problem
B&C	branch and cut
CI	computational intelligence
CSA	cuckoo search algorithm
DE	differential evolution
ETA	estimated time of arrival
FCFS	first come first serve
GA	genetic algorithm
MCT	maritime container terminal
MILP	mixed integer linear programming
MQ	multi-quay
NOB	non-optimal berthing position
NOQ	non-optimal berthing quay
QC	quay crane
QCAP	quay crane allocation problem
RTD	requested time of departure
SA	simulated annealing
PBQ	preferred berthing quay

PBP	preferred berthing position
PSO	particle swarm optimization

References

1. Review of Maritime Transport 2022. United Nations Conference on Trade and Development, New York. 2022. Available online: https://unctad.org/system/files/official-document/rmt2022_en.pdf (accessed on 17 July 2024).
2. Zheng, F.; Li, Y.; Chu, F.; Liu, M.; Xu, Y. Integrated berth allocation and quay crane assignment with maintenance activities. *Int. J. Prod. Res.* **2019**, *57*, 3478–3503. [CrossRef]
3. Michaelides, M.P.; Herodotou, H.; Lind, M.; Watson, R.T. Port-2-port communication enhancing short sea shipping performance: The case study of Cyprus and the Eastern Mediterranean. *Sustainability* **2019**, *11*, 1912. [CrossRef]
4. Lind, M.; Michaelides, M.P.; Robert, W.; Richard, W.T. *Maritime Informatics*; Springer: Berlin/Heidelberg, Germany, 2020.
5. Wang, Z.; Hu, H.; Zhen, L. Berth and quay cranes allocation problem with on-shore power supply assignment in container terminals. *Comput. Ind. Eng.* **2024**, *188*, 109910. [CrossRef]
6. Aslam, S.; Michaelides, M.P.; Herodotou, H. A survey on computational intelligence approaches for intelligent marine terminal operations. *IET Intell. Transp. Syst.* **2024**. [CrossRef]
7. Li, Y.; Chu, F.; Zheng, F.; Liu, M. A Bi-Objective Optimization for Integrated Berth Allocation and Quay Crane Assignment With Preventive Maintenance Activities. *IEEE Trans. Intell. Transp. Syst.* **2020**. [CrossRef]
8. Yu, T.; Qiang, Z.; Benfei, Z. A genetic algorithm based on spatiotemporal conflict between continuous berth-allocation and time-varying specific crane assignment. *Eng. Optim.* **2019**, *51*, 390–411. [CrossRef]
9. Aslam, S.; Michaelides, M.P.; Herodotou, H. Enhanced Berth Allocation Using the Cuckoo Search Algorithm. *SN Comput. Sci.* **2022**, *3*, 1–15. [CrossRef]
10. Bierwirth, C.; Meisel, F. A follow-up survey of berth allocation and quay crane scheduling problems in container terminals. *Eur. J. Oper. Res.* **2015**, *244*, 675–689. [CrossRef]
11. Chang, S.C.; Lin, M.H.; Tsai, J.F. An Optimization Approach to Berth Allocation Problems. *Mathematics* **2024**, *12*, 753. [CrossRef]
12. Xiang, X.; Liu, C. An almost robust optimization model for integrated berth allocation and quay crane assignment problem. *Omega* **2021**, *104*, 102455. [CrossRef]
13. Rodrigues, F.; Agra, A. Berth allocation and quay crane assignment/scheduling problem under uncertainty: A survey. *Eur. J. Oper. Res.* **2022**, *303*, 501–524. [CrossRef]
14. Frojan, P.; Correcher, J.F.; Alvarez-Valdes, R.; Koulouris, G.; Tamarit, J.M. The continuous Berth Allocation Problem in a container terminal with multiple quays. *Expert Syst. Appl.* **2015**, *42*, 7356–7366. [CrossRef]
15. Krimi, I.; Todosijević, R.; Benmansour, R.; Ratli, M.; El Cadi, A.A.; Aloullal, A. Modelling and solving the multi-quays berth allocation and crane assignment problem with availability constraints. *J. Glob. Optim.* **2020**, *78*, 349–373. [CrossRef]
16. Gutierrez, F.; Lujan, E.; Asmat, R.; Vergara, E. Fully fuzzy linear programming model for the berth allocation problem with two quays. In *Uncertainty Management with Fuzzy and Rough Sets: Recent Advances and Applications*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 87–113.
17. Lujan, E.; Vergara, E.; Rodriguez-Melquiades, J.; Jiménez-Carrión, M.; Sabino-Escobar, C.; Gutierrez, F. A Fuzzy Optimization Model for the Berth Allocation Problem and Quay Crane Allocation Problem (BAP+ QCAP) with n Quays. *J. Mar. Sci. Eng.* **2021**, *9*, 152. [CrossRef]
18. Aslam, S.; Michaelides, M.P.; Herodotou, H. Muli-Quay Combined Berth and Quay Crane Allocation Using the Cuckoo Search Algorithm. In Proceedings of the 8th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS), Angers, France, 2–4 May 2024; pp. 220–227.
19. Theofanis, S.; Boile, M.; Golias, M. An optimization based genetic algorithm heuristic for the berth allocation problem. In Proceedings of the 2007 IEEE Congress on Evolutionary Computation, Singapore, 25–28 September 2007; IEEE: Piscataway, NJ, USA, 2007; pp. 4439–4445.
20. Golias, M.M.; Boile, M.; Theofanis, S. Berth scheduling by customer service differentiation: A multi-objective approach. *Transp. Res. Part Logist. Transp. Rev.* **2009**, *45*, 878–892. [CrossRef]
21. Golias, M.M.; Saharidis, G.K.; Boile, M.; Theofanis, S.; Ierapetritou, M.G. The berth allocation problem: Optimizing vessel arrival time. *Marit. Econ. Logist.* **2009**, *11*, 358–377. [CrossRef]
22. Alsoufi, G.; Yang, X.; Salhi, A. Robust berth allocation using a hybrid approach combining branch-and-cut and the genetic algorithm. In Proceedings of the International Workshop on Hybrid Metaheuristics, Plymouth, UK, 8–10 June 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 187–201.
23. Şahin, C.; Kuvvetli, Y. Differential evolution based meta-heuristic algorithm for dynamic continuous berth allocation problem. *Appl. Math. Model.* **2016**, *40*, 10679–10688. [CrossRef]
24. Lee, D.H.; Chen, J.H.; Cao, J.X. The continuous berth allocation problem: A greedy randomized adaptive search solution. *Transp. Res. Part E Logist. Transp. Rev.* **2010**, *46*, 1017–1029. [CrossRef]
25. Ernst, A.T.; Oğuz, C.; Singh, G.; Taherkhani, G. Mathematical models for the berth allocation problem in dry bulk terminals. *J. Sched.* **2017**, *20*, 459–473. [CrossRef]
26. Korekane, S.; Nishi, T.; Tierney, K.; Liu, Z. Neural network assisted branch and bound algorithm for dynamic berth allocation problems. *Eur. J. Oper. Res.* **2024**, *319*, 531–542. [CrossRef]

27. Xu, Y.; Xue, K.; Du, Y. Berth scheduling problem considering traffic limitations in the navigation channel. *Sustainability* **2018**, *10*, 4795. [CrossRef]
28. Li, Y.; Chu, F.; Zheng, F.; Kacem, I. Integrated Berth Allocation and Quay Crane Assignment With Uncertain Maintenance Activities. In Proceedings of the 2019 International Conference on Industrial Engineering and Systems Management (IESM), Shanghai, China, 25–27 September 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
29. Abou Kasm, O.; Diabat, A.; Cheng, T. The integrated berth allocation, quay crane assignment and scheduling problem: Mathematical formulations and a case study. *Ann. Oper. Res.* **2020**, *291*, 435–461. [CrossRef]
30. Hsu, H.P.; Chiang, T.L.; Wang, C.N.; Fu, H.P.; Chou, C.C. A Hybrid GA with Variable Quay Crane Assignment for Solving Berth Allocation Problem and Quay Crane Assignment Problem Simultaneously. *Sustainability* **2019**, *11*, 2018. [CrossRef]
31. Wang, T.; Wang, X.; Meng, Q. Joint berth allocation and quay crane assignment under different carbon taxation policies. *Transp. Res. Part B Methodol.* **2018**, *117*, 18–36. [CrossRef]
32. Chang, D.; Jiang, Z.; Yan, W.; He, J. Integrating berth allocation and quay crane assignments. *Transp. Res. Part Logist. Transp. Rev.* **2010**, *46*, 975–990. [CrossRef]
33. Aslam, S.; Michaelides, M.P.; Herodotou, H. Berth allocation considering multiple quays: A practical approach using cuckoo search optimization. *J. Mar. Sci. Eng.* **2023**, *11*, 1280. [CrossRef]
34. Iris, Ç.; Pacino, D.; Ropke, S.; Larsen, A. Integrated berth allocation and quay crane assignment problem: Set partitioning models and computational results. *Transp. Res. Part E Logist. Transp. Rev.* **2015**, *81*, 75–97. [CrossRef]
35. Yang, X.S.; Deb, S. Cuckoo Search via Lévy Flights. In Proceedings of the World Congress on Nature & Biologically Inspired Computing (NaBIC), Coimbatore, India, 9–11 December 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 210–214.
36. Yang, X.S.; Deb, S. Cuckoo search: Recent advances and applications. *Neural Comput. Appl.* **2014**, *24*, 169–174. [CrossRef]
37. Katoch, S.; Chauhan, S.S.; Kumar, V. A review on genetic algorithm: Past, present, and future. *Multimed. Tools Appl.* **2021**, *80*, 8091–8126. [CrossRef]
38. Hsu, H.P.; Chiang, T.L. An improved shuffled frog-leaping algorithm for solving the dynamic and continuous berth allocation problem (DCBAP). *Appl. Sci.* **2019**, *9*, 4682. [CrossRef]
39. Ting, C.J.; Wu, K.C.; Chou, H. Particle swarm optimization algorithm for the berth allocation problem. *Expert Syst. Appl.* **2014**, *41*, 1543–1550. [CrossRef]
40. Xiang, X.; Liu, C.; Miao, L. A bi-objective robust model for berth allocation scheduling under uncertainty. *Transp. Res. Part E Logist. Transp. Rev.* **2017**, *106*, 294–319. [CrossRef]