Dynamic and Continuous Berth Allocation using Cuckoo Search Optimization

Sheraz Aslam[a], Michalis P. Michaelides[b] and Herodotos Herodotou[c]
Department of Electrical Engineering, Computer Engineering and Informatics, Cyprus University of Technology, Cyprus

Keywords: Berth Allocation Problem, Intelligent Sea Transportation, Cuckoo Search Algorithm, Metaheuristic Optimization, Port Efficiency.

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 maritime 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. In order to improve port efficiency in terms of reducing the total handling cost and late departures, this study investigates the dynamic and continuous berth allocation problem (DC-BAP), where vessels are assigned dynamically as they arrive at their berth locations assuming a continuous berth layout. First, the DC-BAP is formulated as a mixed-integer linear programming (MILP) model. Since BAP is an NP-hard problem and cannot be solved by mathematical approaches in a reasonable time, this study adopts the recently developed metaheuristic cuckoo search algorithm (CSA) to solve the DC-BAP. For validating the performance of the proposed CSA method, we use a benchmark case study and a genetic algorithm solution proposed in recent literature as well as compare our results against the optimal MILP solution. From the simulation results, it becomes evident that the newly proposed algorithm has higher efficiency over counterparts in terms of optimal berth allocation within reasonable computation time.

SCIENCE AND TECHNOLOGY PUBLICATIONS

1 INTRODUCTION

The shipping industry covers 90% of the world seaborne trade movements and 74% of the total goods that are imported or exported in Europe travel with ships (Aslam et al., 2020). According to Hsu et al. (2017), 60% of the total sea transport is based on containers, which is also growing every year by 6.4%. So the maritime container terminal (MCT) serves as an important node in the shipping industry to deal with increasing sea trade. A report presented in Barbosa et al. (2019) stated that worldwide ports have handled almost 701 million twenty-foot equivalent units (TEUs) of containers in 2016. At the same time, the throughput of container ports is also continuously increasing, and the management of MCTs' operations is becoming a challenging task. 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 (De et al., 2020). Since the MCTs have a huge importance in the maritime industry, there is an exigent need to enhance the operational efficiency of MCTs by mitigating the total turnaround service times of vessels and achieving competitive strategy along with customer satisfaction. Furthermore, 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

[a] https://orcid.org/0000-0003-4305-0908
[b] https://orcid.org/0000-0002-0549-704X
[c] https://orcid.org/0000-0002-8717-1691

Aslam, S., Michaelides, M. and Herodotou, H.
Dynamic and Continuous Berth Allocation using Cuckoo Search Optimization.
DOI: 10.5220/0010436600720081
In *Proceedings of the 7th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2021)*, pages 72-81
ISBN: 978-989-758-513-5
Copyright © 2021 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

than 12 hours) (CargoSmart.ai, 2019). According to a recent report presented in UNCTAD (2017), average waiting times for vessels from port-to-berth is 2.2, 2.4, and 2.7 hours in Malaysia, Dubai, and China, respectively. In Michaelides et al. (2019), the effect of time in port is considered by investigating 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 assisting all those 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 (Lind et al., 2019). Hence, MCTs' operators need to employ suitable strategies and approaches for proper utilization of the port resources and to avoid the above-mentioned issues.

These challenges have motivated us to focus on enhancing seaside operations. Thus, this study deals with the berth allocation problem (BAP), which is a well-known problem that aims to assign berthing positions to arriving vessels at the port in order to minimize or maximize the 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 we can have a mix of continuous and discrete berthing layouts (Carlo et al., 2015). In terms of vessel arrivals, there are two main types: 1) static arrivals where all the vessels are assumed to be at the MCT and 2) dynamic arrivals meaning that vessels are not at the MCT; however, the expected time of arrival (ETA) is known for each vessel. This study adopts the continuous berthing layout together with dynamic vessel arrivals (i.e., DC-BAP).

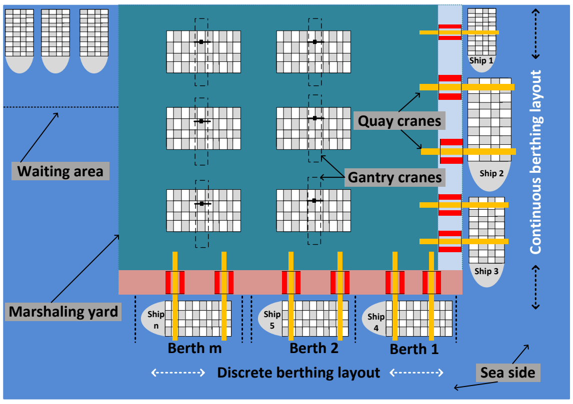


Figure 1: An illustration of MCT with multiple berthing positions assuming discrete and continuous berthing layout.

Literature Review: Several approaches are reported in the literature that deal with the BAP (Bierwirth and Meisel, 2010, 2015). These approaches may provide exact solutions (Jos et al., 2019) or approximate (based on heuristic or metaheuristic) solutions (Kavoosi et al., 2019; Dulebenets, 2017). However, approximate approaches are more popular over exact methods due to their efficiency in terms of computational complexity. The authors of Kavoosi et al. (2019) present a solution of the BAP by employing evolutionary algorithms (EAs), particle swarm optimization (PSO) and differential evolution (DE). An EA-based solution is developed in Dulebenets (2017) to deal with BAP, while the study presented in Xu et al. (2018) proposed a simulated annealing (SA) algorithm for the same problem. Jos et al. (2019) developed a mixed-integer linear programming (MILP) model to deal with the BAP and a genetic algorithm (GA) is developed in Hsu et al. (2019) to solve the BAP. Another study (Chen and Huang, 2017) also develops a GA-based approach to deal with the DC-BAP to minimize penalty costs for late departures. Mauri et al. (2016) also deal with the DC-BAP. The objectives of this study are to minimize the total service cost and the total ship stay time at port. To solve the problem, a variant of neighborhood search method called adaptive large neighborhood search (ALNS) algorithm is developed. The algorithm works based on the principle of destroy and recreate, where at each iteration some solutions are destroyed and new ones are generated in different ways to find the best solution according to fitness criteria.

The work presented in Frojan et al. (2015) deals with an updated form of BAP, where multiple quays and continuous berth layouts are considered. The problem is first formulated as an integer linear model and then solved using GA. To confirm the effectiveness of the proposed method, several simulations are performed with different sized datasets and the results show the effectiveness against the compared approaches. Han et al. (2015) solve the integrated type of berth allocation where the BAP and the quay crane (QC) allocation problem are considered simultaneously. They proposed a multi-objective particle swarm optimization (PSO) algorithm, and a continuous berth layout is considered for berth planning. The primary objectives of this study are twofold: minimizing the range of maximum and minimum QCs to save resources, and reducing the movements of QCs to improve terminal's efficiency. Another study presented in Cahyono et al. (2019) also addresses the integrated BAP and QC assignment problem with the goals of minimizing the total handling and waiting costs. A model predictive algorithm is developed to achieve the desired objectives of this study. To validate the proposed approach, several experiments are conducted on real-time data collected from a container terminal in Jakarta, Indonesia. In this study, a novel cuckoo search algorithm (CSA) is proposed for the first time (according to the best of our knowledge) to solve the dynamic and continuous berth allocation problem (DC-BAP).

Contributions: This study investigates the DC-BAP and aims to mitigate the total service cost of arriving vessels, which includes the total handling cost as well as the waiting times and late departures penalties. We first formulate DC-BAP as a mixed-integer linear programming (MILP) model and solve it using the newly developed cuckoo search algorithm (CSA). The simulation results are compared against two benchmarks proposed in the recent literature: GA and an MILP exact solution. From the provided simulation results, it becomes evident that our proposed CSA shows efficacy over counterparts.

Outline: The remainder of the paper is organized as follows. Section 2 explains the investigated problem and provides its mathematical formulation. Our newly proposed CSA method is described in detail in Section 3 and simulation settings along with results are presented in Section 4. Finally, Section 5 discusses future work and concludes the paper.

2 PROBLEM DESCRIPTION

This section first describes in detail the DC-BAP considered in this work, followed by a mathematical formulation as a mixed-integer linear programming problem. Table 1 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 $B = \{1, 2, \ldots, M\}$ 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 $T = \{1, 2, \ldots, K\}$ that can represent some time duration of interest (e.g., an hour or a 15-minute interval). Finally, let $S = \{1, 2, \ldots, N\}$ denote the set

Table 1: Nomenclature.

Name	Explanation
Acronyms	
BAP	Berth allocation problem
BP	Berthing position
CSA	Cuckoo search algorithm
DC-BAP	Dynamic and continuous BAP
ETA	Estimated time of arrival
ETD	Estimated time of departure
GA	Genetic algorithm
HT	Handling time
LoS	Length of ship
MCT	Maritime container terminal
PBP	Preferred berthing position
QCs	Quay cranes
WC	Waiting cost
WT	Waiting time
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_s	Length of ship s
LDC_s	Late departure cost of s per time period
LDT_s	Late departure time of s
PBP_s	Preferred berthing position of s
W	Length of wharf
WC_s	Waiting cost of s per time period
WT_s	Waiting time of s
Indices	
$B = \{1,2,...,M\}$	Set of available berth positions
$S = \{1,2,...,N\}$	Set of ships
$T = \{1,2,...,K\}$	Set of time periods
b	Berthing position index
s	Individual ship index
t	Single time period index

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. As a result of the increased number of ships in the waiting area, congestion and navigational challenges are created 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) (Bierwirth and Meisel, 2010, 2015). 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.

2.1 Mathematical Formulation

Before disclosing the mathematical formulation of DC-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 completion (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 or can be easily computed.
- Each ship has a preferred berthing position and it is known.
- 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:

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

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,

$$WT_s = BT_s - ETA_s, \quad \forall\, s \in S \quad (2)$$

The second term in Equation (1) corresponds to the total handling cost for loading/unloading containers. The handling time HT_s depends on the total volume of containers to be loaded on the vessel, the number of quay cranes available at this berth, and the average handling productivity of the cranes. Even though we consider the handling time as input in this work, we can easily extend our formulation to compute the handling time. Without loss of generality, we also introduce the term $f(|BP_s - PBP_s|)$, which will penalize the handling cost based on the absolute difference between the assigned berthing position BP_s and the preferred berthing position PBP_s. In this work, we assign a fix penalty when the BP_s is different than the PBP_s, and zero otherwise.

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.

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

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:

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

subject to the following set of constraints:

$$x_{sbt} \in \{0,\ 1\}, \quad \forall\, s \in S,\ b \in B,\ t \in T \quad (5)$$

The variable x_{sbt} is 1 if vessel s is assigned to berthing position b at berthing time t, and 0 otherwise.

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

Constraint (6) ensures that each arrived ship at the MCT will be assigned at a particular berthing position only once during the planning time.

$$\sum_{s' \neq s \in S} \sum_{b=BP_s-L_{s'}+1}^{BP_s+L_s} \sum_{t=BT_s-HT_{s'}+1}^{BT_s+HT_s} x_{s'bt} = 0, \quad \forall\, s \in S, \tag{7}$$

Constraint (7) 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 $300m$. According to constraint (7), 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 $300m$ to $500m$ in the time interval $2h$ to $5h$ as it would overlap with ship s. Visually, this constraint ensures that any two rectangles (denoting the time intervals and berthing positions allocated to vessels) shown in Figure 3 can never overlap.

$$ETA_s \leq BTs, \quad \forall\, s \in S. \tag{8}$$

Constraint (8) warrants that the scheduled berthing time BT_s of ship s must always be later than or equal to its planned time of arrival ETA_s.

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

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.

3 PROPOSED METHODOLOGY

A metaheuristic-based cuckoo search algorithm (CSA) was developed to solve the DC-BAP. In the last decade, CSA has been applied in several domains of computational intelligence and optimization, where it demonstrated promising efficacy. For instance, CSA is implemented for task scheduling in cloud computing (Agarwal and Srivastava, 2018), estimating solar panel model parameters (Chen and Yu, 2019; Kang et al., 2018), electricity load forecasting (Dong et al., 2018), optimal placement of actuators problem (Yang et al., 2018), various scheduling problems (Tein and Ramli, 2010; Chandrasekaran and Simon, 2012; Aslam et al., 2017a,b), and constrained optimization problems (Valian et al., 2013).

CSA is a swarm-based metaheuristic optimization algorithm that was developed by Yang and Deb (2009). 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 (Yang and Deb, 2009):

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, as depicted in Figure 2. 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 shows the procedure of the Cuckoo Search Algorithm, which starts 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 will be performed first, where new solutions are generated by replacing some existing eggs with cuckoo eggs in randomly selected nests (lines #3-8). The replacement follows the rationale that if a cuckoo egg is very similar to a host

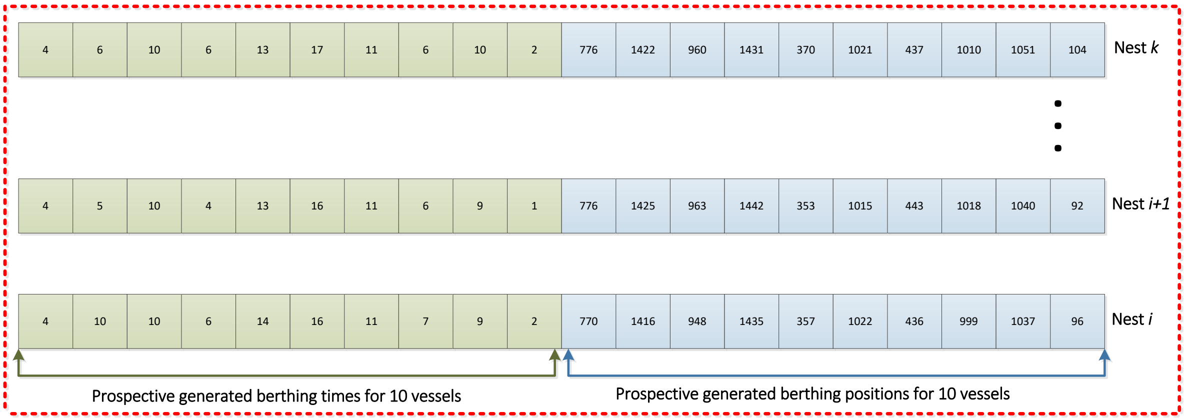


Figure 2: Example solution sets (nests) generated by cuckoo search.

Algorithm 1: Cuckoo Search Algorithm for DC-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**

egg, then this egg has lesser chances to be discovered. Therefore, a random walk is performed through Lévy flights in order to generate new nests (i.e., new solutions).

$$X_i^{(t+1)} = X_i^{(t)} + \alpha \oplus Levy(\lambda), \qquad (10)$$

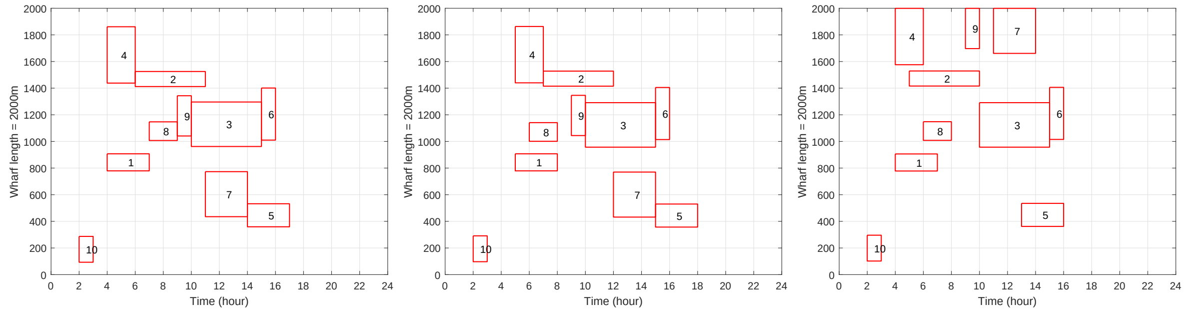
where t denotes the current iteration number, X_i the solution for nest i, and α ($\alpha > 0$) the step size. The $\oplus$ operation denotes entry-wise multiplication. A random walk in Lévy flights is performed from a Lévy distribution with a scale parameter λ (Sanajaoba and Fernandez, 2016). The primary objective 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. Note that the fitness of a possible solution is evaluated based on the objective function of the DC-BAP presented in Equation (4) and accounts for the total processing cost (i.e., a combination of handling, waiting, and late departure penalty costs). Furthermore, some cuckoo eggs may be discovered by host birds. These bad nests will be abandoned and new ones are built so as to enhance the exploration of the search space (lines #9-13). The discovering probability p_α in our work is set to 0.45 as reported in Yang and Deb (2009). Finally, we keep track of the best solution (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.

4 EXPERIMENTAL RESULTS

This section presents the experimental setup, experimental data, and results of the experiments. In addition to the CSA algorithm, we have implemented a state-of-the-art approach (i.e., GA) proposed in recent literature (Salhi et al., 2019) as well as the exact MILP approach. For our experiments, the problem dataset was taken from Şahin and Kuvvetli (2016) and contains ten arriving vessels in a day along with the estimated time of arrival, handling time, estimated time of departure, preferred berthing position, and length for each arriving vessel. The complete dataset is shown in Table 2. The wharf is continuous with a length of 2000m. The GA parameters of maximum generations, population size, crossover rate, and mutation rate are set to 500, 200, 0.5, and 0.4, respectively (Salhi et al., 2019). The implemented algorithms are coded in MATLAB 2019b on a Windows 10 PC with COREi7 processor and 8GB RAM.

Figure 3 presents the berth allocation solutions that are generated by the three implemented algorithms, namely, CSA, GA, and MILP. All three approaches allocate ships over available berthing positions and time slots based on the primary objective of this study, which is to minimize total processing cost as presented in Equation (4). In Figure 3, the vertical

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS



(a) (b) (c)

Figure 3: Berth allocation solution generated by (a) CSA, (b) GA, and (c) MILP. Each rectangle denotes the time intervals and berthing positions allocated to a particular vessel.

Table 2: Dataset used for simulations (Şahin and Kuvvetli, 2016).

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	1	3	102	194

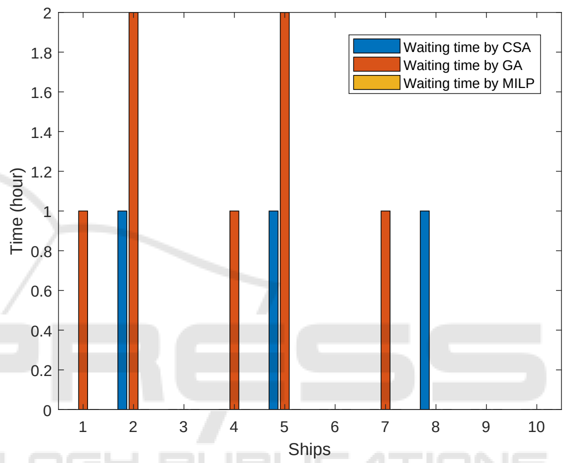


Figure 4: Waiting times when using CSA, GA, and MILP for each vessel.

SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS

axis shows the berthing positions, while the horizontal axis presents time divided into hourly time intervals. Each rectangle in this figure denotes the berthing time periods and berthing positions allocated to an incoming vessel. The label within a rectangle shows the vessel index. For instance, vessel 2, which arrives at time period 5, is assigned to berthing time period 5 by MILP, time period 6 by CSA, and time period 7 by GA. Hence, with MILP, vessel 2 incurs no waiting time, while with CSA, it will have to wait for 1 hour as its preferred berthing position is allocated to vessel 4 (which is optimally assigned). Despite the 1 hour berthing delay, vessel 2 will still achieve its requested time of departure. On the other hand, with GA, vessel 2 will have to wait for 2 hours to berth, which in turn will cause 1 hour delay in its departure. All three methods assign berthing position 1416, which corresponds to the preferred berthing position of vessel 2.

The waiting time for each vessel for the three methods is presented in Figure 4. It can be observed from Figure 4 that no vessel needs to wait for its optimal berthing position (PBP) by employing MILP. However, our newly proposed CSA provides a solution with a maximum waiting time of one time period (hour) for any vessel, while with GA the waiting time for any vessel is up to 2 hours. Furthermore, using CSA, only three vessels need to wait for their optimal berth position assignment; on the contrary, using GA, five vessels need to wait for optimal assignment of berthing position. From this figure, we can conclude that the CSA outperforms GA in terms of reducing waiting time.

Results presented in Figure 5 show the requested departure times for all arriving vessels and the planned departures given by the three implemented algorithms, i.e., CSA, GA, and MILP. It can be clearly seen from this figure that no ship departs late when MILP is used, while only one vessel (vessel 5) is late for one hour with CSA. On the contrary, when GA is employed, two vessels depart late: vessel 2 is late for one hour and vessel 5 for two hours. Once again, we conclude that CSA exhibits higher performance in terms of minimizing late departures compared to GA.

For comparison purposes, Table 3 lists computa-

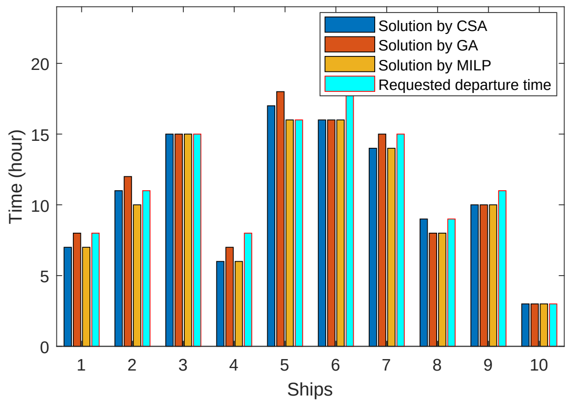


Figure 5: Requested departure time and planned departure times when using CSA, GA, and MILP for each vessel.

tional times along with total processing costs in Euro for all tested methods. Furthermore, we also varied the number of arriving vessels in order to investigate the scalability of the proposed method. We have tested the three methods on three datasets, where 10, 20, and 30 ships are considered, while all other parameters are the same, i.e., length of wharf, the vessel arrival pattern, and berthing layout. It can be seen from Table 3 that MILP gives the optimal solution in terms of minimum processing cost in all cases. For example, results from the 10 vessels show that MILP, as expected, achieves the lowest cost with 275 Euro, followed closely by CSA with 285 Euro, and GA with a significantly higher cost of 310 Euro. A similar pattern is observed for the 20- and 30-vessel scenarios; the solutions proposed by CSA are only slightly costlier (3-5%) than the optimal MILP solution, while the GA solution is much costlier (8-16%).

Table 3: Comparative analysis.

Method	**Total Cost (Euro)**			**Comp. Time (sec)**		
# ships:	10	20	30	10	20	30
CSA	285	580	750	0.09	0.87	1.83
GA	310	635	785	0.08	0.55	1.05
MILP	275	550	725	28.35	47.22	85.02

In terms of computation times, CSA and GA exhibit very similar behaviors, both in terms of absolute numbers and scalability. Specifically, both approaches are extremely efficient with sub-second times for the 10- and 20-vessel cases, and take less than 2 seconds for the 3-vessel case. However, the computation times for MILP are about *two orders of magnitude* higher compared to CSA and GA. While it only takes 1.83 seconds for CSA to find a near-optimal solution for the 30-vessel case, it takes 85 seconds for MILP to find the optimal solution, which is only 3% cheaper than the CSA solution. It is important to note that the above problem sizes are fairly small. For larger, more realistic problem sizes, MILP cannot be used as it has been reported to take over 100 hours of CPU time for real-world instances (Salhi et al., 2019). Such times are certainly not acceptable in the context of MCT operations. From the above discussion, it can be observed that the CSA provides near-optimal solutions in affordable computation times.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we have focused on the continuous berth allocation problem with dynamic vessel arrivals. We have developed a metaheuristic cuckoo search algorithm (CSA) to solve the DC-BAP. Furthermore, two benchmark schemes, a well-known metaheuristic GA and an exact approach (MILP), have also been implemented for comparison purposes. A benchmark dataset is employed from recent literature for the experiments. The results show that our proposed algorithm has higher efficiency over GA in terms of minimum processing cost for all arriving vessels. The newly developed CSA outperforms GA by 8.1%, 8.7%, and 4.5% in terms of reduced cost considering 10, 20, and 30 vessels, respectively. Compared to MILP, our proposed CSA provides a near-optimal solution (<5% from the optimal) at a fraction of computation time. Hence, we conclude that our proposed CSA is an efficient algorithm for near-optimal berth allocation with minimum computational complexity.

In the future, we plan to examine the performance of the proposed CSA method on larger real-world datasets containing several vessels and spanning longer planning time periods (days to weeks). We also plan to extend the modeling to incorporate a hybrid berthing layout that includes both discrete and continuous berthing layouts. Finally, we plan to investigate the application of the CSA in solving the berth allocation problem combined with the related quay crane assignment and scheduling problems.

ACKNOWLEDGEMENTS

This work was supported by the European Regional Development Fund and the Republic of Cyprus through the Cyprus Research and Innovation Foundation (STEAM Project: INTEGRATED/0916/0063).

REFERENCES

Agarwal, M. and Srivastava, G. M. S. (2018). A cuckoo search algorithm-based task scheduling in cloud computing. In *Advances in Computer and Computational Sciences*, pages 293–299. Springer.

Aslam, S., Bukhsh, R., Khalid, A., Javaid, N., Ullah, I., Fatima, I., and Hasan, Q. U. (2017a). An efficient home energy management scheme using cuckoo search. In *International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, pages 167–178. Springer.

Aslam, S., Iqbal, Z., Javaid, N., Khan, Z. A., Aurangzeb, K., and Haider, S. I. (2017b). Towards efficient energy management of smart buildings exploiting heuristic optimization with real time and critical peak pricing schemes. *Energies*, 10(12):2065.

Aslam, S., Michaelides, M. P., and Herodotou, H. (2020). Internet of ships: A survey on architectures, emerging applications, and challenges. *IEEE Internet of Things Journal*, 7:9714–9727.

Barbosa, F., Rampazzo, P. C. B., Yamakami, A., and Camanho, A. S. (2019). The use of frontier techniques to identify efficient solutions for the berth allocation problem solved with a hybrid evolutionary algorithm. *Computers & Operations Research*, 107:43–60.

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

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

Cahyono, R. T., Flonk, E. J., and Jayawardhana, B. (2019). 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.

CargoSmart.ai (2019). Vessels arrivals at China Ports. https://www.cargosmart.ai/en/blog/vessels-arrive-at-top-china-ports-with-shorter-delays-in-2019/.

Carlo, H. J., Vis, I. F., and Roodbergen, K. J. (2015). Seaside operations in container terminals: literature overview, trends, and research directions. *Flexible Services and Manufacturing Journal*, 27(2-3):224–262.

Chandrasekaran, K. and Simon, S. P. (2012). Multi-objective scheduling problem: hybrid approach using fuzzy assisted cuckoo search algorithm. *Swarm and Evolutionary Computation*, 5:1–16.

Chen, L. and Huang, Y. (2017). 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.

Chen, X. and Yu, K. (2019). Hybridizing cuckoo search algorithm with biogeography-based optimization for estimating photovoltaic model parameters. *Solar Energy*, 180:192–206.

De, A., Pratap, S., Kumar, A., and Tiwari, M. (2020). 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.

Dong, Y., Zhang, Z., and Hong, W.-C. (2018). A hybrid seasonal mechanism with a chaotic cuckoo search algorithm with a support vector regression model for electric load forecasting. *Energies*, 11(4):1009.

Dulebenets, M. A. (2017). 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.

Frojan, P., Correcher, J. F., Alvarez-Valdes, R., Koulouris, G., and Tamarit, J. M. (2015). The continuous berth allocation problem in a container terminal with multiple quays. *Expert Systems with Applications*, 42(21):7356–7366.

Han, X., Gong, X., and Jo, J. (2015). A new continuous berth allocation and quay crane assignment model in container terminal. *Computers & Industrial Engineering*, 89:15–22.

Hsu, H.-P., Chiang, T.-L., Wang, C.-N., Fu, H.-P., and Chou, C.-C. (2019). A Hybrid GA with Variable Quay Crane Assignment for Solving Berth Allocation Problem and Quay Crane Assignment Problem Simultaneously. *Sustainability*, 11(7):2018–2038.

Hsu, H.-P., Wang, C.-N., Chou, C.-C., Lee, Y., and Wen, Y.-F. (2017). Modeling and solving the three seaside operational problems using an object-oriented and timed predicate/transition net. *Applied Sciences*, 7(3):218.

Jos, B. C., Harimanikandan, M., Rajendran, C., and Ziegler, H. (2019). Minimum cost berth allocation problem in maritime logistics: new mixed integer programming models. *Sādhanā*, 44(6):149.

Kang, T., Yao, J., Jin, M., Yang, S., and Duong, T. (2018). A novel improved cuckoo search algorithm for parameter estimation of photovoltaic (pv) models. *Energies*, 11(5):1060.

Kavoosi, M., Dulebenets, M. A., Abioye, O., Pasha, J., Theophilus, O., Wang, H., Kampmann, R., and Mikijeljević, M. (2019). Berth scheduling at marine container terminals. *Maritime Business Review*, 5.

Lind, M., Michaelides, M., Ward, R., Herodotou, H., and Watson, R. (2019). Boosting data-sharing to improve short sea shipping performance: Evidence from limassol port calls analysis. Technical Report 35, UNCTAD Transport and Trade Facilitation Newsletter No. 82 - Second Quarter.

Mauri, G. R., Ribeiro, G. M., Lorena, L. A. N., and Laporte, G. (2016). An adaptive large neighborhood search for the discrete and continuous berth allocation problem. *Computers & Operations Research*, 70:140–154.

Michaelides, M. P., Herodotou, H., Lind, M., and Watson, R. T. (2019). Port-2-port communication enhancing short sea shipping performance: The case study of cyprus and the eastern mediterranean. *Sustainability*, 11(7):1912.

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

Salhi, A., Alsoufi, G., and Yang, X. (2019). 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.

Sanajaoba, S. and Fernandez, E. (2016). Maiden application of cuckoo search algorithm for optimal sizing of a remote hybrid renewable energy system. *Renewable energy*, 96:1–10.

Tein, L. H. and Ramli, R. (2010). Recent advancements of nurse scheduling models and a potential path. In *Proc. 6th IMT-GT Conference on Mathematics, Statistics and its Applications (ICMSA 2010)*, pages 395–409.

UNCTAD (2017). UNCTAD: Ports in 2017. https://unctad.org/system/files/official-document/rmt2018ch4_en.pdf.

Valian, E., Tavakoli, S., Mohanna, S., and Haghi, A. (2013). Improved cuckoo search for reliability optimization problems. *Computers & Industrial Engineering*, 64(1):459–468.

Xu, Y., Xue, K., and Du, Y. (2018). Berth scheduling problem considering traffic limitations in the navigation channel. *Sustainability*, 10(12):4795–4816.

Yang, B., Miao, J., Fan, Z., Long, J., and Liu, X. (2018). Modified cuckoo search algorithm for the optimal placement of actuators problem. *Applied Soft Computing*, 67:48–60.

Yang, X.-S. and Deb, S. (2009). Cuckoo Search via Lévy Flights. In *World Congress on Nature & Biologically Inspired Computing (NaBIC)*, pages 210–214. IEEE.

SCIENCE AND TECHNOLOGY PUBLICATIONS