Applied Intelligence
https://doi.org/10.1007/s10489-020-02062-y

A novel mathematical formulation for solving the dynamic and discrete berth allocation problem by using the Bee Colony Optimisation algorithm

Luigi Pio Prencipe[1] · Mario Marinelli[2]

Accepted: 4 November 2020

Abstract

Berth allocation is one of the crucial points for efficient management of ports. This problem is complex due to all possible combinations for assigning ships to available compatible berths. This paper focuses on solving the Berth Allocation Problem (BAP) by optimising port operations using an innovative model. The problem analysed in this work deals with the Discrete and Dynamic Berth Allocation Problem (DDBAP). We propose a novel mathematical formulation expressed as a Mixed Integer Linear Programming (MILP) for solving the DDBAP. Furthermore, we adapted a metaheuristic solution approach based on the Bee Colony Optimisation (BCO) for solving large-sized combinatorial BAPs. In order to assess the solution performance and efficiency of the proposed model, we introduce a new set of instances based on real data of the Livorno port (Italy), and a comparison between the BCO algorithm and CPLEX in solving the DDBAP is performed. Additionally, the application of the proposed model to a real berth scheduling (Livorno port data) and a comparison with the Ant Colony Optimisation (ACO) metaheuristic are carried out. Results highlight the feasibility of the proposed model and the effectiveness of BCO when compared to both CPLEX and ACO, achieving computation times that ensure a real-time application of the method.

Keywords Berth allocation problem · Bee colony optimisation · Container terminal · Berth planning · Combinatorial optimisation

1 Introduction

During the last century, global trade and freight growth impose new challenges and requirements for efficient management of transport processes. Therefore, maritime transport is one of the crucial points of intermodal transportation, concerned with the difficulties of developing more efficient port operations. This paper focuses on solving the Berth Allocation Problem (BAP) by optimising port operations using an innovative model. In the literature, this problem has been studied by several researchers using different approaches. Basically, the BAP is a complex operations research problem based on the assignment of ships to berth areas along a quay [41]. Typically, ships arrive at a port in a specific time window, and port operators indicate their available berths. Models applied to solve the BAP can be divided according to variables related to space and time.

Spatial variables are related to the quay. According to Imai et al. [27] and Bierwirth and Maisel [1], there are three different cases of spatial variables: discrete layout (BAPD), continuous layout (BAPC), and hybrid layout (HBAP). The BAPD is the simplest and the most used model for berth planning. In this model, a quay is divided into several berths with specific lengths, and each berth can serve only one ship at a time. In the BAPC, the quay is not divided into several berths, and each ship can be served in any place without overlapping. In the HBAP, the quay is similar to the BAPD, but ships can occupy a few or more berths according to their length. Recently, Kovač et al. [32] have solved the HBAP, formulated as the Minimum Cost Hybrid Berth Allocation, by using four different metaheuristic approaches, i.e., Evolutionary Algorithms (EAs), Bee Colony Optimisation (BCO), Variable Neighborhood Search (VNS), and the General Variable Neighborhood Search (GVNS).

Temporal variables divide the BAP in two cases related to the ships' arrival time: Static arrival (SBAP) and Dynamic arrival (DBAP) time [25]. In the SBAP, there are no expected arrival times, so it is assumed that all ships are in the harbor at

✉ Luigi Pio Prencipe
luigipio.prencipe@poliba.it

[1] D.I.C.A.T.E.Ch, Polytechnic University of Bari, Via Orabona 4, Bari, Italy

[2] Department of Engineering, University of Sannio, Piazza Roma 21, Benevento, Italy

Published online: 01 December 2020

the beginning of the planning horizon, waiting to be anchored. Thus, the ships' arrival time puts a single constraint on the berthing times, with the possibility of speeding up the specific operations before the expected arrival time. In the DBAP, arrival times are scheduled and assigned to each ship, in order to avoid overlapping. Recently, Kramer et al. [34] have proposed two novel formulations: a time-indexed formulation and an arc-flow formulation for solving a big-size DBAP considering 250 ships and 20 berths instances in a reasonable computation time. Generally, various models aim at minimising the sum of the ships' waiting time or the total staying time for completing the activities, increasing the attractiveness and efficiency of container terminal ports.

Another classification of BAP is formulated including the synchronisation between the BAP and the Quay Cranes Assignment Problem (QCAP), and it has been treated by several authors in the literature ([21, 40, 58, 60, 61]).

The problem formulated in this work deals with the Discrete and Dynamic Berth Allocation Problem (DDBAP). Generally, container terminals are highly dynamic systems, and terminal managers have to decide different port operations in short periods [56]. The right decision-making in a short time basis is crucial for port management optimisations. For this reason, an efficient optimisation algorithm that will provide an optimal berth assignment solution in real-time is a current challenge. Umang et al. [55] proposed a real-time formulation for solving the BAP with stochastic arrival and handling time on a rolling planning horizon aiming at minimising the total realised cost of the updated berthing schedule.

Heuristics and metaheuristics are the most used approaches for solving the DDBAP. Metaheuristics are general frameworks used to build heuristic algorithms for NP-hard optimisation problems in reasonable computation time. In the literature, several metaheuristic approaches were proposed, such as Genetic Algorithms [18, 28, 53], Particle Swarm Optimisation [54], Tabu Search [35] and Variable Neighborhood Search [23].

This paper proposes a novel mathematical formulation expressed as a Mixed Integer Linear Programming (MILP) for solving the DDBAP, in order to find an optimal berth assignment for a given schedule. We applied a Swarm Intelligence algorithm, the Bee Colony Optimisation (BCO), since it resulted in being very effective in solving large-sized combinatorial problems [6, 44, 52] and, to the best of our knowledge, it has not been applied to the DDBAP yet. In order to assess the performance of the model, we introduced a new set of instances based on Livorno port real data. We compared the exact solutions obtained for the proposed MILP formulation by using CPLEX with near-optimal solutions obtained by the proposed BCO algorithm. Furthermore, we applied the proposed model to a real case study (Livorno port in Italy), and we compared the results obtained by the BCO with the Ant Colony Optimisation (ACO) metaheuristic, another well-known Swarm Intelligence algorithm.

The paper is organised as follows. Section 2 presents the literature review related to the DDBAP. Section 3 introduces the proposed MILP formulation, while Section 4 describes the proposed metaheuristic based on the BCO algorithm for solving the DDBAP. Section 5 is devoted to a numerical application for comparing BCO solutions with CPLEX exact ones. Section 6 shows the application to a real case study and compares results obtained by both BCO and ACO metaheuristics. Finally, Section 7 gives concluding remarks and proposes some future developments.

2 Literature review on the DDBAP

A brief introduction of BAP has already been presented in Section 1, while in this section we aim to present a brief state-of-the-art focused on the Dynamic variant of the Discrete BAP. The most recent BAP survey paper in the literature is presented by Bierwirth and Maisel [2].

A state-of-the-art on the dynamic variants of the DBAP is necessary for highlighting the contribution of the proposed work. Since DDBAP is an NP-hard problem due to the formulation structure and to the problem size [26, 27, 48], it is necessary to apply heuristic or metaheuristic methods as a resolution approach to obtain near-optimal solutions. A summary of proposed DDBAP mathematical formulations and algorithm approaches is needed. Imai et al. [25] were the first authors who addressed the DDBAP by using a Lagrangian relaxation heuristic as a resolution approach for solving numerical experiments. Hansen and Oğuz [22] implemented a DDBAP equivalent Mixed Integer Program (MIP) correcting and extending the model proposed by Imai et al. [25]. They provided a VNS metaheuristic for solving large instances to near optimality. Thereafter, Hansen et al. [23] extended the mathematical formulation proposed by Hansen and Oğuz [22] in terms of objective function, variables, and constraints introducing the Minimum Cost Berth Allocation Problem (MCBAP) considering dynamic ships' arrival and discrete quay division with the aim to minimise ships' waiting and handling times. They proposed a VNS metaheuristic for solving the MCBAP medium and large size instances and demonstrated that VNS outperformed other three metaheuristics, i.e. Multi-Start, Genetic Algorithms (GAs) and Memetic search algorithm (MA). Monaco and Sammarra [45] improved the DDBAP mathematical formulation by Imai et al. [25] introducing a compact and stronger formulation dealing with the same problem. They used a Lagrangian heuristic algorithm for solving large-sized instances reaching near-optimal solutions in a reasonable computation time. Issam et al. [29] extended the Imai's DDBAP mathematical formulation as a multi-objective MILP including ships' draft and greenhouse gas emissions evaluation. Xu et al. [59] proposed a MILP formulation for solving the DDBAP and the SDBAP considering water depth and tidal effect with the aim to minimise the ships' completion time. They developed two heuristic algorithms to solve the dynamic and static DBAP, respectively, and they tested the model performance on numerical experiments. Recently, Lassoued and Elloumi [38] proposed a MILP formulation for optimising the DDBAP in bulk ports. They compared the proposed MILP model with a real case study dataset by using CPLEX solver on small instances.

The most applied algorithm approaches are GA-based metaheuristics for solving large and real data instances. Nishimura et al. [47] were the first who applied GAs to address the DDBAP. Imai et al. [26] extended the formulation proposed by Imai et al. [25] including service priority constraints and applying GAs metaheuristic as the solution approach. Han et al. [20] implemented a non-linear mathematical formulation to minimise the total turnaround time and to improve the terminal operation efficiency by using GAs and a hybrid optimisation strategy combining GA and Simulated Annealing (GASA), respectively. Zhou et al. [62] proposed a non-linear programming model considering ships' total waiting and service times as stochastic parameters. They used a GAs and designed a greedy algorithm for improving ships' assignment to available berths. Golias et al. [14] proposed a MILP formulation with two objective functions for minimising ships' late departure costs and for maximising ships' early and timely departure benefits within a time window. Imai et al. [28] proposed a two-objective mathematical formulation considering ships' service quality and berths' utilisation objective functions by minimising delay in ships' departure and total service time, respectively. They applied GAs and Subgradient Optimisation (SP) approaches for solving the DDBAP. Theofanis et al. [53] implemented a linearised version of the DDBAP formulation proposed in Imai et al. [26], and they compared two heuristics, i.e. GAs and the Optimisation-Based GA (OBGA), on a small dataset. Golias et al. [15] proposed a multi-objective combinatorial optimisation problem considering ships' priority for minimising the total service time. They developed a GA-based metaheuristic to solve large and real data instances. Golias et al. [19] proposed an additional non-linear MIP formulation for minimising ships' emissions, waiting time, and delayed departures by using a GA-based metaheuristic to solve real data instances. Saharidis et al. [49] formulated a bi-level optimisation problem introducing a hierarchy among the different objective functions. They used an improved interactive GA-based metaheuristic, i.e. the *k*-th best algorithm, for solving real data instances. Golias et al. [16] proposed a lamda-optimal based heuristic algorithm, namely the Defined Neighborhood heuristic, for solving berth scheduling subproblems minimising the total weighted service time. They also presented a GAs metaheuristic as an alternative solver for the cases in which no-efficient objective optimiser is available to solve sub-problems. Golias and Haralambides [17] proposed a non-linear DDBAP mathematical formulation for minimising ships' late departure total costs and waiting time, and for maximising ships' early departure total premiums simultaneously. They assumed different variable penalty/premium costs functions based on contractual agreements between the liner shipping company and the terminal operator. Additionally, they used a GA-based metaheuristic algorithm to solve computational examples based on four different berth scheduling policies. Golias et al. [18] proposed a bi-level bi-objective optimisation problem for minimising the average and range of total service times depending on ships' arrival and handling time uncertainties. They developed a GA-based metaheuristic algorithm for solving big-sized numerical experiments. Simrin and Diabat [51] proposed a non-linear MIP for solving the DDBAP with the aim to minimise the ships' total handling and waiting times. Moreover, they linearised the previous model and developed a GA metaheuristic as a resolution approach for solving small and large set of instances.

In the literature, other authors dealt with the DDBAP by using different metaheuristic approaches. Cordeau et al. [4] proposed two mathematical formulations considering both discrete and continuous cases with the aim to minimise the ships' total weighted service time. They developed a Tabu Search (TS) metaheuristic for solving discrete and continuous cases for big-sized instances while they applied CPLEX solver for small-sized instances. Buhrkal et al. [3] improved the mathematical formulation by Cordeau et al. [4] by introducing a Generalized Set Partitioning Problem (GSPP) mathematical formulation. Lalla-Ruiz et al. [34] proposed a hybrid heuristic combining a TS metaheuristic with an artificial intelligence method, namely the Path Relinking algorithm, for solving the DDBAP. They compared exact solutions of the GSPP mathematical formulation proposed by Buhrkal et al. [2] and near-optimal solutions of the TS metaheuristic proposed by Cordeau et al. [4]. Furthermore, de Oliveira et al. [4] proposed an alternative Clustering Search method using the Simulated Annealing (SA) metaheuristic applied on the mathematical formulation by Cordeau et al. [4]. Lin and Ting [42] proposed two versions of the SA algorithm, i.e., with and without restarting strategy, for solving the DDBAP by using the mathematical formulation by Buhrkal et al. [3]. They tested both SA algorithms on several small and medium instances in the literature. Lin et al. [43] proposed the Iterated Greedy (IG) heuristic for solving the DDBAP minimising ships' total service time. Lee and Jin [39] dealt with the DDBAP considering the optimisation between feeder ships schedule and terminal operations in a container transhipment terminal. They proposed a mixed integer programming model and a MA approach, i.e. the combination between GAs and TS.

Karafa et al. [31] proposed a modified version of the DDBAP model proposed by Golias et al. [16] considering ships' handling time as a stochastic parameter. They formulated a bi-objective optimisation problem with the aim to minimise the berth schedule risk and to maximise the berth throughput by using an Evolutionary EA-based heuristic and a simulation-based Pareto front pruning algorithm. Dulebenets [9] proposed a non-linear MIP model for solving the DDBAP with the aim to minimise the ships' total weighted service time. He developed a modified version of MA with a Deterministic Parameter Control (MA-DPC) algorithm and compared it with other four metaheuristics, i.e., MA, EA, SA, and VNS algorithms by testing it on numerical instances. Furthermore, Dulebenets et al. [13] proposed a MILP model for solving the DDBAP minimising the ships' total service cost, including the total carbon dioxide emission cost. They developed a Hybrid Evolutionary Algorithm (HEA) for solving a set of numerical instances. Dulebenets [10] proposed a MILP for solving the DDBAP including service priority with the aim to minimise the ships' total weighted service cost. He developed an Adaptive Evolutionary (AEA) metaheuristic with a parameter control strategy to test the model effectiveness through extensive numerical experiments. Dulebenets et al. [12] proposed a MILP model aimed at minimising the ships' total weighted turnaround time and the ships' total weighted late departure. They improved the previous AEA algorithm proposed by Dulebenets [12] with a Self-Adaptive Evolutionary Algorithm (SAEA). Additionally, Dulebenets [11] proposed a MILP model aiming at minimising the ships' total weighted service cost. He developed an Adaptive Island Evolutionary Algorithm (AIEA) which can execute simultaneously multiple EAs in parallel on its islands based on an adaptive mechanism. Ting et al. [54] proposed a MIP model for solving the DDBAP as a vehicle routing type problem with the aim to minimise the ships' total service time. They developed a Particle Swarm Optimisation (PSO) metaheuristic algorithm to test and validate their model through two different size sets of benchmark instances in the literature. Lalla-Ruiz and Voß [36] and Lalla-Ruiz and Voß [37] proposed a Partial Optimisation Metaheuristic Under Special Intensification Conditions (POPMUSIC) metaheuristic for solving the DDBAP applying the mathematical formulation proposed by Cordeau et al. [4]. They proposed two variants of POPMUSIC metaheuristic, i.e. POPMUSIC and POPMUSIC-G, applying different initial solution methods, i.e. Random-Greedy method and First-Come First-Served Greedy, respectively. Recently, Nishi et al. [46] proposed a Dynamic Programming-Based Matheuristic (DP-Math) based on POPMUSIC metaheuristic to solve the DDBAP proposed by Cordeau et al. [4]. They compared DP-Math metaheuristic with GSPP and dynasearch algorithms on large scale instances. Issam et al. [30] proposed a DDBAP mathematical formulation with the aim to minimise the ships' total staying time at a container terminal. They developed a Modified Sailfish Optimiser (MSFO) metaheuristic and compared MSFO with other heuristic-based approaches in the literature on a set of instances proposed by Cordeau et al. [4]. Wang et al. [57] proposed a DDBAP mathematical formulation minimising the ships' total service time considering tidal effects. They developed a Levy-Flight (LF) metaheuristic to solve the proposed model and compared LF with other state-of-the-art exact and heuristic methods on large scale instances.

In Table 1, we report a summary of the state-of-the-art as an extension of the survey presented by Kovač [32], in which main differences in models and metaheuristic algorithms proposed for solving the DDBAP are highlighted.

According to Table 1, the proposed mathematical formulation can be classified as discrete (*disc*), and draft ships dependent (*draft*) as a spatial attribute; dynamic (*dyn*), and due time-dependent (*due*) as a temporal attribute; and position-dependent handling time (*pos*) as a handling time attribute. Among the above-mentioned authors, Zhou et al. [62], Han et al. [20], Xu et al. [59], and Issam et al. [29] have considered draft constraints in their mathematical formulations. Accordingly, as highlighted by Sheikholeslami et al. [50], we considered the compatibility between ships' draft and berths' draft since it is relevant, especially for shallow ports where tide effects and low-draft access (e.g., channel port access) could influence the berth planning negatively.

Thus, considering the existing literature, the contribution of this paper is summarised as follows:

1. A novel mathematical formulation expressed as a MILP for solving the DDBAP was proposed. Additionally, we introduced a new parameter that considers the berth approaching/leaving time, to the best of our knowledge, not considered in the literature yet.
2. A metaheuristic for solving the DDBAP, the BCO, was proposed since there are few Swarm Intelligence algorithms applied for solving the DDBAP in the literature. Furthermore, a new set of benchmark instances, based on Livorno port real data, for the DDBAP was introduced to test the feasibility of the proposed optimisation model by using an exact solver and the BCO.
3. An application to a real case study (Livorno port) considering a long-time horizon (from 1 up to 4 weeks) was introduced. We compared the real monthly berth planning with the solution obtained by the proposed BCO and another well-known Swarm Intelligence method, the ACO, to assess the effectiveness of the proposed approach.

3 Problem description and formulation

In this paper, we propose a novel mathematical formulation for solving the DDBAP, formulated as a MILP. The formulation was obtained starting from the Non-Linear Mixed Integer Programming (NLMIP) model reported in

Table 1 Problems and metaheuristic algorithms for the DDBAP

Problem description	Article	Solution algorithm	Problem dimension		
			# of ships	# of berths	Time horizon
disc | *dyn* | *pos*	Golias et al. [15]	GA	40 to 80	5 to 10	1 to 2 weeks
	Golias and Haralambides [17]	GA	inter arrival time	5	1 week
	Saharidis et al. [49]	GA	50	5 to 10	1 to 2 weeks
	Golias et al. [18]	GA	inter arrival time	4 to 5	1 week
	Ting et al. [54]	PSO	25 to 60	5 to 13	–
	Lalla Ruiz et al. [35]	TS	25 to 60	5 to 13	600 time units
	Golias et al. [16]	GA	40 to 80	5	1 to 2 weeks
	Golias et al. [19]	GA	inter arrival time	5	–
	Hansen et al. [23]	VNS	10 to 200	10 to 20	–
	Theofasis et al. [53]	GA	20–25 per week	5	1 to 2 weeks
	Golias et al. [14]	GA	9	2	–
	Imai et al. [28]	GA	24	4	1 week
	Simrin and Diabat [51]	GA	30 to 80	10 to 30	–
	Dulebenets [9]	MA-DPC	6 to 16	2 to 4	2 weeks
	Dulebenets et al. [13]	HEA	11 to 90	2 to 6	–
	Dulebenets [10]	AEA	5 to 42	4 to 6	–
	Dulebenets et al. [12]	SAEA	5 to 100	2 to 8	–
	Dulebenets [11]	AIEA	6 to 85	2 to 6	–
	Issam et al. [30]	MSFO	60	13	–
disc | *dyn* | *pos, stoch*	Karafa et al. [31]	EA	inter arrival time	5	1 week
disc | *dyn, due* | *pos*	Lalla-Ruiz and Voß [36]	POPMUSIC	40 to 55	5 to 7	600 time units
	Lalla-Ruiz et al. [37]	POPMUSIC-G	60	13	–
	Nishi et al. [46]	DP-Math	150	15	–
	Cordeau et al. [4]	TS	25 to 35	5 to 10	–
	de Oliveira et al. [5]	SA	60	13	–
	Lin and Ting [42]	SA	25 to 35	5 to 10	–
	Lin et al. [43]	IG	25 to 35	5 to 10	–
disc | *dyn, due* | *fix*	Lee and Jin [39]	MA	15 to 40	3 to 8	–
disc, draft | *dyn* | *pos*	Han et al. [20]	GA	7	2	–
disc, draft | *dyn, due* | *pos*	Zhou et al. [62]	GA	25 to 100	5 to 8	–

Appendix 1 in order to provide an equivalent linear model for algorithms comparison. In both formulations, the objective function minimises the total ships' staying time. Our model considers the following assumptions: the port, at its initial state, is empty, and each ship corresponds at least to one compatible berth due to the discrete distribution model of quays. Furthermore, compatibility between ships and berths is related to geometric constraints, such as length and draft. Ships arrive at the port in the harbor area according to scheduled arrival times and arranged in chronological order. Arrival, service, and total ships' staying times are related to a fixed time reference, while berth approaching/leaving, handling, and waiting times are fixed time intervals. Berth approaching/leaving, and handling times are calculated as the average of observed values, respectively. The warehouse capacity for the storage of goods and/or containers is unlimited. The number and performance of cranes are not considered. Finally, ships can move freely within the port: priority and any additional waiting times due to internal congestion have been excluded. The list of notations used for the MILP formulation is reported in Table 2 including notations used in the NLMIP (Appendix A).

All definitions, as depicted in Fig. 1, are given as follows:

- Arrival time: the arrival time of a ship to a given port in the harbor area.
- Waiting time: the time spent by a ship, assigned to a berth, in waiting for the previous ship to leave the same berth.
- Berth approaching/leaving time: the total time necessary for a ship to approach and leave its assigned berth. This

Table 2 List of notations used in problem formulations

Notation	Description
Sets	
S	set of all ships, $S = \{1, 2, \ldots, n_s\}$, ordered according to the schedule
B	set of all berths, $B = \{1, 2, \ldots, n_b\}$
Parameters	
n_s	number of ships in the schedule
n_b	number of berths
l_i^s	length of the i-th ship, $i \in S$
d_i^s	draft of the i-th ship, $i \in S$
l_j^b	length of the j-th berth, $j \in B$
d_j^b	maximum acceptable draft of the j-th berth, $j \in B$
t_i^a	arrival time of the i-th ship, $i \in S$
t_i^h	handling time of the i-th ship, $i \in S$
t_{ij}^b	berth approaching/leaving time of the i-th ship to the j-th berth, $i \in S, j \in B$
M	large positive number
Decision variables	
x_{ij}	binary decision variable of assignment of the i-th ship to the j-th berth; x_{ij}= 1 if the i-th ship anchored to the j-th berth; x_{ij}= 0, otherwise. $i \in S, j \in B$
t_{ij}^w	waiting time auxiliary variable of the i-th ship to the j-th berth, $i \in S, j \in B$
t_{ij}^W	waiting time of the i-th ship to the j-th berth, $i \in S, j \in B$
t_i^B	berth approaching/leaving time auxiliary variable of assignment of the i-th ship, $i \in S$
t_{ij}^s	service time of the i-th ship to the j-th berth, $i \in S, j \in B$
t_{ij}^t	total staying time of the i-th ship to the j-th berth, $i \in S, j \in B$
s_i	leaving time of the i-th ship, $i \in S$

time includes all mooring manoeuvres starting and finishing at the harbor area.

- Handling time: the time spent by port operators for loading or unloading goods of a ship, assigned to a berth.
- Service time: the sum of waiting, berth approaching/leaving, and handling times of a ship assigned to a berth.
- Leaving time: the port leaving time is the sum of arrival and service time of a ship assigned to a berth.
- Total staying time: the cumulative sum of arrival and service times of all ships assigned to their respective berths.

3.1 The MILP mathematical formulation

In order to find the global optimal solution of the DDBAP using an exact solver, we modelled the problem as a MILP whose mathematical formulation is given as follows:

$$\min \sum_{i=1}^{n_s} t_i^t \tag{1}$$

s.t.

$$\sum_{j=1}^{n_b} x_{ij} = 1, \forall j \in B \tag{2}$$

$$s_i = t_i^w + t_i^a, \forall i \in S \tag{3}$$

$$s_i \geq t_i^a, \forall i \in S \tag{4}$$

$$s_i + M \cdot (1 - x_{ij}) \geq s_k + t_k^h + t_{kj}^b - M \cdot (1 - x_{kj}), \forall i \in S, \forall k \in S, \forall j \in B, k > i \tag{5}$$

$$l_i^s \cdot x_{ij} \leq l_j^b, \forall i \in S, \forall j \in B \tag{6}$$

$$d_i^s \cdot x_{ij} \leq d_j^b, \forall i \in S, \forall j \in B \tag{7}$$

$$t_i^B \geq t_{ij}^b \cdot x_{ij}, \forall i \in S, \forall j \in B \tag{8}$$

$$t_i^t = t_i^a + t_i^h + t_i^B + t_i^w, \forall i \in S \tag{9}$$

$$x_{ij} \in \{0, 1\}, \forall i \in S, \forall j \in B \tag{10}$$

The proposed MILP formulation (1)–(10) aims at minimising the total staying time by the objective function (1). Constraints (2) ensure the uniqueness of the assignment. We define the leaving time s_i in constraints (3), while constraints (4) and (5) ensure that s_i must be consistent with all previous arrival times. We introduce a large positive number M in constraints (5) in order to avoid unfeasibility and to faster reach the optimal solution. Constraints (6) and (7) ensure the compatibility between the length and the draft of the ships with respect to those of the berths. Constraints (8) evaluate the berth approaching/leaving time auxiliary variables t_i^B

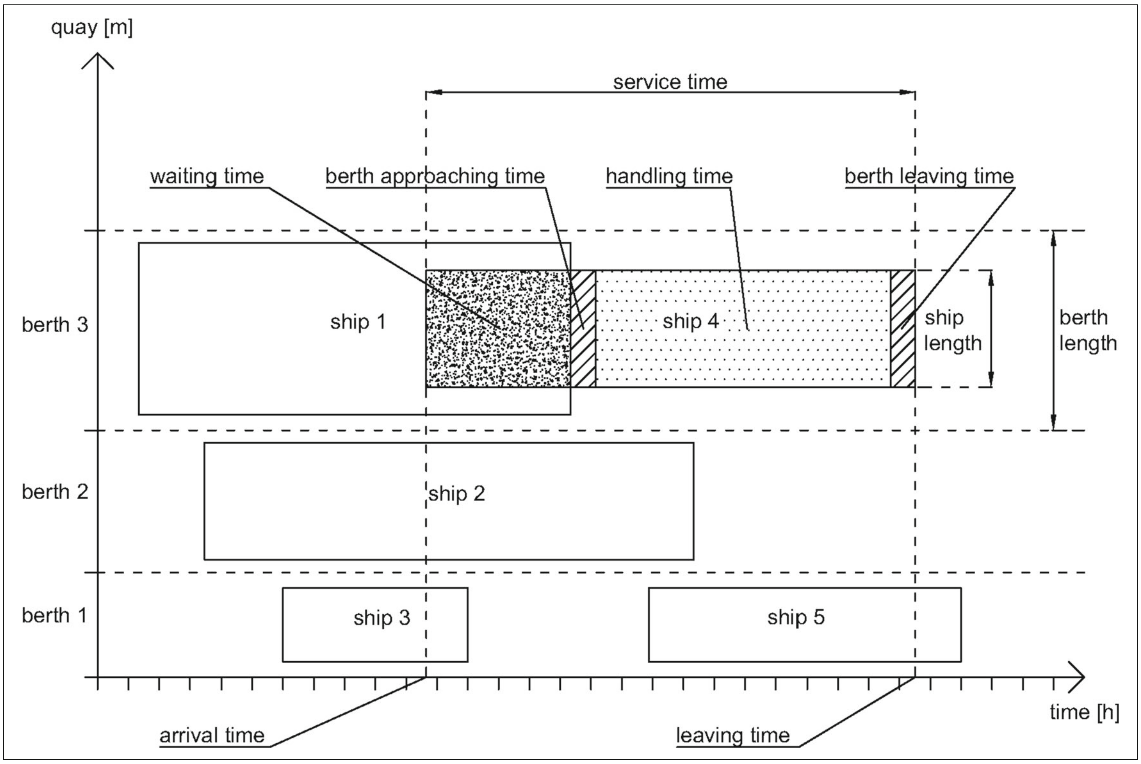

Fig.1 A space-time representation of the proposed DDBAP model

related to the assignment of the i-th ship to the j-th berth, allowed by the right-side term $t_{ij}^{b}\cdot x_{ij}$. Finally, constraints (9) and (10) define the total staying time decision variables t_i^t and the binary nature of the decision variables x_{ij}, respectively.

4 The Bee Colony Optimisation approach

In order to face the complexity of the problem, we applied the BCO metaheuristic proposed by Teodorović et al. [52] to solve the DDBAP.

Generally, the BCO is based on bees' natural collaborative behaviour for solving the complex combinatorial optimisation problem. We used, in this case, artificial bees for performing the search process. Firstly, bees are located in the hive and then they start to interact when the search process is undertaken. Each artificial bee undertakes a set of moves through stages determining a partial solution of the problem. These partial solutions remain until some other suitable ones have been found. At each stage, bees' movements are called forward and backward step. During the forward step, they generate partial solutions through previous individual or collective experiences, while during the backward step, they return to the hive for participating in the decision-making process. At the end of the stages, we completed an iteration generating one or more feasible solutions. The search process continues until a fixed maximum number of iterations is reached. When applied to the DDBAP, each bee (b) represents a solution, i.e., a feasible assignment of ships to available berths. We can represent an assignment matrix using a three-dimensional structure, as shown in Fig. 2a.

The complete list of all parameters used for BCO is reported in Table 3.

In the initialisation, we define the maximum number of iterations (n_k) and the number of bees (n_{bee}). Each solution is constructed through stages following a given time schedule according to the dynamic nature of the problem. Thus, the number of stages (G) is equal to the number of ships (n_s) in the schedule, and a stage represents a single ship-to-berth assignment. As stated before, each stage is composed of a forward and backward step. During a forward step, each bee finds partial assignments of the ships to the berths. All the partial solutions are identified observing the constraints of the optimisation problem, and the associated fitness value is given by the objective function (1). During a backward step, each bee could improve its partial solution or move to a better one. At the end of each iteration, a bee constructs a feasible solution. Then, a new iteration starts searching for new solutions until the maximum number of iterations is reached. The flowchart of the proposed method is shown in Fig. 2b. In Algorithm 1, we reported the proposed BCO pseudo-code to better understand the forward/backward process related to each iteration k of the algorithm.

Algorithm 1. Pseudo-code applied to solve the DDBAP

```
Initialisation. An empty solution is assigned to each bee. Set $n_k, n_{bee}, n_c, n_r$
procedure
    for each iteration $k \leftarrow 1, n_k$ do
        for each stage $u \leftarrow 1, n_s$ do
            for each bee $b \leftarrow 1, n_{bee}$ do
                // forward step //
                for each berth $j \leftarrow 1, n_c$ do
                    evaluate the fitness of each berth according to model (1)-(10)
                    $j=j+1$
                end for
                // backward step //
                choose a partial solution through the recruiting process according to Eq.(11)-(13)
                $b=b+1$
            end for
            $u=u+1$
        end for
        save the best solution with the highest fitness
        $k=k+1$
    end for
    save the global solution
end procedure
```

As a result, a path of the created artificial network corresponds to a berth assignment found by a bee. At the end of each iteration, all the possible solutions found are evaluated, referring to the associated fitness value, and the best assignment is saved. Then, a new iteration starts searching for new solutions until the maximum number of iterations n_k is reached [44]. We can represent the decision space as an artificial network composed of nodes and stages, as shown in Fig. 3a and b.

The roulette wheel selection [24], applied to the BCO algorithm, is a common algorithm used to select an item proportional to its probability. The main inspiration for its development derives from the example of roulette used in gambling. Each solution can be chosen, and the probability of selection (the size of a roulette slot) depends on its fitness value, which is one value of the objective function. In practice, the size of the circular sector of the roulette associated with each solution is determined by the ratio between the

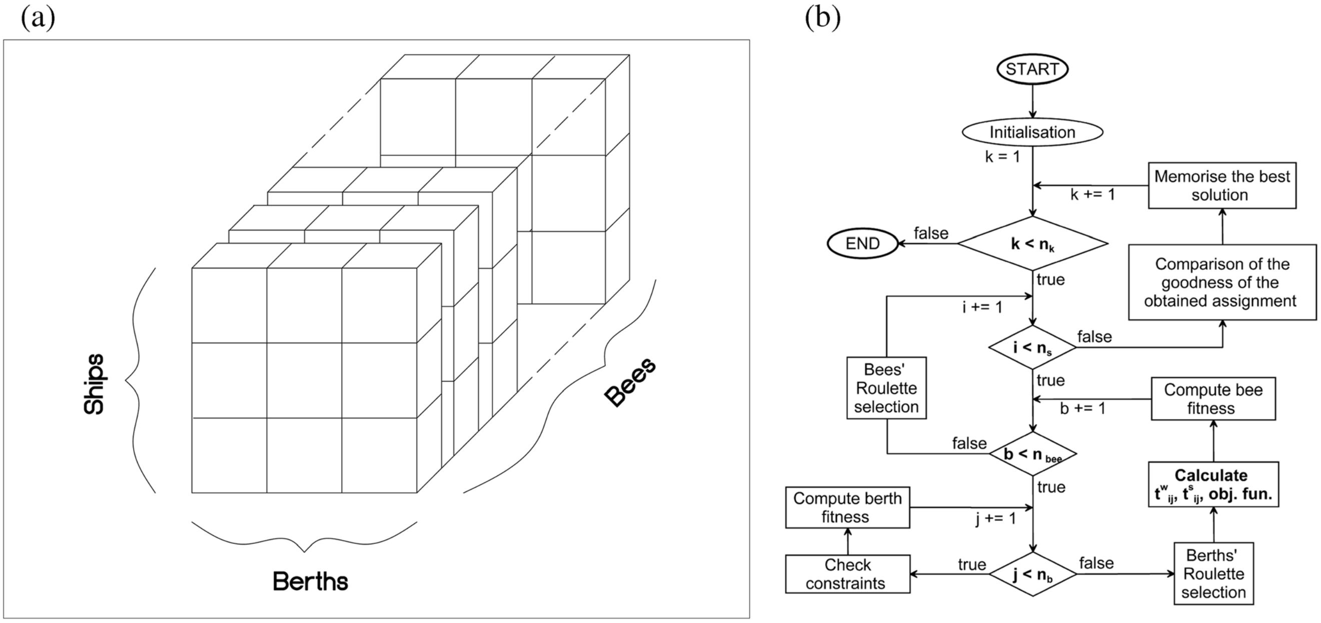

Fig. 2 **a** Three-dimensional assignment matrix structure; **b** flowchart of the proposed method

Table 3 List of Notations used in BCO algorithm

Notation	Description
Sets	
C	set of all possible bees' movements in the forward step, $C = \{1, 2, \ldots, n_c\}$
R	set of recruiter bees, $R = \{1, 2, \ldots, n_r\}$
I	set of all iterations, $I = \{1, 2, \ldots, n_k\}$
G	set of all stages, $G = \{1, 2, \ldots, n_s\}$
Parameters	
n_c	number of possible bees' movements in the forward step
n_r	number of possible bees' recruiters in the recruitment process
n_k	number of iterations
n_{bee}	number of bees
Decision variables	
p_b^u	bee's probability to follow the b-th constructive movements at step u, $u \in G$
$O_{\max}$	maximum normalised value of the objective function of partial or complete solution generated by the b-th bee
O_b	normalised value of the objective function of partial or complete solution generated by the b-th bee, $b \in N$
O_r	normalised value of the objective function of partial or complete solution generated by the r-th recruiter bee, $r \in R$
$Q_{\max}$	maximum value of the objective function of partial or complete solution generated by the b-th bee, $b \in N$
$Q_{\min}$	minimum value of the objective function of partial or complete solution generated by the b-th bee, $b \in N$
Q_b	value of the objective function of partial or complete solution generated by the b-th bee, $b \in N$

normalised value of the corresponding objective function and the sum of the normalised values of the objective function for all the solutions found. On the one hand, a solution with a better value than the objective function is more likely to be selected; on the other, there is a probability that it will be eliminated following a subsequent search process. The probability that a partial or complete solution of b can be chosen by any unengaged bee is equal to the following formulation:

$$p_b = \frac{O_b}{\sum\limits_{r=1}^{n_r} O_r}, \forall b \in R \tag{11}$$

The probability that the b-th bee, at the beginning of the next forward step, remains loyal to its own total or partial solution found in previous steps is represented as follows:

$$p_b^{u+1} = e^{-\left(\frac{O_{\max}-O_b}{u}\right)}, \forall b \in B, \forall u \in G \tag{12}$$

The normalisation of O_b, related to the minimisation of the objective function (1), is calculated by the following equation:

$$O_b = \frac{Q_{\max}-Q_b}{Q_{\max}-Q_{\min}}, \forall b \in B \tag{13}$$

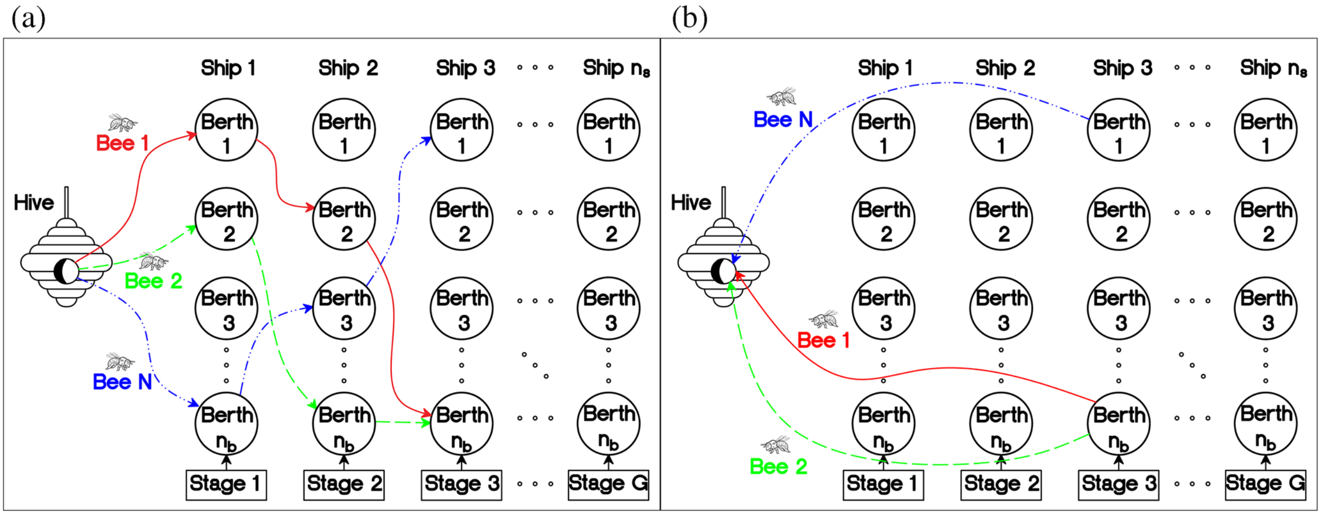

Fig. 3 **a** Forward step and **b** Backward step

Table 4 Results comparison between CPLEX and BCO

Instance	*Problem size*		*CPLEX*		*BCO*		*Gap (%)*
	# of ships	*# of berths*	*f* [min]	*t* [s]	*f* [min]	*t* [s]	
R25–5	25	5	137,130	2.3	137,130	0.7	0.00
R25–10	25	10	139,051	1.5	139,051	1.2	0.00
R25–15	25	15	142,138	2.4	142,138	1.8	0.00
R25–20	25	20	143,725	2.7	143,812	2.3	0.06
R50–5	50	5	467,341	139.9	469,253	2.4	0.41
R50–10	50	10	460,408	189.2	460,656	4.4	0.05
R50–15	50	15	462,711	40.9	463,021	6.5	0.07
R50–20	50	20	469,892	125.6	470,264	8.6	0.08
R75–5	75	5	970,215	7200	973,526	4.9	0.34
R75–10	75	10	958,576	2143	958,997	9.5	0.04
R75–15	75	15	969,749	536.2	970,420	14.3	0.07
R75–20	75	20	969,021	7200	969,639	18.9	0.06
R100–5	100	5	1,650,677	7200	1,657,824	8.6	0.43
R100–10	100	10	1,643,871	2844.2	1,644,913	16.6	0.06
R100–15	100	15	1,638,401	7200	1,639,266	31.2	0.05
R100–20	100	20	1,641,994	7200	1,642,888	33.5	0.05

Notes. f denotes the ships' total staying time in minutes. t denotes the average computation time in seconds.

The normalised value O_b represents the fitness value of a partial or complete solution. According to Eq. (11)–(13) and a random number generator, each artificial bee decides whether to become an engaged follower or to continue exploring its own solution. If the random number chosen is less than the calculated probability, the bee remains faithful to its solution. Otherwise, if the random number is greater than the probability, the bee becomes unengaged. A higher O_b value corresponds to a better-generated solution and to a greater probability that the bee remains faithful to the solution discovered previously. The higher the index value of the step u the higher the probability that the bee remains faithful to its previously found solution, according to Eq. (12).

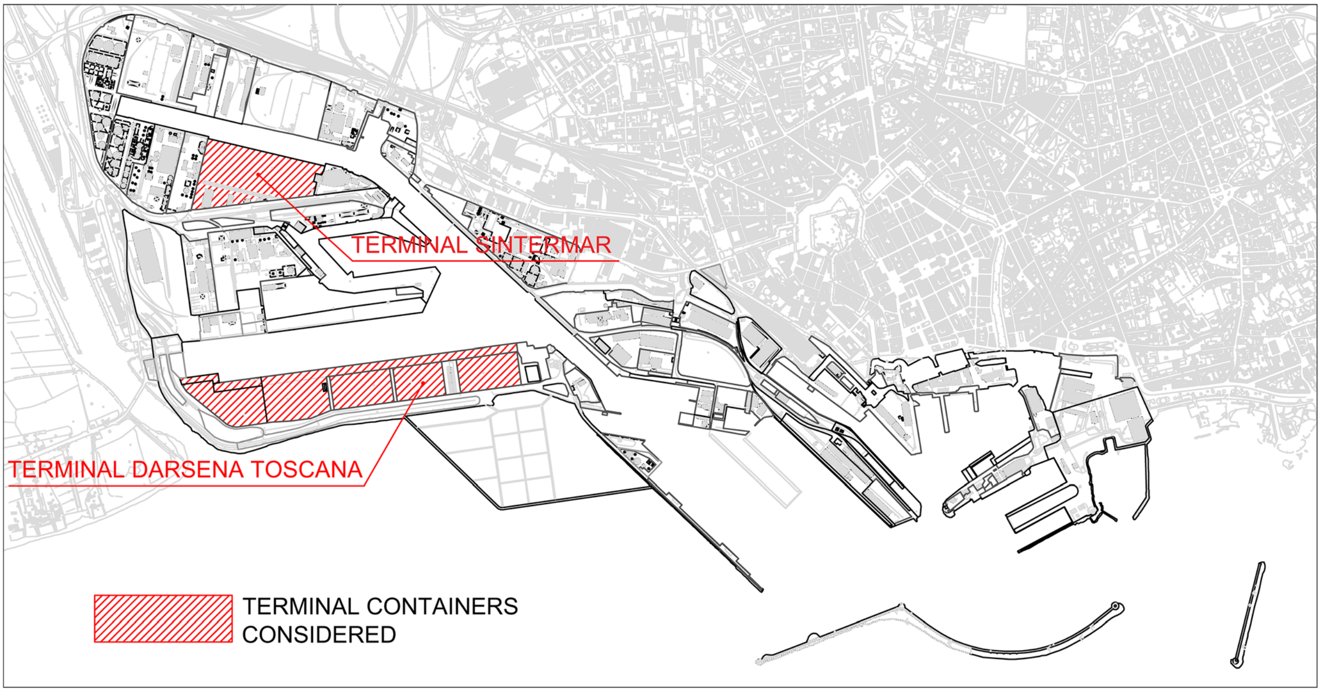

Fig. 4 Container terminal quays considered in the Livorno port (Italy)

Table 5 Comparison between observed and computed times considering 1000 iterations and 100 bees over 10 runs – monthly berth plan

	Observed (days)	Computed (days)	Improvement (%)
		Min	
Obj. function	2144	2015	6
Total service time	107	97	9
Total waiting time	12	2	82

5 Numerical application

In order to check the feasibility of the proposed optimisation model, we compared the BCO with an exact solver for the DDBAP as formulated above. We tested both algorithms on a PC equipped with an Intel i7 2.4GHz CPU and 4 Gb of RAM. The BCO algorithm was developed in C++, while CPLEX was used as an exact solver for the MILP. We introduced a new set of instances based on a Livorno port scheduling real dataset considering 25, 50, 75, 100 ships and 5, 10, 15, 20 berths. Generated instances are available at the following URL: https://bit.ly/37avUC7. The proposed benchmark instances are obtained as follows: the arrival time is considered as a fixed value for all instances, while all the other parameters are assigned randomly from the real dataset. Moreover, according to problem constraints, at least one berth must be compatible with all ships in terms of length and draft.

Concerning the BCO algorithm, we set $n_k = 1000$, $n_{bee} = 10$, $n_c = 1$, $n_r = n_{bee}$, while computation times (t) are related to the average CPU time over 10 runs. Regarding CPLEX, we set the time limit for finding the global optimum equal to 7200 s. The results comparison is reported in Table 4.

We can observe that for small-sized instances, the exact solution is achieved by both approaches. Furthermore, for medium and large-sized instances, we can observe that results in terms of the objective function gap give low values (0.41% in the worst case). In contrast, the comparison in terms of CPU time is relevant. The proposed BCO approach assures near-optimal results in low computation time, thus, it could be used for real-time applications in port management operations. Accordingly, the proposed model could avoid data uncertainties by updating the berth scheduling process in real-time.

6 Application to a real case and results

We applied the proposed method to the real case of the Livorno port by monitoring and acquiring data referred to the container terminals of Darsena Toscana (5 berths) and Sintermar (3 berths), as shown in Fig. 4. We used a time schedule of 28 days referred to the berth planning of February 2016 and considered a total of 140 container ships. Acquired data included arrival times, staying times, ships' length and draft, berths' destination, and status referred to a daily berthing planning and extended to a monthly time schedule. In order to better evaluate the effectiveness of the proposed method, we compared the obtained results with the real schedule of the Livorno port and the Ant Colony Optimisation (ACO) metaheuristic. The comparisons were carried out executing 10 runs of the BCO algorithm. The results of the comparison with the Livorno port assignment are reported in Table 5 in terms of objective function value, service time, and waiting time. We can observe that the percentage of improvement in terms of the total service time is 9% over 4 weeks. This represents a significant result, considering the wide schedule time interval observed. Moreover, we obtained a saving in terms of total waiting time of about 10 cumulated days when compared to the real schedule.

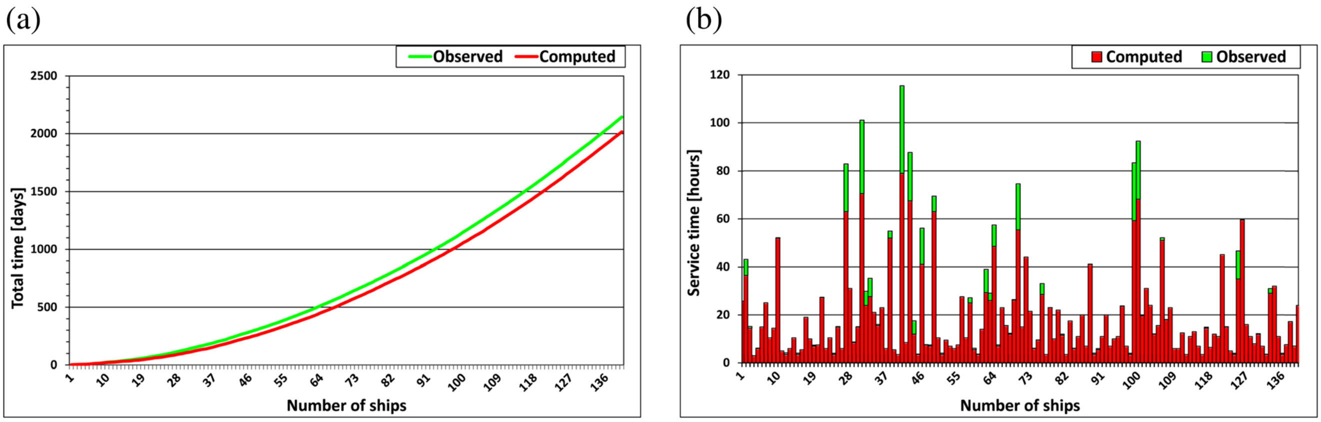

Fig. 5 Comparison between observed and computed (**a**) ships' total staying time (obj. function), (**b**) service time – monthly berth plan

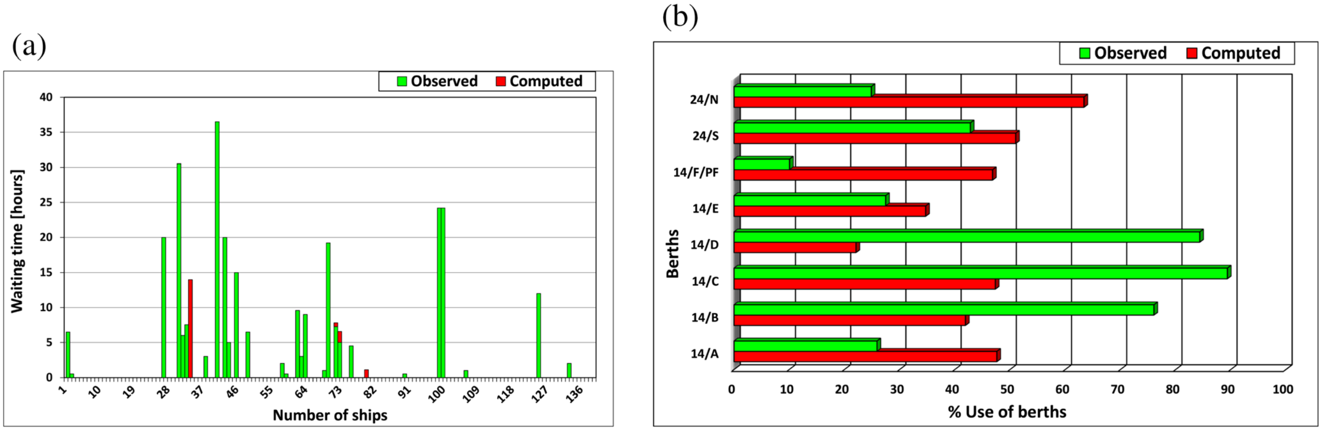

Fig. 6 Comparison between observed and computed (**a**) ships' waiting time and (**b**) the percentage of berths' usage – monthly berth plan

The optimisation of the total staying time improves the performance of the container terminal and, consequently, of the whole port for better ships' management. The lower the objective function, the greater the number of ships that the port can serve for the same time interval, thus increasing the volume of the handled goods.

Figure 5a shows a comparison between the observed and computed total staying times. We can observe significant improvements in terms of service time (Fig. 5b) and waiting time (Fig. 6a). The latter resulted to be about 80% lower than the observed times. In Fig. 6b, we can highlight a resulting homogeneous distribution of the ships over the berths.

In the second comparison, we adapted the Ant Colony Optimisation (ACO) to the DDBAP by developing it in C++ as the proposed BCO. The ACO is one of the most used and known Swarm Intelligence method metaheuristics, firstly proposed by Dorigo [7]. It is an algorithm based on the behaviour of ants seeking a path between their colony and a source of food, suitable to search for an optimal path in a graph. Further details can be found in Dorigo and Gambardella [8]. We applied the ACO to the Livorno port schedule and executed 10 runs as the previous comparison. The results of the comparison are reported in Table 6. We can observe that the proposed BCO found better results than ACO in terms of minimum and average objective function value expressed in hours. In particular, we obtained a saving of 3.5 h in terms of minimum value and about 3 h on average. Standard deviation values are almost the same, which means that both metaheuristics are characterised by high robustness.

Finally, the computation performance of the BCO in terms of CPU time (seconds) are shown in Table 7 and Fig. 7. We can assert that the obtained computation times, on average 4 min in the worst case, are compatible with real-time applications for container ports.

7 Conclusions

In this paper, we proposed a novel mathematical formulation to solve the Discrete and Dynamic Berth Allocation Problem (DDBAP). Starting from a Non-Linear Mixed Integer Programming (NLMIP) we formulated the DDBAP as a Mixed Integer Linear Programming (MILP) to find the global optimum with an exact solver. The objective function minimises the total ships' staying time, introducing a new parameter related to the berth approaching/leaving time. Moreover, a solution method based on the Bee Colony Optimisation (BCO) was introduced for its capabilities in solving complex

Table 6 Comparison between ACO and BCO considering 1000 iterations and 100 ants/bees over 10 runs – monthly berth plan

	Obj. function (hours)		Improvement (hours)
	ACO	BCO	
Min	48,371.7	48,368.2	3.5
Avg	48,373.7	48,370.9	2.8
Std. dev.	1.61	1.75	−0.14

Table 7 Comparison of the BCO average CPU time (s) over 10 runs varying the number of iterations and bees

No. Artificial bees	No. Iterations				
	1	10	100	1000	10,000
1	3.31	3.41	4.28	6.86	13.83
10	3.35	4.86	6.78	27.72	230.30
100	4.69	9.24	27.26	222.60	2112.00

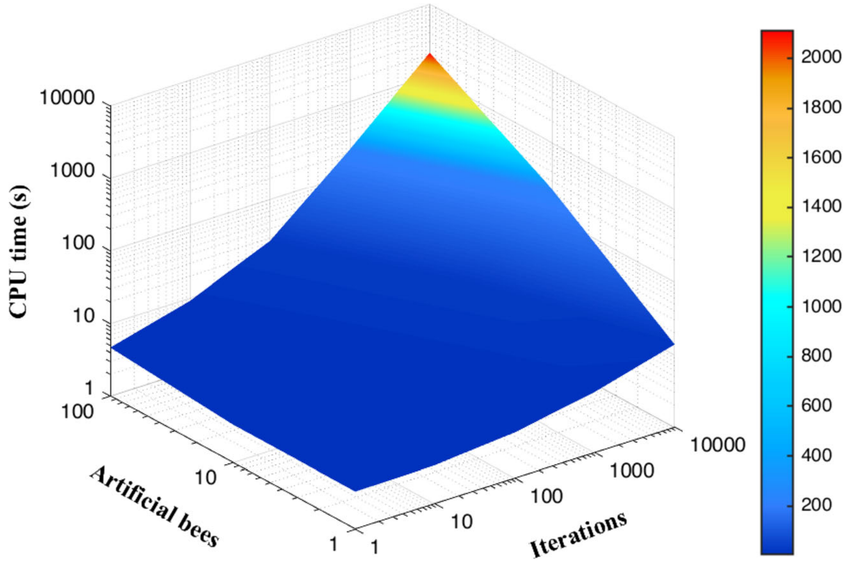

Fig. 7 Computational time values varying the number of iterations and artificial bees

problems. In order to check the feasibility of the proposed optimisation model, we introduced a new set of instances based on Livorno port (Italy) real data. We compared the optimal solutions of the proposed MILP, obtained using the exact solver CPLEX, with near-optimal solutions obtained by the proposed BCO. Results highlighted the effectiveness of the BCO in terms of near-optimal solutions (average gap of 0.12%) in a reasonable low computation time (on average 10.3 s). Furthermore, we applied the proposed model to the real case of the Livorno port in Italy over a time horizon of 4 weeks (February 2016). We compared the proposed BCO with another well-known Swarm Intelligence metaheuristic, the Ant Colony Optimisation. Results highlighted the effectiveness of the proposed approach in terms of average improvement obtained over 10 runs. We evaluated improvements in terms of objective function value, service, and waiting times. As a result, we obtained a saving of 10 cumulative days when compared to the real berth assignment of the Livorno port. Moreover, BCO resulted in being more effective than ACO, obtaining an improvement of about 3 h on average. Finally, computation performance highlighted the possibility of a real-time application of the proposed model to support port operators in the berth assignment task.

From a practical point of view, several benefits can be achieved. It is possible to serve a greater number of ships in a certain time interval by minimising the total staying time. Thus, the port performance increases proportionally in terms of volume of goods, economic benefits due to anchorage fees, port taxes, possible freight surcharges, cost reduction of port operators, and resource management. Other benefits can also be referred to shipping companies. The reduction of waiting time can result in different cost benefits, such as a reduction in personnel costs, possible anchored surcharges, maintenance, and fuel costs. Furthermore, non-monetizable costs such as water noise and air pollution can be also reduced.

Further developments will concern the integration of the proposed DDBAP model with the Quay Crane Assignment Problem (QCAP) for container terminals and to include data uncertainty, budget limit, and priority, in the mathematical formulation. To this aim, the NLMIP formulation, reported in Appendix 1, can serve as a basis to develop extended non-linear models that embed other optimisation problems in port management like the QCAP. Additionally, the comparison with other metaheuristic approaches would be considered in future research.

Funding Open access funding provided by Politecnico di Bari within the CRUI-CARE Agreement.

Appendix 1. The NLMIP mathematical formulation

According to the notations reported in Table 2 and the assumptions presented in Section 3.1, the NLMIP formulation aiming at minimising the total staying time is given as follows:

$$\min \sum_{i=1}^{n_s}\sum_{j=1}^{n_b} t^t_{ij}\cdot x_{ij} \tag{14}$$

s.t.

$$\sum_{j=1}^{n_b} x_{ij} \le 1, \forall i \in S \tag{15}$$

$$\sum_{j=1}^{n_b} l^s_i\cdot x_{ij} \le l^b_j, \forall i \in S \tag{16}$$

$$\sum_{j=1}^{n_b} d^s_i\cdot x_{ij} \le d^b_j, \forall i \in S \tag{17}$$

$$\sum_{j=1}^{n_b} l^s_i\cdot x_{ij} + d^s_i\cdot x_{ij} \le l^b_j\cdot x_{ij} + d^b_j, \forall i \in S \tag{18}$$

$$t_{ij}^{W} = \max\left\{0, -t_i^a + \sum_{k=i-1}^{1} t_{kj}^{t} \cdot x_{kj}\right\}, \forall i \in S, \forall j \in B, t_0^a = 0, t_0^h = 0, t_{0j}^b = 0 \quad (19)$$

$$t_{ij}^{s} = t_{ij}^{W} + t_{ij}^{b} + t_i^h \quad (20)$$

$$t_{ij}^{t} = t_i^a + t_{ij}^{s} \quad (21)$$

$$t_{ij}^{w} \geq t_{ij}^{W}, \forall i \in S, \forall j \in B \quad (22)$$

$$t_{ij}^{w} \leq M \cdot \left(1 - \left|x_{ij} - x_{(i-1)j}\right|\right), \forall i \in S, \forall j \in B \quad (23)$$

$$x_{0j} = 0, \forall j \in B \quad (24)$$

$$x_{ij} \in \{0, 1\}, \forall i \in S, \forall j \in B \quad (25)$$

The objective function (14) minimises the total staying time. Constraints (15) ensure the uniqueness of the assignment between the *i*-th ship and the *j*-th berth, i.e., a ship can only be assigned to a single berth. Constraints (16)–(18) are related to the compatibility between the length and the draft of the ships with respect to those of the berths. The waiting times t_{ij}^{W} are calculated by constraints (19) and they are calculated as the maximum value between zero and the difference between the total staying time of the *(i-1)*-th ship and the arrival time of the *i*-th ship assigned to the same *j*-th berth. Constraints (20) and (21) define the service time t_{ij}^{s} and the total staying time, respectively. Constraints (22) and (23) define the assignment of the waiting time auxiliary variable introducing a large positive number *M*. Accordingly, the waiting time auxiliary variables t_{ij}^{w} are the minimum value between the waiting times t_{ij}^{W} and *M*. Consequently, constraints (24) and (25) introduce three possible cases, referred to the value of the decision variables x_{ij}:

1 $x_{ij} = 1, x_{(i-1)j} = 1$: t_{ij}^{w} is equal to the waiting time t_{ij}^{W} according to the constraints (22);

2 $x_{ij} = 0$, $x_{(i-1)j} = 0$: t_{ij}^{w} is equal to zero according to constraints (19);

3 $x_{ij} = 1, x_{(i-1)j} = 0$, or vice versa: t_{ij}^{w} is equal to zero, according to the constraints (23).

Constraints (24) ensure that the decision variables in constraints (19) and (23) with the subscript *i*-1 are equal to zero when $i = 1$. Finally, constraints (25) define the binary nature of the decision variables.

References

1. Bierwirth C, Maisel F (2010) A survey of berth allocation and quay crane scheduling problems in container ports. Eur J Oper Res 202: 615–627
2. Bierwirth C, Meisel F (2015) A follow-up survey of berth allocation and quay crane scheduling problems in container terminals. Eur J Oper Res 244(3):675–689
3. Buhrkal K, Zuglian S, Ropke S, Larsen J, Lusby R (2011) Models for the discrete berth allocation problem: a computational comparison. Transp Res E 47(4):461–473
4. Cordeau J-F, Laporte G, Legato P, Moccia L (2005) Models and tabu search heuristics for the berth-allocation problem. Transp Sci 39(4):526–538
5. de Oliveira RM, Mauri GR, Lorena LAN (2012) Clustering search for the berth allocation problem. Expert Syst Appl 39(5):5499–5505
6. Dell'Orco M, Marinelli M, Altieri MG (2017) Solving the gate assignment problem through the fuzzy bee Colony optimization. Transp Res C 80:424–438
7. Dorigo M (1992) Optimisation, Learning and Natural Algorithms. Ph.D. Thesis, Politecnico di Milano, Italy
8. Dorigo M, Gambardella LM (1997) Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans Evol Comput 1:53–66
9. Dulebenets MA (2017) A novel memetic algorithm with a deterministic parameter control for efficient berth scheduling at marine container terminals. Marit Business Rev 2(4):302–330
10. Dulebenets MA (2018) Application of evolutionary computation for berth scheduling at marine container terminals: parameter tuning versus parameter control. IEEE Trans Intell Transp Syst 19:25–37
11. Dulebenets MA (2020) An Adaptive Island evolutionary algorithm for the berth scheduling problem. Memetic Computing 12:51–72
12. Dulebenets MA, Kavoosi M, Abioye OF, Pasha J (2018) A self-adaptive evolutionary algorithm for the berth scheduling problem: towards efficient parameter control. Algorithms 11(7):1–35
13. Dulebenets MA, Moses R, Ozguven EE, Vanli A (2017) Minimizing carbon dioxide emissions due to container handling at marine container terminals via hybrid evolutionary algorithms. IEEE Access 5:8131–8147
14. Golias M, Boile M, Theofanis S (2007) The berth allocation problem: a formulation reflecting time window service deadlines. Proceedings of the 48th Annual Transportation Research Forum, Boston, Massachusetts, 15-17 March 2007
15. Golias M, Boile M, Theofanis S (2009a) Berth scheduling by customer service differentiation: a multi-objective approach. Trans Res Part E: Logist Trans Rev 45:878–892
16. Golias M, Boile M, Theofanis S (2010) A lamda-optimal based heuristic for the berth scheduling problem. Trans Res Part C: Emerg Technol 18(5):794–806
17. Golias M, Haralambides H (2011) Berth scheduling with variable cost functions. Marit Econ Logist 13(2):174–189
18. Golias M, Portal I, Konur D, Kaisar E, Kolomvos G (2014) Robust berth scheduling at marine container terminals via hierarchical optimisation. Comput Oper Res 41:412–422
19. Golias M, Saharidis G, Boile M, Theofanis S, Ierapetritou M (2009b) The berth allocation problem: optimizing vessel arrival time. Marit Econ Logist 11(4):358–377
20. Han M, Li P, Sun J (2006) The algorithm for berth scheduling problem by the hybrid optimization strategy Gasa. Proceedings of the 9th International Conference on Control, Automation, Robotics and Vision, Singapore, Singapore, 5-8 December 2006
21. Han Xl, Lu Zq, Xi Lf (2010) A proactive approach for simultaneous berth and quay crane scheduling problem with stochastic arrival and handling time. Eur J Oper Res 207(3):1327–1340

22. Hansen P, Oğuz C (2003) A note on formulations of the static and dynamic berth allocation problems. In: Les Cahiers du GERAD G2003–30. Montréal, HEC
23. Hansen P, Oğuz C, Mladenovic N (2008) Variable neighborhood search for minimum cost berth allocation. Eur J Oper Res 191(3): 636–649
24. Holland JH (1975) Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor, MI
25. Imai A, Nishimura E, Papadimitriou S (2001) The dynamic berth allocation problem for a container port. Transp Res B 35:401–417
26. Imai A, Nishimura E, Papadimitriou S (2003) Berth allocation with service priority. Transp Res B 37:437–457
27. Imai A, Sun X, Nishimura E, Papadimitriou S (2005) Berth allocation in a container port: using a continuous location space approach. Transp Res B 39:199–221
28. Imai A, Zhang JT, Nishimura E, Papadimitriou S (2007) The berth allocation problem with service time and delay time objectives. Marit Econ Logist 9(4):269–290
29. Issam EH, Azza L, Mohamed EM, Kaoutar A, Yassine T (2017) A multi-objective model for discrete and dynamic berth allocation problem. Proceedings of the 2nd International Conference on Big Data, Cloud and Applications, Tetouan, Morocco, March 2017, pp 1–5
30. Issam EH, Azza L, Mohamed EM, Tabaa YA (2019) A modified sailfish optimizer to solve dynamic berth allocation problem in conventional container terminal. Int J Ind Eng Comput 10(4): 491–504
31. Karafa J, Golias M, Ivey S, Saharidis G, Leonardos N (2013) The berth allocation problem with stochastic vessel handling times. Int J Adv Manuf Technol 65(1–4):473–484
32. Kovač N, Stanimirović S, Davidović T (2018) Metaheuristic Approaches for the Minimum Cost Hybrid Berth Allocation Problem. In: Konstantopoulos C, Pantziou G (eds) Modeling, computing and data handling methodologies for maritime transportation. Intelligent Systems Reference Library (131):22–68
33. Kovač N (2017) Survey metaheuristic approaches for the berth allocation problem. Yugoslav J Oper Res 27:265–289
34. Kramer A, Lalla-Ruiz E, Iori M, Voß S (2019) Novel formulations and modeling enhancements for the dynamic berth allocation problem. Eur J Oper Res 278(1):170–185
35. Lalla-Ruiz E, Melian-Batista B, Moreno-Vega J (2012) Artificial intelligence hybrid heuristic based on tabu search for the dynamic berth allocation problem. Eng Appl Artif Intell 25(6):1132–1141
36. Lalla-Ruiz E, Voß S (2014) Towards a matheuristic approach for the berth allocation problem. Learning and Intelligent Optimization, Springer, Cham, pp 218–222
37. Lalla-Ruiz E, Voß S (2016) POPMUSIC as a matheuristic for the berth allocation problem. Ann Math Artif Intell 76:173–189
38. Lassoued R, Elloumi A (2019) The discrete and dynamic berth allocation problem in bulk port. In: 6th international conference on control, decision and information technologies, pp 1976–1980
39. Lee DH, Jin J (2013) Feeder vessel management at container transshipment terminals. Transp Res Part E: Logist Transp Rev 49(1): 201–216
40. Li MW, Hong WC, Geng J, Wang J (2017) Berth and quay crane coordinated scheduling using multi-objective chaos cloud particle swarm optimization algorithm. Neural Comput & Applic 28:3163–3182
41. Lim A (1998) The berth planning problem. Oper Res Lett 22:105–110
42. Lin S-W, Ting C-J (2013) Solving the dynamic berth allocation problem by simulated annealing. Eng Optim 46(3):308–327
43. Lin S-W, Ying K-C, Wan S-Y (2014) Minimizing the Total Service Time of Discrete Dynamic Berth Allocation Problem by an Iterated Greedy Heuristic. Sci World J:1–12
44. Marinelli M, Dell'Orco M, Sassanelli D (2015) A metaheuristic approach to solve the flight gate assignment. Transp Res Procedia 5:211–220
45. Monaco MF, Sammarra M (2007) The berth allocation problem: a strong formulation solved by a Lagrangean approach. Transp Sci 41(2):265–280
46. Nishi T, Okura T, Lalla-Ruiz E, Voß S (2020) A dynamic programming-based matheuristic for the dynamic berth allocation problem. Ann Oper Res 286(1-2):391-410
47. Nishimura E, Imai A, Papadimitriou S (2001) Berth allocation planning in the public berth system by genetic algorithms. Eur J Oper Res 131:282–292
48. Pinedo M (2008) Scheduling: theory, algorithms, and systems. 4th Edition, Springer-Verlag, New York
49. Saharidis G, Golias M, Boile M, Theofanis S, Ierapetritou M (2010) The berth scheduling problem with customer differentiation: a new methodological approach based on hierarchical optimization. Int J Adv Manuf Technol 46(1–4):377–393
50. Sheikholeslami A, Mardani M, Ayazi E, Arefkhani H (2020) A dynamic and discrete berth allocation problem in container terminals considering tide effects. Iranian Journal of Science and Technology, Transactions of Civil Engineering 44(1):369–376
51. Simrin A, Diabat A (2015) The dynamic berth allocation problem: a linearized formulation. RAIRO-Oper Res 49(3):473–494
52. Teodorovic D, Dell'Orco M (2005) Bee Colony optimization – a cooperative learning approach to complex transportation problems. In: Advanced OR and AI Methods in Transportation, Proceedings of the 10th Meeting of the EURO Working Group on Transportation, Poznan, Poland, pp 51–60
53. Theofanis S, Boile M, Golias M (2007) An optimisation based genetic algorithm heuristic for the berth allocation problem. Proceedings of 2007 IEEE Congress on Evolutionary Computation, Singapore, pp 4439-4445
54. Ting C-J, Wu K-C, Chou H (2014) Particle swarm optimisation algorithm for the berth allocation problem. Expert Syst Appl 41(4):1543–1550
55. Umang N, Bierlaire M, Erera AL (2017) Real-time management of berth allocation with stochastic arrival and handling times. J Sched 20:67–83
56. Ursavas E (2015) Priority control of berth allocation problem in container terminals. Ann Oper Res 2015
57. Wang R, Nguyen TT, Li C, Jenkinson I, Yang Z, Kavakeb S (2019) Optimising discrete dynamic berth allocations in seaports using a levy flight based meta-heuristic. Swarm Evol Comput 44:1003–1017
58. Xiang X, Liu C, Miao L (2018) Reactive strategy for discrete berth allocation and quay crane assignment problems under uncertainty. Comput Ind Eng 126:196–216
59. Xu D, Li C-L, Leung JY-T (2012) Berth allocation with time-dependent physical limitations on vessels. Eur J Oper Res 216(1): 47–56
60. Zhen L, Liang Z, Zhuge D, Lee LH, Chew EP (2017) Daily berth planning in a tidal port with channel flow control. Transp Res B Methodol 106:193–217
61. Zhou P, Kang H (2008) Study on berth and quay-crane allocation under stochastic environments in container terminal. Syst Eng-Theory Pract 28(1):161–169
62. Zhou P, Kang H, Lin L (2006) A dynamic berth allocation model based on stochastic consideration. Sixth World Congress Intel Control Autom WCICA 2006(2):7297–7301

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

LuigiPio Prencipe received his BSc degree in Civil Engineering from the Polytechnic University of Bari, Italy, in 2013. He received his MSc degree in Civil Engineering (Transportation curriculum) from the Polytechnic University of Bari, Italy, in 2016. He is currently pursuing the Ph.D. degree in Risk and Environmental, Territorial and Building Development at the Polytechnic University of Bari, Italy. His current research interests include sustainable mobility, linear programming, and optimization models and algorithms to solve transportation and logistics problems.

Mario Marinelli received the BSc and MSc degrees in Computer Engineering from the Polytechnic University of Bari, Italy, in 2006 and 2007, respectively, and the Ph.D. degree in Transportation Engineering from the same university in 2013. He is currently associate professor at the University of Sannio, Benevento, Italy. His research interests include artificial intelligence techniques to simulate the human factor in transport planning, metaheuristic approaches to solve optimization problems in transportation and logistics.