

Article

Enhanced Ant Colony Algorithm for Discrete Dynamic Berth Allocation in a Case Container Terminal

Meng Yu, Yaqiong Lv ^{*}, Yuhang Wang and Xiaojing Ji

School of Transportation and Logistics Engineering, Wuhan University of Technology, Wuhan 430063, China

* Correspondence: y.q.lv@whut.edu.cn

Abstract: Berth allocation is a critical concern in container terminal port logistics, involving the precise determination of where and when arriving vessels should dock along a quay. With berth space limitations and a continuous surge in container handling demands, ensuring an effective berth allocation is paramount for the smooth and efficient operation of container ports. However, due to the randomness of vessel arrival times and uncertainties surrounding container ship loading capacities, berth allocation problems (BAP) often present discrete and dynamic challenges. This paper addresses these challenges by considering real-world terminal operational factors, formulating relevant assumptions, and establishing a model for dynamic berth allocation and efficient ship berthing scheduling. The primary motivation stems from the parallels observed between the BAP problem and ant foraging path selection, leading to the proposal of a novel Parallel Search Structure Enhanced Ant Colony Algorithm (PACO). A proper set of parameters of the algorithm are selected based upon sensitivity analyses on the convergence and parallelism efficiency of the algorithm. To validate our method, a real-world case-container terminal operation in Shanghai Port was studied. The experimental comparison results show that the PACO algorithm outperforms other commonly used algorithms, making it more effective and efficient for the Discrete Dynamic Berth Allocation Problem (DDBAP).

Keywords: berth allocation; parallel search structure; ant colony algorithm; container terminal; dynamic scheduling



Citation: Yu, M.; Lv, Y.; Wang, Y.; Ji, X. Enhanced Ant Colony Algorithm for Discrete Dynamic Berth Allocation in a Case Container Terminal. *J. Mar. Sci. Eng.* **2023**, *11*, 1931. <https://doi.org/10.3390/jmse11101931>

Academic Editors: Hsuan-Shih Lee and Po-Hsing Tseng

Received: 4 August 2023

Revised: 18 September 2023

Accepted: 1 October 2023

Published: 7 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The economic globalization accelerates the development of the marine transportation industry, which necessitates higher efficiency and productivity in container terminals. A container terminal in a harbor logistics operation consists of many physical resources such as containers, vessels, handling equipment, berths, container yards, and communication establishment [1]. The loading/unloading work is carried out by quay cranes (QCs), yard cranes (YCs), and container trucks (CTs) at a container terminal [2]. Referring to Figure 1, containers are delivered at a terminal by trucks and are stored in the terminal yard [3]. A vessel, loaded with containers, is allocated to a berth once it reaches the port. After that, its containers are unloaded and are temporally stored at the yard until being moved to trucks, trains, or vessels to be further delivered. QCs standing on the rail tracks along the quayside carry on the transshipment of containers from the waterside to the quayside [4].

One key issue in a particular container terminal (CT) is the scheduling of the quay space and serving time to ships that must be handled at a terminal [4,5], which is generally categorized as the berth allocation problem (BAP). Each of the vessels will go through the following processes in a particular CT: in-bound to the port, waiting until the scheduled berthing time, mooring at the assigned berth, containers' loading/unloading, and then leaving the port. Apparently, the main problem is how to allocate the corresponding number of the berth to arriving vessels at the port, providing the berth time and location within a planning horizon period. The decision horizon is generally one week, while this may be updated based on the arrival and departure situation of vessels [6].

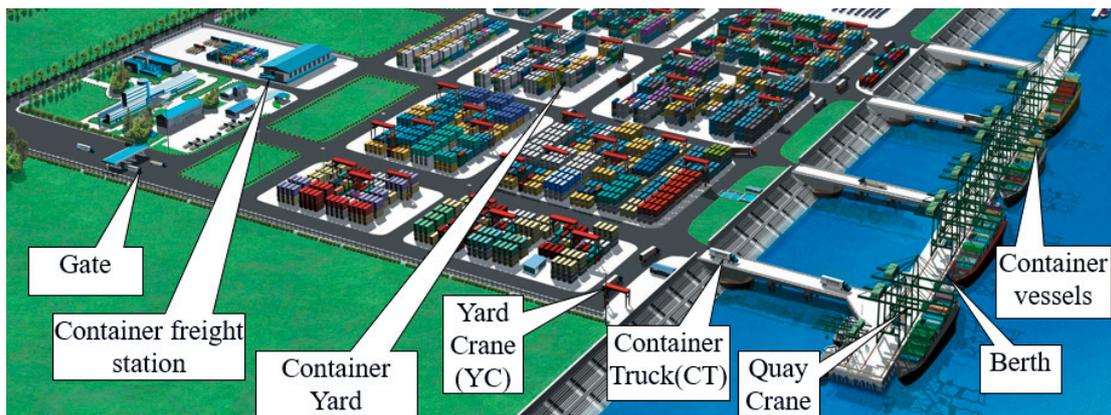


Figure 1. Layout of a typical container terminal.

BAP problems can be divided into static (SBAP) or dynamic (DBAP) [7]. The static BAP model assumes that all vessels reach the terminal earlier than the starting planning time, while a dynamic model defines the ships arriving at an arbitrary time of the planning period given future arrival data [8]. The BAP problem can be further divided into discrete and continuous variants [9]. In discrete cases, strictly one vessel can be accepted for each berthing location, that is, “berth”—one of the partitioned quay sections, at one time, regardless of its size. For the continuous cases, the quay is not segmented, i.e., ships can berth at arbitrary places within the boundaries of the quay, and consequently, the corresponding berth planning is more complex than that in a discrete case with a superiority of better quay space’s utilization [10].

Figure 2 shows the berth operation timeline for each vessel. The service time of a vessel at the port includes the waiting time between the arrival of the vessel and the vessel berthing time as well as the handling time for containers loading or unloading. For BAP, as an NP-hard scheduling problem [11], with the aim of a high efficiency in CT operations, researchers keep seeking for its better solution. Nevertheless, there is a distinct lack of comprehensive solutions that cater to the dynamic nature of container terminals. These solutions need to consider the evolving vessel type dynamics, arrival time, and distinctive handling time seamlessly. This gap underscores the need for a novel algorithmic approach that can adeptly address these multifarious factors.

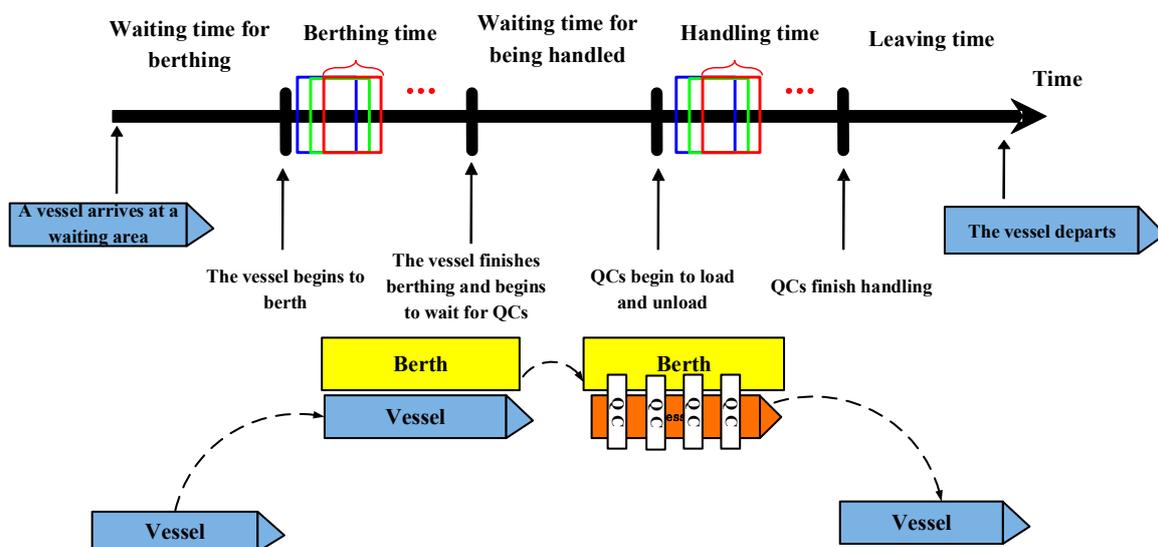


Figure 2. Timeline of vessels staying at a port.

Addressing this research gap begets pertinent questions: How can an algorithm that seamlessly integrates varying vessel types, safety regulations, and operational complexities while maintaining optimization efficiency be devised? How does the proposed algorithm perform across diverse operational conditions and constraints? Answering these questions is imperative to bridge the divide between theoretical optimization and the practical realities of container terminal operations.

In the development of intelligent optimization algorithms, ant colony algorithms have been proven to be effective in solving various types of NP-hard problems [12]. On the basis of reviewing research work, this work's primary contribution lies in the formulation and validation of the Parallel Search Structure Enhanced Ant Colony Algorithm (PACO). Designed to surmount the limitations of conventional methods, PACO introduces an innovative approach inspired by ant foraging behaviors. Its parallel search structure enhances exploration, minimizing the risk of converging towards local optima. Furthermore, this study introduces a holistic model that effectively marries theoretical optimization with the intricate dynamics of real-world container terminal operations.

The rest of the paper is organized as follows. Section 2 expatiates on the related literature review; the detailed description of the problem and the mathematics models of DDBAP are addressed in Section 3; Section 4 introduces the ant colony algorithm and argues for the parallel ant colony algorithm for solving the DDBAP models; the case experiments and the analysis results are illustrated in Section 5; and the summary of the work is drawn in the last section.

2. Literature Review

The berth allocation problem (BAP) entails assigning incoming vessels to berth positions. Moreover, the BAP problems are often discrete and dynamic owing to the randomness of the arriving time of those container vessels and their loading capacities. At present, many scholars have proposed new solutions and methods for the berth allocation problem. Issam, E. H. et al. [13] investigated the Dynamic Berth Allocation Problem (DBAP) and provided a multi-objective mathematical model; the validity of the model was verified by numerical experiments. Hsu, H. P. [14] proposed a hybrid particle swarm optimization (PSO) algorithm, which combined the improved PSO algorithm with the event-based heuristic algorithm to solve the DDBAP problem and dynamic QCAP (DQCAP) seaside operation planning problems. Hu, Z. H. [15] proposed a multi-objective genetic algorithm (moGA) based on two-part representation to solve the two-objective model. Lin, S. W. et al. [16] studied a DDBAP problem, whose objective is to minimize the total service time, and introduced an iterative greedy (IG) algorithm to solve it. Türkoğulları, et al. [17] focused on the integrated seaside operations, namely, the coordination of the berth assignment, quay crane allocation, and quay crane scheduling problems For DBAP, Simrin et al. [18] introduced a new mathematical model, where a non-linear mixed integer program was established to formulate the problem, converting it to be an equivalent mixed integer program (MIP). Kovač N et al. [19] proposed an enhanced evolution algorithm for the BAP to optimize the total waiting time of all ships. Kordić S et al. [20] provided an accurate combinatorial method for addressing the DBAP problem and the Hybrid BAP with given vessel loading/unloading times based on the Sedimentation Algorithm (SA). Budipriyanto A. et al. [21] studied the influence of uncertainty on the coordination between berths. It was verified that the cooperative strategy can improve the total waiting time, container processing time, and total vessel turnover time. Xiang, X. et al. [22] examined the berth allocation problem by analyzing uncertainties, and a dual-target robust BAP model focused on economic effectiveness and consumers' satisfaction was developed.

It is not difficult to see from the above research that many scholars have designed many different algorithms to solve the BAP problem from different directions. Since continuously arriving vessels can be considered as jobs and discrete berths can be considered as parallel machines, the DDBAP problem can be converted as a parallel-machine scheduling problem [23], which is NP-hard in a strong sense [24]. The general object of berth planning is to

obtain quick and credible services for vessels, which are referred to the various objective functions in the research articles. Among them, models are always formulated to optimize the sum of the waiting and processing time of ships (i.e., the time of staying in a port) [4]. Thus, in this paper, DDBAP is still the focus, and the mathematics model that dynamically allocates the berth and schedule vessels to the moor berth efficiently under the goal of optimizing the sum of the waiting as well as the total processing time of vessels in the port during a certain time is proposed with full consideration of related constraints.

Many heuristic algorithms, such as the Genetic Algorithm (GA), Differential Evolution (DE), and Simulated Annealing (SA), have been applied to solve the BAP, as mentioned previously. Ant Colony Optimization (ACO) has successfully shown feasibility in many NP-hard combinatorial optimization problems, such as vehicle routing problems, quadratic assignment problems, other scheduling problems, etc. [24]. GA offers global optimization capabilities, yet it may be slower for dynamic scenarios. DE excels in non-linear optimization but could struggle with complexity. SA has a probabilistic advantage but requires parameter tuning. Ant Colony Optimization (ACO) is adept at combinatorial problems but can be slow in large-scale scenarios. To better solve these problems, the improved ACO is then proposed based on the original ACO algorithm. It increases the number of slave processors and makes multiple processors perform calculations at the same time. It can effectively improve the deficiencies of the ACO with the ease of sinking into the local optimum.

After comparing and analyzing several typical logistics scheduling algorithms [25–31], Randall, M et al. examined several parallel decomposition strategies and applied them to a particular problem, that is, the travelling salesman problem, leading to a result of better efficiency and computing speeds [26]. Yu M et al. proposed a predictive two-stage path planning algorithm [27,28]. It alleviates traffic congestion, avoids collisions, improves the efficiency of automated terminal loading and unloading, and considers road congestion factors. Li H T et al. proposed a dynamic ant colony genetic algorithm on the basis of the extensive research of the ant colony algorithm, genetic algorithm, and other related algorithms [29]. Based on the efficiency of enterprise workshop distribution operations, the allocation of the shop floor location is studied, and a coordinated optimization model of automatic guided vehicle berth assignment and material box allocation based on a genetic algorithm is proposed in the simulation environment [30]. Niu M et al. considered the time and space constraints between the ship entry and exit and ship berthing, focused on the realistic limitations caused by the berthing problem caused by the ship berthing priority, designed an immunogenetic algorithm to solve this, and gave a berth allocation scheme, a berthing scheme, and a ship entry and exit scheduling scheme [31]. The basic concepts and ideas of the genetic algorithm are introduced. At the same time, the function of some algorithms derived from the genetic algorithm is proposed. Zhang M et al. proposed an improved ACO algorithm, namely, 3D-PACA, which was implemented to identify and scan 3D terrains and achieved certain results [32].

To sum up, it can be seen from the above research that the parallel ACO algorithm, as an improvement of the ACO algorithm, has been applied in other fields and can be used to solve the practical problems. Therefore, it motivates us to apply it to the dynamic berth allocation problems.

3. Problem Statement and Model Formulation

Since each container has a different transit time from its original storage area to its assigned berth, each vessel has a different handling time at different berths. However, the vessel handling time is assumed to be deterministic in most published BAP models because the vessel handling time has been considered in different ways:

- (a) It is known beforehand and regarded as immutable, i.e., it is fixed.
- (b) It depends on where the vessels berthed.
- (c) It depends on the quantity of cranes serving the vessels.
- (d) The working schedule of the cranes is highly relied on.

(e) All the mentioned conditions (b–c) need to be handled at the same time.

In this paper, the handling time relies on the number of cranes serving the vessel, the transshipment volume of containers, and the working performance of the assigned cranes. There are m vessels (i.e., vessels $1, 2, \dots, m$) and m^b berths (i.e., berths $1^b, 2^b, \dots, m^b$). Assume that each vessel is served by only one berth at a time and each berth can handle one vessel at a time. If a vessel g starts serving at a berth, it will remain there continuously for the next time unit.

The DDBAP considered here is considered to be a parallel machine scheduling problem, with vessels being jobs and berths being machines [33]. Job j has a processing time p_j , a lease date a_j , a weight w_j , and a set of machines that are capable of processing the job j . The processing time p_j , is determined by the quantity of working quay cranes and the efficiency of the cranes (the processing time depends on the quantity of cranes serving the vessels and the crane efficiency). To define our problem mathematically, the problem of sequencing vessels at a single berth is defined first. And then, the problem of allocating and sequencing vessels at multi-berths is modeled mathematically.

3.1. Notation

All notations of this paper are described in alphabetic order, including model-related notations and algorithm-related notations.

Notation

Sets

B : Set of berths, $B = \{1, 2, \dots, n^b\}$;

C : Set of QCs, $C = \{1, 2, \dots, n^c\}$;

J : Set of incoming vessels during a planning period, $J = \{1, 2, \dots, n\}$;

Model-Related Notations

n : Number of incoming vessels within a planning period, positive integer;

n^b : Number of available berths in a port;

n^c : Number of quay cranes;

n_j^{cap} : Container capacity of vessel j ;

n_j^l : Number of containers to be loaded/unloaded on vessel j ;

O_i : An ordered subset of vessels to be loaded/unloaded at berth i ;

t_{ij}^b : Moment of vessel j starting to berth at berth i ;

t_j^w : Arrival moment of vessel j to the waiting area, $t_j^w \geq 0$;

T_j^b : Time duration of a vessel to berth and leave, assumed to be fixed and the same for different vessels, $T_j^b > 0$;

t_{ij}^{fl} : The finishing moment of loading/unloading vessel j at berth i , $t_{ij}^{fl} = t_{ij}^l + T_{ij}^l$;

t_{ij}^l : The beginning moment of loading and unloading vessel j at berth i ;

T_{ij}^l : Duration of loading and unloading vessel j at berth i ; a function of n_j^l , n_j^{cap} , and n_j^c ;

PACO-Related Notations

d_{ij} : Time distance between vessels j and j' ;

$e(j, j')$: Edge connecting i and j ;

$t_k(i)$: The length of time during which the k th ant starts from berth i to the end through all sections of the path;

Q : Constant, the amount of information released by an ant after the complete path search is complete;

L^k : The path length through which the k th ant passes;

n^{ant} : Quantity of ants in a colony;

n^{con} : Quantity of iterations of convergence;

n_{max}^{itr} : Maximum allowable number of iterations;

n^{itr} : Current number of iterations;

$n_k(s)$: Notation of the number of nodes passed by the current ant k in the step s ;

$p_{ij}^k(o_i)$: Probability of ants k choosing vessel j' from vessel j as the next candidate within the ordered subset of berth i ;

α : Adjustable parameters controlling the relative impact of pheromone trajectories $\tau_{jj'}(t)$;

β :	Adjustable parameters controlling the relative impact of heuristic desirability;
ρ :	Constant, a coefficient of evaporation; $0 < \rho < 1(1 - \rho)$ expresses pheromone decay during t and $t + 1$;
$\tau_{jj'}(t)$:	The intensity of the trail on edge (j,j') at time t ;
$\eta_{jj'}(t)$:	A heuristic measure of the desirability of adding edge (j,j') to the solution being built;
$\Delta\tau_{jj'}$:	A total incremental quantity of the pheromones laid by all ants on edge (j,j'') ;
$\Delta\tau_{jj'} = \sum_{k=1}^m \Delta\tau_{jj'}^k$	
$\Delta\tau_{jj'}^k$:	An incremental quantity of pheromones laid by the k th ant on edge (j,j') during time t and time $t + 1$, which is a function of the length L_k of tour $T_k(t)$:
$\Delta\tau_{jj'}^k = \begin{cases} \frac{Q}{L^k} & \text{if the } k\text{th ant passes path } (i, j) \\ 0 & \text{else} \end{cases}$	

Variables

n_j^c :	Decision variable; number of quay cranes assigned to the vessel j ;
x_{ijo} :	Binary decision variable, with a value of one when vessel j at berth i is served at service time t and a value of 0 otherwise;

3.2. Single-Berth Model

A single-berth allocation is just to sequence coming vessels in order to minimize the turn-around time ($i = 1$). The scheme of berth allocation can be realized by the following model:

$$\text{Min } T_C = \sum_{o \in O} \sum_{j \in J} ((t_{1j}^b - t_j^w) + (t_{1j}^l - (T_j^b + t_{1j}^b)) + T_{1j}^l + 2T_j^b) x_{1jo} \tag{1}$$

$$\sum_{o \in O} x_{1jo} \leq 1, \quad \forall j \in J \tag{2}$$

$$\sum_{o \in O} \sum_{j \in J} x_{1jo} = 1 \tag{3}$$

$$(t_{1j}^b - t_j^w) x_{1jo} \geq 0, \quad \forall j \in J \tag{4}$$

$$(t_{1j}^l - (T_j^b + t_{1j}^b)) x_{1jo} \geq 0, \quad \forall o \in O \tag{5}$$

$$x_{1jo} \in \{0, 1\}, \quad \forall j \in J, \forall o \in O \tag{6}$$

The objective function (1) is to minimize the total time of the ship in a certain period that is the overall waiting time (the sum of time spent waiting to be berthed and operated by quay cranes), handling time T_{ij}^l , and berthing/departure time. Constraints (2) mean that each vessel has only one chance of berthing. Bind (3) ensures that one vessel or less is restricted to the moor at one berth at any moment. Bind (4) is the setup time it takes for vessel j to be processed with loading/unloading, which is no less than 0 to ensure that each vessel must be served after arrival. Bind (5) means the vessel j waiting time; this is no less than 0. Bind (6) announces a decision variable x_{1jk} , being a 0–1 variable [28].

3.3. Multi-Berth Model

Mostly, there is more than one berth in every container terminal, so the multi-berths allocation problem is to solve not only how to sequence the loading/unloading of vessels but also to allocate berths to vessels to berth first.

The berth allocation model follows the following assumptions:

1. The vessels arrival time is random;
2. The vessel loading and unloading times are derived from the number of containers, the quantity of QCs, the productivity of QCs, and other related factors;

3. The mooring space must conform to the real conditions of the vessel (depth and length of water) of the constraints;
4. The vessels moving from one berth to another will not be considered, where each vessel has only one berth opportunity;
5. The number of assigned QCs does not exceed the number of QCs allowed to work simultaneously;
6. When a large quantity of Quay Cranes (QCs) operates on a single vessel simultaneously, they can influence each other, potentially affecting the overall utilization rate of QC s. In cases where more than two terminal container cranes are operating simultaneously, one of them operates at a reduced capacity, typically around 90%.

The berth assignment plays a significant role in the minimization of the turnaround time, which is one of the key indicators in evaluating the port operations. The purpose of berth allocation can be formulated as Equation (7).

$$\text{Min } T_C = \sum_{i \in B} \sum_{o \in O} \sum_{j \in J} ((t_{ij}^b - t_j^w) + (t_{ij}^l - (T_j^b + t_{ij}^b)) + T_{ij}^l + 2T_j^b) x_{ijo} \tag{7}$$

$$\sum_{j \in J} x_{ijo} \leq 1, \quad \forall i \in B, \forall o \in O \tag{8}$$

$$\sum_{i \in B} \sum_{o \in O} \sum_{j \in J} x_{ijo} = 1 \tag{9}$$

$$T_j^c = \frac{n_j^l}{n_j^c p_1 k_1 k_2 (1 - k_3)}, \quad \forall j \in J \tag{10}$$

$$n_j^c = \begin{cases} 1 & n_j^l \leq 150 \\ 2 & 151 \leq n_j^l \leq 500 \\ 3 & 501 \leq n_j^l \leq 700 \\ 4 & 701 \leq n_j^l \leq 1000 \\ 5 & 1001 \leq n_j^l \leq 2000 \\ 6 & n_j^l \geq 2001 \end{cases}, j \in J \tag{11}$$

$$p_j^c = \begin{cases} 22 & \text{containers/hour } n_j^{cap} \leq 500TEU \\ 28 & \text{containers/hour } 6500TEU < n_j^{cap} < 500TEU, \quad j \in J \\ 30 & \text{containers/hour } n_j^{cap} \geq 6500TEU \end{cases}$$

Here, p_1 is the service efficiency of the quay crane loading and unloading container vessel j ; k_1 is the conversion coefficient of TEUs; k_2 is the co-operation rate of quay cranes; when $n_j^c = 1$, $k_2 = 1$; otherwise, $k_2 = 90\%$; k_3 is the reordering rate of containers for container vessels.

$$(t_{ij}^b - t_j^w) x_{ijo} \geq 0, \quad \forall j \in J \tag{12}$$

$$(t_{ij}^l - (T_j^b + t_{ij}^b)) x_{ijo} \geq 0, \quad \forall j \in J \tag{13}$$

$$D_i^b \geq D_j^s \tag{14}$$

$$L_i^b \geq L_j^s \tag{15}$$

D_j^s is the draft depth of vessel j ; D_i^b is the draft depth of berth i ; L_j^s is the length of vessel j ; L_i^b is the length of berth i ;

$$x_{ijo} \in \{0, 1\}, \quad \forall j \in J, \forall i \in B, \forall o \in O \tag{16}$$

The terms of the equation were given in Section 3.1. The objective in (7) is to minimize the total time of the container vessels in a port during a certain period, that is, the total waiting time and processing time T_{ij}^l . Constraint (8) represents that each vessel has a chance to berth. Constraint (9) ensures that no more than one vessel is allowed to moor at one berth at any moment. Constraint (10) is the estimation of the vessel handling time; Constraint (12) is the loading and unloading time of vessel j , which is non-negative to ensure that each vessel must be serviced upon arrival [28]. Constraint (13) means the vessel j waiting time, which is non-negative. Constraints (14) and (15) emphasize that the distribution must conform to the physical conditions of the berth (depth and length). Constraint (16) declares that the decision variable x_{ij0} is a 0–1 variable.

4. Solution Algorithm

In this section, the basic principle of the ACO algorithm is discussed first, and then the modified ant colony algorithm for DDBAP is proposed. By comparing the parallel ant colony algorithm with the traditional ant colony algorithm, the parallel ant colony algorithm and its advantages in solving the berth allocation problem are analyzed.

4.1. The Basic Principle of the Ant Colony Algorithm

Berths are the main resources in the container terminal, berth allocation is the primary content of the formulation of the terminal loading and unloading operation plan, and the operation arrangements of the container terminal are all carried out around the arrangement of the berth allocation plan. The robustness of berth allocation stability in uncertain environments is important.

Due to the limited number of berths at the terminal, the container terminal needs to make overall arrangements in advance for the berthing location and berthing time of the ships to be arriving during the planning period to improve the overall operational efficiency and service quality of the terminal. According to the shipping company's estimated arrival plan, the terminal can obtain the estimated arrival time, estimated departure time, and estimated loading and unloading time of each ship during the planning period and formulate the corresponding berth allocation plan accordingly. However, in the actual process, the actual arrival time of the ship is often affected by a variety of uncertainties, and there will be a certain deviation relative to the expected arrival time. However, the scheduling plan for the other facilities at the terminal is based on the original berth allocation plan, and this temporary adjustment will result in loading and unloading difficulties in terms of both time and cost. Moreover, due to the inevitable blindness and short-sightedness of temporary adjustments, frequent adjustments will seriously disrupt the normal operation order of the dock and reduce the overall operation efficiency of the wharf. Therefore, when formulating the berth allocation plan, it is one-sided to simply take the service efficiency of the ship as an evaluation index, and the thinking on the anti-interference ability of the plan in an uncertain environment should also be added, that is, the robustness of the plan. The ant colony algorithm has strong parallelism and robustness when solving the berth allocation problem.

At present, the intelligent algorithms used to solve difficult problems with high complexity in time and space mainly include genetic algorithms, neural network algorithms, taboo search algorithms, simulated annealing algorithms, and ant colony algorithms. Genetic algorithms take the population as the solution unit, and the solution efficiency is high and widely used. Neural network algorithms have strong self-learning and adaptive abilities, but the algorithm is slow to converge. The probability of the contraindicated search algorithm seeking the optimal solution is large, and the algorithm convergence is good, but it has a strong laziness regarding the initial solution. The solution process of the simulated annealing algorithm is simple and easy to combine with other algorithms, but the solution time is long, and the performance of the algorithm is more sensitive to the selection of parameters. The ant colony algorithm has strong parallelism and good robustness.

Therefore, when formulating the berth allocation plan, it is one-sided to simply take the service efficiency of the ship as an evaluation index, and the thinking on the anti-interference ability of the plan in an uncertain environment should also be added, that is, the robustness of the plan, so the text considers the use of the ant colony algorithm.

In ant colony optimization (ACO), it is known that ants communicate through aromatic substances called pheromones, which are inspired by the collective behavior of ant colonies searching for food [34,35]. Ants that find food return to their nests and mark their tracks with pheromones. By laying and tracking these pheromone tracks, ants can find a good proximity of the shortest path between the food source and nest [36].

The berth allocation and scheduling problem exhibits strong similarities with the ant colony routing problem. In this context, each vessel can be seen as a dynamic node seeking the optimal route to its destination within the terminal. It is well-documented that the ant colony algorithm, when applied to routing problems, operates in a distributed manner, offering positive feedback and demonstrating global convergence [37]. The idea is that if ants must choose among different paths at a given point, the ants that are strictly chosen by the preceding ants (i.e., those with high path levels) will be selected with a higher probability.

4.2. The Basic Principle of the Parallel Ant Colony Algorithm

For large-scale optimization problems, solving with a parallel ant colony algorithm will shorten the computation time and build a good foundation for the practical application of the algorithm. In an artificial ant colony algorithm, each ant is independent of the construction of the solution, and it can communicate with others indirectly through pheromones. This natural parallelism of the ACO algorithm provides a good foundation for the parallel implementation of the algorithm [38]. Therefore, it is feasible to transform the serial ant colony algorithm into a parallel ant colony algorithm.

The basic idea of the parallel ant colony algorithm is to divide one ant colony (m ants) in the original serial algorithm into P ant colonies. The number of ants in each ant colony is m/P , and there is a processor for each sub-ant colony. The number of ants in different ant colonies can be equal or unequal. If the processor performance is the same, the number of ants is generally equal.

To solve the problem that the ant colony algorithm tends to fall into local optimal solutions, each ant colony in a parallel ant colony algorithm separately searches the solutions [39]. Each subprocessor follows the basic principles of the ACO algorithm and calculates the best solution. After each iteration, each sub-processor needs to communicate with the main processor and other sub-processors to obtain information from them. When the termination condition of the parallel ACO algorithm is met, one of the processors collects the optimal solution of each sub-ant colony and outputs the global optimal solution which is shown as Figure 3.

4.3. PACO for BAP

Given n set of nodes $\{1, 2, \dots, n\}$ and the travel time between every two nodes, the goal is to find a route that takes the minimum time to travel around every node and return to the starting point. If each node is considered as one of the graph nodes, and the time $t_{jj'}$ is sought for a weight connecting vertices v_j and $v_{j'}$, then the objective is to discover the shortest cycle within the complete graph with n nodes.

Artificial ant k ($k = 1, 2 \dots m$) decides the direction in which to move according to the amount of information on each path in the process of movement. Unlike the real ant system, the ant colony system has a memory function, a certain vision in a discrete environment. With the time passed, the information before was gradually faded, so after $n^{i^{th}}$ moments when the ant completes a cycle path, the amount of information should be adjusted. Each artificial ant behavior should be in accordance with the following rules.

- (1) According to the concentration of hormones on the path, the ant selects the next path with the corresponding probability.

- (2) The ant no longer chooses nodes for the next path, which have been passed in this cycle, by using a data structure (tabu matrix) to control.
- (3) After completing a cycle, the ant releases pheromones in corresponding concentrations according to the length of the path and updates the pheromone concentration of the traversed path. In the initial stage, m ants are randomly placed on nodes with an equal amount of information on each path.

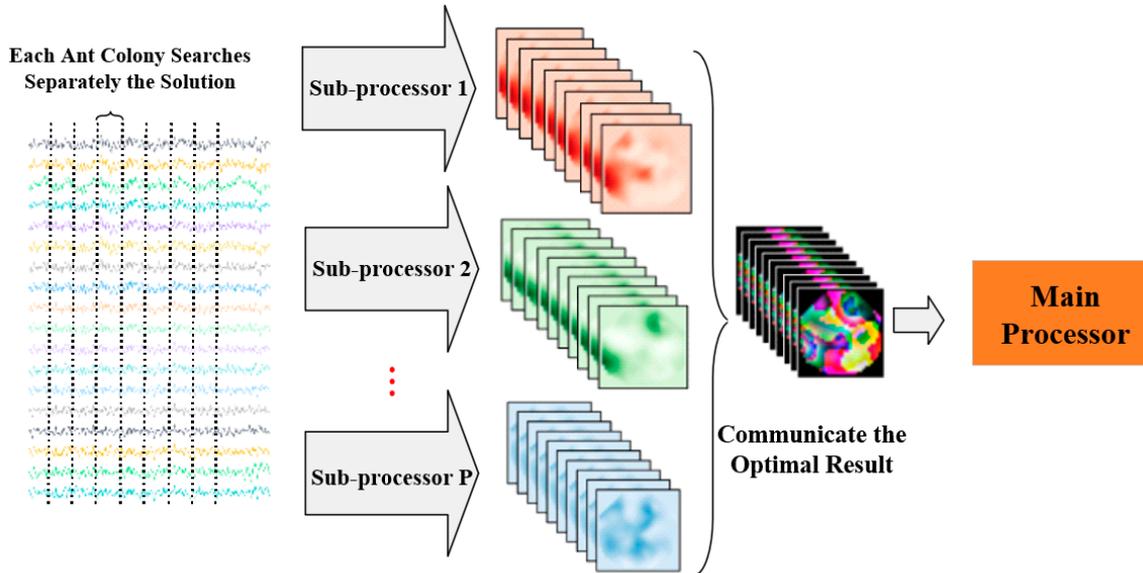


Figure 3. The schematic diagram of PACO.

Suppose $\tau_{jj'}(0) = \tau_0$ (τ_0 is constant).

Ants autonomously choose the next path node based on the amount of information on the path. At the moment t , ant k from node j to node j' is the probability given as Equation (17).

$$p_{jj'}^k = \begin{cases} \frac{\tau_{jj'}^\alpha \eta_{jj'}^\beta}{\sum_{j \in allowed_k} \tau_{ij'}^\alpha \eta_{ij'}^\beta}, & j \in allowed_k \\ 0, & else \end{cases} \quad (17)$$

where $allowed_k = \{1, 2, \dots, n\} - tabu_k$ means all nodes are permitted to go to the next step, and $tabu_k$ records all of the passed nodes of ant k . When all nodes are put into $tabu_k$, ant k will finish one cycle, and the path traversed by ant k is a solution to the problem. $\eta_{jj'}$ is a heuristic factor that represents the desired degree for the ant to move from node i to node j , where $\eta_{jj'} = 1/C_{jj'}$ and $C_{jj'}$ is the required time passing from node j to node j' . Parameters α and β control the relative significance of the trajectories and [40]. Thus, the transition probability is a balance between the visibility and trajectory strength at time t .

Let $\tau_{jj'}(t)$ be the intensity of the trail on edge (j, j') at time t . Each ant chooses the next node at time t , where it will be at time $t + 1$. At this point, the tension trajectory is updated as Equation (18).

$$\tau_{ij}(t + 1) = \rho \tau_{ij}(t) + \Delta \tau_{ij} \quad (18)$$

Here, the coefficient ρ is such that $(1 - \rho)$ represents the evaporation of the trail between time t and $t + n$.

$$\Delta \tau_{ij} = \sum_{k=1}^n \Delta \tau_{ij}^k \quad (19)$$

where $\Delta\tau_{ij}^k$ is the number per unit of the length of the trail substance (the pheromone in real ants) laid on the edge (j,j') by the k -th ant between time t and $t + n$; it is given by Equation (20).

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L^k} & \text{if the } k\text{-th ant goes from } i \text{ to } j \text{ between time } t \text{ and } t + 1 \\ 0 & \text{otherwise} \end{cases} \quad (20)$$

where L^k is the tour length of the k -th ant, which is equal to the sum of time taken on each route for the k -th ant.

The greater the total path time is, the lower the unit pheromone concentration released. Obviously, the ants do not release pheromones in any path they had not experienced.

In the ant-density model, trajectories of quantity Q are left on the edge (j,j') when an ant goes from i to j . In the ant-quantity model, an ant leaves a quantity Q/d_i of the trail on the edge (j,j') when it goes from i to j .

Therefore, in the ant-density model, the quantity per unit of length of the trail substance is as shown in Equation (21):

$$\Delta\tau_{j'j}^k = \begin{cases} Q & \text{if the } k\text{-th ant goes from } j \text{ to } j' \text{ between time } t \text{ and } t + 1 \\ 0 & \text{otherwise} \end{cases} \quad (21)$$

And in the ant-quantity model, the amount of trail material per unit length is as shown in Equation (22):

$$\Delta\tau_{j'j}^k = \begin{cases} \frac{Q}{d_{j'j}} & \text{if the } k\text{-th ant goes from } j \text{ to } j' \text{ between time } t \text{ and } t + 1 \\ 0 & \text{otherwise} \end{cases} \quad (22)$$

From these definitions, it is clear that the increase in edge trajectories (j,j') as ants go from j to j' is independent of $d_{j'j}$ in the ant density model, whereas in the model, it is inversely proportional to $d_{j'j}$ in the ant-quantity model (i.e., shorter edges are made more desirable by ants in the ant-quantity model).

In addition, some parameters including α and β in the ant colony algorithm have a great influence on the performance of the algorithm. Through the analysis of parameters in the ant system by Oliver30, M. Dorigo has found that when $\alpha = \{0.5, 1\}$, $\beta = \{1, 2, 3, 4, 5\}$, the ant system can always converge to the optimal solution, and when m (the number of ants) is close to n (the number of nodes), the algorithm has a better performance.

4.3.1. Solution Encoding

A fixed-length solution is used in PACO to represent (Figure 4). It encodes the order of vessels scheduling at each berth, not the berthing time. The order in which a vessel is serviced does not imply that the vessel can only berth after the service of its predecessor has been completed. It is used to specify the order in which a vessel begins to berth, i.e., the vessel's berthing time must be greater than or equal to the berthing time of its predecessor. Given the dispatch order of vessels, their berthing time can be easily calculated by considering Constraints (4)–(6) described in Section 4.3 of the problem formulation.

4.3.2. PACO for the Berth Allocation Problem

Berth assignment is known as a parallel machine scheduling problem, which is not only about assigning berths to vessels but also about sequencing the service for the vessel [28]. Considering the interaction between berths and storage yards, the collaborative optimization of the berth allocation and storage yard is studied from the perspective of ships in a certain planning period [40]. Each vessel is considered as a node (equivalent to a city in the TSP problem).

The travelling salesman problem (also called the travelling salesperson problem or TSP) asks the following question: “Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?” It is an NP-hard problem in combinatorial optimization and is important in theoretical computer science and operations research. The steps of PACO for the berth allocation problem are followed in Figure 5 [41].

During the initialization phase, suppose the time $t = 0$ and the cycle number $n^{itr} = 0$, and set the maximum number of cycles n^{itr}_{max} . m ants are placed on n nodes (arriving container vessels to berth), the initial quantity of information on each edge is (j, j') , and $\tau_{ij'}(0) = const$, where the initial quantity per unit $\Delta\tau_{ij'}(0) = 0$.

Fitness function evaluation steps:

- (1) Individual ants select elements (container vessel) j to go according to the probability calculated through the state transition probability formula (3), where $j \in tabu_k, c_{ij'}$ is a time equal to vessel j 's time of completion minus vessel j 's arrival time, and l_k is the path of all nodes the ant has walked, namely, the total time of all vessels staying in the port.
- (2) Modify the tabu search list pointer that is for ant j to move to the new element (container vessel) after choosing, and put the element (container vessel) into the individual ant's tabu search list.
- (3) If the elements (container vessels) in Set C have not been traversed, which means $k < m$, then $k = k + 1$; otherwise, continue.

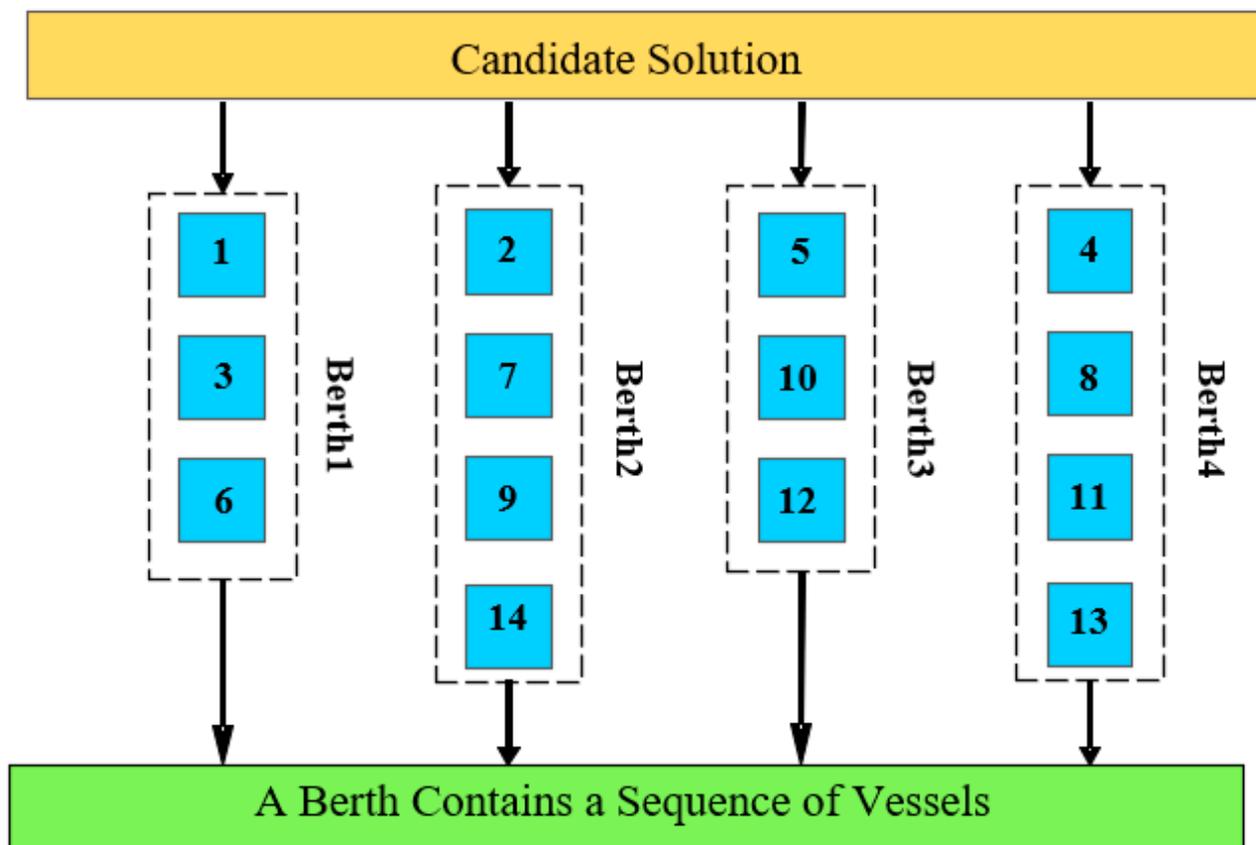


Figure 4. Fixed-length solution representation.

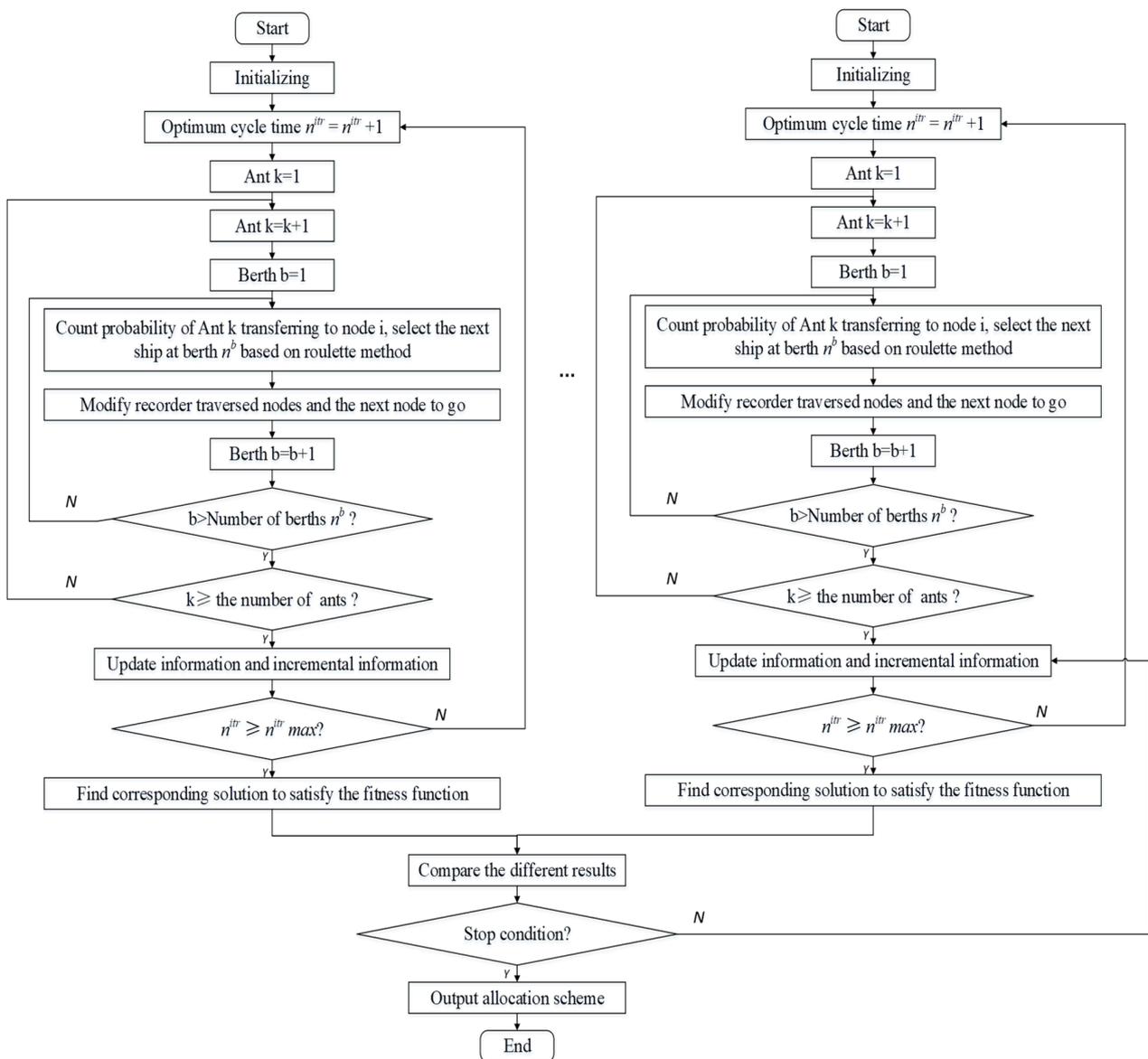


Figure 5. Flowchart of PACO for the berth allocation problem.

5. Numerical Experiments

In this section, the proposed PACO algorithm is verified based on a series of experiments. This part of the research data is obtained by the investigation of Shanghai Port. By analyzing the berth allocation data of container terminals for 24 consecutive hours, the experimental results are reliable and authentic.

5.1. Determination of Parameter Values

In order to conduct numerical analysis, it is necessary to first determine the values of a few parameters.

Based on a survey at a four-berth container terminal of the Shanghai Port (time, reference number, and/or other information about the survey), when a group of 20 vessels came into a port at the same time, a certain period was chosen at random. Table 1 shows the time intervals in between the vessels' arrival (to simplify, the first vessel's arrival time is used as a reference, and the time is set as 0) and the processing time for loading/unloading.

Table 1. Intervals between the arrivals and times required for loading/unloading.

No. of Vessels	1	2	3	4	5	6	7	8	9	10
Inter-arrival time (h)	0.0	13.5	8.0	5.0	2.5	0.5	0.5	1.5	5.0	0.0
Loading/unloading time (h)	12.9	11.7	8.7	19.7	22.0	11.5	9.8	10.8	10.8	9.8
No. of Vessels	11	12	13	14	15	16	17	18	19	20
Inter-arrival time (h)	14.0	8.8	7.3	5.0	1.5	13.5	6.0	3.25	1.8	1.5
Loading/unloading time (h)	9.7	9.8	11.0	12.5	14.7	10.7	6.8	11.3	10.0	10.3

Note: The time interval between vessels arriving at the port refers to the time interval between two ships in a row, such as the time interval between vessels one and two, which is 13.5 h, and the time interval between ships two and three, which is 8 h.

5.2. Results of Single-Berth Allocation

5.2.1. Based on ACO

Using the arrival time and loading/unloading time of the 20 known vessels, the ant colony algorithm can calculate the total time of arrival of all vessels in the port. In this experiment, $Q = 100$; $\alpha = 1$, $\beta = 5$, $\rho = 0.5$; the maximum number of iterations is 200 times and the ant population C^a equals 5.

By MATLAB programming, the ant colony algorithm can solve the problem of berth allocation. The total time of the 20 vessels in the port is 1095.02 h, and the berthing order is as follows:

1–2–3–6–10–8–11–14–15–19–16–20–17–13–7–9–18–12–4–5

The abscissa shown in Figure 6 represents the number of iterations, while the ordinate reflects the total time spent in the port by all vessels. As the number of iterations grows, the total port time of the boats steadily lowers and ultimately stabilizes, as seen in Figure 6.

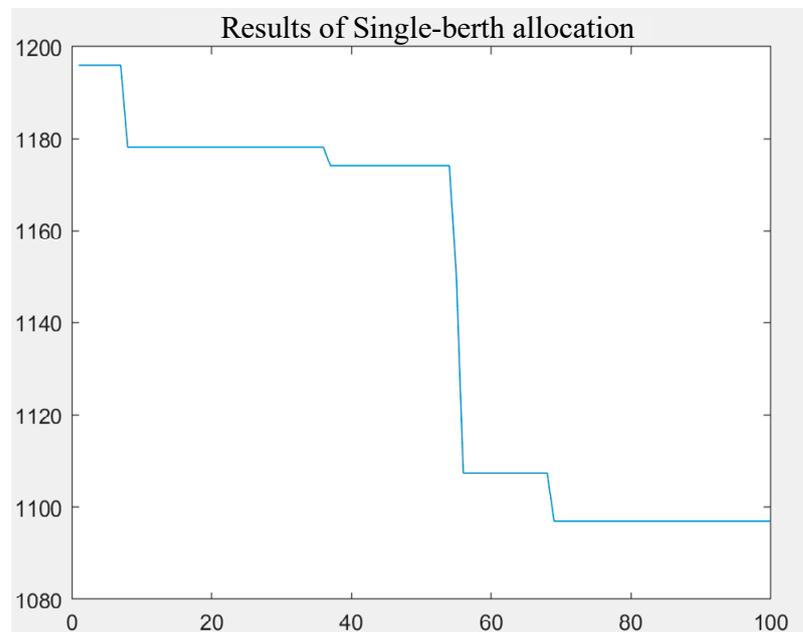


Figure 6. Results of single-berth allocation based on ACO.

5.2.2. Based on PACO

Referring to the basic concepts of the parallel ACO algorithm and the process of solving the single-berth problem, the parallel ACO algorithm is programmed in MATLAB. In this experiment, six computers are used as the sub-processors. The calculated total ship processing time in the port is at least 1054.93 h. The order of the berthing of the ship corresponding to this result is:

1–2–3–8–6–9–11–10–7–14–16–19–12–13–17–18–15–20–4–5

The abscissa represents the number of iterations, while the ordinate reflects the total time spent in the port by all vessels. As the number of iterations grows, the total port time of boats steadily lowers and ultimately stabilizes, as seen in Figure 7.

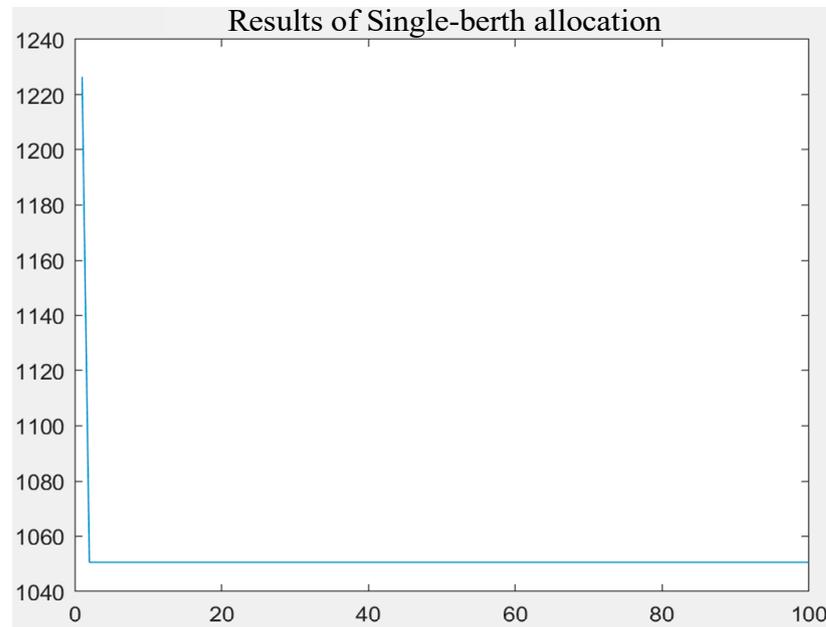


Figure 7. Results of single-berth allocation based on PACO.

Compared with the result of the ant colony algorithm, it is shown that the PACO algorithm outperforms the normal ACO algorithm. The minimum total ship processing time has been improved from 1095.02 h to 1054.93 h, which makes the berthing time reduce by more than 2 h per vessel. And the computing efficiency is highly improved, with a five-times-faster convergence capability. Then, the PACO algorithm is proved to be feasible in solving the problem of berth allocation in the port. When it is applied to actual production, the operating efficiency of the container terminal can be improved.

5.3. Results of Multi-Berth Allocation

5.3.1. Based on ACO

The total actual stay time of the above 20 vessels in the port is 304. In 4 h, the PSO model of the linear decreasing inertia of inertial weights is adopted, the maximum number of iterations is set to 100 times, the best result obtained by continuously optimizing 10 times is 257.2 h, and the worst result is 262 h.

According to the first-come-first-service rule, the actual total time of 20 vessels staying in the harbor is 257.2 h. The specific allocation results are as follows:

Berth1: 1–4–8–13–17

Berth2: 2–5–12–18

Berth3: 3–7–10–11–15–20

Berth4: 6–9–14–16–19

The ant colony optimization is employed to optimize the berth scheduling problem. Since Dorigo has found that when $\alpha = \{0.5, 1\}$, $\beta = \{1, 2, 3, 4, 5\}$, the ant system can always converge to the optimal solution, and when m (the number of ants) is close to n (the number of nodes), the algorithm has a better performance, so the parameters are assumed as follows:

$$Q = 100; \alpha = 1, \beta = 5, \rho = 0.5;$$

Set the maximum number of iterations to 200 times with the ant population C^a equaling 5.

According to the first-come-first-service rule, the actual total time of 20 vessels staying in the harbor is 257.2 h. The specific allocation results are as follows:

Berth1: 1–5–12–16–20

Berth2: 2–6–9–13–17

Berth3: 3–7–8–11–15–19

Berth4: 4–10–14–18

The ordinate shown in Figure 8 represents the average time of a vessel in the port in four berths. Compared with the single-berth order of vessels, it can be seen that the sum of the total port time of four berths is less than the total time of single-berth berths. It is found that the addition of berths can improve the waiting time of the vessels, so as to increase the loading and unloading efficiency of the vessels.

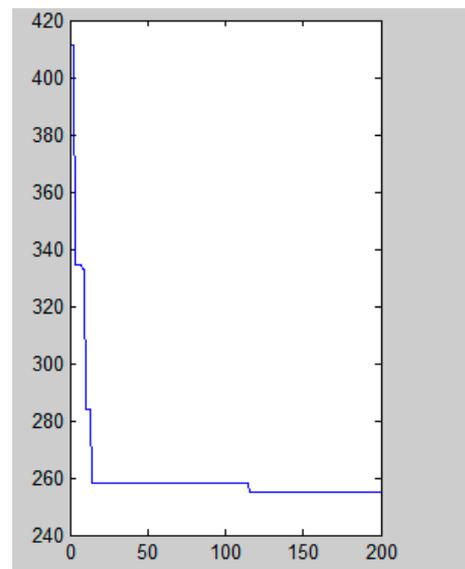


Figure 8. Results of multi-berth allocation based on ACO.

5.3.2. Based on PACO

In a parallel ant colony algorithm, the value of the parameter is the same as that of the ant colony algorithm, which can ensure the consistency of the experiments, and it is helpful for comparative analysis. The results show the best results for 251.83 h. The scheduling Gantt chart is shown in Figure 9, and the corresponding berth assignment of the vessel berthing plan is as follows:

Berth1: 1–4–10–13–17

Berth2: 2–6–9–15–18

Berth3: 3–7–8–14–19

Berth4: 5–11–12–16–20

The ordinate shown in Figure 10 represents the average time of the vessel in the port in four berths. Compared with the single-berth parallel ant colony algorithm, with the same principle of the PACO algorithm, six computers are used to calculate the vessels' time in the port. Based on the results of different sub-processors, it is found that the final result of the PACO algorithm is superior to that of the ACO algorithm in terms of both the objective value and computing efficiency.

5.4. Sensitivity Analysis

In order to verify the validity of the number of ant colonies, under the same conditions as those of other experiments, only one parameter was changed, and several other experiments were carried out. When the ant population C^a was changed, the experimental results are shown in Table 2 and Figure 11.

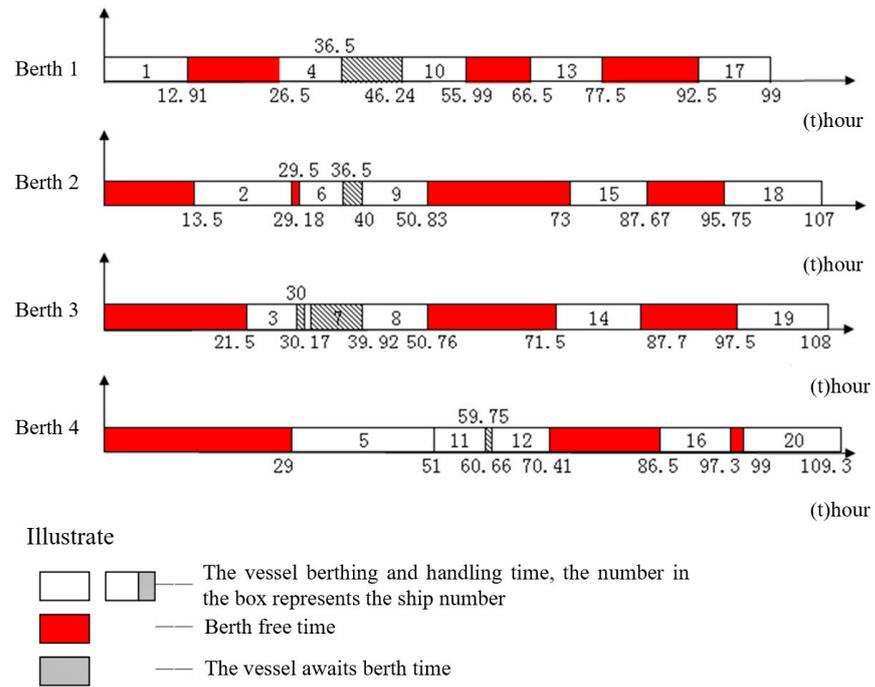


Figure 9. Results of multi-berth allocation based on PACO.

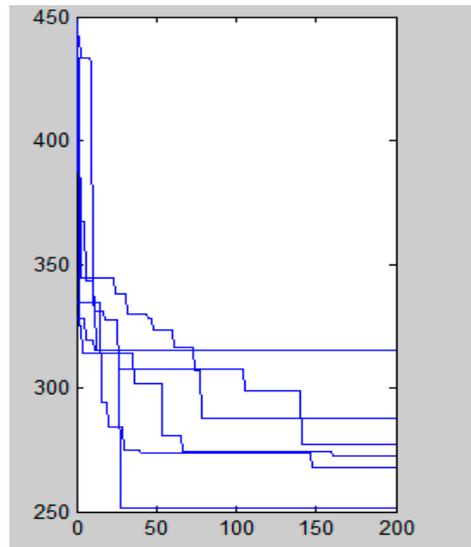


Figure 10. Results of multi-berth allocation based on PACO.

The data in Table 2 also show that the smaller the quantity of ants, the earlier the initial convergent algebra without changing other parameters. Since berth scheduling is a parallel machine problem, each ant can only walk N^b/C^s nodes (in this case, the number of nodes is five). Experiments show that when the number of ants is less than the number of nodes, the optimization is easily convergent. As the number of ants increases, the calculated time decreases and the convergence speed increases. At the same time, increasing the number of ants will make the calculation time for each generation longer; thus, it increases the burden on the processor. From the benchmarking results in Table 2, we found that when the number of ants was chosen to be 5 or 10, the calculation result was better and the calculation time was shorter. In other words, in the parallel ant colony algorithm calculation, the number of ants should be approximately equal to the number of nodes.

Table 2. Berth scheduling optimization results by changing C^a (hour).

C^a	The Optimal Value (the Total Residence Time of All Vessels in the Port)	Started Convergence Algebra
1	257.75	7
2	256.84	5
3	256.42	5
4	257.76	1
5	251.83	5
10	251.83	3
20	251.83	2
30	251.83	2
40	251.83	1
50	251.83	1

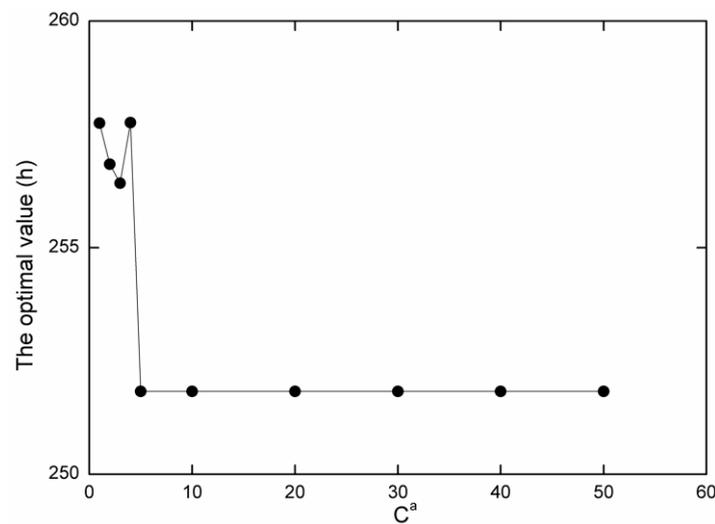


Figure 11. Berth scheduling optimization results line chart by changing C^a .

After determining the number of ants, the next step is to determine the values of α and β in the parallel ant colony algorithm to ensure optimal calculation results. Therefore, C^a was defined as five in this experiment, and the influence of the values of α and β on the results was studied. When $\alpha = 0.5$ or 1 , $\beta = \{1, 2, 3, 4, 5, 6\}$, the total residence time of all vessels in the port and the started convergence algebra were calculated, respectively. The result is as shown in Table 3:

Table 3. Berth scheduling optimization results by changing α, β (hour, $C^a = 5$).

α	β	The Optimal Value (the Total Residence Time of All Vessels in the Port)	Started Convergence Algebra
0.5	1	257.35	53
0.5	2	251.83	15
0.5	3	251.83	21
0.5	4	251.83	3
0.5	5	251.83	3
0.5	6	256.84	6
1	1	262.92	22
1	2	260.33	21
1	3	255.51	6
1	4	256.60	5
1	5	251.83	3
1	6	256.76	2

As can be seen in Figure 12, regardless of whether α is equal to 0.5 or 1, as long as $\beta = 5$, the minimum total residence time of all vessels in the port is obtained, and the minimum optimal value is 251.83 h.

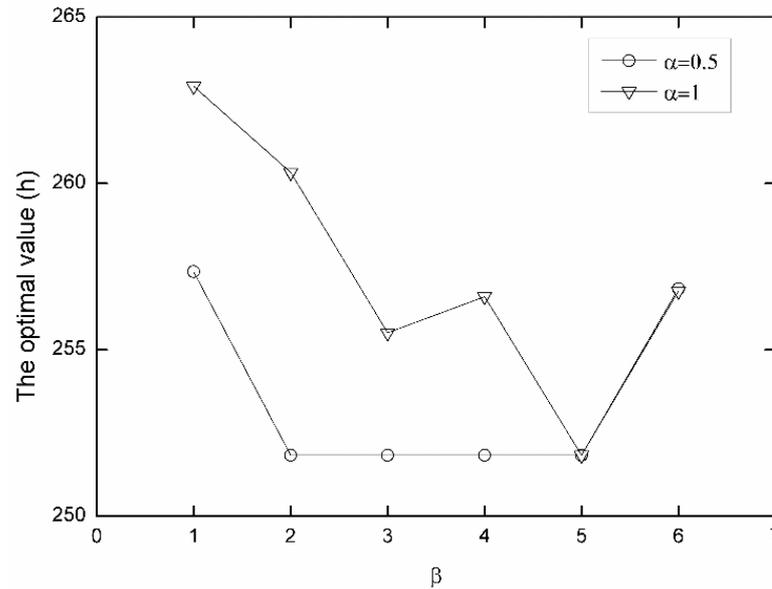


Figure 12. Berth scheduling optimization results line chart by changing α, β (hour, $C^a = 5$).

In comparison to the previous experiment, the value of C^a in this experiment was changed to 10 and we obtained all the corresponding berth scheduling operations results by varying α and β as presented in Table 4. As we can see in Figure 13, the values of α and β have little effect on the experimental results. In other words, $\alpha = 0.5$ or $1, \beta = \{2, 3, 4, 5\}$, and the total residence time of all vessels in the port can be minimized.

Table 4. Berth scheduling optimization results by changing α, β (hour, $C^a = 10$).

α	β	The Optimal Value (the Total Residence Time of All Vessels in the Port)	Started Convergence Algebra
0.5	1	256.18	29
0.5	2	251.83	22
0.5	3	251.83	8
0.5	4	251.83	2
0.5	5	251.83	2
0.5	6	251.83	17
1	1	256.10	14
1	2	251.83	6
1	3	251.83	5
1	4	251.83	2
1	5	251.83	2
1	6	255.85	2

Through the above two sets of experiments, an obvious rule can be found: if both C^a and α remain unchanged, as the value of β increases, the started convergence algebra gradually decreases. Moreover, if β is too small, it is poorly convergent. In addition, if other conditions are consistent, $\alpha = 0.5$ makes the optimization result easier to obtain than $\alpha = 1$. Overall, the optimal values obtained from the second set of experiments ($C^a = 10$) are more stable. To consider the overall calculation results and the optimized speed, the values of $C^a, \alpha,$ and β should be 10, 0.5, and 5.

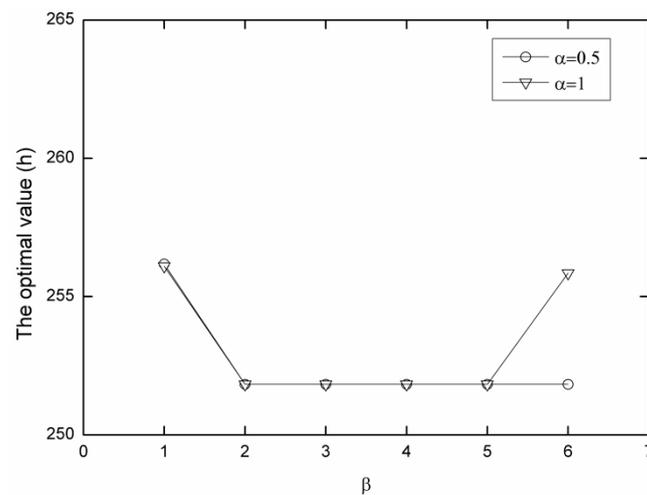


Figure 13. Berth scheduling optimization results line chart by changing α, β (hour, $C^a = 10$).

5.5. Benchmarking Analysis with Other Algorithms

To further validate the superiority of our proposed algorithm, we conducted experiments on the same case using both the Genetic Algorithm (GA) and Particle Swarm Optimization (PSO).

In solving the berth allocation problem for the given case, we applied a Genetic Algorithm (GA). GA is an iterative search technique designed to find optimal solutions through successive generations. To adapt GA effectively to our specific case, we meticulously selected and fine-tuned its control parameters, including the population size, crossover probability, mutation probability, and maximum number of generations. Through experimentation, we determined that a population size of 30, a crossover probability of 0.8, and a mutation probability of 0.1, with 20 generations, yielded highly promising results. This parameter configuration struck a balance between maintaining population diversity and converging towards optimal solutions. As a result, our GA optimization achieved a total port time of 254.17 h for the 20 vessels in the case.

In a parallel effort, we employed Particle Swarm Optimization (PSO) to address the berth allocation problem in the same case scenario. PSO is a population-based optimization algorithm inspired by the social behaviors of birds and fish. In PSO, a population of potential solutions, represented as particles, navigates the search space to discover the optimal solution. Each particle adjusts its position based on its individual experiences and the experiences of neighboring particles. The primary goal of PSO is to find the best solution by iteratively updating particle positions and velocities. Upon running the PSO algorithm and optimizing the berth allocation, we calculated the total vessel time for the port. The outcome of our experiments revealed that, under the PSO algorithm’s optimization, the total vessel time for the port was reduced to 257.25 h.

Comparative results can be found in Table 5, highlighting the performance differences among the actual condition, GA, PSO, ACO, and our proposed PACO.

Table 5. Comparison results for the case.

	Actual	GA	PSO	ACO	PACO
The total port time for the 20 vessels in the case (h)	304.4	254.17	257.25	257.2	251.83

These comparisons underscore the effectiveness of our proposed Enhanced Ant Colony Algorithm (PACO) in addressing the challenges posed by the Discrete Dynamic Berth Allocation Problem (DDBAP) in the specified case.

6. Conclusions

The problem of berth allocation for incoming vessels is discrete and random, which brought difficulties in obtaining the optimum solution and largely affected the port operation efficiency. To address this issue, this paper built up a dynamic BAP model referring to the truly practical environment of berth allocation in container terminals. A parallel ant colony algorithm catering to berth allocation is designed to allocate berths for vessels arriving at the right time. With the principle of the ant routing process, a parallel ant colony algorithm is delivered to handle the BAP problem. Regarding the prematurity and convergence of the optimization algorithm, the experimental analysis and tuning of the parameters in the ant colony algorithm are carried out. To validate the feasibility of the approach, a case study is provided.

The existing ship scheduling scheme and berth allocation scheme are relatively single, and there is no strong flexibility or universality; for this reason, this paper proposes a dynamic berth allocation scheme, which can be achieved under certain conditions and can reduce the ship's stay time in a port to a certain extent. In the numerical examples, it can be found that the PACO algorithm effectively avoids the deficiencies of the premature convergence of the ACO algorithm when solving the berth allocation problem. The superiority of the PACO algorithm is proved. In addition, the different parameters of the PACO algorithm are studied and compared to find the optimal value when solving the problem of berth allocation. This not only ensures the accuracy of the calculation results but also improves the calculation speed. Without increasing the number of large berths, the service level of the terminal can be effectively improved based on better allocation planning via the proposed method, and the shortage of berths at terminals can be alleviated. Moreover, we conducted comprehensive comparisons with other common optimization algorithms to showcase the superiority of the PACO algorithm. While our study considered limited practical factors and operated under specific assumptions, we acknowledge the need for future research to involve more practical constraints. In particular, we plan to conduct experiments on a broader range of instances to ensure the robustness and generalizability of our approach. This will further enhance the applicability of our algorithm to real-world scenarios and contribute to the ongoing development of berth allocation solutions.

Author Contributions: Conceptualization, M.Y. and Y.L.; methodology, M.Y.; software, Y.W.; validation, X.J., M.Y. and Y.L.; formal analysis, M.Y.; investigation, Y.L.; resources, M.Y.; data curation, M.Y.; writing—original draft preparation, M.Y. and Y.L.; writing—review and editing, Y.L.; funding acquisition, M.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This paper has been funded by the National Key R&D Program of China (2020YFB1712400) and the National Natural Science Foundation of China (52272423).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

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

References

1. Lu, B.; Hua, G. Optimizing berth operation system in container terminal by simulation. In Proceedings of the IEEE International Conference on Logistics, 2017, Sydney, NSW, Australia, 24–27 July 2016.
2. Yu, M.; Wang, S.; Cai, Y. A dispatching method for trucks at container terminal by using fuzzy-cnp concept. In Proceedings of the 2010 International Conference on Logistics Engineering and Intelligent Transportation Systems, Wuhan, China, 26–28 November 2010; pp. 1–4.
3. Roy, D.; de Koster, R. Stochastic modeling of unloading and loading operations at a container terminal using automated lifting vehicles. *Eur. J. Oper. Res.* **2018**, *266*, 895–910. [[CrossRef](#)]
4. Bierwirth, C.; Meisel, F. A follow-up survey of berth allocation and quay crane scheduling problems in container terminals. *Eur. J. Oper. Res.* **2015**, *244*, 675–689. [[CrossRef](#)]

5. Bierwirth, C.; Meisel, F. A survey of berth allocation and quay crane scheduling problems in container terminals. *Eur. J. Oper. Res.* **2010**, *202*, 615–627. [[CrossRef](#)]
6. Frojan, P.; Francisco, C.J.; Alvarez, V.R.; Koulouris, G.; Manuel, T.J. The continuous berth allocation problem in a container terminal with multiple quays. *Expert Syst. Appl. Int. J.* **2015**, *42*, 7356–7366. [[CrossRef](#)]
7. Jakub, W.; Maciej, D. Selecting algorithms for large berth allocation problems. *Eur. J. Oper. Res.* **2020**, *283*, 844–862.
8. Lin, S.W.; Ting, C.J. Solving the dynamic berth allocation problem by simulated annealing. *Eng. Optim.* **2014**, *46*, 308–327. [[CrossRef](#)]
9. Mauri, G.R.; Lorena LA, N.; Lorena LA, N.; Laporte, G. An adaptive large neighborhood search for the discrete and continuous berth allocation problem. *Comput. Oper. Res.* **2016**, *70*, 140–154. [[CrossRef](#)]
10. Rodriguez-Molins, M.; Salido, M.A.; Barber, F. Robust scheduling for berth allocation and quay crane assignment problem. *Math. Probl. Eng.* **2014**, *2014*, 834927. [[CrossRef](#)]
11. Čustić, A.; Sokol, V.; Punnen, A.P.; Bhattacharya, B. The bilinear assignment problem: Complexity and polynomially solvable special cases. *Math. Program.* **2017**, *166*, 185–205. [[CrossRef](#)]
12. Liu, Y.; Gao, C.; Zhang, Z. Solving NP-Hard Problems with Physarum-Based Ant Colony System. *Trans. Comput. Biol. Bioinform.* **2017**, *14*, 108–120. [[CrossRef](#)]
13. Issam, E.H.; Azza, L.; Yassine, T. A multi-objective model for discrete and dynamic Berth Allocation Problem. In Proceedings of the 2nd International Conference on Big Data, Cloud, and Applications, Tetouan, Morocco, 29–30 March 2017.
14. Hsu, H.P. A HPSO for solving dynamic and discrete berth allocation problem and dynamic quay crane assignment problem simultaneously. *Swarm Evol. Comput.* **2016**, *271*, 156–168. [[CrossRef](#)]
15. Hu, Z.H. Multi-objective genetic algorithm for berth allocation problem considering daytime preference. *Comput. Ind. Eng.* **2015**, *89*, 2–14. [[CrossRef](#)]
16. Lin, S.W.; Ying, K.C.; Wan, S.Y. Minimizing the total service time of discrete dynamic berth allocation problem by an iterated greedy heuristic. *Sci. World J.* **2014**, *2014*, 218925. [[CrossRef](#)]
17. Türkoğulları, Y.B.; Taşkın, Z.C.; Aras, N.; Altınel, İ.K. Optimal berth allocation, time-variant quay crane assignment and scheduling with crane setups in container terminals. *Eur. J. Oper. Res.* **2016**, *2016*, 985–1001. [[CrossRef](#)]
18. Simrin, A.; Diabat, A. The dynamic berth allocation problem. *RAIRO—Oper. Res.* **2015**, *49*, 473–494. [[CrossRef](#)]
19. Kovac, N.; Davidovic, T.; Stanimirovic, Z. Evolutionary algorithm for the minimum cost hybrid berth allocation problem. In Proceedings of the International Conference on Information, Intelligence, Systems and Applications, Corfu, Greece, 6–8 July 2015; pp. 1–6.
20. Kordić, S.; Davidović, T.; Kovač, N.; Dragović, B. Combinatorial approach to exactly solving discrete and hybrid berth allocation problem. *Appl. Math. Model.* **2016**, *40*, 8952–8973. [[CrossRef](#)]
21. Budipriyanto, A.; Wirjodirdjo, B.; Pujawan, I.N.; Gurning, S. A simulation study of collaborative approach to berth allocation problem under uncertainty. *Asian J. Shipp. Logist.* **2017**, *33*, 127–139. [[CrossRef](#)]
22. Xiang, X.; Liu, C.; Miao, L. A bi-objective robust model for berth allocation scheduling under uncertainty. *Transp. Res. Part E Logist. Transp. Rev.* **2017**, *106*, 294–319. [[CrossRef](#)]
23. Eisen, H.E.; Van der Lei, J.E.; Zuidema, J.; Koch, T.; Dugundji, E.R. An Evaluation of First-Come, First-Served Scheduling in a Geometrically-Constrained Wet Bulk Terminal. *Front. Future Transp.* **2021**, *23*, 709822. [[CrossRef](#)]
24. Bai, Y.; Lv, Y.; Zhang, J. Smart mobile robot fleet management based on hierarchical multi-agent deep Q network towards intelligent manufacturing. *Eng. Appl. Artif. Intel.* **2023**, *124*, 106534. [[CrossRef](#)]
25. Wang, Z.H.; Zhi, Y.; Yang, X.F. Multi-agent and ant colony optimization for ship integrated power system network reconfiguration. *J. Syst. Eng. Electron.* **2022**, *2*, 489–496. [[CrossRef](#)]
26. Randall, M.; Lewis, A. A Parallel Implementation of Ant Colony Optimization. *J. Parallel Distrib. Comput.* **2015**, *62*, 1421–1432. [[CrossRef](#)]
27. Yu, M.; He, Z.; Li, D.; Yin, L. Research on multi-AGV path planning in automated container terminal. In Proceedings of the 2019 5th International Conference on Transportation Information and Safety (ICTIS), Liverpool, UK, 14–17 July 2019.
28. Yu, M.; Wang, S. The Application Research of Container Berth Allocation Based on Ant Colony Algorithms. In Proceedings of the ICTIS 2011 Wuhan China: Multimodal Approach to Sustained Transportation System Development: Information, Technology, Implementation, Wuhan, China, 30 June–2 July 2011; pp. 1755–1761.
29. Li, H.T.; Zhen, N.M.; Peng, T.S. Dynamic Ant Colony Genetic Algorithm Based on TSP. *Mach. Des. Manuf.* **2019**, *12*, 147–149.
30. Zhang, X.J.; Wang, K.; Cao, M.Y. Coordinated Optimization of Berth Assignment and Box Allocation in Simulation Environment. *Mod. Manuf. Eng.* **2022**, *3*, 13–18.
31. Niu, M.; Wang, Z.Y.; Li, Y.J. Considering the integration optimization of berth allocation and ship scheduling using priority. *High-Tech Commun.* **2020**, *30*, 972–981.
32. Zhang, M.; Jiang, Z.; Wang, L.; Yao, Y. Research on Parallel Ant Colony Algorithm for 3D Terrain Path Planning. In *Asian Simulation Conference*; Springer: Singapore, 2017; pp. 74–82.
33. Xu, D.; Li, C.L.; Leung JY, T. Berth allocation with time-dependent physical limitations on vessels. *Eur. J. Oper. Res.* **2012**, *216*, 47–56. [[CrossRef](#)]
34. Dorigo, M.; Di Caro, G.; Gambardella, L.M. Ant algorithms for discrete optimization. *Artif. Life* **1999**, *5*, 137–172. [[CrossRef](#)]

35. Dorigo, M.; Stützle, T. Ant colony optimization: Overview and recent advances. In *Handbook of Metaheuristics*; Springer: Cham, Switzerland, 2019; pp. 311–351.
36. Geza, K.; Balazs, L.; Janos, J. Parallel Ant Colony Algorithm for Shortest Path Problem. *Period. Polytech. Civ. Eng.* **2019**, *63*, 243–254.
37. Chen, X.; Dai, Y. Research on an Improved Ant Colony Algorithm Fusion with Genetic Algorithm for Route Planning. In Proceedings of the IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chongqing, China, 12–14 June 2020.
38. Huang, W.; Huang, M.; Wang, H.; Software, S.O.; University, J.N. Application research of ant colony algorithm based on association rules in cross-selling. *Comput. Digit. Eng.* 2014. [[CrossRef](#)]
39. Chowdhury, S.; Marufuzzaman, M.; Tunc, H.; Bian, L.; Bullington, W. A modified Ant Colony Optimization algorithm to solve a dynamic traveling salesman problem: A case study with drones for wildlife surveillance. *J. Comput. Des. Eng.* **2019**, *6*, 368–386. [[CrossRef](#)]
40. Guo, W.; Ji, M.; Zhu, H. Collaborative Optimization of Berth Allocation and Yard Storage in Container Terminals. *Traffic Transp. Eng.* **2019**, *7*, 56–70.
41. Ge, Y.; Wang, A. Spatial scheduling strategy for irregular curved blocks based on the modified genetic ant colony algorithm (MGACA) in shipbuilding. *Int. J. Prod. Res.* **2018**, *56*, 3099–3115. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.